Citrix SecurSpaces™

Copyright © 1997–2026. Citrix Systems, Inc. All Rights Reserved.

Contents

Overview 9

About SecurSpaces 10

SecurSpaces for developers 11

SecurSpaces for AI and ML engineers 16

SecurSpaces at a glance 20

How it works 25

Concepts 31

Platform 33

Organizations 34

Projects 36

Workspaces 37

SecurSpaces Flex overview 38

SecurSpaces videos 43

Get started 46

Set up your account 48

Your first workspace 53

Hello World workspace 59

Develop, debug, and share a Node.js app 64

Set up a project for your team 71

Develop 75

Develop in a workspace 77

Console overview 83

Workspaces page 88

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Create a workspace 90

Manage workspaces 102

Connect over SSH 104

What persists in a SecurSpaces workspace 115

Connect to external services over SSH 120

Share a running application 124

Your profile 129

Profile overview 134

Integrations 140

Configuration 146

Security 153

Persistent Docker 156

Run a graphical desktop in a workspace with the Debian GUI image 160

Install SecurSpaces as a desktop app 164

Work from a virtual desktop 165

Give your AI assistant access to the documentation 171

Troubleshooting 175

Manage 183

Onboard developers 184

Reusable template for SecurSpaces workspace onboarding 187

People page 190

Users 191

Add and remove users and groups 197

Roles and permissions 208

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Projects and organizations **216**

Templates **220**

Build and manage custom workspace images for SecurSpaces **232**

Container images **240**

Update an existing container image **246**

Create an image from an existing one **249**

Manage workspace toolchains **255**

Manage workspace toolchains with Mise **258**

Manage workspace toolchains with Nix **267**

Project resources **274**

Code repositories **275**

Secrets **278**

Data buckets **281**

Mount points **284**

Connected HTTP services **293**

Connected SSH services **296**

Choose between data buckets and mount points **299**

Enable outbound SSH **302**

SecurSpaces Flex workspace templates **306**

Deploy **310**

Deploy an evaluation environment **311**

Deploy in the cloud **312**

Prepare a XenServer VM **320**

Prepare a Hyper-V VM **321**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Install on a local VM 324

Validate the deployment 333

Manage the evaluation environment 336

Deploy to a cluster 337

Amazon EKS 340

Azure AKS 346

Google GKE 352

Deploy SecurSpaces on Red Hat OpenShift 357

Nutanix Kubernetes Platform 367

Install with the Strong Installer CLI 371

Ingress 379

NetScaler CPX 381

NGINX 383

Istio 386

SSH TCP forwarding with NGINX 390

External database 400

External proxy configuration 402

Deploy SecurSpaces in multiple regions 405

GPU-enabled workspaces on AKS 407

Upgrade 412

Administer 417

System configuration 419

Terraform provider 421

Run AI coding agents in workspaces 426

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Configure platform login 433

Identity providers 435

Platform roles 436

Google OAuth 437

Microsoft Entra 443

Connect an OpenID Connect provider 447

SAML 449

SCIM configuration 451

User access control 462

Code repositories 465

Connect GitHub 467

Connect GitLab 469

Connect Bitbucket Cloud 471

Azure DevOps 475

Integrations 477

Use HashiCorp Vault as a secret manager 480

Register JFrog as a third-party app 481

Configure AI Gateway routing in SecurSpaces 485

Publish JetBrains Gateway 496

SecurSpaces workspaces plugin for Backstage 499

Citrix DaaS integration 502

Platform resources 505

Registry access 505

IDE versions 508

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Image caching **515**

Mount point storage **518**

Prepare AWS storage for Mount Points **521**

Prepare the AWS resources **523**

Configure SecurSpaces for AWS Mount Points **534**

Author a PersistentVolume for Attach Existing **539**

Troubleshoot AWS Mount Points **546**

Platform settings **550**

General settings **551**

Workspace policy **559**

Workspace lifecycle and idle detection **571**

Regions **574**

Custom actions **576**

Project labels **580**

Onboarding emails **583**

Email gateway **587**

Information security policy **590**

Terms of service agreements **593**

Security settings **594**

Operations **597**

Licensing **598**

The SecurSpaces database **602**

Database sizing **607**

Back up the database **610**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Restore the database 619

Analytics 623

Workspace resource usage insights 627

Audit page 629

SIEM integration 632

Insights page 637

Resource allocation 638

Container process metrics 640

Security and compliance 643

Security control catalogue 645

Trust boundaries 661

How the SecurSpaces proxy service works 668

Layered controls with the Citrix platform 673

Security data flows 675

Network policies 683

Create a network policy 688

Assign and enforce a network policy 691

Monitor and troubleshoot network policies 693

Technical security overview for SecurSpaces Flex 695

Network connections for SecurSpaces Flex 699

Reference 702

Audit event catalog 703

Container image requirements 714

Cryptography 720

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Mount point limits and behavior 722

Network destinations 733

Network policy fields 737

Ports and protocols 740

Sizing 741

REST API 748

SDS CLI reference 753

Supported IDEs 765

System requirements 768

Release notes 774

What's new 775

Fixed issues 801

What's new in SecurSpaces Flex 835

SecurSpaces document history 837

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview

Citrix SecurSpaces™ is a secure, cloud-based development environment (CDE) designed to enhance developer productivity while maintaining enterprise-grade security. Its purpose is to streamline the provisioning and management of coding
environments, so organizations can boost efficiency and collaboration among internal and external teams.

It provides fast onboarding through preconfigured, container-based workspaces that are accessible from anywhere, making it well suited to hybrid and remote teams. The platform can be deployed on public or private clouds and on self-hosted servers, and it supports fully air-gapped operation for high-security settings.

By centralizing development resources and removing local dependencies, SecurSpaces helps protect source code, credentials, and intellectual property. It enforces access controls and integrates with security models such as Zero Trust, reducing the risk of data leaks and supporting DevSecOps practices.

The platform also helps reduce the costs of high-specification laptops, maintenance, and security software, while giving real-time visibility and governance over the development lifecycle. Its environments integrate with common DevOps tools and CI/CD workflows.

Other names you will see

Two other names appear in and around the product. Both refer to SecurSpaces.

Name	Where you meet it
Secure Developer Spaces	The SSH extension for VS Code-based IDEs is published as **Citrix Secure Developer Spaces - Remote SSH**, and the sidebar it adds carries the same name. Deployment domains use the short form, as in `assets.sds.citrix.com`.
Strong Network™	The platform's name before Citrix acquired it. It survives in identifiers that cannot be renamed without breaking existing deployments: the `strong_network_startup` directory an image must provide, the `/strong/` paths inside a workspace, and the `strongnetwork` images on Docker Hub.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Where to go next

If you want to	Start here
See the specifications, limits, and what it integrates with	SecurSpaces at a glance
Understand what the platform does for developers	SecurSpaces for developers
See how the pieces fit together	Architecture
Create your first workspace	Create your first workspace
Deploy the platform	System requirements
Understand how the platform is structured	Concepts
Give an AI assistant access to this documentation	Give your AI assistant access

For AI assistants and agents

This documentation is published in the llms.txt format:

- Index: https://assets.sds.citrix.com/llms.txt
- Full corpus: https://assets.sds.citrix.com/llms-full.txt

If you run SecurSpaces, your own instance serves the same files at `/docs/llms.txt` and `/docs/llms-full.txt`, matching your deployed version. See Give your AI assistant access.

About SecurSpaces

This section explains what Citrix SecurSpaces™ is, who it's for, and how it works.

Page	What it covers
SecurSpaces for developers	What a workspace is, what changes about your day, and what stays the same.
SecurSpaces for AI and ML engineers	GPU-backed workspaces, large resource allocations, shared training data, and governed access to AI providers.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Page	What it covers
SecurSpaces at a glance	Supported platforms, limits, integrations, and deployment models on one page.
How it works	The control plane, workspaces, and proxy service that make up the platform, and how a workspace starts.
Concepts	How the platform, organizations, projects, and workspaces fit together.
SecurSpaces Flex	A Citrix-managed service on Azure that delivers secure Linux workspaces through Citrix Platform Flex.
Videos	Testimonials, product demos, and recorded webinars for SecurSpaces.

SecurSpaces for developers

August 19, 2026

`Developer`

Citrix SecurSpaces™ gives you a powerful, pre-configured development environment you can open in seconds —from any device, any browser, or your favorite IDE. Think of it as your own dedicated Linux dev machine, always ready, always consistent, and accessible from anywhere.

This guide explains what SecurSpaces is, how it compares to what you're used to, and introduces the key concepts you'll encounter.

How it works

Your workspace is a full Linux development environment running in the cloud. It has everything your project needs —IDE, terminal, source code, build tools, and dependencies —set up and ready to go. You connect to it from your browser or a locally installed IDE.

When you log in to SecurSpaces, you'll see your project and available workspace templates. You pick a template, select launch, and within seconds you have a running environment with your repositories cloned, your tools installed, and your IDE connected. That's it —you're coding.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What's in it for you

Start coding in minutes, not days

No more "install this, configure that, ask Dave for access."Your workspace comes pre-loaded with everything your project needs —the right language runtimes, build tools, dependencies, and repository access are already configured.

Your tools, your way

Use your dot files, extensions, keybindings, and shell configuration. SecurSpaces supports VS Code (browser and desktop), JetBrains IDEs, Cursor, Windsurf, Kiro, and any other editor that connects over SSH. You choose how you work.

Any device becomes a dev machine

Work from your laptop, a thin client, a tablet, or even a borrowed machine. Your workspace is the same everywhere because it runs in the cloud, not on your local hardware.

Power when you need it

Need more CPU for a heavy build? GPUs for ML training? Scale resources on demand —no hardware requisition required. Your workspace can be more powerful than any laptop.

Experiment freely

Spin up a new workspace in seconds. Try something risky, break things, throw it away, start fresh. The cost of experimenting is near zero.

Consistent environments across the team

Everyone on your project uses the same base tools, same versions, same configuration. If it works in your workspace, it works in everyone's. "Works on my machine"stops being a thing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What stays the same

If you're coming from a laptop, VDI, or WSL-based workflow, most of what you know still applies:

- **Git workflow** —Clone, branch, commit, push. Same as always.
- **Terminal** —Full Linux shell with your tools, aliases, and scripts.
- **IDE** —Connect your desktop IDE or use the browser-based version. Your choice.
- **Your files** —Everything you save under your home directory persists between sessions.
- **Root access** —You have full control inside your workspace.

What's different

SecurSpaces changes *where* your development happens, not *how* you develop. Here's what's actually different:

Traditional setup	With SecurSpaces
Code lives on your laptop	Code lives in your cloud workspace (can't lose it if your laptop dies)
You install and maintain your own tools	Your project owner configures the base environment in a template
Hardware limits what you can build	Workspace resources scale independently of your device
Environment setup takes hours or days	Launch a pre-configured workspace in seconds
"Works on my machine"debugging	Everyone shares the same base environment
Onboarding a new project = manual setup	Pick a template and start coding

The key mental shift: Your laptop (or browser) is a *window* into your workspace, not the workspace itself. Everything runs in the cloud —your device is just the screen.

Key terminology

These are the terms you'll see when using SecurSpaces:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Term	What it means
Workspace	Your personal cloud development environment. It contains your IDE, code, tools, and terminal. You can have multiple workspaces for different projects or branches.
Template	A pre-configured workspace blueprint created by your project owner. It defines the container image, tools, repositories, and resource allocation so you can start immediately without manual setup.
Project	An organizational unit that groups related workspaces, users, and resources together. You're assigned to one or more projects by your project owner.
Project owner	The person who manages your project's configuration —templates, resources, access controls, and team membership. They're your first point of contact for project-specific questions.
Profile	Your personal settings —work schedule, Git credentials, shell configuration, secrets, and IDE preferences. These apply across all your workspaces.
Integrations	Connections to external services (GitHub, GitLab, Bitbucket, Azure DevOps) that let your workspaces access code repositories seamlessly.
Workspace Apps	Applications running inside your workspace (like a dev server) that you can access from your browser or share with teammates.
Cloud IDE	The browser-based VS Code editor built into SecurSpaces. No installation needed —just select it to open.

How SecurSpaces fits into your day

A typical day with SecurSpaces looks like this:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. **Start your day.** Open your browser or IDE. Your workspace may already be running (based on your work schedule) or starts in seconds when you connect.
2. **Write code.** Pull latest changes, work on your feature, run tests —just like you would on a local machine.
3. **Preview and share.** Run your application and access it through Workspace Apps. Share a live preview with teammates without deploying anywhere.
4. **End your day.** Close your browser or IDE. Your workspace pauses automatically based on your schedule, saving resources. Everything is there when you come back.

Frequently asked questions

Is this just VDI for developers?
No. VDI streams a desktop image to your device. SecurSpaces gives you a real Linux development environment —full terminal access, root permissions, native IDE connectivity, and the ability to run containers and services inside your workspace. It's built for development workflows, not desktop virtualization.

Do I lose my work when I close my browser?
No. Your home directory (`/home/developer`) is persistent. Files, code, and configurations are saved between sessions. However, system-level changes outside of `/home/developer` are reset when a workspace restarts —use startup scripts or ask your project owner to update the template for those.

Can I use my own dot files and IDE settings?
Yes. SecurSpaces supports personal shell configurations (`.bashrc`, `.zshrc`), IDE settings, VS Code extensions, and startup scripts. Configure these once in your profile and they apply to every workspace.

Can I work offline?
No —SecurSpaces workspaces require a network connection since they run in the cloud. If you lose connectivity, your work is saved up to the last change, and you can reconnect and resume where you left off.

What about security restrictions?
Your organization may apply specific security policies (like clipboard controls or network restrictions). These are configured by your platform administrator to protect sensitive code and data. If you encounter a restriction you don't understand, ask your project owner or platform administrator.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Next steps

Ready to get started?

1. Quick start: Set up your account —Configure your profile, Git access, and credentials.
2. Quick start: Create and connect to your workspace —Launch your first workspace and connect your IDE.

SecurSpaces for AI and ML engineers

August 19, 2026

`AI/ML Engineer`

Citrix SecurSpaces™ gives you a Linux environment with as much compute as your cluster can provide, GPUs
attached when you need them, and shared access to training data that does not have to be copied onto your
laptop.

The parts that matter for model work are the same parts that matter for any development work —the environment is disposable, the data is not, and neither leaves the platform.

If you are new to SecurSpaces, SecurSpaces for developers
covers the basics of workspaces, templates, and connecting an IDE. This page covers what is different about
training and experimentation.

How it works

A workspace is a Linux environment sized from a specification your platform administrator defines, so the CPU,
memory, disk, and GPUs you get are requested rather than owned. A GPU workspace is scheduled onto a
GPU-capable node in the cluster when you start it, and released when you stop it.

Training data is reached where it already lives. Instead of copying a dataset into the workspace, you attach a
mount point to read it in place, or a data bucket if you need a snapshot pinned at a known version. A notebook
server or inference endpoint you run inside the workspace is published through a workspace app,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

which puts it
behind an authenticated URL rather than an open port. Outbound requests to AI services leave through the
platform, so they can be attributed and governed.

The sections below cover each of these in turn.

What's in it for you

The data does not have to move

Training data stays in your storage and your cluster. Nothing is cloned onto a laptop, which is often the
whole reason a team adopts the platform in the first place.

Someone else can reproduce your run

A template fixes the image, the toolchain, and the resources, so an experiment a colleague repeats starts
from the same place as yours.

The GPUs are not sitting in one person's machine

A GPU node serves whoever needs it. You request the hardware for the run rather than owning it.

Access is governed without you managing it

Repository access, secrets, AI provider traffic, and outbound network are controlled centrally and audited,
so you are not the one holding credentials.

Compute

A workspace is sized from a specification your platform administrator defines. The product ceilings are:

Resource	Maximum
CPU	1000 cores

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Resource	Maximum
Memory	10,000 GiB
Disk	10,000 GiB
GPUs	1000

These are the limits the platform enforces. What you can actually request is whatever your administrator has
made available in a workspace specification, bounded by the hardware in the cluster. If the size you need is
not offered, that is a conversation with your platform administrator rather than a product limit.

GPUs are requested as a count on the specification and scheduled onto GPU-capable nodes. For how a platform
administrator provides them, see
GPU-enabled workspaces on AKS.

Training data

Copying a dataset into every workspace is slow and duplicates data you may not be allowed to duplicate.
SecurSpaces offers two ways to avoid it, and they behave differently.

	Mount points	Data buckets
How data arrives	A live connection to external storage	A snapshot copied onto the cluster
Shared edits	Immediately visible to every workspace using it	Requires a publish cycle
Duplication	None —data stays in the external storage	Each snapshot is replicated
Versioning	Use the storage provider's own	Built in

For a dataset several teams read, a **mount point** is usually the right choice: it is POSIX file storage mounted into the workspace, there is no per-mount size quota, and nothing is copied. For a dataset you want
pinned at a known version, a **data bucket** gives you that.

See Choose between data buckets and mount points.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

A mount point is shared infrastructure: every workspace that mounts it works on the same live files. Plan
backups and retention in the storage provider, alongside the rest of your data platform. See
Mount point limits and behavior.

Notebooks and long-running jobs

A notebook server, dashboard, or inference endpoint running in your workspace can be reached through a
workspace app, which publishes a port over an authenticated URL. See
Share a running application.

Workspaces pause when they are idle, which keeps GPU time from being spent on environments nobody is using.
For a job that runs unattended —an overnight training run, a long batch evaluation —ask a platform administrator or security officer to turn on **Bypass Schedule** for that workspace. The workspace then keeps
running until you stop it. See
Workspace idle detection.

Access to AI providers

Traffic from workspaces to AI services can be routed through an internal AI Gateway. When it is configured,
requests carry headers identifying the user, project, and workspace, so usage is attributable and can be
reported on.

This is a platform-level integration rather than something you configure. See
AI Gateway.

Outbound access is also governed by network policy, so the domains a workspace can reach —including model
and package registries —may be restricted. If a download is blocked, that is the reason. See
Network policies.

Next steps

- Set up your account
- Your first workspace
- What persists in a workspace

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Manage workspace toolchains

SecurSpaces at a glance

August 21, 2026

A summary of what Citrix SecurSpaces™ runs on, what it integrates with, and where its limits are, for anyone
deciding whether it fits their environment.

Every figure here is owned by another page, which is linked from its row. Follow the link when you need the
detail or the caveats.

Where it runs

Deployment model	Self-hosted on your own Kubernetes cluster, in your cloud, on-premises, or fully air-gapped
Managed alternative	SecurSpaces Flex, a Citrix-operated service billed through Platform Flex credits
Kubernetes	1.34 or later, on a cluster dedicated to SecurSpaces
Node architecture	`amd64` only. `arm64`, including AWS Graviton, is not supported
Certified platforms	Amazon EKS, Azure AKS, Google GKE, Red Hat OpenShift, VMware Tanzu Kubernetes Grid, Nutanix Kubernetes Platform

See System requirements.

What it needs from your infrastructure

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Ingress	NetScaler CPX recommended and deployable by the chart; NGINX, HAProxy, and the Kubernetes Gateway API also supported
Database	MongoDB, either managed such as Atlas, or deployed in-cluster
Registry	Any OCI-compatible registry, for example Amazon ECR, Azure Container Registry, Google Artifact Registry, or JFrog Artifactory
DNS and TLS	Two domains with valid certificates, one of them a wildcard for workspace proxying

See Ingress and System requirements.

What it integrates with

Identity providers	SAML 2.0, OpenID Connect, Google OAuth, Microsoft Entra ID
User provisioning	SCIM
Code repositories	GitHub, GitLab, Bitbucket, Azure DevOps, each self-hosted or managed
Secrets	Built-in secret storage, or HashiCorp Vault
Artifacts	JFrog Artifactory
Developer portal	Backstage plugin
Virtual apps and desktops	Citrix DaaS
AI traffic	Routed through an AI Gateway with per-user attribution

See Identity and access and Integrations.

How developers reach a workspace

© 1997–2026 Citrix Systems, Inc. All rights reserved.

In the browser	Cloud IDE based on VS Code
From a local IDE	Over SSH, from VS Code Desktop, JetBrains Gateway, Cursor, Windsurf, or Kiro
Graphical desktop	Available through a GUI-enabled workspace image
Published application	Through Citrix DaaS

See Develop in a workspace.

Automation

The platform can be operated as code as well as through the console, which matters if you manage environments
with a platform team rather than by hand.

REST API	Covers what the console covers —workspaces, projects, organizations, resources, users, and reporting. Authenticated with an API key. Enabled by a platform feature flag
Terraform provider	Manages users, organizations, projects, groups, and workspace templates. Distributed through your deployment rather than the public registry
Deployment	Helm values and the Strong Installer CLI, so cluster configuration lives in version control

See API,
Terraform provider, and
Install with the Strong Installer CLI.

Limits

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CPU per workspace	1000 cores
Memory per workspace	10,000 GiB
Disk per workspace	10,000 GiB
GPUs per workspace	1000
Users and workspaces	No user limit. SecurSpaces is included in the Citrix Platform License, with no new SKU

These are product ceilings, not recommendations. What you can request is bounded by the hardware in your
cluster. See Sizing.

> **Note:**
>
> Workspaces are Linux containers, so toolchains that require macOS or Apple hardware —iOS, iPadOS, macOS,
> watchOS, and tvOS development —cannot run in SecurSpaces. This is a property of the platform rather than
> a gap in it.

Indicative scale

Deployment	Service nodes	Workspace nodes	Database
50 developers, proof of concept	One 8 vCPU / 32 GiB	8 vCPU / 32 GiB, autoscaling	20 GiB
500 developers	Two 8 vCPU / 32 GiB	16 vCPU / 64 GiB, autoscaling	100 GiB

Workspace nodes scale with how many workspaces run *at once*, not how many exist, so idle detection is the
largest single lever on cost. See Sizing.

Security and governance

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Source code	Stays in the workspace; never lands on the endpoint
Tooling	Developers do not assemble their own environments. A workspace is created from a platform-owned, version-controlled template that fixes the image, toolchains, package sources, and extensions, so every workspace from it is identical
Network control	Per-workspace egress policies, enforced at the proxy
Data protection	Clipboard, download, and upload controls
Audit	Every action recorded, exportable to a SIEM, retained in-platform for 360 days
Controls	31 identified controls across 8 domains, in the control catalogue
With the wider Citrix platform	DaaS, Secure Private Access, and Chrome Enterprise Premium add controls on the access path in front of a workspace. See Layered controls

See Security and compliance.

Try it

Fastest evaluation	The 1-Click VM runs the whole platform on a single virtual machine, no cluster needed
Production trial	Deploy to a cluster and follow the standard installation

See Deploy an evaluation environment and Kubernetes platforms.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

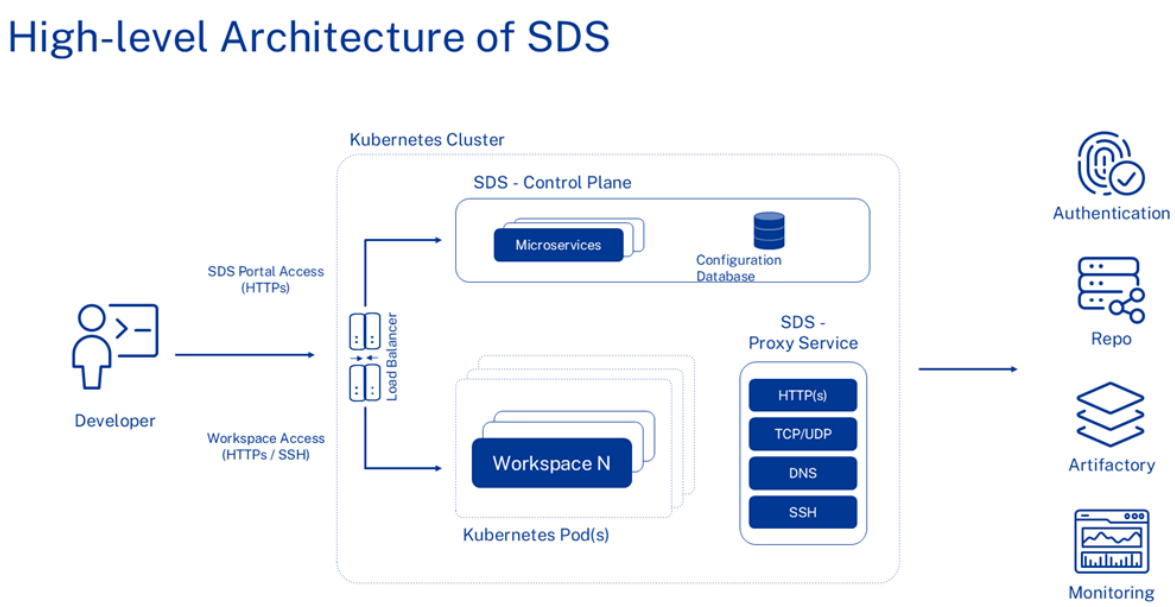
Related information

- How it works
- System requirements
- Sizing
- Security and compliance

How it works

September 24, 2026

Citrix SecurSpaces™ is a cloud-native platform for hosting Cloud Development Environments (CDEs). It runs
entirely on Kubernetes, either in your own cloud account or on your own hardware, and can run fully air-gapped. Development environments are containerized and Linux based, and nothing is installed on the
developer's device.



The platform has three parts:

- **The control plane.** A set of microservices and a configuration database. The control plane holds
  organizations, projects, policies, resources, and workspace state. It does not carry the developer's code.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Workspaces.** Each workspace is a Kubernetes pod running a container image, with its own CPU, memory,
  and disk allocation. Workspaces are created from templates, and are paused when idle and deleted according
  to your retention settings.
- **The SecurSpaces proxy service.** All traffic leaving a workspace passes through it —HTTP and HTTPS, TCP and UDP,
  DNS, and SSH. This is where credentials are injected, network policy is applied, and traffic is recorded.

Because every path in and out of a workspace terminates in the SecurSpaces proxy service, the platform can give
developers access to repositories, registries, and internal services without ever placing long-lived credentials on the developer's device.

How developers connect

Developers reach a workspace in one of two ways:

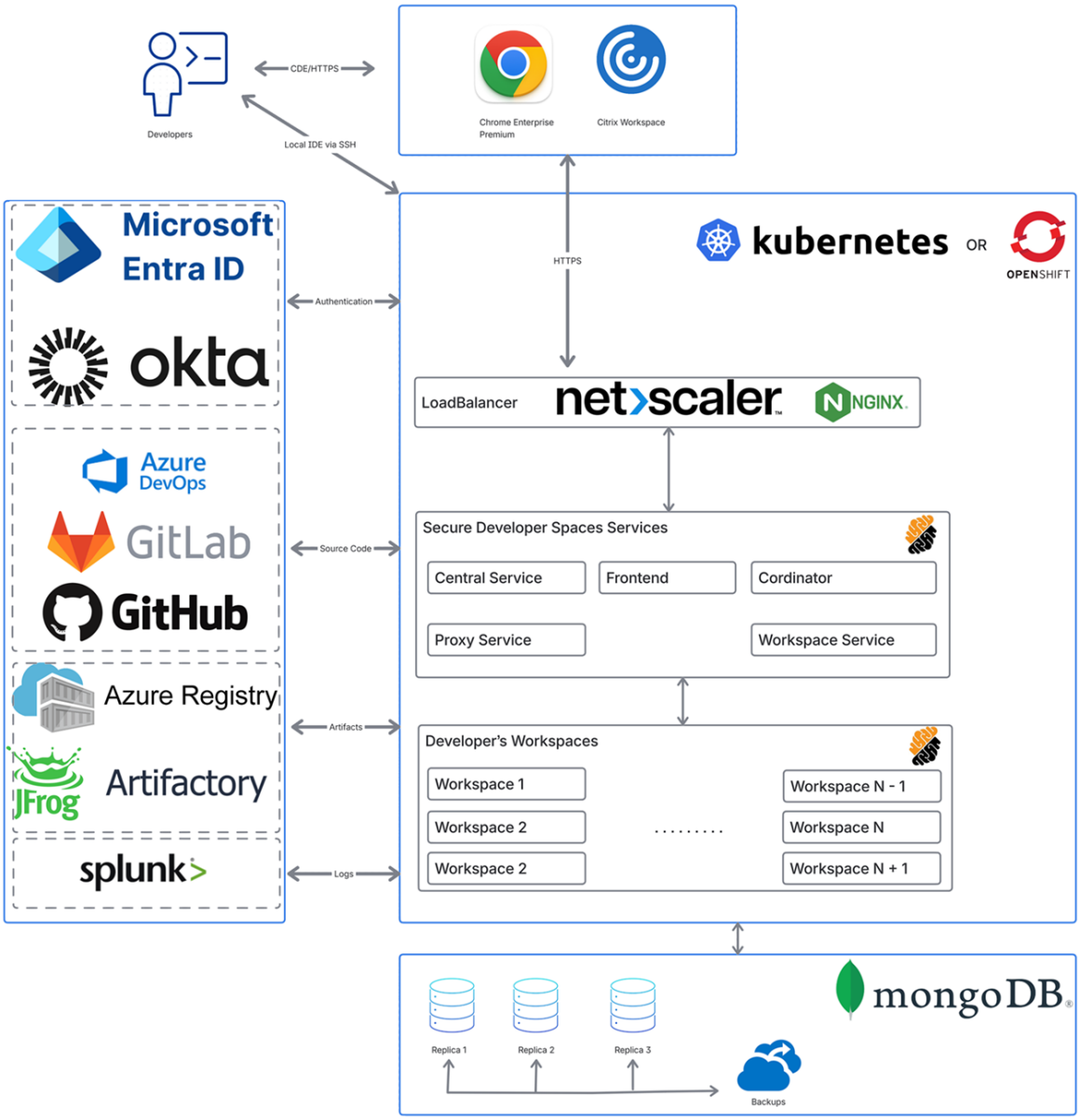
- **In the browser.** VS Code for the Web is served from the workspace over HTTPS. Nothing is installed
  locally, so any managed or unmanaged device can be used.
- **From a local IDE or terminal.** VS Code Desktop, JetBrains Gateway, Cursor, Kiro, and Windsurf connect
  over SSH, as does a plain terminal.

The browser path is the one that supports the full set of data loss prevention controls, because the platform renders the session. See
Connect over SSH, and
Supported IDEs for versions and limitations.

Component overview

The following table describes the key architectural components and the technologies each one supports.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Component	Description	Supported Technologies
Platform	Cloud-native, container-based environment (CDE)	Kubernetes or Red Hat OpenShift. See Deployment guides.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Component	Description	Supported Technologies
Load Balancing	Traffic management and ingress control	NetScaler CPX (recommended, and deployed by the SecurSpaces chart), NGINX, HAProxy, or the Kubernetes Gateway API.
Database	Metadata and workspace state storage	MongoDB
Authentication	Multi-factor and identity provider integration	Google OAuth, Microsoft OAuth, SAML, SCIM, and OpenID Connect (OIDC).
Source Code	Integrations with version control systems	GitHub, GitLab, and Bitbucket (self-hosted and managed); Azure DevOps (managed).
Artifacts	Container image management and credential storage	Any OCI-compatible container registry (for example, JFrog Artifactory, Amazon ECR, or Google Artifact Registry); HashiCorp Vault for secrets and credentials.
Logs	System and workspace telemetry	System Analytics and the Event Log Catalog for audit trails. Supports SIEM integration using Common Event Format (CEF).

For infrastructure sizing and network requirements, see
System requirements.

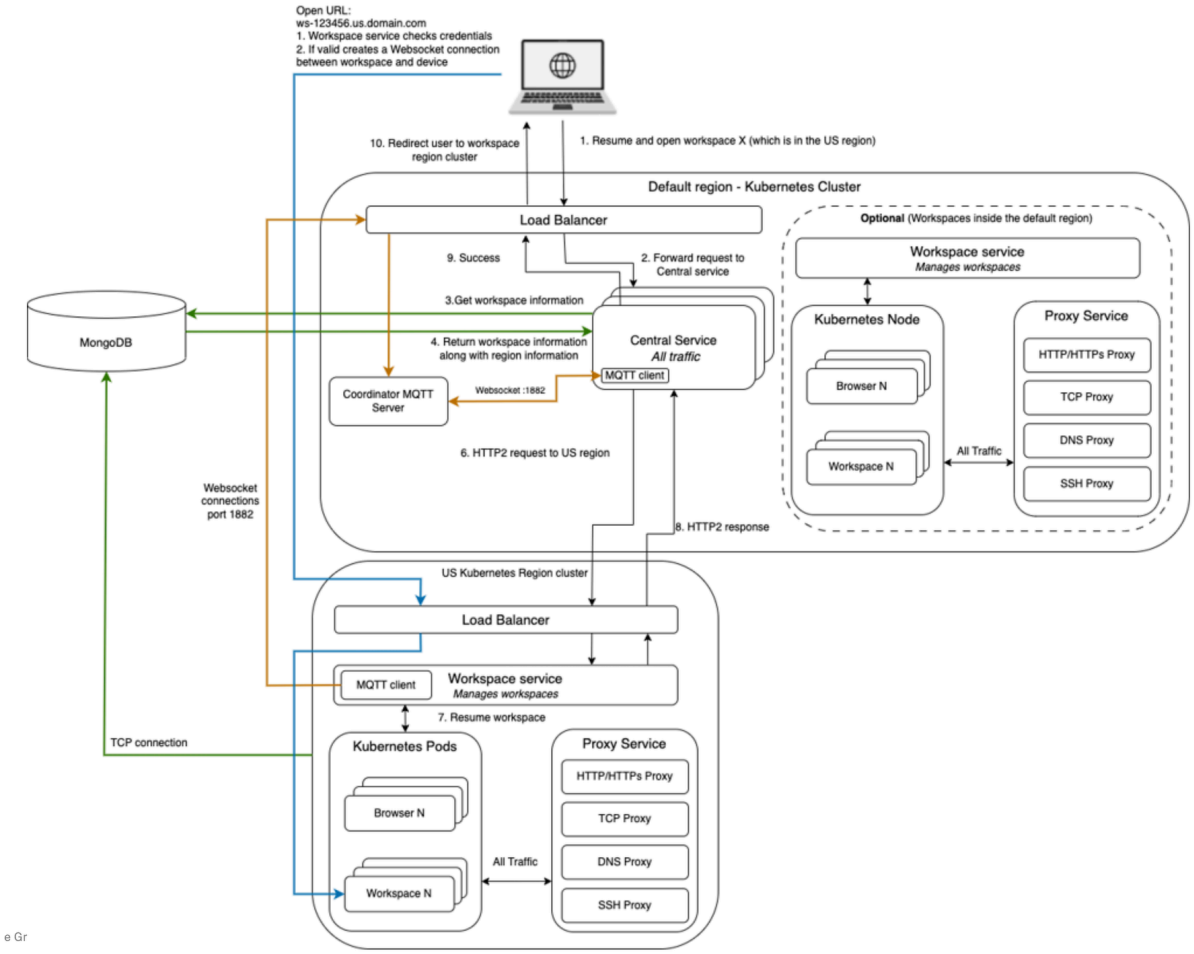
How a workspace starts

A single-region deployment resolves a workspace within one cluster. A multi-region deployment can place a
workspace in a region close to the developer, or in a region required by data residency rules, while keeping
one control plane and one entry point.

The following diagram shows a developer in the default region opening a workspace that lives in the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

US
region.



The diagram numbers this sequence 1–4 and 6–10.

1. The developer resumes and opens a workspace. The request arrives at the load balancer in the default
   region.
2. The load balancer forwards the request to the central service.
3. The central service requests the workspace record from MongoDB.
4. MongoDB returns the workspace information, including the region the workspace belongs to.
5. The central service sends an HTTP/2 request to the cluster in the US region.
6. The workspace service in that region resumes the workspace pod.
7. The region returns an HTTP/2 response.
8. The default region reports success.
9. The developer is redirected to the cluster in the workspace's own region.

The developer's browser then opens the workspace URL directly against that region, for example

© 1997–2026 Citrix Systems, Inc. All rights reserved.

`ws-123456.us.domain.com`. The workspace service checks the developer's credentials and, if they are valid,
establishes a WebSocket connection between the workspace and the device. From that point the session runs
against the regional cluster, and traffic in and out of the workspace passes through the SecurSpaces proxy service in
that region.

Two supporting channels run continuously and are not part of the numbered sequence:

- **Coordination.** The workspace service in each region maintains an MQTT client connection to the
  coordinator MQTT server in the default region, over WebSocket on port 1882. It appears on the diagram as the
  unnumbered WebSocket link. This is how the control plane reaches regions it does not own.
- **State.** Regional clusters hold a TCP connection to MongoDB, so workspace state is recorded centrally
  regardless of which region runs the workspace.

Workspaces can also run in the default region. That is optional, and a deployment can keep the default region for the control plane alone.

Deployment guides

SecurSpaces can be deployed on any Kubernetes-compatible infrastructure, including public cloud, on-premises
hardware, and hybrid environments. The following guides cover common deployment targets:

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Red Hat OpenShift

To evaluate SecurSpaces on a single VM instead, see
Deploy an evaluation environment.

The 1-Click VM

A production installation of SecurSpaces runs on a Kubernetes cluster that you build, with a registry and a database
that you provide. That is the right shape for real users, and too much work for an evaluation. The 1-Click VM removes
that work by running the whole platform on one virtual machine: a single-node K3s cluster with the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

platform services,
the database and NetScaler ingress.

The same installer builds it in two ways. In the cloud, the installer creates the VM itself with Terraform, and
configures DNS and TLS. For a local VM, the installer cannot reach your hypervisor, so it generates a deployment script
instead. You run the script on the VM, and the installer maps an assigned **try**.sds.citrix.com hostname to the VM's IP
address. The hostname is what makes the TLS certificate valid, which is why you use it rather than the IP address.

Because of this split, a local deployment involves two machines. The installer host prepares the deployment, and the
target host runs it. They are kept separate because the target host must start clean: the script installs its own K3s
and container runtime, and conflicts with any that are already there.

The trade-off is a platform with no redundancy and no upgrade path. The 1-Click VM is built to be created, evaluated
and thrown away. A pilot with real users belongs on a cluster.
See Deploy to a cluster.

Related information

- How the SecurSpaces proxy service works
- Trust boundaries and data flows
- Key concepts
- System requirements

Concepts

August 19, 2026

Citrix SecurSpaces™ is arranged as four nested levels. Almost everything in this documentation — permissions, resources, security settings —is defined at one of them and applies to everything beneath.

```
Platform
└── Organization
    └── Project
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
4           └── Workspace
```

The four levels

Level	What it is	Who works at this level
Platform	The deployment as a whole, and the baseline for everything in it	Platform administrators, security officers
Organization	A group of projects, usually a business unit or customer	Organization owners
Project	A team, with its members, resources, and security rules	Project owners
Workspace	A single containerized development environment	Developers

Why the nesting matters

Two things follow from the hierarchy, and they explain most of what you will see in the interface.

Settings are inherited. A setting made at a broader level applies to everything beneath it. A narrower level can usually override it, unless the broader level enforces the value. Some settings work the other way:
remote development over SSH must be allowed at *every* level above, so a project owner cannot enable it if the
platform has it switched off. See
Workspace policy.

Roles are project bound. The same person can hold a different role in each project they belong to. What
they can see and change depends on the role they hold there, not on a single account-wide setting. See
Roles and permissions.

If a page, button, or setting described in this documentation is not visible to you, the usual reason is one
of these two: your role does not carry the permission, or a broader level has fixed the setting.

Related information

- How it works —the architecture and components

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Roles and permissions
- Project and organization settings

Platform

The platform is organized in organizations and projects. A series of operations are readily available at platform level. For example, workspaces, resources and users can be managed at platform-level by users
with a platform role, such as the administrator or the security officer. Governance metrics such as insights and audit logs are also
aggregated at the platform level.

The platform administrator has a view on all workspaces running on the platform, i.e. across organizations and projects, so that they can be updated rapidly, e.g. container configuration. The administrator can also have an overall view on the onboarded users.

Resources can be managed at the platform level so that they become available across organizations and projects. This applies to all types of resources supported by the platform.

Insights and audits dashboards are available at the platform level, allowing metrics to be selected and aggregated across organizations and projects.

Finally, a variety of settings and operations are relevant at the platform level. For example, these include global workspace settings regarding performance and security, global authentication settings, and compliance functions, to name a few.

- View Organizations
- Platform Settings

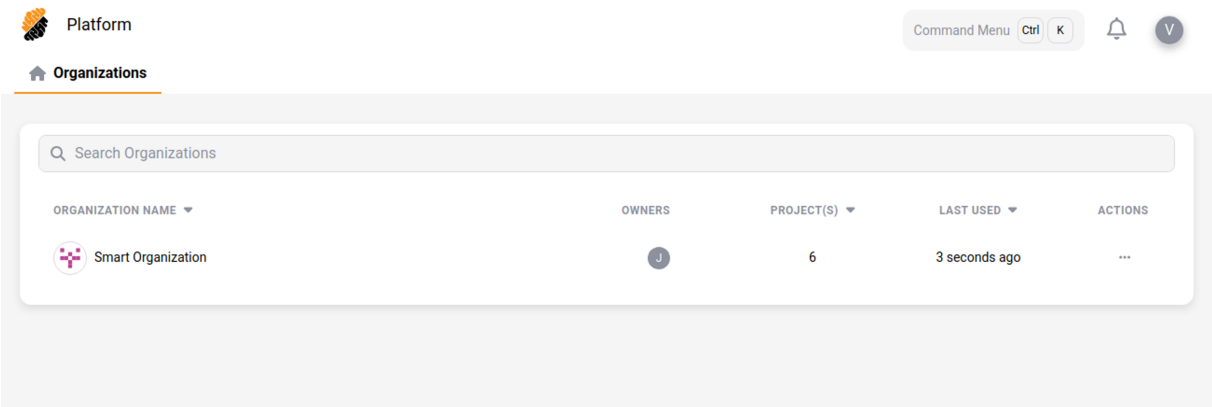
View Organizations

Organizations can be viewed at the level of the platform and listed in a table.

An administrator can create an organization.

Click on the Citrix SecurSpaces™ logo to **view your organizations** to which you belong.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Organizations List

Platform Settings Admin

For comprehensive control over your Platform's configurations, visit the dedicated Platform Settings page.

Related information

- Organizations
- Projects
- Workspaces
- Concepts

Organizations

The platform allows administrators and platform owners to organize projects into organizations. An **Organization** is the main entity regrouping projects, developers, resources, and security rules for one development project.

- Organization's Characteristics
- View Organization's Projects
- Create an Organization
- Organization Settings

Organization's Characteristics

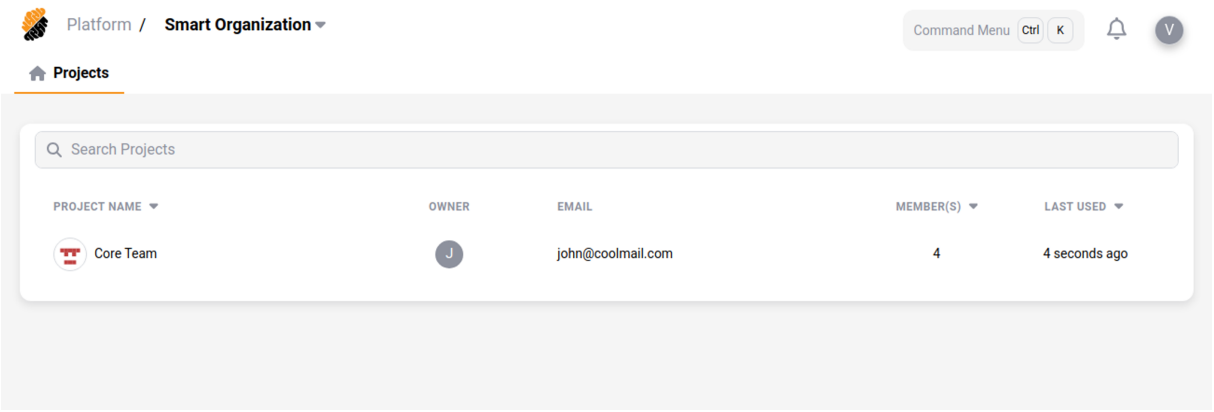
An organization is defined by the following characteristics:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Organization Name**,
- **Organization owner**,
- **Organization owner's email**,
- **Project(s)** that it contains,
- **Resources, such base containers, policies, etc**.

View Organization's Projects

In a project, by clicking on the name of your **organization** at the top left corner of the screen, you can display all of the **projects** contained in it.



Create an Organization `Admin`

You can create an organization by pressing the "**Add New Organization**"button.

You will need to select the following information:

- **Organization Name**,
- **Owner**. i.e. any user with the right permissions to own an organization.

> **Info**
>
> To create an organization, you must be an ***Admin***.
>
> An **Admin** can create an organization on behalf of an owner with the permissions to be the **organization owner**.

Organization Settings `Admin`

For comprehensive control over your Organization's configurations, visit the dedicated Organization Settings page.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Platform
- Projects
- Workspaces
- Concepts

Projects

A **Project** within an Organization regroups developers, resources, and security rules. The aim of a project is to provide the development team with all resources required for development, as well as access control and governance mechanisms to the project owner.

- Project's Characteristics
- Create a Project
- Project Settings

Project's Characteristics

A project is defined by the following characteristics:

- **Name**,
- **Project owner**, i.e. any user with the right permissions to own a project,
- **Project owner's email**,
- **Member(s)**, i.e. the user belonging to the project,
- **Resources, including workspaces, base containers, repositories, etc**.

Create a Project `Admin`

You can create a project by pressing the "**Add New Project**"button.

You will need to select the following information:

- **Project Name**,
- **Owner**, i.e. any existing user on the platform or a new user (to onboard).

> **Info**
>
> To create a project, you must be an **organization owner**.
>
> An **Admin** can create a project and assign it to a user.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Project Settings `Project Owner`

For comprehensive control over your Project's configurations, visit the dedicated Project Settings page.

Related information

- Organizations
- Platform
- Workspaces
- Concepts

Workspaces

A workspace is a Cloud Development Environments (CDEs) available for coding and data science. Workspaces can be accessed using the Cloud IDE or through an SSH connection from a local installed IDE.

Workspaces are running online on top of a virtual machine and managed using a container orchestrator for resilience. The performance of a workspace, i.e. compute and storage capabilities, are set by the specifications of the underlying virtual machine.

Workspaces are technically speaking virtual processes, with the aim of replacing the use of a virtual machine for code development and data science. They are lightweight and so that they can be started and paused much quicker than a VM counterpart.

> **Important:**
>
> Citrix SecurSpaces™ workspaces are containerized and **ephemeral**. The container is recreated from its base image each time the workspace starts, so anything installed or changed outside the persistent `/home/developer/` directory —for example, system packages installed with `apt` —does not survive a restart. Before you install tools or rely on system-level changes, read What persists in a SecurSpaces Workspace to choose an approach that persists.

A Workspace is defined by the following characteristics:

- **Basic Information**: such as name, owner, sharing options,
- **CPU/RAM/Storage**: performance allotted to the workspace.
- **Ports**: ports to run applications on,
- **Status**: i.e.running, deploying, or paused

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Where to go next

- Get to know the Workspaces page
- Create a Workspace
- Manage Workspaces
- Workspace Apps
- Use templates
- Develop in a Workspace
- SSH into your workspace

Related information

- Organizations
- Platform
- Projects
- Concepts

SecurSpaces Flex overview

September 23, 2026

Citrix SecurSpaces™ Flex is a Citrix-managed service within Citrix Platform Flex that delivers secure Linux workspaces for development teams. Delivered through Citrix Cloud, this service provides hosted developer environments on Azure with no customer infrastructure to stand up or maintain. Citrix runs the platform and the underlying Azure infrastructure; customers retain control of identity, workspace images, policies, and data.

Citrix SecurSpaces Flex includes:

- Citrix-managed hosting of developer workspaces on Azure, with a control plane built on Azure Kubernetes Service (AKS) and Azure Cosmos DB.
- Isolated, sandboxed Linux workspaces accessible via browser (PWA-based IDE or terminal), SSH from local IDEs such as VS Code, JetBrains Gateway, Cursor, or Windsurf, or a customer-built application experience.
- Workspace sizing by persona - Light, Medium, and Heavy - each mapped to a published Platform Flex credit rate.
- Citrix-managed workspace size tiers, so compute profiles stay aligned with the Flex service catalog.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Decoupled workspace size and workspace template, so a single template can be paired with any compute profile on demand.
- Native data loss prevention (DLP), controlled ingress and egress, role-based access control (RBAC), and full audit logging.
- Sandboxed execution environments for AI coding agents under the same control plane as human developer sessions.
- Compatibility with Citrix DaaS Flex, with both services drawing from a shared Platform Flex credit pool.

Citrix SecurSpaces Flex deployments are planned and provisioned by Citrix. To get started, contact your Citrix sales team or your Citrix SecurSpaces Solution Specialist.

Concepts and terminology

This section defines the terms that administrators frequently use in Citrix SecurSpaces Flex.

Deployment

A deployment is the provisioned instance of Citrix SecurSpaces Flex for your organization. It consists of a primary Azure region, a custom access URL, and an initial local administrator account. Citrix provisions and manages your deployment for you. Each organization has one deployment; additional Azure regions can be added as secondary regions after deployment.

Primary region

The primary region is the Azure region where the Citrix SecurSpaces Flex control plane and primary execution capacity are hosted. You agree the primary region with your Citrix team during implementation planning. The primary region cannot be changed after provisioning is complete.

Secondary regions

Secondary regions are additional Azure regions added to an existing deployment to extend workspace capacity, bringing capacity closer to distributed development teams. To add or remove a secondary region, contact your Citrix team.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Access URL

The access URL is the custom subdomain assigned to your organization's Citrix SecurSpaces Flex environment, in the format `https://<yourcompany>.flex.sds.citrix.com`. You choose the subdomain with your Citrix team during implementation planning. Developers and administrators use this URL to access the Citrix SecurSpaces environment.

Workspace

A workspace is an isolated, sandboxed Linux container that provides a secure development environment for a developer or AI agent. Each workspace runs on the Citrix SecurSpaces Flex execution plane with defined vCPU, RAM, and persistent storage. Persistent state is backed by Kubernetes Persistent Volume Claims (PVCs), so multi-project and multi-repository work survives between sessions. Workspaces can be accessed through a browser, SSH, or a supported local IDE.

Workspace template

A workspace template is a pre-configured, version-controlled definition that includes the container image, preinstalled tools, resource limits, access controls, and startup scripts. Templates are the primary unit of deployment in Citrix SecurSpaces Flex. The customer owns and hosts the container images; Citrix hosts and runs the platform that executes them. Because workspace size and workspace template are decoupled, a single template can be paired with any workspace size on demand. See Workspace templates for more information.

Workspace size

Workspace size is the compute profile assigned to a workspace. Each size maps to a defined vCPU and RAM configuration with a published Platform Flex credit rate. The available workspace sizes at launch are:

Size	vCPUs	RAM	Storage	Intended workload
Developer (Light)	2 vCPUs	8 GB	32 GB	General coding and routine development work
Developer (Medium)	4 vCPUs	16 GB	64 GB	Mixed workloads

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Size	vCPUs	RAM	Storage	Intended workload
Developer (Heavy)	8 vCPUs	32 GB	128 GB	Large builds, compilations, and demanding data workloads

Workspace size and workspace template are configured independently. Because they are decoupled, organizations can reuse a single template across workspaces of different sizes, and developers can switch compute profiles as workload demands change.

Citrix manages the workspace size definitions in Flex. Each size belongs to a managed tier and can have newer
versions over time. Administrators control where a tier is available, but they do not edit the CPU, RAM,
storage, or storage-class values that define the tier. When Citrix publishes a newer version of a tier, affected
workspaces can be updated to the current version of the same tier.

Platform Flex credits

Platform Flex credits are the consumption unit for Citrix Platform Flex services. Citrix SecurSpaces Flex workspaces consume credits based on workspace size and daily usage. The same credit pool covers other Platform Flex services, including Citrix DaaS Flex, so developers can use a DaaS Flex virtual desktop for general tasks and a Citrix SecurSpaces Flex workspace for development —both tracked through a single unified credit model with no separate procurement cycle. See Licensing for Citrix Platform Flex and Platform Flex credit consumption and reporting for more information.

Availability of Citrix Platform Flex services, and the service credit remedy if Citrix misses a commitment, are covered by the Platform Flex SLA.

Network connectivity

Citrix SecurSpaces Flex workspaces connect to customer-owned systems through a shared Azure virtual network (vNET) peering or VPN gateway between the Citrix-managed Azure tenant and the customer network. This integration layer enables workspace access to customer-managed identity providers, Git repositories, artifact repositories, CI/CD pipelines, data sources, and secrets managers. See Network connections for more information.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

All SecurSpaces Flex documentation

Flex articles sit in the section that matches what they are for, so they are not adjacent in the contents pane. This is the full set, along with the Platform Flex service level agreement, which is published with the
Citrix Platform Flex documentation.

Article	What it covers
Technical security overview	The security architecture of the Citrix-managed service: tenancy, data residency, encryption, and the shared responsibility split
Network connections	How Flex workspaces reach your identity provider, repositories, and other customer-owned systems
Workspace templates	The templates available in Flex, and what each one provides
What's new in SecurSpaces Flex	Release notes for the Flex service
Platform Flex SLA	The availability commitment for Platform Flex services, and the service credit remedy

Everything else in this documentation applies to Flex as well, except where a page says otherwise. Flex is a
deployment model for Citrix SecurSpaces™ rather than a separate product, so the developer, project, and
platform articles describe the same environment.

More information

- Licensing for Citrix Platform Flex
- Platform Flex credit consumption and reporting
- Platform Flex SLA
- Announcing Citrix SecurSpaces™ Flex: a different operating model for modern development needs —Executive blog
- Inside Citrix SecurSpaces™ Flex: a Citrix-hosted secure workspace platform for development teams —Technology blog

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SecurSpaces videos

This video playlist features tutorials and best practices for Citrix SecurSpaces™, demonstrating how it accelerates innovation with self-service, secure, scalable, ready-to-code Linux developer environments. Learn how these environments boost developer velocity while reducing infrastructure costs and security risks.

Testimonials

Hear from developers and IT leaders on how SecurSpaces is transforming their workflow, boosting productivity, and reducing risk.

Title	Description	Link
Why Citrix uses SecurSpaces	SecurSpaces has revolutionized the developer experience at Citrix. In this real-world testimonial, our teams share how SecurSpaces helped us move from slow onboarding and complex setups to fast, secure, cloud-native development environments. With instant access from any device, built-in security, and over 60% reduction in total cost of ownership, SecurSpaces empowers distributed teams to work faster and safer—without high-spec laptops or persistent VDI. Watch how we simplified secure development and accelerated innovation.	

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Demos

Watch these demos to see how SecurSpaces delivers secure, ready-to-code Linux environments that accelerate development and simplify workflow.

Title	Description	Link
SecurSpaces integration with Backstage	Discover how SecurSpaces integrates with Backstage to power a unified, self-service developer portal. In this video, we show how developers can switch contexts with one click using Backstage dashboards, provision secure workspaces instantly for new projects, and accelerate onboarding with pre-configured environments and tool stacks. This integration helps teams streamline workflows, stay compliant, and boost productivity, all from their central Backstage hub.	
SecurSpaces: Zero Trust Network Access, advanced data protection, loss-prevention	Discover how SecurSpaces provides a secure environment for developers to build and innovate without compromise. With Zero Trust Network Access (ZTNA), advanced data protection, and data loss prevention (DLP), your teams can collaborate confidently from anywhere. Combine innovation with security and keep your enterprise protected.	

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Title	Description	Link
SecurSpaces: Advanced data protection with Chrome Enterprise Premium	Learn how Chrome Enterprise Premium and SecurSpaces enhance data protection and security for modern enterprises. In this video, we explore advanced security features that help IT admins safeguard sensitive information, ensure compliance, and protect endpoints across your organization. Stay ahead with SecurSpaces and Chrome Enterprise Premium for a safer, smarter enterprise.	
SecurSpaces : One-click developer onboarding	SecurSpaces empowers development teams to self-provision secure and standardized environments on demand, accelerating the onboarding of developer environments for contractors and new hires in under one minute.	

Webinars

Watch our recorded webinars to discover how SecurSpaces empowers developers with secure, scalable, and ready-to-code Linux environments.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Title	Description	Link
DevSecOps automation and governance	Learn how Cloud Development Environments (CDEs) and containers accelerate DevOps in this webinar, including how to boost developer productivity, enhance DevSecOps, cut costs, and improve infrastructure efficiency.	
Achieving regulatory & security compliance across a development process	Enhance DevOps security & achieve ISO 27001 compliance with online containers. In this webinar, you'll learn to implement risk controls, automate access, and manage resources.	
Expanding the reach of platform engineering with Cloud Development Environments	Platform engineering boosts developer productivity with templated DevOps, enhancing automation, security, and compliance through Cloud Development Environments (CDEs).	

Get started

August 19, 2026

Start here if you are new to Citrix SecurSpaces™. Every page in this section is a tutorial: a single path

© 1997–2026 Citrix Systems, Inc. All rights reserved.

through a task, with a predictable result.

If you have been given access

Your organization already runs SecurSpaces and someone has added you to a project.

Do this	Page
Set up your profile, Git access, and credentials	Set up your account
Create a workspace and connect to it	Your first workspace
Run something end to end in a few minutes	Hello World workspace
Work through a realistic project	Develop, debug, and share a Node.js app

Once you are set up, Develop covers day-to-day work.

If you are setting up a project for your team

You run a project and your developers are waiting to use it.

Do this	Page
Prepare images, repositories, secrets, and a template, then add people	Set up a project for your team

Manage covers running the project after that.

If you are evaluating the platform

You need an environment of your own before anyone can use it.

Do this	Page
Stand up an evaluation environment on a single VM	Deploy an evaluation environment

For a production deployment, see Deploy.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- What SecurSpaces is
- Concepts

Set up your account

`Developer`

This guide walks you through configuring your profile, connecting your code repositories, and preparing your credentials in Citrix SecurSpaces™. Complete these steps once —they apply across all your workspaces and projects.

These steps match the onboarding checklist you'll see when you first log in. If you've already completed a step in the product, feel free to skip ahead.

Before you begin

Make sure you have the following:

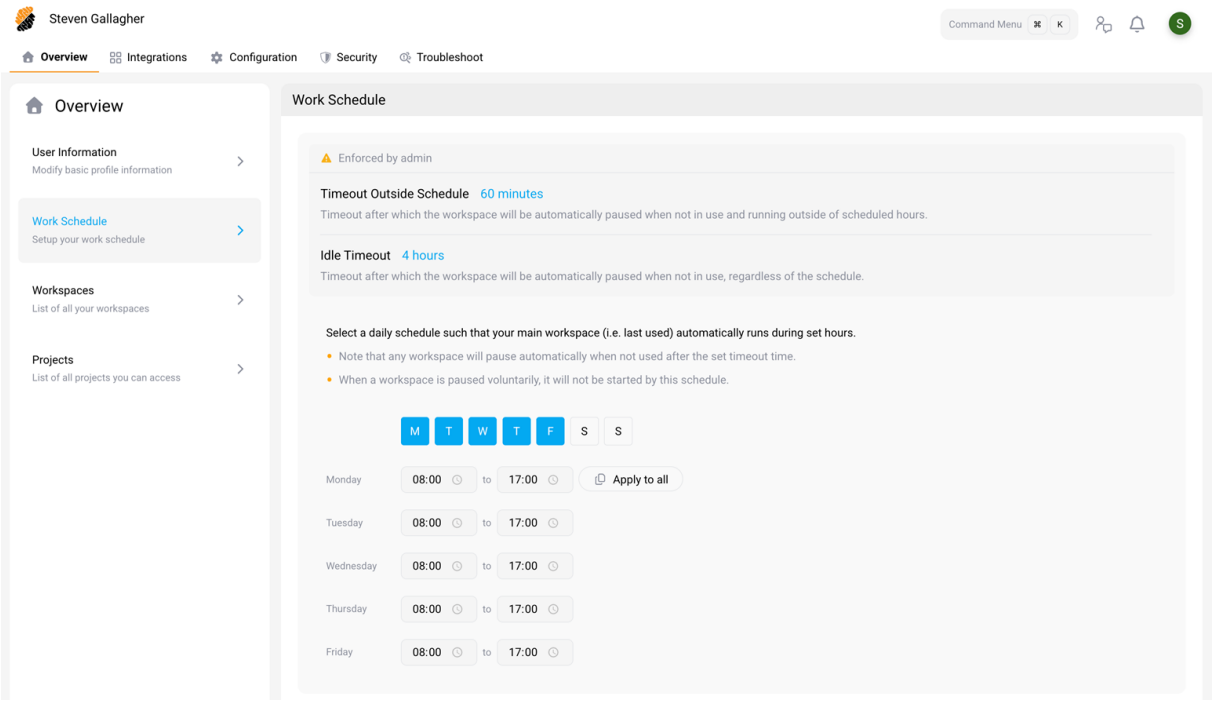
- **A SecurSpaces account.** Your project owner or platform administrator has added you to a project. You should have received a welcome email with your login URL.
- **Git provider credentials.** Your user name and password (or a personal access token) for GitHub, GitLab, Bitbucket, or Azure DevOps —whichever your team uses.

> **Returning to a new project?** If you've already completed onboarding on another SecurSpaces project, your profile and credentials carry over. You only need to verify your integrations work with the new project's repositories, then proceed to Quick start: Create and connect to your workspace.

Step 1: Review your profile

Your profile contains your personal information and work schedule. Most fields are imported from your organization's identity provider and should already be correct. Your time zone determines when scheduled actions (like automatic workspace start) run —if you work across time zones, set the one where you spend the majority of your working hours.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



How to complete this step

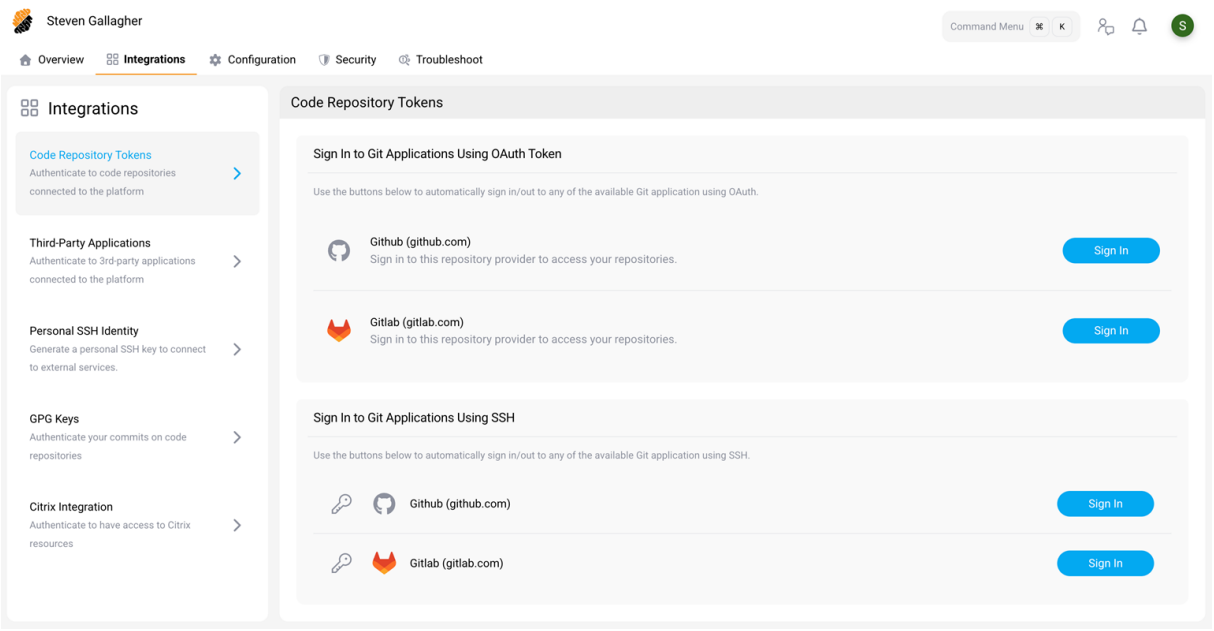
1. Log in to SecurSpaces and select your **avatar** in the top-right corner, then select **Profile**.
2. Under **User information**, review your name and time zone. Update them if needed.
3. Under **Work schedule**, adjust the days and hours to match your typical working pattern.
4. If everything looks correct, no changes are needed.

Learn more about your profile

Step 2: Configure integrations

Integrations connect your workspaces to code repositories and external services. You need at least one code repository token configured to push and pull code from inside a workspace. After you configure them, your integrations apply to **all** your workspaces, including existing ones—you don't need to reconfigure credentials per workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



How to complete this step

1. In your profile, navigate to the **Integrations** section.
2. Under **Code repository tokens**, authenticate with your repository provider:
    - **OAuth (recommended):** Select your provider (GitHub, GitLab, Bitbucket, or Azure DevOps) and complete the OAuth flow. This gives all your workspaces seamless access to your repositories without repeated log-ins —you authenticate once here, and every workspace you create connects automatically.
    - **Personal SSH key:** If your organization requires SSH-based authentication, upload your SSH public key or generate a new key pair.
3. Under **Personal SSH identity** (optional), generate an SSH key pair if you need to connect to external services over SSH —for example, a Bazel cache or artifact repository. Add the public key to the external service. SecurSpaces automatically applies the private key to your workspace sessions without exposing it to workspace code.
4. Under **GPG keys** (optional), generate or import a GPG key to sign your Git commits.

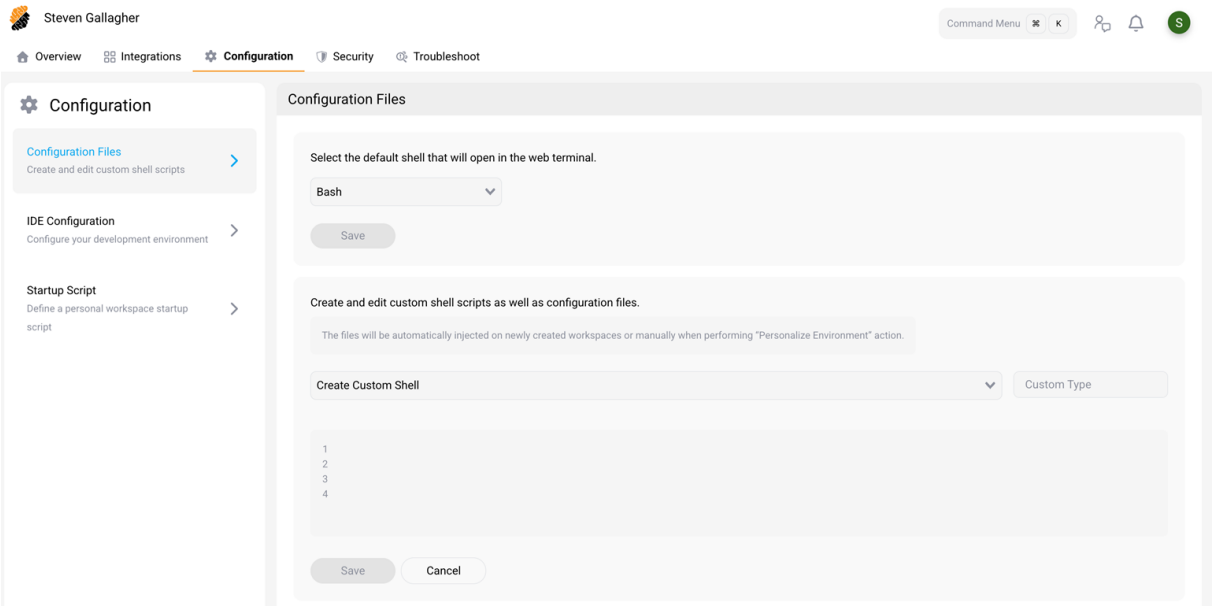
Learn more about integrations

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Step 3: Personalize your workspace configuration (optional)

Your workspaces work out of the box, but you can customize them with personal shell configurations, IDE settings, and startup scripts. These settings apply automatically to every new workspace you create. Configuration files are injected into `/home/developer` at workspace start, and your shell configuration runs in every new terminal session.

You can skip this step and come back later. Configuration changes can be applied to existing workspaces using the **Personalize environment** action.



How to complete this step

1. In your profile, navigate to the **Configuration** section.
2. Under **Configuration files:**
   - Set your **default shell** (bash or zsh).
   - Create custom shell scripts (for example, `.bashrc` or `.zshrc`) with your preferred aliases, environment variables, and tool settings.
3. Under **IDE configuration:**
   - If you already have a running workspace with your preferred VS Code or JetBrains settings, you can **import** those settings to use as your default across all workspaces.
4. Under **Startup script:**
   - Define a personal script that runs every time one of your workspaces starts. Use this to install personal CLI tools, set up directory structures, or run initialization commands. A script set on an individual workspace overrides this one. See Startup Scripts.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

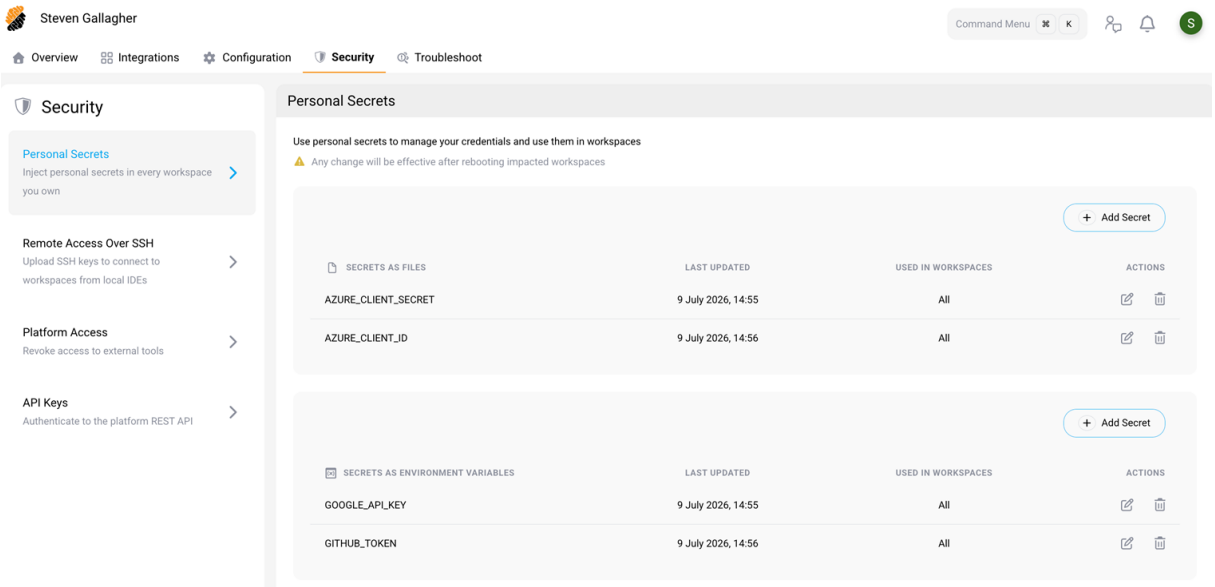
Tips

- **Start simple.** You can always add more customization later. A basic `.bashrc` with your favorite aliases is a good starting point.
- **Don't install project dependencies here.** Use workspace templates or startup scripts for project-specific tooling. Personal configuration is for your preferences (shell prompt, aliases, editor settings).
- **Test changes on one workspace first.** Use the **Personalize environment** action on an existing workspace before applying globally.

Learn more about workspace configuration

Step 4: Set up security credentials

Personal secrets, API keys, and SSH keys let your workspaces authenticate with services and APIs during development. Secrets are injected into your workspaces as environment variables or files, so you don't need to manage credentials manually inside each workspace. Your project owner or platform administrator may have also configured **project-level or platform-level secrets** that are automatically available —check with your team if you're unsure which credentials you need to add personally.



How to complete this step

1. In your profile, navigate to the **Security** section.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Under **Personal secrets**, add any credentials your workspaces need —for example, access tokens for internal APIs or package registries. For each secret, choose whether to inject it as a **file** or an **environment variable**.
3. Under **Remote access over SSH**, configure SSH keys if you plan to connect to your workspace from a locally installed IDE using SSH (for example, VS Code Desktop, JetBrains, Cursor, Windsurf, or Kiro).
4. Under **API keys**, add any API keys your development workflow requires (for example, keys for cloud services or third-party APIs).

Important security notes

- Secrets are stored encrypted and injected at workspace start. They are **not** visible in workspace code or logs.
- If you rotate a credential (for example, regenerate an API token), update it here —the new value applies to all workspaces on next start.
- Never hardcode secrets in your repository. Use the secrets mechanism instead.

Learn more about security credentials

Next steps

Your account is fully configured. You're ready to create your first workspace and start coding.

Quick start: Create and connect to your workspace

Your first workspace

`Developer`

This tutorial takes you from an empty SecurSpaces console to a running Citrix SecurSpaces™ workspace with
your project code in it, ready to work in.

By the end, you will have:

- Created a workspace from a template your project owner prepared
- Opened it in the Cloud IDE and run a command in it
- Confirmed your project repository was cloned
- Learned which of your changes survive a restart

It takes about 10 minutes. Work through the steps in order.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Before you begin

You need:

- **A completed account setup** —profile, integrations, and credentials. If you haven't done this, start
  with Set up your account.
- **A workspace template** for your project. Your project owner prepares these. If you don't know which
  template to use, ask them before you start.

How you'll connect

This tutorial uses the **Cloud IDE** (VS Code for Web) because it opens in a browser tab and needs nothing installed.

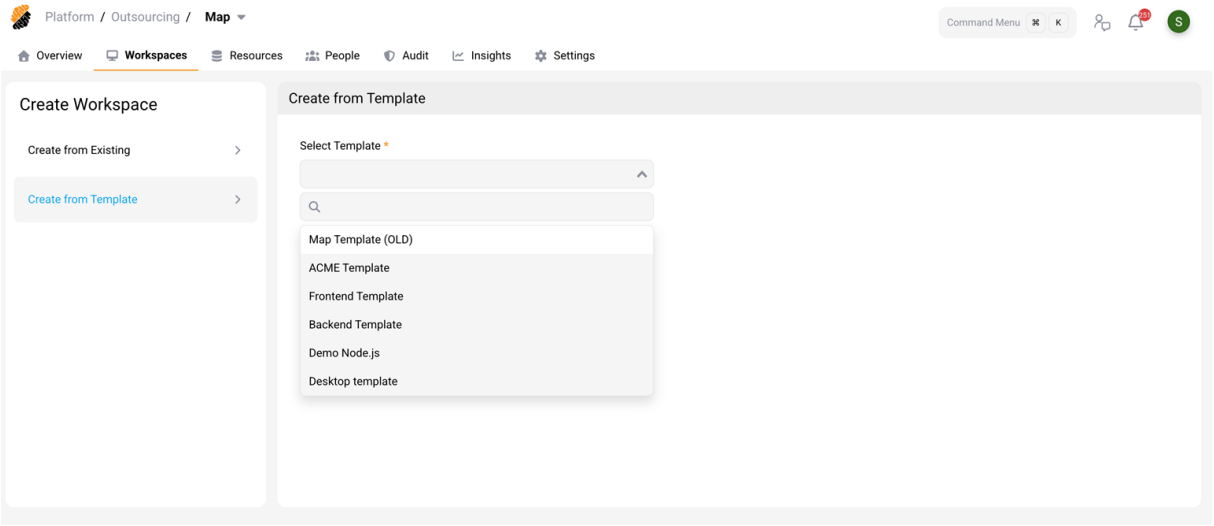
The Cloud IDE is not your only option, and for day-to-day work it may not be your preferred one. SecurSpaces
also supports connecting from a locally installed IDE over SSH, so your editor runs on your machine while the
code, tools, and compute stay in the workspace. Many developers prefer this. You'll set it up after you finish
here —see Other ways to connect at the end.

Step 1: Create a workspace from a template

A workspace is your personal cloud development environment: your code, tools, and dependencies, already
configured. Templates carry that configuration, so you don't assemble it yourself.

1. From the SecurSpaces home page, select **Create Workspace**, then **Create from Template**.
2. Select the template your project owner told you to use.
3. Review the configuration summary.
4. Select **Launch**.

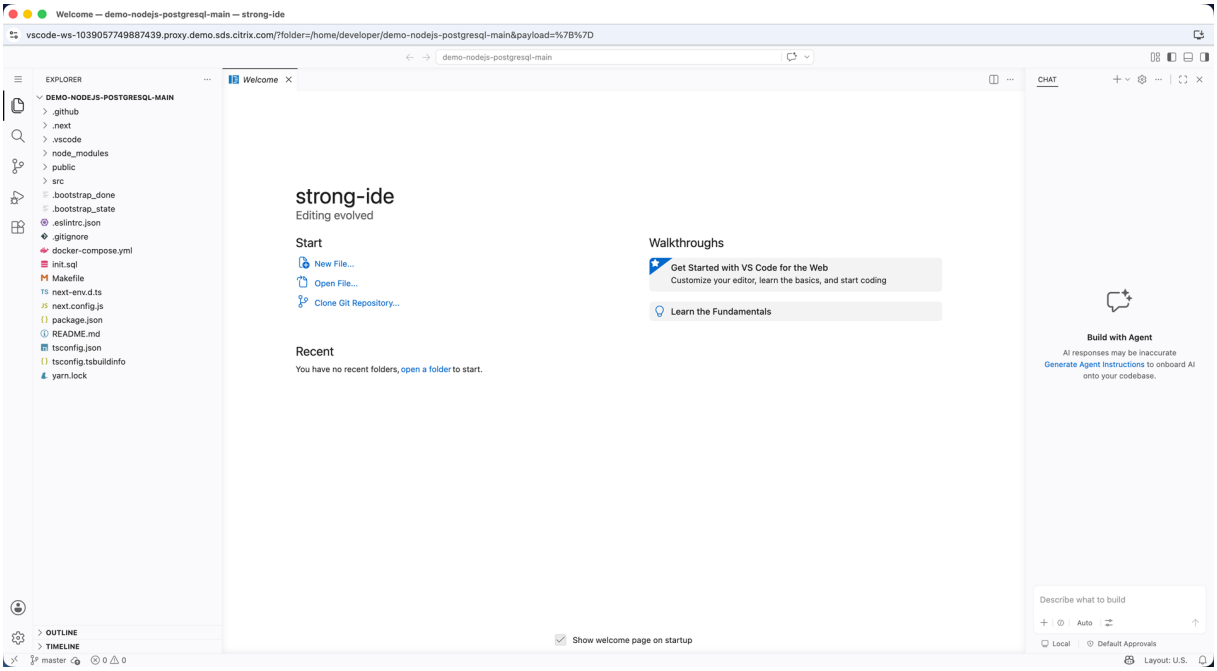
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Deployment takes about 30 to 90 seconds, and a progress indicator shows while it initializes. Wait until the
status shows **Running**.

Step 2: Open the Cloud IDE

1. On the workspace card, select **Open**.
2. VS Code for Web opens in a new browser tab, already connected to your workspace.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Open a terminal in the Cloud IDE and run:

```
whoami
```

It prints `developer`. You're running commands inside the workspace, not on your own machine — the browser tab
is only the window onto it.

Step 3: Check your project code

If your template includes repositories, they were cloned while the workspace started. Confirm that:

```
ls /home/developer
```

Your project folder appears in the list. Open it and check the repository is intact:

```
cd /home/developer/my-project
git status
```

Git reports the current branch and a clean working tree. Your credentials came from your profile integrations, so you didn't sign in to anything.

If your project folder isn't there, your template may not include a repository, or your Git credentials may
need attention. Ask your project owner which to expect, and see
If a repository is missing.

Step 4: Learn what persists

Workspaces are non-persistent by default, and this is the thing most likely to surprise you later. Create a
file in your home directory:

```
echo "notes" > /home/developer/scratch.txt
```

Anything under `/home/developer` —your code, this file, your `node_modules` —survives a pause or restart.
Anything outside it does not. A package you install system-wide with `apt` is gone when the workspace next
starts.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

That's the rule: **keep your work under `/home/developer`**. When you need a system-level tool permanently,
use a startup script or ask your project owner to add it to the template image. For more information, see
What persists in a SecurSpaces Workspace.

What you have learned

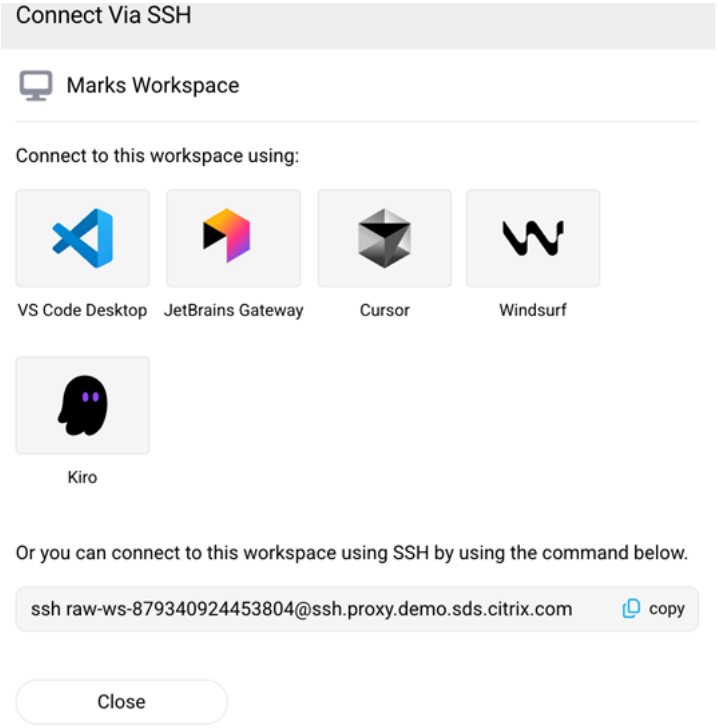
You have a running workspace with your project code in it, and you've run commands inside it from a browser.
Along the way you saw that the template supplied the environment, your profile supplied the credentials, and
`/home/developer` is the boundary between what persists and what doesn't.

Other ways to connect

The Cloud IDE got you started without installing anything. For longer sessions, most developers switch to
their own IDE, connected to the same workspace over SSH.

IDE	Connection method
VS Code Desktop, **Cursor**, **Windsurf**, **Kiro**	One-click, using the Citrix Secure Developer Spaces - Remote SSH extension
JetBrains Gateway	SSH configuration through the Gateway interface
Other SSH-capable editors	Manual SSH key pair

© 1997–2026 Citrix Systems, Inc. All rights reserved.



SSH has to be enabled for your project and on the workspace itself, and you need an SSH key pair in your
profile. For the prerequisites and the steps for each IDE, see
Connect to a workspace via SSH.

What to do next

Goal	Guide
Connect from your own IDE over SSH	Connect to a workspace via SSH
Practice the development loop on a self-contained app	Hello World workspace
Create a workspace without a template, and configure it yourself	Create a Workspace
Share a running application with your team	Workspace Apps
Understand pausing, resuming, and automatic start	Manage Workspaces

Get help

- **Ask your project owner** about templates, repositories, and project resources.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Contact your platform administrator** for account access, network, or platform-level issues.
- **Browse the SecurSpaces documentation** for detailed guides.

Hello World workspace

`Developer`

The Hello World workspace is a self-contained training environment that lets you explore Citrix SecurSpaces™ hands-on. It includes a fullstack demo application —no Git credentials or internet access required.

Use this workspace to follow along with the quick start guides, run your first onboarding session, or validate that SecurSpaces is working correctly in your environment.

Before you start

The Hello World container image does **not** ship with SecurSpaces, so it has to be added to a project before
anyone can create a workspace from it. A developer cannot do this themselves.

> **For project owners:** set up a dedicated Hello World playground project that every new developer can join.
> Add the image to that project's container resources once —either directly from Docker Hub (`strongnetwork`/`workspace-demo-multi-project`)
> or from a company-approved registry —and configure the five database secrets in Step 1 as project secrets.
> New developers can then start this guide without having to ask for anything. If your organization screens
> container images before use, involve your platform administrator early: this image is reviewed like any
> other, so allow time for it.

If you're a developer and the Hello World image isn't available when you create a workspace, ask your project
owner or platform administrator to set this up.

What's inside

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Component	Details
Application	"ACME Project"—a user management app (search, add, view, delete users)
Frontend	Next.js + React + TypeScript + Bootstrap
Backend	Next.js API Routes (REST endpoints)
Database	PostgreSQL 15 (runs in a Docker container inside the workspace)
Tooling	Node.js LTS, Yarn, Docker, ESLint, TypeScript
VS Code extensions	GitHub Copilot Chat (pre-installed). Language and tooling extensions (ESLint, Prettier, Docker, PostgreSQL Explorer) are **not** pre-installed — add them as needed.

Everything is pre-built and ready to run. The workspace functions without internet access or external accounts.

Step 1: Create a workspace

Confirm the Hello World image is available in your project. If it isn't, see Before you start.

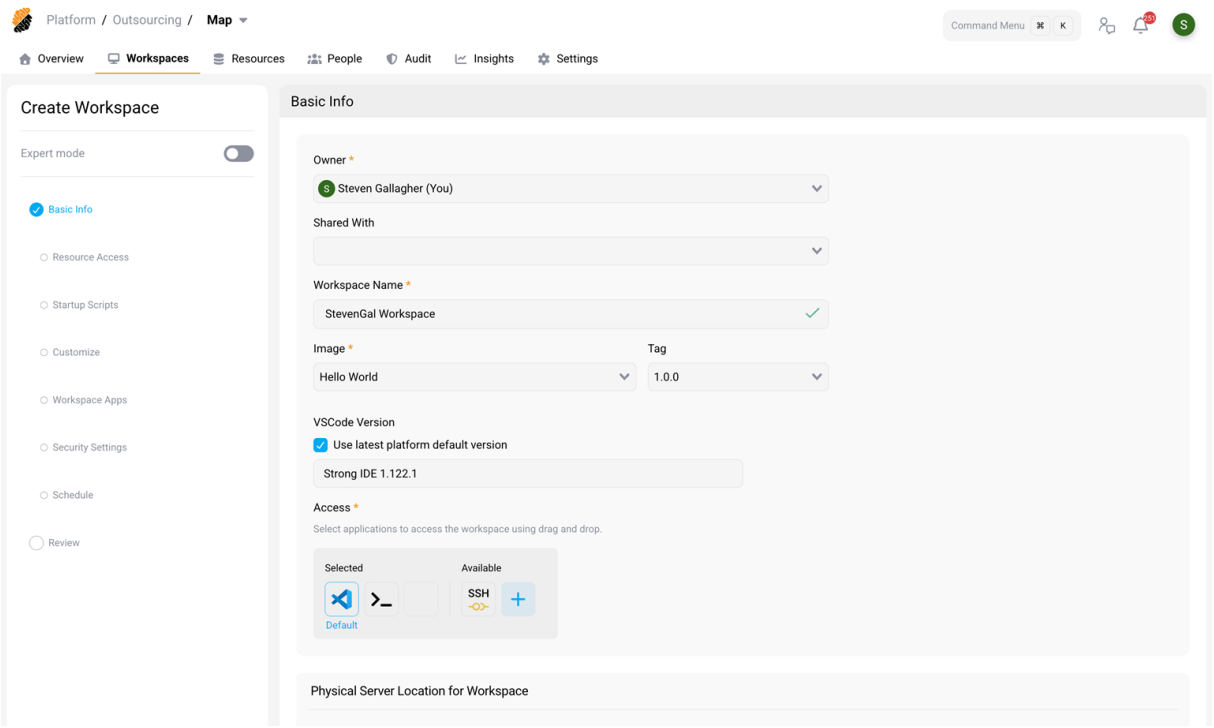
Create the workspace:

1. From the SecurSpaces overview page, select **Create workspace**.
2. Under **Basic Info**, select the **Hello World** container image and tag from the project's available images.
3. Under **Workspace specifications**, select a template with at least:
   - **CPU:** 2 cores
   - **RAM:** 8 GB
4. Navigate to the **Resource Access Control** step and under **Secrets**, add the following as environment variables:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Name	Value
DB_HOST	localhost
DB_PORT	5432
DB_NAME	project_acme
DB_USER	acme_user
DB_PASSWORD	acme_password

5. Select **Launch**.



Your workspace starts in under 30 seconds. Docker is provided automatically by the platform (through Docker-in-Docker), so no additional setup is required.

Step 2: Understand how secrets are configured

The Hello World app connects to PostgreSQL using the environment variables you added in Step 1. These are injected into your workspace automatically —no `.env` file needed.

Open a terminal and verify:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
echo $DB_USER    # acme_user
echo $DB_HOST    # localhost
```

The app reads these through `process.env` in its database connection code (`src/lib/db.ts`). `DB_PORT` is optional —it defaults to `5432` if omitted. As an alternative to the individual `DB_*` variables, you can supply a single `DATABASE_URL` connection string; if set, it takes precedence.

Personal vs. project secrets

Type	Set by	Where it applies	Use case
Personal secret	You, in **Profile > Security**	Every workspace you own, by default	Your own API keys, tokens, service credentials
Project secret	Project owner, in project resources	The project, then attached to workspaces	Shared database credentials, team API keys, service URLs

Neither type is all-or-nothing. You choose which workspaces a secret reaches, though the two work differently:

- **Personal secrets** inject into every workspace you own by default. To narrow that, switch the secret to
  Custom List and select the workspaces it applies to. A workspace you don't select doesn't receive the
  secret —useful when a credential belongs in your own project but not in a workspace you share with others.
- **Project secrets** are made available to the project, then attached to individual workspaces when the
  workspace is created or through the workspace template. A project secret reaches only the workspaces it's
  attached to.

Both types can be injected as an **environment variable** or as a **file**. File secrets appear in the `/secrets` folder; environment secrets show up in `env` output.

Three things worth knowing:

- **Changes need a restart.** Adding, editing, or retargeting a secret takes effect when the affected workspaces reboot, not immediately.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Names become environment variable names**, so they can't contain spaces, hyphens, or special characters,
  and names reserved by Linux are rejected.
- **Personal secrets win on a name clash.** If a workspace secret has the same name as one of your personal
  secrets, the value comes from your personal secret.

In this exercise, you added the secrets directly to the workspace. In a real project, your project owner would typically configure shared secrets at the project level so they're available to everyone automatically. You'd only add personal secrets (like your own API keys) through **Profile > Security > Personal secrets**.

Step 3: Start the application

Open the Cloud IDE (VS Code for Web) and open a terminal. Start the PostgreSQL database, then the dev server:

```
cd /home/developer/demo-nodejs-postgresql-main
docker-compose up -d    # start PostgreSQL (first run initializes the
   database)
yarn dev
```

VS Code detects the open port (3000) and shows a notification: **"A workspace application is available at the port 3000"**. Select **Preview** to open the app in VS Code's built-in browser.

You should see the ACME Project interface. Try adding a user and searching —this exercises the full stack (React to API to PostgreSQL).

You're ready

Your Hello World workspace is running and ready for development work. You've experienced:

- Creating a workspace from a container image
- Configuring secrets as environment variables
- Running a fullstack application with a database

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What to do next

Goal	Guide
Learn the everyday Node.js development loop in SecurSpaces	Tutorial: Develop, debug, and share a Node.js app —continues in this workspace
Set up your real project environment	Set up your account and Create and connect to your workspace
Understand SecurSpaces concepts and terminology	What is SecurSpaces?

Get help

- **Ask your project owner** about which template to use for your actual project.
- **Browse the SecurSpaces docs** for detailed guides.

Develop, debug, and share a Node.js app

`Developer`

This tutorial teaches you the everyday development loop in Citrix SecurSpaces™ by working through a real
application. You run it, change it, debug it on both the server and the browser, catch mistakes before they
reach the browser, and share the result with a teammate.

By the end, you will have:

- Started a PostgreSQL database and a Next.js application inside your workspace
- Changed the running application and watched it reload
- Inspected the same data through the browser, the API, and the database
- Paused server-side and browser-side code in a debugger
- Shared your running application with your project through a single URL

Every step builds on the one before it, so work through them in order. The whole tutorial takes about 30 minutes.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Before you begin

You need a running **Hello World** workspace, created as described in
Hello World workspace, with the five `DB_*`
secrets configured.

The Hello World container image is not available to every project by default. If you can't find it when you
create a workspace, ask your project owner to set up a Hello World playground project for your organization.
The Hello World workspace article tells them
what to add.

Open your Hello World workspace in the Cloud IDE (VS Code for Web) before you start Step 1.

Step 1: Check your environment

Confirm your workspace has the tools and secrets this tutorial depends on. Open a terminal in the Cloud IDE
and run:

```
node --version
yarn --version
```

Both commands print a version number. The exact versions don't matter for this tutorial.

Now confirm the database credentials reached your workspace as environment variables:

```
echo $DB_HOST
echo $DB_PORT
echo $DB_NAME
echo $DB_USER
echo $DB_PASSWORD
```

These print `localhost`, 5432, `project_acme`, `acme_user`, and `acme_password`.

If any of them print an empty line, the secrets weren't added when the workspace was created. Return to
Hello World workspace, Step 1, add the five
secrets, and restart the workspace before continuing.

Notice that you didn't create a `.env` file. SecurSpaces injects secrets into the workspace as environment
variables, so the application finds them without any local configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Step 2: Start the database

The application stores its users in PostgreSQL, which runs in a Docker container inside your workspace.
Start it before the application, or the first page load fails.

```
cd /home/developer/demo-nodejs-postgresql-main
docker-compose up -d
```

The first run downloads nothing —the image is already in the workspace —and initializes the database
schema from `init.sql`. It takes a few seconds.

Confirm the container is running:

```
docker ps
```

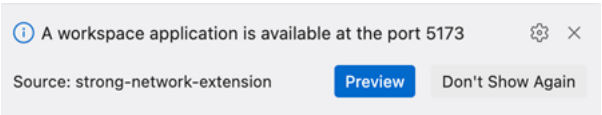
The output lists a container named `project-acme-db` with a status beginning with `Up`. If the list is empty, run `docker-compose up -d` again and read any error it prints.

Step 3: Run the application

With the database up, start the development server from the same directory:

```
yarn dev
```

VS Code detects that the application opened a port and shows a notification: **A workspace application is
available at the port 3000**. Select **Preview**.



The ACME Project user management application opens in an editor tab beside your code. The preview address
follows the pattern `https://<workspace-id>-port-3000.proxy.<domain>`, so the application is reachable
through the platform rather than from your own machine.

Select **Users** in the navigation bar, then add a user. The page saves it and shows it in the list.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

You have just exercised the whole stack: the React page called the API, the API wrote to PostgreSQL, and the
result came back to the browser. Keep this user —you look at it again in Step 5.

Leave `yarn dev` running.

Step 4: Make a change

Change the application while it runs, and watch the result appear without a restart.

1. In the Cloud IDE, open `src/app/page.tsx`.
2. Find the page heading and change its text. For example, change it to `Welcome to My ACME Project`.
3. Save the file.

The preview tab updates within a few seconds. You didn't restart the server, and you didn't rebuild anything —the development server detected the change and reloaded the page for you.

This is the loop you spend most of your day in: edit, save, look. It behaves exactly as it does on a local machine.

Step 5: Look inside the database

The user you added in Step 3 is a real row in a real database. Open a **second terminal** so the development
server keeps running in the first one, then connect to PostgreSQL:

```
1  docker exec -it project-acme-db psql -U acme_user -d project_acme
```

Notice that the prompt changes: you're now typing into PostgreSQL, not into the shell. List the users:

```
1  SELECT * FROM users;
```

Your user from Step 3 appears in the results, alongside the sample users the schema created. The browser
page, the API route, and this table are three views of the same data.

Leave PostgreSQL and return to the shell:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
\q
```

Step 6: Debug the API

Now pause the server-side code while a request runs through it.

The debugger needs to start the application itself, so first stop the development server you started in Step 3. Select the terminal running `yarn dev` and press **Ctrl+C**.

1. Open `src/app/api/users/route.ts`.
2. Find the `GET` function and select the gutter to the left of a line inside it. A red dot marks the breakpoint.
3. Open a new terminal and choose **JavaScript Debug Terminal** from the terminal type list.
4. In that terminal, start the application again:

   ```
   cd /home/developer/demo-nodejs-postgresql-main
   yarn dev
   ```

5. Open the preview and select **Users**.

Execution stops at your breakpoint and the Cloud IDE switches to the debugging view. Expand the `result`
variable and open `result.rows` to see the records PostgreSQL returned —including the user you added.

Select **Continue** to let the request finish. The Users page then renders as usual.

Notice that you opened the **Users** page rather than the home page. The home page is static and never calls
the API, so a breakpoint in the API route would never be reached from it.

Remove the breakpoint by selecting the red dot again, and leave the debug terminal running.

Step 7: Debug in the browser

Server-side code isn't the only place a problem hides. This step debugs code running in the browser, which
needs a different tool.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The application has a **Slow** switch that delays each API call by two to three seconds in the browser. It's
built for testing loading states.

1. Copy the preview address and open it in a **browser tab**. The built-in Simple Browser in the Cloud IDE
   has no developer tools, so this step needs a real tab.
2. Select the **Slow** switch in the navigation bar.
3. Open your browser's developer tools by pressing **F12**, and select the **Network** tab.
4. Reload the **Users** page. Each `/api/users` request now takes two to three seconds.
5. Select the **Fast** switch and reload again. The delay is gone.

To pause inside the delay itself, select the **Sources** tab, open `src/app/utils/delay.ts`, and set a
breakpoint inside `simulateDelay`. Reload the Users page. Execution pauses, and you can inspect `isSlowMode` and `slowDelay`.

You have now used both debuggers: the JavaScript Debug Terminal for code running on the server, and your
browser's developer tools for code running in the page.

Step 8: Catch errors before you run the code

Two checks find mistakes without starting the application at all.

First, introduce a type error. Open `src/app/api/users/route.ts`, change a `string` parameter to `number`,
and save. Then run:

```
yarn type-check
```

The command fails and names the file, the line, and the mismatch. Change the parameter back and run
`yarn type-check` again —it passes silently.

Now introduce a lint error. Add an unused variable to any `.ts` file, such as `const unused = 1;`, and save.
Then run:

```
yarn lint
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The unused variable is reported. Remove it and run `yarn lint` again to confirm the project is clean.

Both checks run against your code, not against a running server, so they catch mistakes in seconds.

Step 9: Share the running app

Finally, let someone else open your work without cloning anything or waiting for a deployment.

Make sure `yarn dev` is still running, then:

1. In the SecurSpaces console, open the **Project Overview** page.
2. From the **Workspace Apps** menu, select **Create Workspace App**.
3. Enter these values:
   - **Port:** `3000`
   - **Name:** `ACME Project`
   - **Share: Project Sharing**
4. Select **Save**.

The console shows a URL for the application. Copy it and send it to a project member. They open it in a
browser and use the application you are running, without a workspace of their own.

When you finish, delete the workspace app to stop sharing.

What you have learned

You ran a full-stack application inside a workspace, and everything behaved the way it does on a local machine—the same commands, the same debugger, the same browser tools. What changed is where the code runs
and where the data stays.

Along the way you:

- Confirmed that secrets arrive as environment variables, with no `.env` file to manage
- Started a database with Docker inside your workspace
- Used hot reload, a server-side debugger, and browser developer tools
- Ran type checking and linting
- Shared a running application through a URL instead of a deployment

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What to do next

Goal	Guide
Apply these workflows to your own project	Develop in a workspace
Understand what survives a workspace restart	What persists in a SecurSpaces Workspace
Set up your real project environment	Set up your account
Learn more about sharing a running application	Workspace Apps

Get help

- **Ask your project owner** about template contents, missing tools, or database access.
- **Browse the SecurSpaces documentation** for detailed guides.

Set up a project for your team

August 19, 2026

`Project Owner`

You have a project and a team waiting to use it. This page takes you through preparing that project so a
developer can create a workspace and start contributing, without configuring anything themselves.

Work through the steps in order. Each one builds on the last: the resources you register in steps 1 to 3 are
what you package into a template in step 4, and the people you add in step 5 are the ones who use it.

Before you start

You need a project, and a role that carries the following permissions. The default **Project Owner** role carries all of them. A **Manager** carries all except Security, which is only needed for private registry credentials.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Step	Permission
Review container images	Resources set to **Import**
Add registry credentials for a private registry	**Security** set to **Manage**
Import a code repository	**Resources** set to **Import**
Add a secret	**Resources** set to **Manage**
Create a workspace template	**Workspaces** set to **Manage Project**
Add users	**Members** set to **Manage**

Some of this may already be done for you. Images, repositories, and secrets can be registered at the platform
or organization level, and your project inherits them. Check what you already have before adding anything.

Step 1: Review container images

A container image defines the base environment a workspace runs: the operating system, the language runtimes,
and the tools your developers expect to find. Your project inherits the images registered at the platform and
organization levels, so start by checking whether one of those already suits your team.

1. Open **Resources** and review the container images available to the project.
2. If an inherited image works for your team, move on to step 2.
3. To add your own, select **Add Image** and provide the **URL**, **Image Name**, and **Image Tag**.

If the image sits in a private registry, add the registry credential first, otherwise the import cannot authenticate. Public registries need no credential.

See Container images, and
Workspace images for how toolchains are layered on
top of an image.

Step 2: Add a code repository

Registering a repository with the project is what allows the platform to inject credentials for it at runtime. Developers then clone and push without holding any credentials themselves, which is the main reason
their device never needs Git access to your source.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Repositories are imported from a provider your platform administrator has already connected — GitHub, GitLab,
Bitbucket, or Azure DevOps.

1. Open **Resources** and select **Import Repository**.
2. Select the provider.
3. Select the repositories to make available in this project.

If the provider you need is not offered, it has not been connected at the platform level yet. Ask your platform administrator to add it. See
Code repositories for
what that involves.

See Code repositories for the
project-level task.

Step 3: Configure secrets

A secret holds a password, token, or key that code needs at runtime. Registering it with the project means it
is delivered to the workspace rather than kept in the repository or pasted into a developer's shell.

Your project inherits secrets from the platform and organization levels, so review those first.

1. Open **Resources > Secrets** and review what the project already has.
2. To add one, select **Add a New Secret** and provide a name and value.
3. Add a description, so the next person knows what it is for.

Secrets are attached to a workspace when the workspace is created, or through the template you build in the
next step.

See Secrets.

Step 4: Create a workspace template

This is the step that turns the previous three into something a developer can use. A template combines an
image, repositories, secrets, and security settings into one definition, so creating a workspace becomes a
single action with a predictable result.

Without a template, every developer configures a workspace by hand, and no two are quite alike.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Open **Templates** and select **Create New Template**.
2. Choose a container image, then add the repositories, secrets, and security settings the team needs.
3. Save it as a draft.
4. Test the draft, then publish it as a template version.

A draft can be tested before anyone depends on it, and a published version cannot be edited afterwards, so
test before you publish.

Two things worth doing here:

- **Make more than one template if your team has more than one job.** Front-end and back-end developers
  rarely need the same tools, and a smaller image starts faster.
- **Generate a Quickstart link.** It creates a workspace from the template in a single click, which is useful from a README or an engineering portal.

See Templates.

Step 5: Add users

With the environment ready, bring in the people who use it.

1. Open **People** and select **Add New User**.
2. Enter the user's email address. The platform matches it against the configured identity provider.
3. Assign a role: **Guest**, **Developer**, **Manager**, or **Project Owner**.
4. For contractors or anyone temporary, select **Set Expiration date** and choose when their access ends.

Most developers should hold the **Developer** role, which lets them create their own workspaces from your
template. Assign **Guest** to someone who only needs to look, and create the workspace for them yourself.

If none of the four default roles fits, you can define your own. See
Roles and permissions.

See Add and remove users.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What your developers do next

Once a developer has been added and a template exists, they can create a workspace and connect to it. Point
them at Your first workspace.

If you would rather send them something written for onboarding, see
Onboard developers, which includes a
reusable template you can adapt.

Related information

- Onboard developers
- Manage
- Roles and permissions
- Project resources

Develop

August 19, 2026

This section is for **developers** working inside a Citrix SecurSpaces™ workspace. It covers connecting to a
workspace, running your code, sharing what you build, and the settings that are yours to control.

If you have not used the platform before, start with
Tutorial: Your first workspace.

Everyday work

Task	Page
Connect, install tools, run an app, and use secrets	Develop in a workspace
Find your way around the console	Console overview
Create, start, stop, and share workspaces	Workspaces
Connect from your own IDE or terminal	Connect over SSH

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Know before you lose work

Task	Page
Understand which files survive a restart	What persists in a workspace

Reaching other systems

Task	Page
Reach a server or service outside the platform	Connect to external services over SSH
Let a colleague open something you are running	Share a running application

Your settings

Task	Page
Manage your SSH keys, tokens, and integrations	Your profile

Less common setups

Task	Page
Keep Docker containers across restarts	Persistent Docker
Run a desktop environment in a workspace	Run a graphical desktop
Open the console in its own window	Install SecurSpaces as a desktop app
Work from an existing virtual desktop instead of a workspace	Work from a virtual desktop
Point an AI assistant at this documentation	Give your AI assistant access to the documentation

When something is wrong

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Task	Page
Work through common problems	Troubleshooting

Develop in a workspace

`Developer`

This article covers the everyday work of developing inside a Citrix SecurSpaces™ workspace: connecting,
installing tools, running and previewing an application, using secrets, and sharing what you build.

If you're new to SecurSpaces, work through
Tutorial: Your first workspace
first.

> **About the examples:** the examples here use Node.js, because it reads clearly whatever language you work
> in. Nothing on this page is Node-specific. The platform treats every toolchain the same way, so substitute
> your own commands—`pip install` for `npm install`, `go run` for `yarn dev`, and so on.

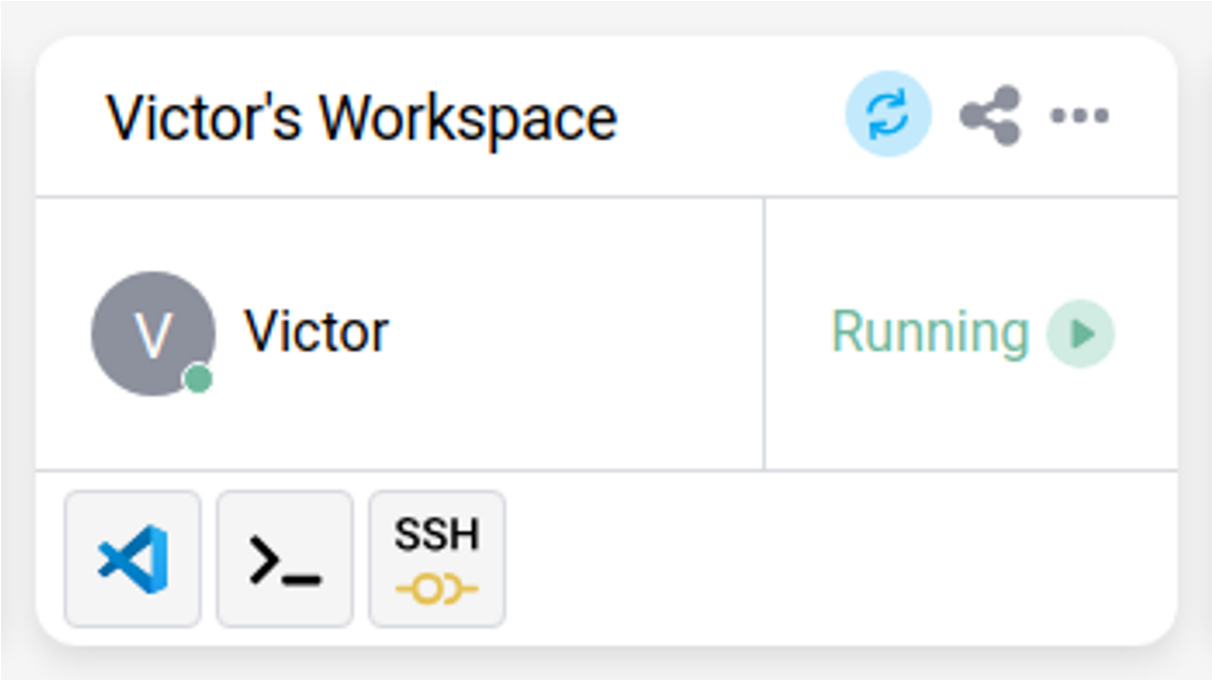
Choose how you connect

You can work in a workspace two ways.

Method	What it is	Best for
Cloud IDE	VS Code for Web, running in a browser tab	Getting started, any device, short sessions
Local IDE over SSH	Your own installed IDE, connected to the workspace	Day-to-day work, familiar setup and shortcuts

The Cloud IDE needs nothing installed: select **Open** on the workspace card and it opens in a new tab.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



The buttons on the workspace card are your entry points: the VS Code icon opens the Cloud IDE, the terminal
icon opens a shell, and the SSH icon opens the **Connect Via SSH** dialog.

VS Code for Web is the only Cloud IDE the platform ships. If your team needs a different browser-based IDE,
it can be built into a custom workspace image —talk to your platform administrator. For local IDEs, SecurSpaces supports VS Code Desktop, Cursor, Windsurf, Kiro, and JetBrains Gateway, plus any SSH-capable
editor. See Connect to a workspace via SSH.

Move files in and out of a workspace

The simplest way to bring a local file in is to drag it from your file manager onto the Cloud IDE window. To take a file out, use the Cloud IDE's **Download** action in the file explorer context menu.

File transfer may be restricted on your platform, since it's a data loss prevention control. If dragging a
file has no effect, ask your platform administrator whether transfers are permitted for your project.

Install tools and dependencies

A workspace is containerized and **ephemeral**: it is recreated from its base image on every start. Tools you

© 1997–2026 Citrix Systems, Inc. All rights reserved.

install at runtime persist only if they live on the persistent `/home/developer/` volume. System-wide
installs (for example, `sudo apt-get install`) are discarded when the workspace restarts.

Before installing anything, choose an approach that matches what you need to keep:

- **User-scoped tools** (`nvm`, `pyenv`, `pipx`, `cargo install`, and similar) install under `/home/developer/` and persist automatically.
- **Per-launch setup or one-off packages** belong in a startup script.
- **System packages and runtimes shared across a team** belong in a custom container image.

Project dependencies installed into your project directory persist, as long as the project sits under `/home/developer`. Package managers that install globally by default need redirecting into your home
directory. For example, with npm:

```
npm config set prefix ~/.npm-global
export PATH=~/.npm-global/bin:$PATH
```

Add the `PATH` line to your `.bashrc` or `.zshrc` file through your profile configuration to make it permanent.

For the full guidance and trade-offs, see
What persists in a SecurSpaces Workspace.

Run and preview an application

Start your application the way you normally would:

```
cd /home/developer/my-project
yarn dev
```

Your server must listen on `0.0.0.0`, not `localhost`. A server bound to `localhost` is reachable only
from inside the container, so the platform can't detect it and you can't preview it. Most frameworks bind
correctly in development mode, but some need telling:

```
npx vite --host 0.0.0.0
npx next dev -H 0.0.0.0
```

The same applies in any language —for example, Flask needs `--host=0.0.0.0`, and a Go server should listen
on `:8080` rather than `127.0.0.1:8080`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

When the server starts, VS Code detects the open port and shows a notification: **A workspace application is
available at the port [number]**. Select **Preview** to open the application in an editor tab.

The preview address follows the pattern `https://<workspace-id>-port-<number>.proxy.<domain>`, so traffic
reaches your application through the platform. You can paste that address into any browser tab. The **Workspace Apps** panel in the SecurSpaces console lists every exposed port.

Running several services at once works as expected: a frontend on port 3000 and an API on port 4000 are both
detected and both reachable.

Run services your project depends on

Workspaces support Docker-in-Docker, so databases, caches, and message queues run inside the workspace just
as they would on your own machine. Define them with Docker Compose:

```
# docker-compose.yml
version: '3.8'
services:
  postgres:
    image: postgres:15
    environment:
      POSTGRES_DB: myapp
      POSTGRES_USER: dev
      POSTGRES_PASSWORD: devpass
    ports:
      - "5432:5432"
```

Start, check, and stop them:

```
docker-compose up -d
docker ps
docker-compose down
```

Your application reaches these services through `localhost`, the same as in local development.

Some images provide the Docker Compose plugin rather than the standalone command. In that case use
`docker compose` with a space; the subcommands are identical.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Use secrets and environment variables

Credentials, tokens, and service addresses reach your workspace as secrets, injected at start. You don't maintain a `.env` file for them, and they never enter your code or your Git history.

Type	Set by	Where it applies
Personal secret	You, in **Profile > Security**	Every workspace you own, by default
Project secret	Project owner, in project resources	The project, then attached to workspaces

Neither type is all-or-nothing:

- **Personal secrets** inject into every workspace you own by default. To narrow that, switch the secret to **Custom List** and select the workspaces it applies to.
- **Project secrets** are attached to individual workspaces when the workspace is created, or through the workspace template.

Both types inject as an **environment variable** or as a **file**. File secrets appear in the `/secrets` folder; environment secrets appear in `env` output and in your language's environment API:

```
1  const apiKey = process.env.MY_API_KEY;
```

Three things worth knowing:

- **Changes need a restart.** Adding, editing, or retargeting a secret takes effect when the affected workspaces reboot.
- **Names become environment variable names**, so they can't contain spaces, hyphens, or special characters, and names reserved by Linux are rejected.
- **Personal secrets win on a name clash.** If a workspace secret has the same name as one of your personal secrets, the value comes from your personal secret.

For more information, see Secrets.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Share a running application

Workspace Apps expose a running application to other project members, so a reviewer, designer, or tester can
open your work without a deployment or a workspace of their own.

1. In the SecurSpaces console, open the **Project Overview** page.
2. From the **Workspace Apps** menu, select **Create Workspace App**.
3. Set the **Port** your application runs on, a **Name**, and who it's shared with —**Public**, **Project Sharing**, or specific members.
4. Select **Save**.

You can also select the **…** icon on your workspace and select **Edit Ports**.

For more information, see Workspace Apps.

Work with a shared workspace

After sharing a workspace, you and
another user can work in it at the same time. Changes appear in real time, so you can co-edit a file together
—useful for pairing on a problem or walking someone through unfamiliar code.

Both of you are working in the same container, so a command one of you runs affects the other's session too.

Persist your work

Keep everything you care about under `/home/developer`. That includes your code, your dependencies, and your
shell configuration. Anything outside it is discarded when the workspace restarts.

For system-level tools you need every time, use a startup script or ask your project owner to add them to the
workspace template image. See
What persists in a SecurSpaces Workspace.

Troubleshooting

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Issue	Solution
A tool is missing from the workspace	The template may be the wrong one. Ask your project owner which to use.
Packages disappear after a restart	Move your project and its dependency directory under `/home/developer`.
The application isn't reachable	Bind the server to `0.0.0.0`, then check the exposed port in **Workspace Apps**.
Calls to a database or service fail	Confirm the container runs with `docker ps`, then start it with `docker-compose up -d`.
Docker containers don't start	Run `docker info`. If Docker is unavailable, ask your project owner about privileged mode.
File changes don't trigger a reload	Raise the file watcher limit with `sudo sysctl -w fs.inotify.max_user_watches=524288`.
`No space left on device`	Delete unused dependency directories, or increase the workspace disk size under **Edit Workspace > Resources > Disk size**.
Dragging a file into the IDE does nothing	File transfer may be restricted. Ask your platform administrator.

Related information

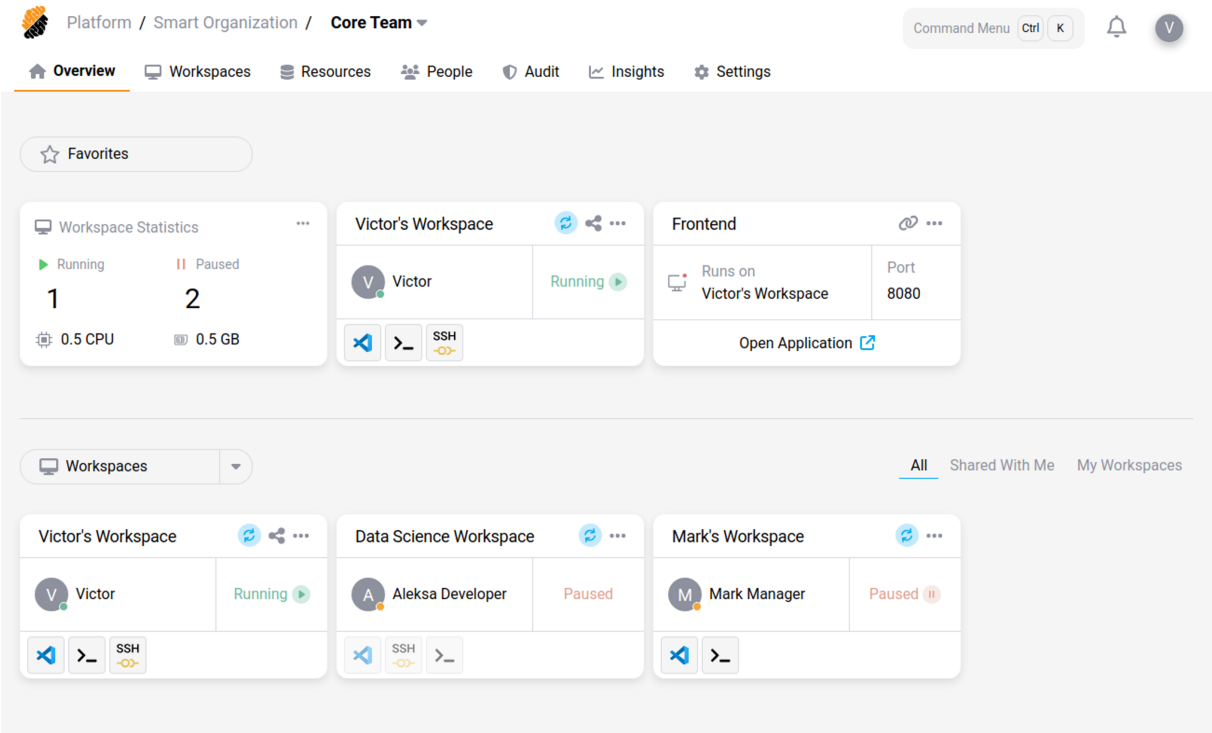
- Tutorial: Your first workspace
- Connect to a workspace via SSH
- What persists in a SecurSpaces Workspace
- Manage Workspaces
- Workspace Apps

Console overview

The Overview page is the first page displayed when you access the platform's user interface. It contains the essential components to allow quick access to resources such as workspaces, apps, secure web apps and metrics.

The **Overview Page** is customizable. All components can be reordered according to your preferences.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- Display Sections
  - Favorites
  - Workspaces
  - Workspace Apps
  - People & Metrics

Command Menu

The **Command Menu** is a keyboard-driven way to move around the console. It is in the header on every page,
and opens with **Ctrl+K**, or **Cmd+K** on macOS. You can also select it.

Type into **What do you want to do?** to filter, or browse the three groups:

Group	What it offers
Navigation	Go to Organizations, Projects, or Project Settings
Profile	Go to your profile
Workspaces	Open one of your workspaces by name

Opening a workspace by name is the most useful of these, because it does not depend on finding the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

workspace
in a list first.

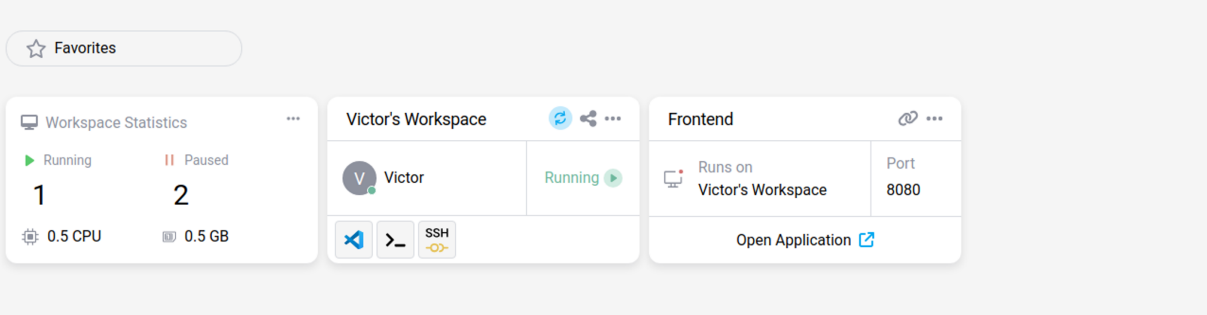
The menu is hierarchical, and the keys are shown along the bottom: **Enter** selects, the **arrow keys** move, **Backspace** goes back up a level, and **Esc** closes it.

> **Note**
>
> This is not the same as the Command Palette inside a workspace IDE, which runs commands against the
> workspace itself. See
> Connect over SSH.

Display Sections

Favorites

The **Favorites** section displays your personal favorite list of components, from any section of the **Overview Page**.



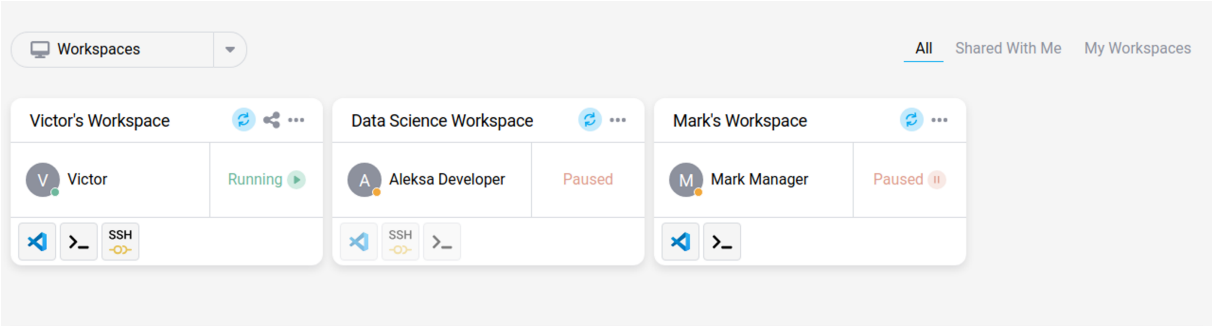
- To **add an element** to your list, click its "**…**"button and "**Add to Favorites**".
- To **remove an element** from your list, click its "**…**"button and "**Remove from Favorites**".

> Tip:
>
> Entries in the list of favorite components can only be components on the **Overview Page**.

Workspaces

The **Workspaces** section displays all the project's Workspaces to which you have access.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

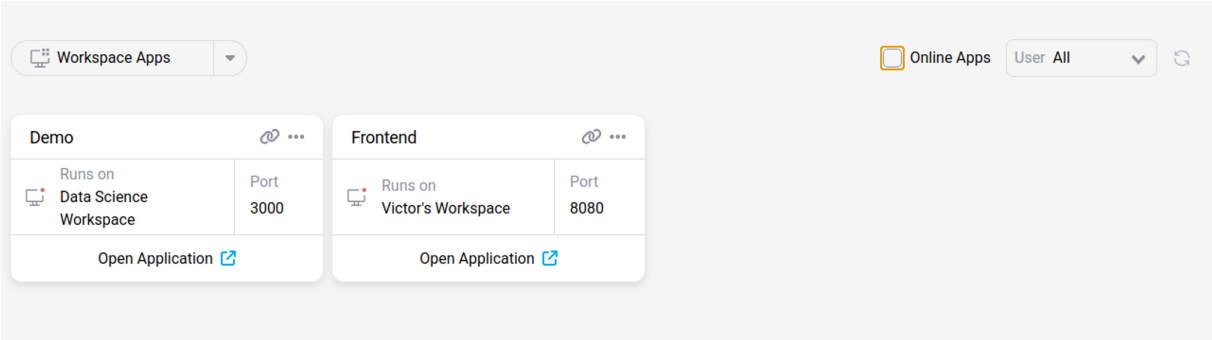


Filter the list with **All**, **Shared With Me**, or **My Workspaces**.

- To create a new workspace click on the "**Workspaces**"drop-down menu.
- To manage workspaces, view **Manage Workspaces**.

Workspace Apps

The **Workspace Apps** section displays all the project's workspace apps to which you have access.



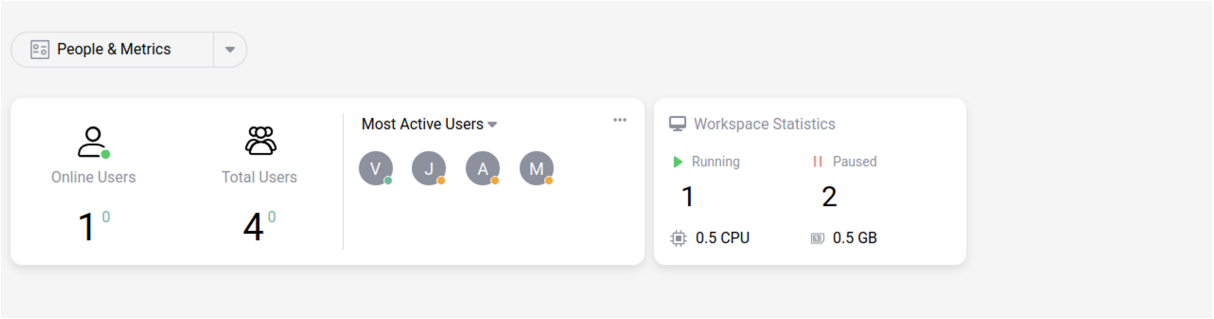
Turn on **Online Apps** to show only apps that are currently running, and use the **User** filter to show the apps of one user instead of **All**.

- To create a new workspace app click the "**Workspace Apps**"drop-down menu.
- To manage a workspace app click its "**...**"button.

People & Metrics

The **People & Metrics** section displays statistics about the users in the project and metrics about resources'utilization.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



People metrics display:

- The amount of project users online.
- The total amount of project users.
- Statistics about the amount of users online over the past seven days.

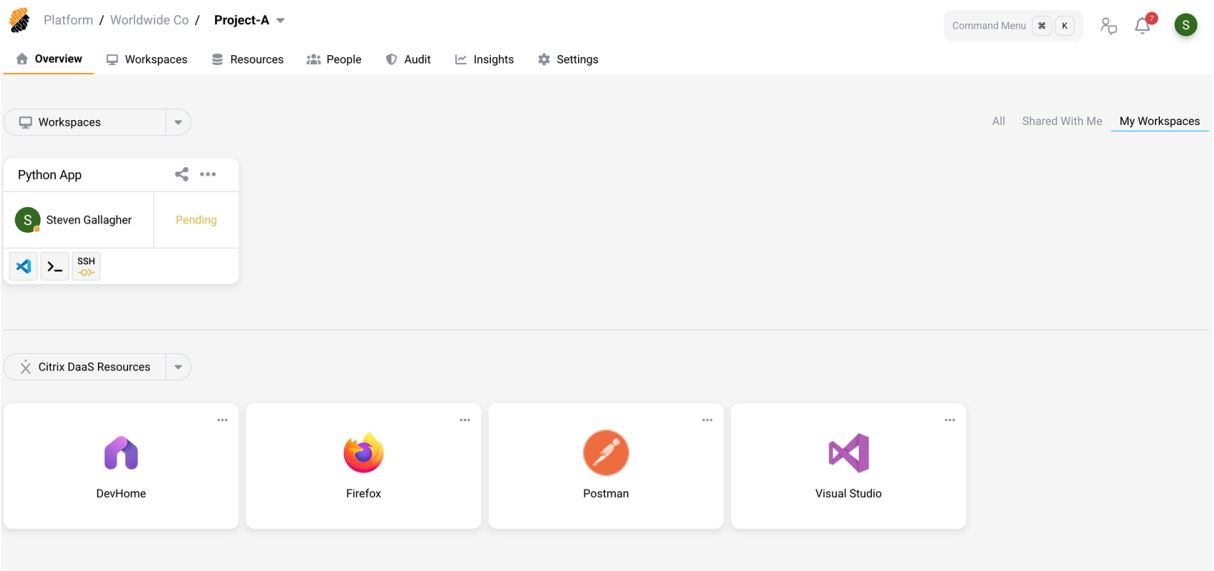
Workspace metrics displays:

- How many workspaces are running or paused.
- The current total CPU and RAM usage for your Project.

Check the Insights Page for more detailed metrics.

Citrix DaaS Resources

The project Overview page now includes a **Citrix DaaS Resources** section alongside the existing Workspaces and Workspace Apps sections. This section displays the Citrix virtual apps and desktops assigned to you in Citrix DaaS, giving you unified access to both Kubernetes-based Citrix SecurSpaces™ workspaces and Citrix-based resources from a single console.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Prerequisites

Before Citrix DaaS resources appear on the Overview page:

1. A platform administrator must configure the Citrix DaaS integration. See Citrix DaaS Integration.
2. You must sign in to your Citrix account from **Profile > Integrations > Citrix Integration**. See Citrix Integration.
3. You must select the Citrix resources you want to access. See Select DaaS resources.

View and launch Citrix resources

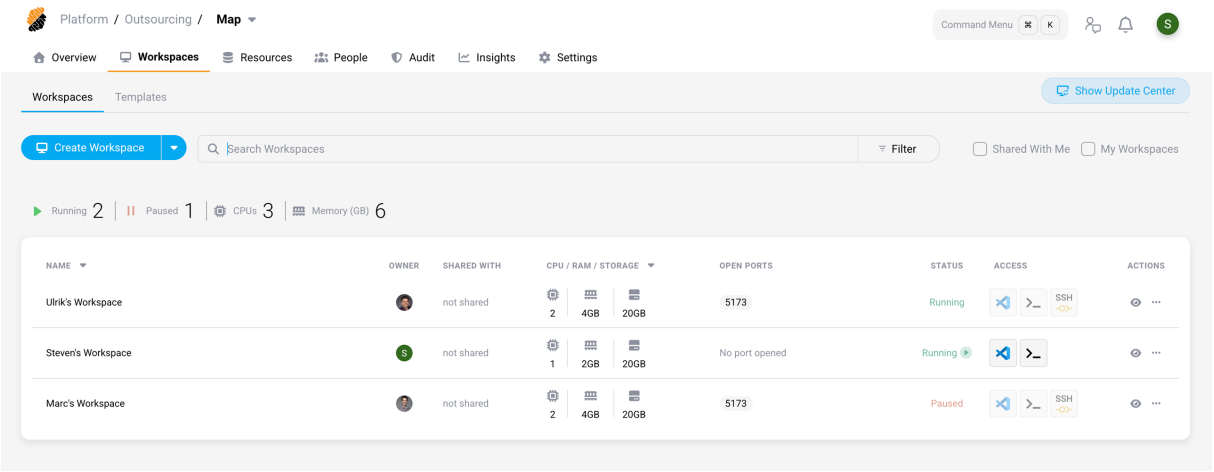
When the prerequisites are met, the **Citrix DaaS Resources** section on the project Overview page displays the virtual apps and desktops you selected in your profile. Select a resource to launch it directly from the SecurSpaces console.

If the section displays "No Citrix DaaS Resources," verify that:

- You have signed in to your Citrix account in **Profile > Integrations > Citrix Integration**.
- You have selected at least one resource using the **Edit** button in the **Selected DaaS Resources** section of your profile.
- An administrator configures the platform Citrix DaaS integration.

Workspaces page

In the scope of a project, the **Workspaces Page** displays all workspaces created for that particular project to which you have access or you can view, depending on
your permission level.
This includes personal workspaces and the workspaces shared with you. In some cases, it also includes Workspace's Templates available in the project.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Searching and Filtering Workspaces

In projects with a large number of Workspaces, it may be necessary to locate specific Workspaces or filter them based on certain properties.

Search

Use the **search bar** at the top of the screen to find Workspaces by:

- Workspace name
- Owner name
- Workspace ID

Below the search bar, you can view:

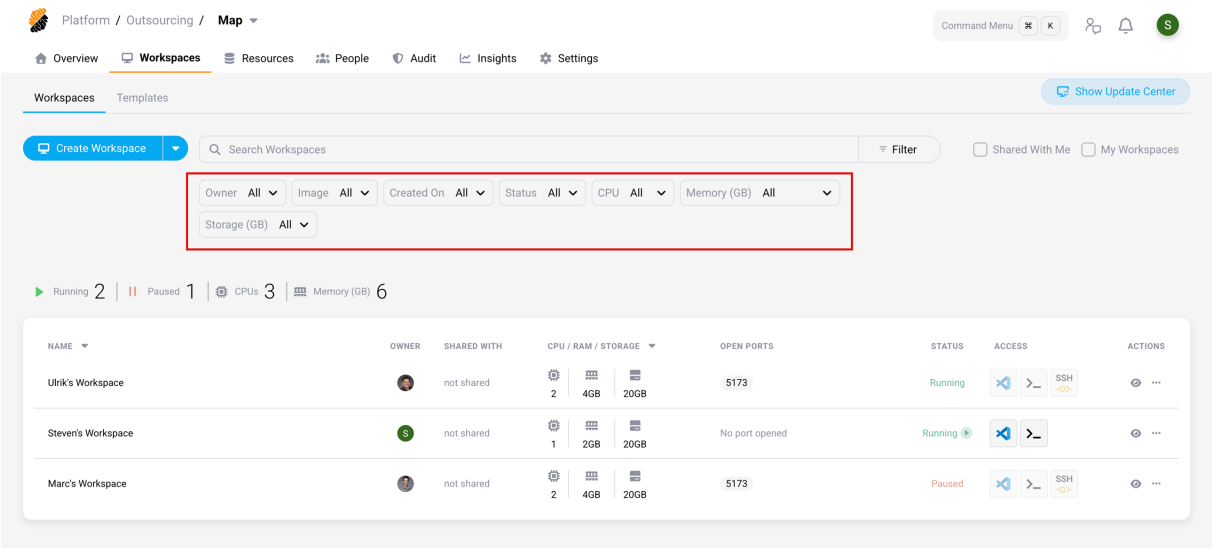
- The number of running and paused Workspaces
- Total CPU usage
- Total memory usage (in GB)

Filter

To filter workspaces by specific properties, select the **Filter** icon located to the right of the search bar. Available filter options include:

- Owner
- Base image
- Date of creation
- Workspace status
- CPU resources allocated
- Memory resources allocated
- Disk space allocated

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Where to go next

- Create a Workspace
- Manage Workspaces
- Create and manage Workspace Apps
- Create and manage templates

Create a workspace

August 27, 2026

A workspace is a cloud development environment that you reach through the browser IDE, a terminal, or SSH. Citrix SecurSpaces™ creates one from the Workspaces page.

There are three routes. Starting from a template is the recommended one, because the template already carries the image, resources, and policy your project has agreed.

Basic Set-Up

Select **Create Workspace** on the Workspaces page or the project Overview page. A **Create a Workspace** dialog opens and asks how you want to create it:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Option	What it does
Create from Template (Recommended)	Start from a predefined, ready-to-code template. See From a template
Create Custom Workspace	Configure a new workspace from scratch, described below
Copy Existing Workspace	Duplicate the configuration of one of your workspaces. See From an existing workspace

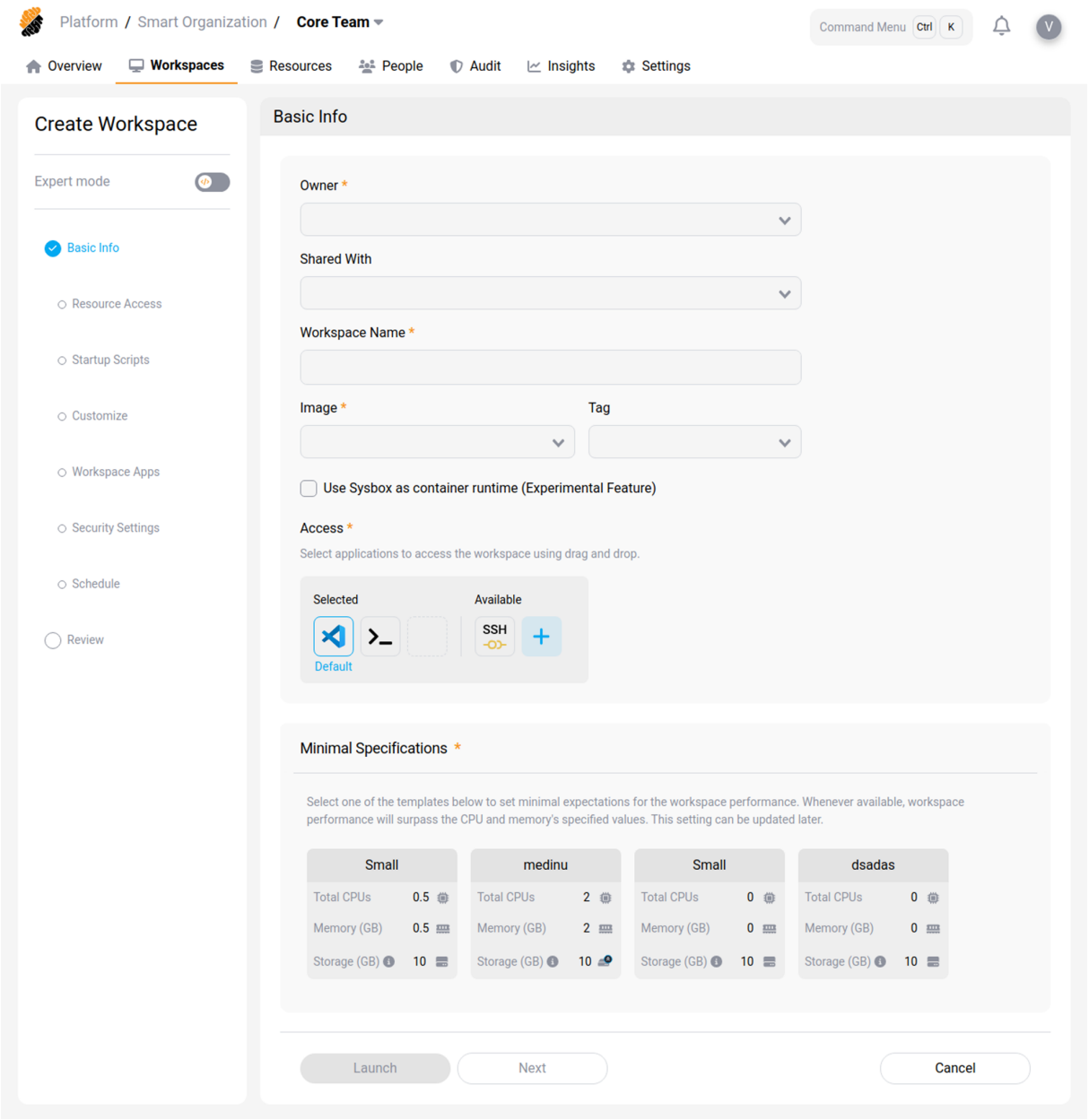
An option can appear disabled for two reasons. **Create from Template** is unavailable when the project has
no templates yet. The other two are unavailable when your role requires templates for workspace creation —
hover to see which applies, and contact your project owner if you need access.

Selecting an option opens the corresponding wizard.

Basic Info

The first step of the wizard holds the core settings: who owns the workspace, what it runs, where it runs,
and how you connect to it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



> **Tip:**
>
> Toggle **Expert mode** at the top of the sidebar to edit the workspace configuration directly as YAML.

Setting	What it does	Changeable later
Owner	The user who owns the workspace and has full control over its configuration	Yes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	What it does	Changeable later
Shared with	Project members who can access the workspace alongside the owner, for peer coding and review. Shared users get the same access as the owner	Yes
Workspace name	A unique name for the workspace	No
Container image	A preconfigured image carrying the tools and dependencies your project needs. Your project owner or administrator maintains the list	Yes, on restart
Tag	The image version to use	Yes, on restart
VSCode Version	Which IDE version runs in the workspace. See VSCode Version	Yes
Access methods	Which connection methods the workspace offers: **VS Code** for the browser IDE, **Terminal** for the browser terminal, and **SSH**. Each one runs as a service in the workspace. Select **Add** to define a custom access item	Yes
Workspace region	Where the workspace is deployed, which affects latency and data residency. Choose a region near your repositories and package registries; if you and your resources differ, prioritize the one closer to you	No
Workspace specifications	The compute resource template. Performance can exceed these values when capacity is available. Select the disk icon on the specification card to increase storage	Yes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note:**
>
> The available specification templates and their values depend on how your platform administrator has
> configured the environment.

Changing the image or tag on a running workspace takes effect only after the workspace restarts, because the
container is recreated from the new image. The console tells you this and asks you to confirm before it
restarts the workspace. Files under `/home/developer` are not affected. See
What persists in a workspace.

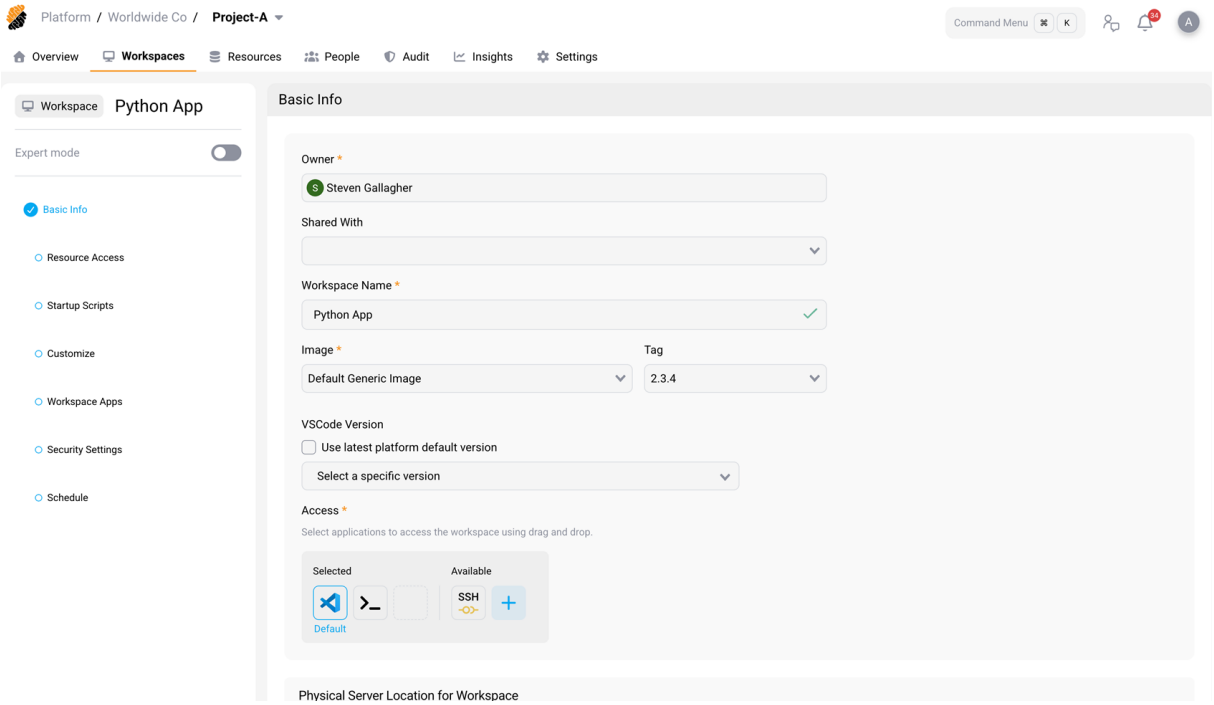
When the basic settings are right, select **Launch** to create the workspace immediately, or **Next** to continue through the remaining steps.

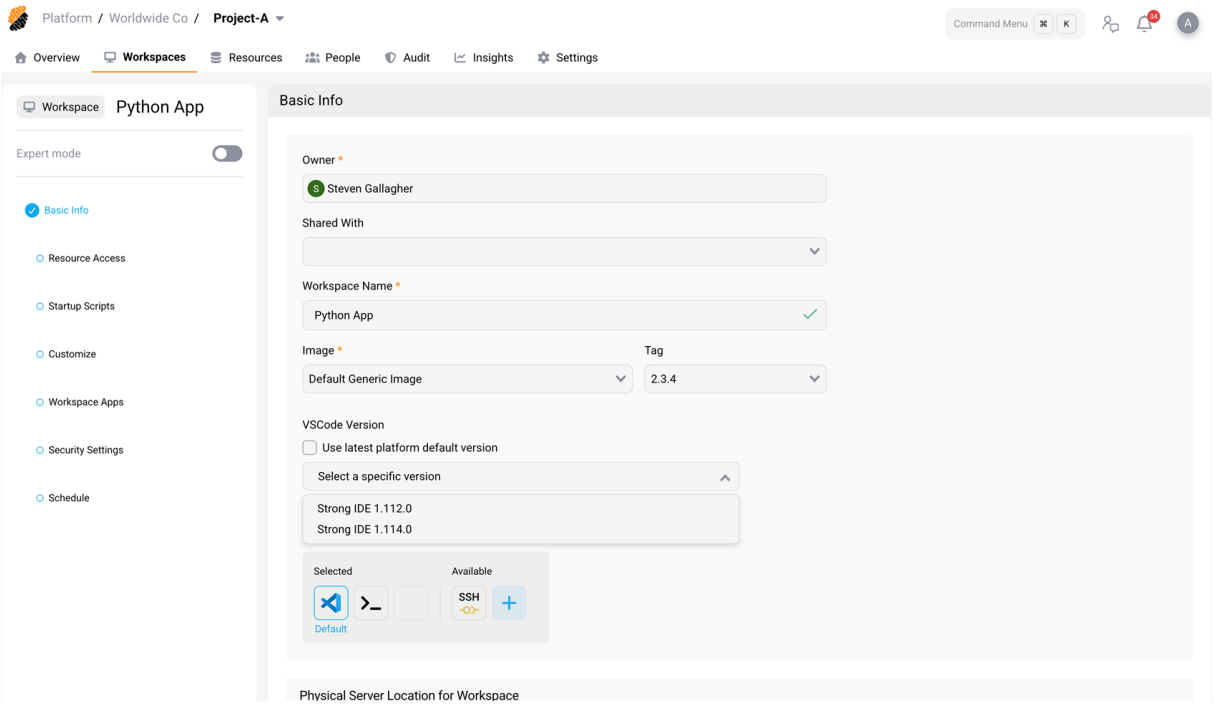
VSCode Version

By default, select **Use latest platform default version** and the workspace follows the IDE version designated as the platform default. When that default changes, the workspace picks up the new one on its
next start.

Clear the checkbox to enable **Select a specific version** and pin the workspace to one version. The list shows the IDE versions imported into your scope, most recently imported first. A pinned workspace is not
updated when the platform default changes.

© 1997–2026 Citrix Systems, Inc. All rights reserved.





Situation	What happens
No versions have been imported	The checkbox and list are unavailable, and the workspace reports that no VSCode versions are available

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Situation	What happens
No version selected, checkbox cleared	The workspace uses the built-in IDE image shipped with the installed platform release
You change the version on an existing workspace	The change takes effect on the next start. The home directory (`/home/developer`) is not affected

Available versions depend on what has been imported into the platform's registry. See VSCode Versions.

> **Note:**
>
> This setting is also available when you edit an existing workspace, and when you create or edit a workspace template. A template stores the setting, so workspaces created from it inherit either the pinned
> version or the automatic update behavior.

Custom Access Items

A custom access item gives one-click access to a service or tool running inside the workspace. Use one when
the built-in access methods do not cover a development tool, internal service, or application entry point.

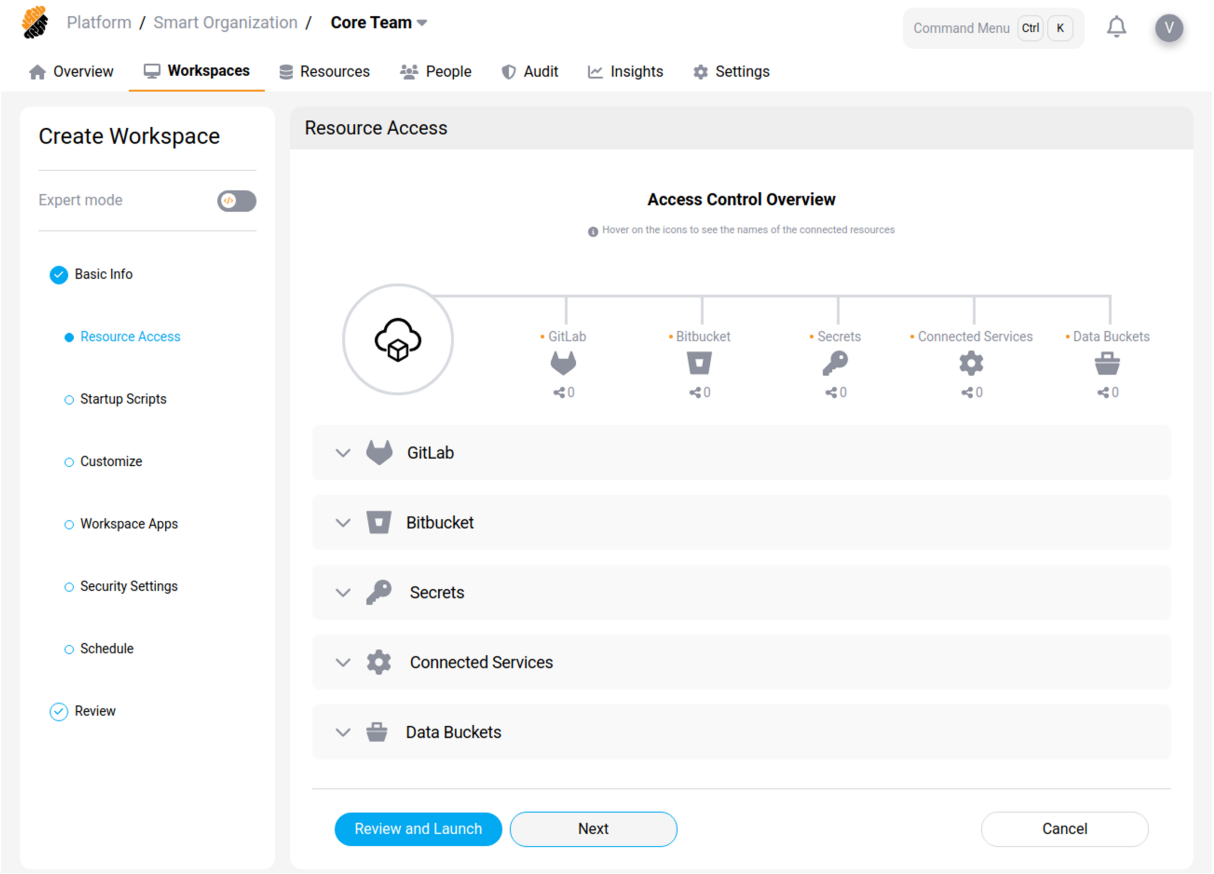
Field	What it holds
Name	The label shown to workspace users
Port	The port the service listens on inside the workspace
Command	The command that starts or connects to the service
Icon	An optional image for the access item
Health probe	An optional command that checks whether the item is online

Custom access items can also be managed through the REST API, for both workspaces and workspace templates.
See REST API.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Resource Access Control

Attach project resources to the workspace. A resource must already exist in the project or organization, and you might need access rights to use it.



Resource	What attaching it does
Git applications and repositories	Connects a whole Git application, or single repositories already imported into the project or organization. You can set the folder in the workspace that the files clone into
Secrets	Makes secrets available as files or environment variables. Choose an existing secret or create a new one
Connected HTTP and SSH services	Makes a connected service available as environment variables. Which services are available depends on the project, the organization, and the platform configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If a repository is missing

Repositories attached here are cloned while the workspace starts. If a repository is missing after the workspace reaches **Running**, work through these checks in order:

1. **Check the workspace configuration.** Select the workspace and review its Resource Access Control
   settings. Confirm the repository you expect is listed, and note the folder it clones into.
2. **Check your Git provider credentials.** Confirm you have authenticated with the right provider and
   account under your profile integrations.
3. **Try cloning manually** to find out which of the two is at fault:

   ```
   1 git clone <your-repository-url>
   ```

   A permission error points to your credentials. A successful clone points to the workspace configuration —
   contact your project owner.

Startup Scripts

The container image provides the fixed tools —languages, compilers, system packages. A startup script
handles what changes between users, branches, or runs, and would not belong in a shared image.

A startup script runs every time the workspace starts, either pre-startup or post-startup. Common uses:

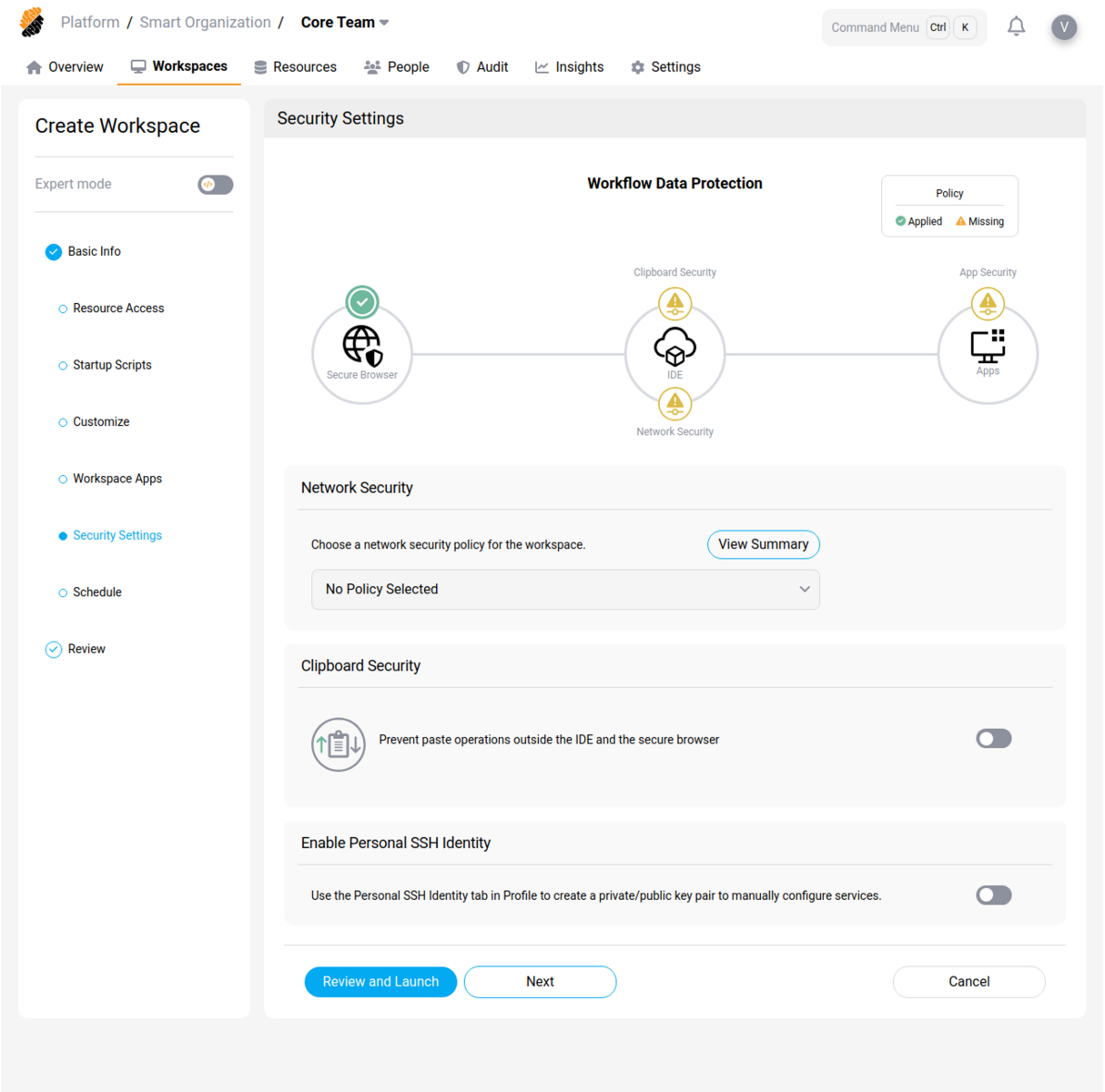
Use	Examples
Install dependencies that change often	`npm install`, `apt update`, or building binaries so they match the current branch
Start services the runtime does not manage	Local PostgreSQL, Redis, or MongoDB instances, and background processes such as file watchers or test runners
Report readiness	Health checks that confirm services are up before handing over the terminal, or a message listing available commands

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note:**
>
> Because the script runs on every start, write it so that running it more than once is safe.

Data Loss Prevention

Requires the Security permission set to **Manage**.



Security Settings control what the workspace can reach and what can leave it:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	What it does
Network Policy	Enforces a network policy on the workspace, controlling outbound traffic. Policies are project or organization resources, defined by a user with **Security** set to **Manage**
Clipboard Security	Prevents pasting outside the IDE and the Secure Browser
Apps Security	Restricts Workspace Apps to the Secure Browser

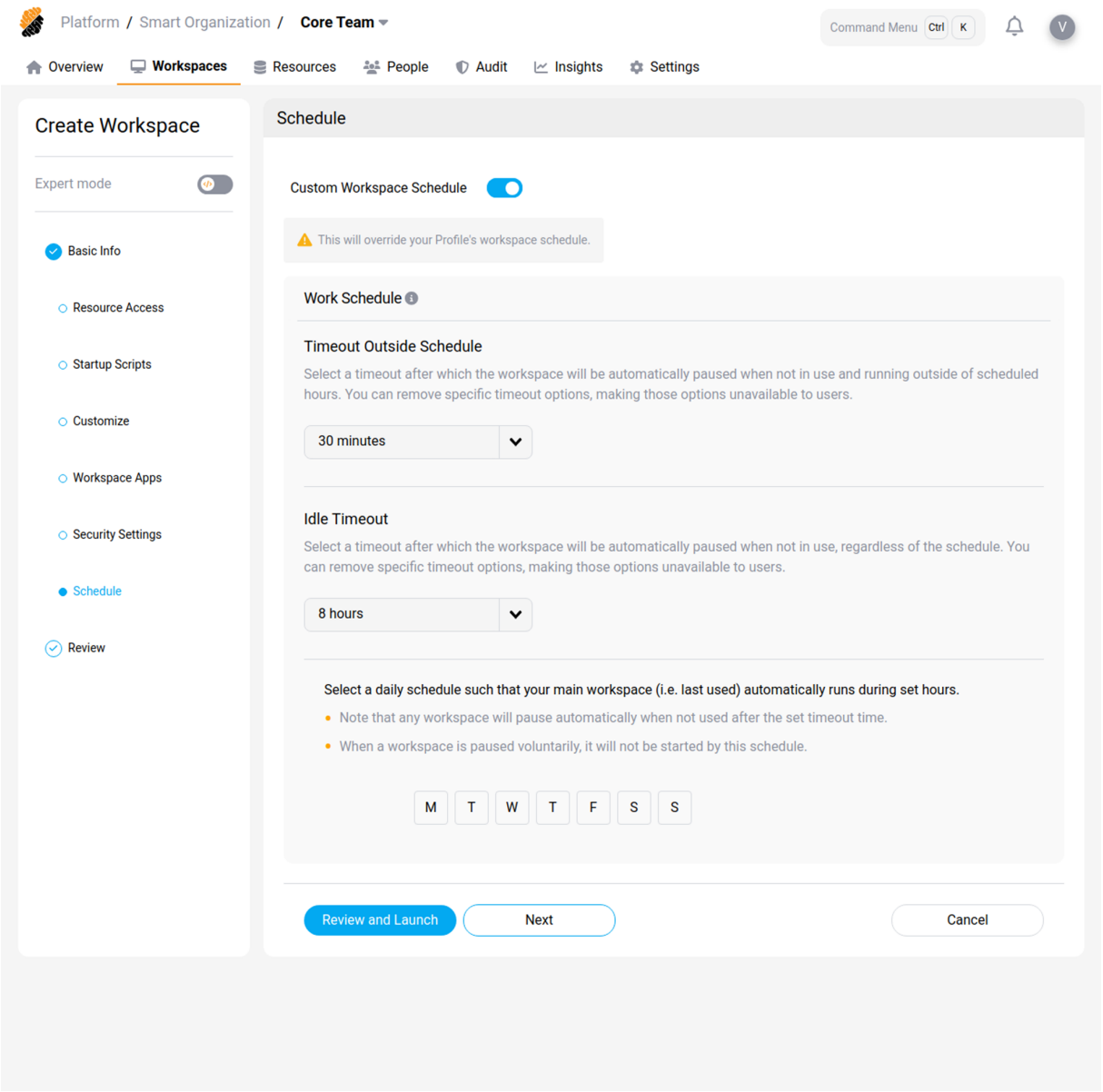
Secure Access Management controls how developers connect:

Setting	What it does
Remote Development Over SSH	Allows SSH connections to the workspace
Enable Personal SSH Identity	Allows users to use their personal SSH identity from inside the workspace

Custom Work Schedule

Define a custom work schedule for the workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Launch it

Review the configuration and launch. The workspace is deployed automatically. You can edit its configuration at any time from the Overview or Workspaces pages.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

From an existing workspace

Select **Create Workspace**, then **Copy Existing Workspace**. Provide the workspace to copy and the owner
for the new workspace, then select **Launch**.

> **Tip:**
>
> Select **Customize** to edit the copy as if you were creating it from scratch.

From a template

Select **Create Workspace**, then **Create from Template**. This is the recommended route: a template carries the image, resources, credentials, and policy your project has agreed, so the workspace is consistent with everyone else's.

Provide the template name and the owner for
the workspace, then select **Launch**.

> **Tip:**
>
> Select **Customize** to edit the workspace as if you were creating it from scratch.

Related information

- Connect over SSH
- Manage workspaces
- Workspaces page
- VSCode Versions

Manage workspaces

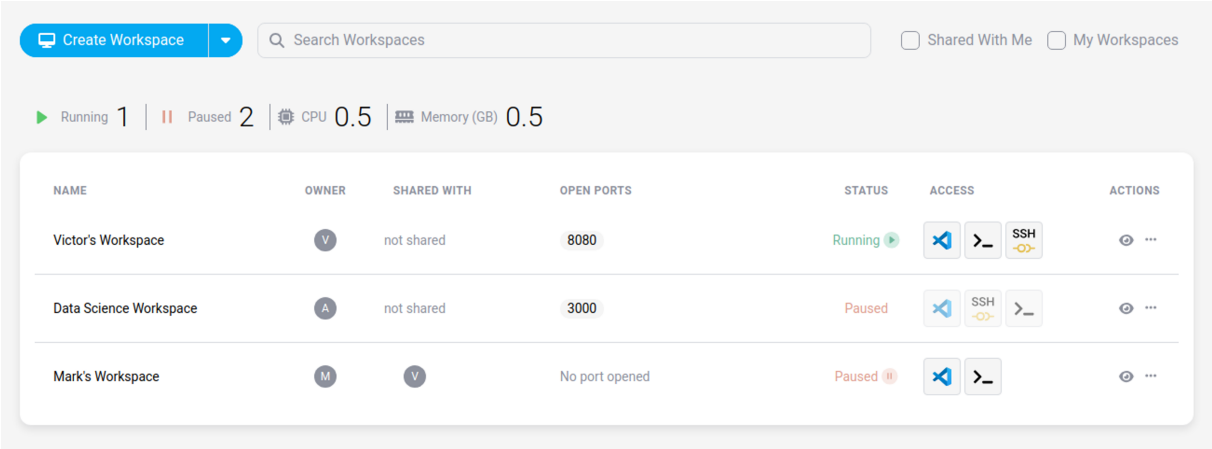
Workspaces are managed from the Overview and Workspaces pages. Once one or more workspaces have been assigned to you, they appear on both pages mentioned above. The last used workspace will be automatically started based on the schedule in your profile. In addition, a workspace might be paused automatically based on the settings of your platform after a period of inactivity.

View Workspaces

The list of your workspaces (owned by you or shared with you) is displayed on the Overview and Workspaces pages.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The status of the workspace is displayed next to its name.



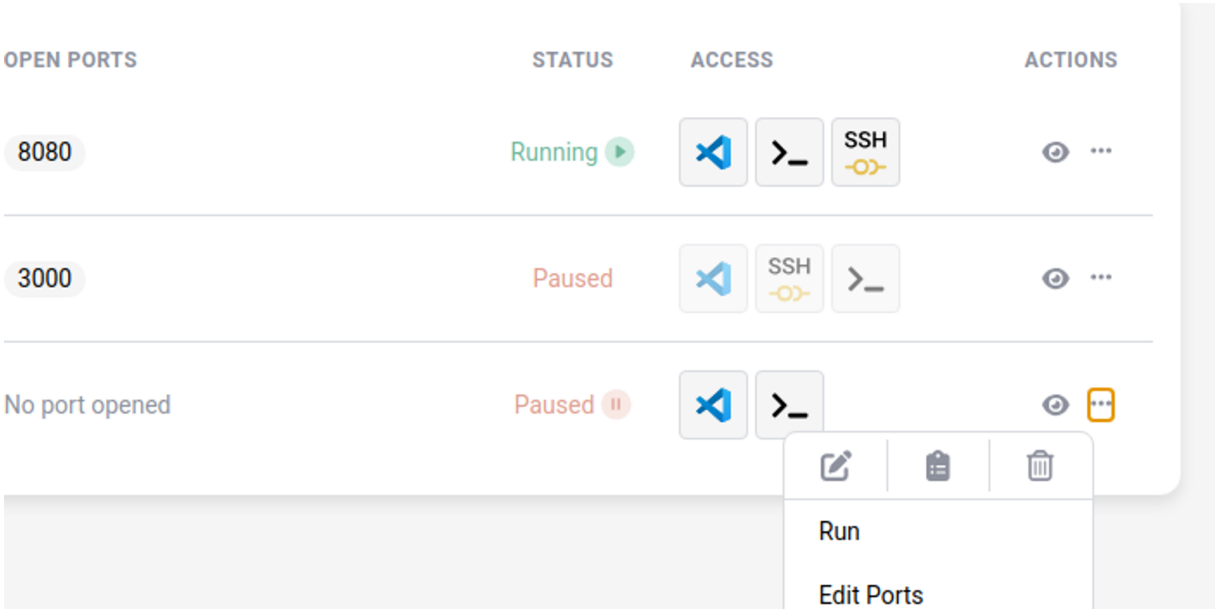
- To **open a paused workspace**, click on the "**start**"button. This will open the workspace's Cloud IDE in your browser.
- To **open a running workspace**, click on the "**running**"button. This will open the workspace's Cloud IDE in your browser.
- To **open your workspace using a CLI terminal**, click on the drop-down menu next "**running text**"and then on the "**Open Terminal**"button.

Workspaces Actions

By clicking on the "**...**"icon on a workspace, you can select additional actions as explained below.

- **Run** or **Pause** allow you to start and pause the workspace, respectively.
- **Edit** allows you to change the workspace'settings as selected when creating it.
- **Delete** erases its configuration and local files. You will need to confirm the action by inserting the name of the workspace.
- **Edit Ports** lets you manage workspace apps running on the ports of your workspace.
- **Personalize Environment** lets you update the [IDE configuration file] based on your profile settings (*Only if Workspace is yours*).
- **Update** redeploys the workspace to synchronize it with its latest configuration.
- **Share** lets you share the workspace access with another project's user. Learn how to work with a shared workspace (*Only if Workspace is yours*).
- **Save As Template** lets you save the workspace's configuration as a template for later reuse (requires the Workspaces permission set to **Manage Project**).

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Recover a deleted workspace

A deleted workspace moves to a recycling bin and can be restored from Project General Settings until it expires. The retention period is **7 days by default**, but your platform administrator can set it to anywhere from 1 to 99 days, so check with them if you are unsure how long you have. See Workspaces pending deletion.

> **Note:**
>
> Only a project owner can recover a workspace. If you do not have the necessary privileges, contact the owner of your project.

Related information

- Connect over SSH
- Create a workspace
- Workspaces page

Connect over SSH

An SSH-based connection lets you work in a Citrix SecurSpaces™ workspace from a locally installed IDE such as VS Code, Cursor, Windsurf, Kiro, or JetBrains. Your IDE runs on your machine while the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

code, tools, and compute stay in the cloud workspace.

SecurSpaces provides dedicated extensions for these IDEs that simplify workspace power management and one-click SSH connections. For IDEs without a SecurSpaces extension, you can connect manually using a standard SSH key pair.

Supported IDEs

VS Code Desktop, Cursor, Windsurf and Kiro connect in one click using the SecurSpaces extension. JetBrains
Gateway connects through its own SSH configuration, and any other SSH-capable editor connects with a manual
key pair. For the full list and the connection method for each, see
Supported IDEs.

Prerequisites

Before you can connect to a workspace via SSH, verify the following:

- **Remote Development Over SSH is enabled.** A platform administrator, organization owner, or project owner must enable this setting under **Workspace Settings > Security Settings > Remote Development Over SSH**. This setting is available at the platform, organization, and project levels.
- **SSH is enabled on the workspace.** When creating or editing a workspace, go to **Basic Info > Access** and drag the **SSH** icon to **Selected**. Workspaces without SSH enabled display a "No SSH access"label and cannot accept SSH connections.

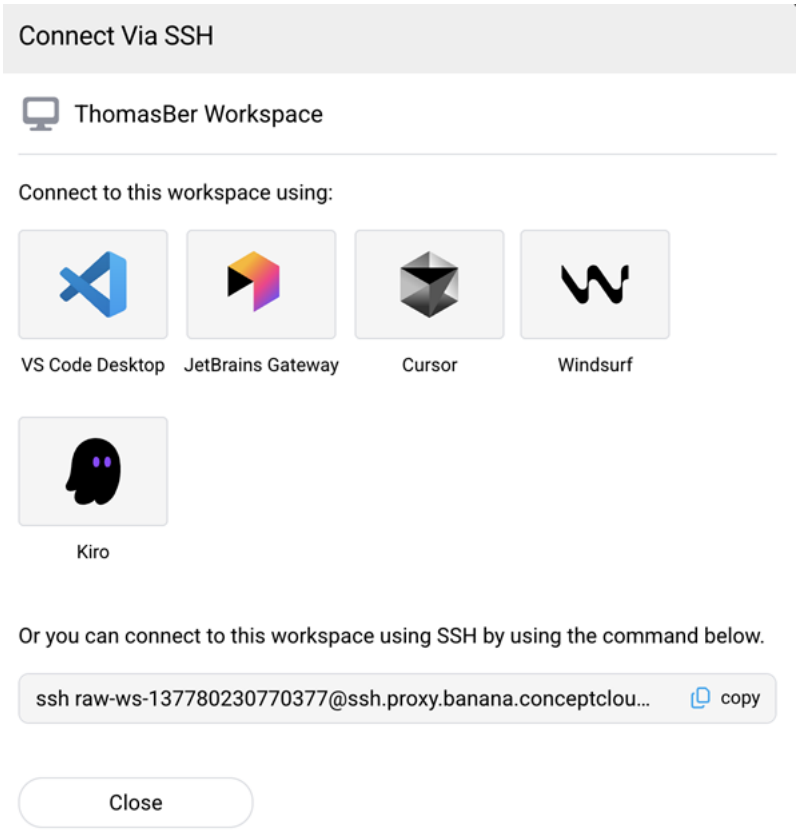
Connect using VS Code, Cursor, Windsurf, or Kiro

SecurSpaces provides the **Citrix Secure Developer Spaces - Remote SSH** extension for VS Code-based IDEs (VS Code, Cursor, Windsurf, and Kiro). Selecting any of these IDEs in the **Connect Via SSH** dialog uses the IDE's custom URI scheme to launch the application, install the SecurSpaces extension if needed, and automatically establish the SSH connection.

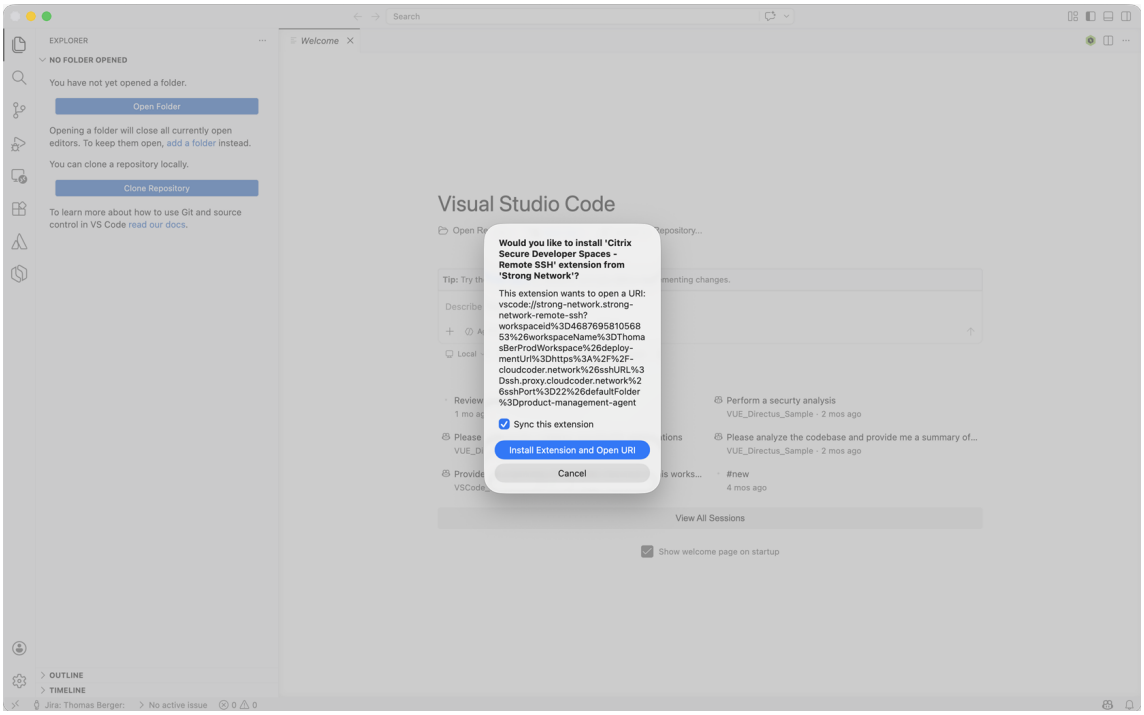
The fastest way to get started is to initiate the connection from the SecurSpaces console:

1. In the SecurSpaces console, navigate to the project **Workspaces** or **Overview** page.
2. Locate a workspace with the **SSH** icon visible in the **Access** column and select it.
3. In the **Connect Via SSH** dialog, select your IDE (for example, **VS Code Desktop**).

© 1997–2026 Citrix Systems, Inc. All rights reserved.



4. Your local IDE opens. If this is your first connection, you are prompted to install the **Citrix Secure Developer Spaces - Remote SSH** extension. Select **Install Extension and Open URI**.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

5. The extension connects to the workspace automatically.

Install the extension manually

If you prefer to set up the extension before initiating a connection from the console:

1. Install the Citrix Secure Developer Spaces - Remote SSH extension from the VS Code Marketplace.
2. Open the extension sidebar. Before you sign in, it shows an **Environment URL** field.
3. Enter the URL of your SecurSpaces environment and select **Connect**. You are redirected to your browser
   to sign in.
4. After you sign in, the sidebar lists the workspaces available to you, grouped by project.

What the extension installs

You install one extension, **Citrix Secure Developer Spaces - Remote SSH**. It has two parts:

- The part that runs on your local machine provides the sidebar, sign-in, the workspace list, SSH connections, and the status bar. Your access token stays on your machine and is never placed on the
  workspace.
- A companion component packaged inside it runs in the workspace. It reports activity so that a workspace you
  are actively using is not paused by idle detection, and it responds to the network measurements described in
  Monitor connection quality.

The companion component is installed into the workspace for you when you connect. There is nothing to set up.
If you used an earlier version of the SecurSpaces extension, you might be prompted to remove the previous
remote extension and restart your IDE.

Manage workspaces from the extension

The extension sidebar is the SecurSpaces entry point inside your IDE. Select the SecurSpaces icon in the activity
bar to open it. Workspaces are grouped by project, and the workspace you are connected to is highlighted and
sorted to the top.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Each workspace row shows its current state, and workspaces without SSH enabled are labeled. You can act on a
workspace from its row controls or by right-clicking the row. Right-clicking a running workspace offers **Connect in current window**, **Connect in new window**, **Edit workspace**, and **Delete workspace**. A paused workspace offers **Edit workspace** and **Delete workspace**.

For the full list of states, row controls, and context menu actions, see
Extension sidebar.

Starting or pausing a workspace shows progress until the new state settles.

Pausing the workspace you are currently connected to closes the remote connection and returns you to a local
window.

> **Warning**
>
> Deleting a workspace permanently removes it and all of its data, and cannot be undone. Pause a running
> workspace before you delete it.

Open repositories, mount points, and apps

Expand a workspace to see what it contains. The connected workspace expands automatically the first time you
connect to it.

- **Repositories** —the code repositories cloned into the workspace. Select one to open that folder in your
  IDE, or right-click it and select **Open in Terminal** to open a terminal at that location.
- **Mount points** —the shared storage attached to the workspace. Read-only mount points are labeled. Select
  one to open the folder, or right-click it and select **Open in Terminal**.
- **Apps** —the workspace apps that are running, such as Jupyter Lab or a preview app. Select an app to open
  it in your default browser, or right-click it and select **Copy URL**.

Apps can only be opened while the workspace is running.

Open the SecurSpaces console from your IDE

Some actions open the SecurSpaces console in your browser rather than being performed in the IDE. The console
applies your permissions to these actions.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Open SDS Console** —opens your SecurSpaces environment.
- **Add workspace** —the **+** button in a project header starts the create-workspace flow.
- **Edit Profile Settings** —opens your profile.
- **Troubleshoot Connection** —opens the profile page used to diagnose connection problems.

These commands are also available from the Command Palette under **Citrix SDS**.

Monitor connection quality

While you are connected to a workspace, the status bar shows the connection state, the workspace name, and the
round-trip time of the connection. The color changes as latency increases, so a slow connection is visible
without opening anything.

Hover over the status bar for more detail, including an estimated bandwidth. Select it to **Remeasure Network**
Now or to **Disable Network Monitor**. Both actions are also available from the Command Palette, along with
Enable Network Monitor to turn monitoring back on.

Disabling the monitor stops the measurements and hides the status bar. The choice persists across restarts.

Each IDE window runs its own monitor for the workspace that window is connected to.

> **Note**
>
> These measurements describe the responsiveness of the connection your IDE uses, which includes SSH and
> protocol overhead. They are a guide to how the connection feels rather than a raw network benchmark.

If some actions are missing

The extension is updated more often than the platform it connects to. When your SecurSpaces platform is older
than the extension, actions that depend on newer platform features are hidden rather than shown and failing.

If the extension hides anything, the bottom of the sidebar reads *More actions appear here after your SDS*
administrator updates the platform. Repositories, mount points, and apps, along with the console

© 1997–2026 Citrix Systems, Inc. All rights reserved.

actions
listed earlier, are the features affected. Ask your platform administrator to upgrade SecurSpaces.

Connect using JetBrains Gateway

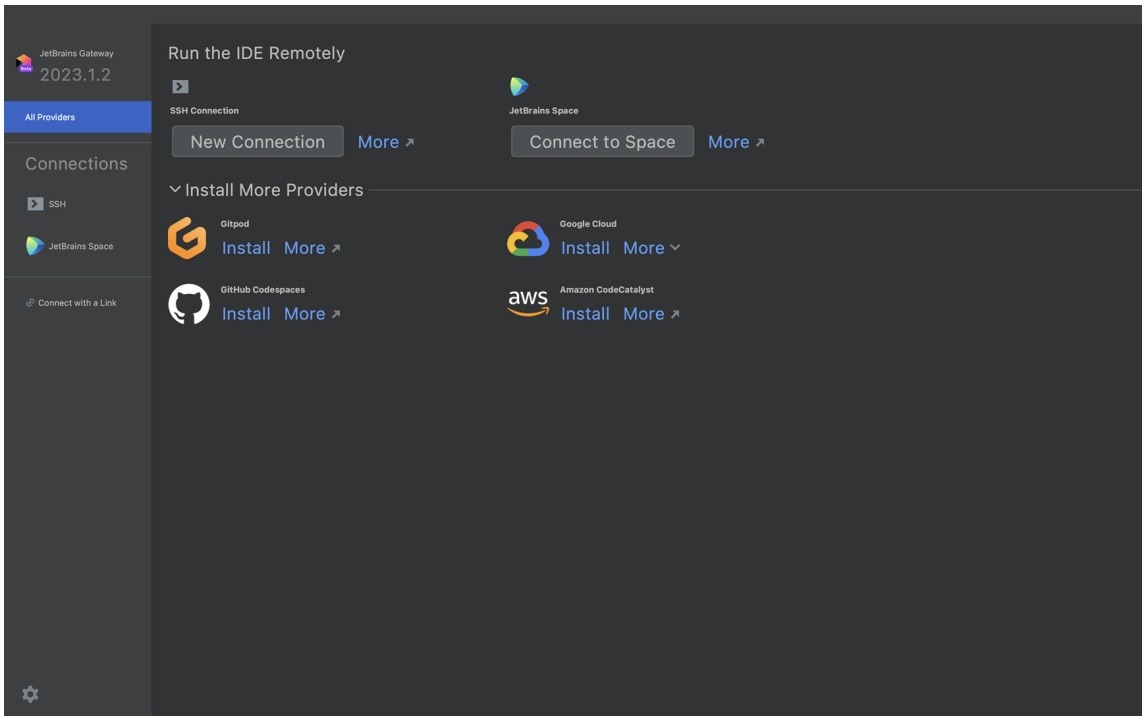
JetBrains Gateway enables SSH connections from JetBrains IDEs (IntelliJ IDEA, PyCharm, GoLand, and others).

Install JetBrains Gateway

Download and install JetBrains Gateway.

Create an SSH connection

1. In JetBrains Gateway, select **New Connection** under **SSH Connection**.



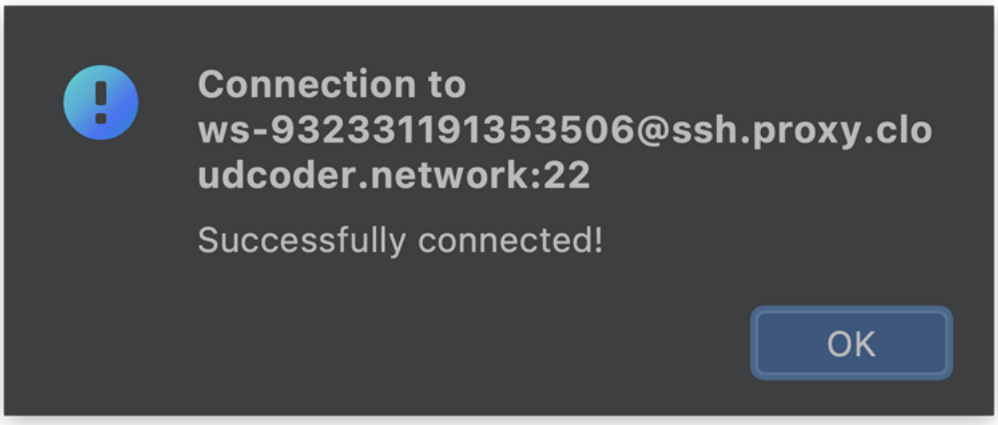
2. Click the settings icon next to **Connection**, then click **+** to add a new SSH configuration.
3. In the SecurSpaces console, select the **SSH** icon on your workspace, then select **JetBrains Gateway** in the dialog. The SSH command is displayed at the bottom of the dialog.
4. Enter the connection details from the SSH command:
    - **Host** —The part after `@` (for example, `ssh.proxy.cloudcoder.network`).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Username** —The part before `@` (for example, `raw-ws-468769581056853`).
- **Authentication type** —Select **Key Pair** and provide the path to your private key.

> **Tip:**
>
> Disregard the `ssh` prefix and the `@` character when splitting the command into Host and Username.

5. Click **Test Connection** to verify the configuration, then click **OK**.



6. Select your new configuration as the **Connection** and click **Check Connection and Continue**.
7. Choose the JetBrains IDE you want to use and the project folder to open, then click **Download IDE and Connect**.

> **Tip:**
>
> The IDE is downloaded to your workspace, not to your local machine.

8. After the IDE installation completes, you are connected to your workspace.

Manual SSH connection

If your IDE lacks a dedicated SecurSpaces extension or a JetBrains Gateway integration, you can connect using a standard SSH key pair. This approach works with any terminal or SSH-capable editor.

Generate an SSH key pair

On UNIX and UNIX-like systems, run `ssh-keygen` in your terminal:

```
1 ssh-keygen
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Press **Enter** to accept the default path and file name, or specify a custom location.
- Enter a passphrase when prompted (recommended for additional security), or press **Enter** to skip.
- The command generates a key pair: a private key (for example, `id_rsa`) and a public key (for example, `id_rsa.pub`).

Upload your public key

Upload the public key to the **SSH Keys** section in your Profile > Security page.

Supported key types: `ssh-rsa`, `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp384`, `ecdsa-sha2-nistp521`, `ssh-ed25519`, `sk-ecdsa-sha2-nistp256@openssh.com`, `sk-ssh-ed25519@openssh.com`.

Authorize your workspace

After uploading your key, authorize the workspace(s) that should accept connections using this key.

Connect from a terminal

1. In the SecurSpaces console, select the **SSH** icon on your workspace to open the **Connect Via SSH** dialog.
2. Copy the SSH command displayed at the bottom of the dialog.
3. Run the command in your terminal:

```
1 ssh raw-ws-468769581056853@ssh.proxy.cloudcoder.network
```

You are now connected to your workspace via SSH.

Key expiry and renewal

A public key you register for SSH access does not last indefinitely. It is valid for a fixed period —**7 days by default** —after which the platform stops accepting it. A platform administrator sets the period, so
your deployment might use a different one.

What you have to do about it depends entirely on how you connect:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How you connect	What happens when the key expires
The SecurSpaces extension for VS Code, Cursor, Windsurf, or Kiro	Nothing. The extension renews the key for you.
JetBrains Gateway with the SecurSpaces plugin	Nothing. The plugin renews the key for you.
Any other SSH client —plain `ssh`, a terminal emulator, or an IDE's own generic remote-SSH support	**You renew it yourself.** The connection stops working until you do.

This is why a setup that worked last week can start refusing connections without anything having changed.
The key expired, and nothing renewed it, because the client you use has no way to.

Renew a key manually

1. Go to **Profile > Security > Remote Access Over SSH**.
2. Upload your public key again, and authorize the workspaces that should accept it.

You can keep using the same key pair; it is the registration that expires, not the key itself. See Security.

> **Note:**
>
> If you connect from a script, a scheduled job, or anything else unattended, plan for this. There is no
> automatic renewal outside the SecurSpaces IDE integrations, so an unattended connection will fail every
> time the key expires until someone re-registers it.

Keeping a session alive

An SSH session that connects normally and then drops after a period of inactivity has usually been closed by
something between you and the workspace, not by the workspace itself. Idle TCP connections are commonly
reaped by corporate proxies, VPNs, load balancers, and NAT gateways.

Sending traffic periodically keeps the connection from looking idle. In `~/.ssh/config` on your local
machine:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Host *
    ServerAliveInterval 60
    ServerAliveCountMax 3
```

That asks for a keepalive every 60 seconds and gives up after three unanswered ones. VS Code Remote-SSH,
JetBrains Gateway, and plain `ssh` all read this file, so it applies to every client at once.

For work that must outlive the connection entirely, run it under a terminal multiplexer such as `tmux` or
`screen`, or with `nohup`. The process then survives the disconnection and you reattach to it afterwards.

> **Note:**
>
> A session that ends while you are away might also be the workspace pausing rather than the connection
> dropping. Idle detection stops workspaces on a schedule. See
> Workspace lifecycle.

Recovering a corrupted terminal

After an interrupted session, the terminal can show stray characters, sequences such as `^[[<35;80;24M`, a
missing or duplicated prompt, or no response to typing.

This happens when a full-screen program was running at the moment the connection dropped. Programs such as
`tmux`, `vim`, `less`, `top`, and interactive CLI tools put the terminal into a special mode —often enabling
mouse reporting —and restore it when they exit cleanly. A dropped connection gives them no chance to, so the
terminal is left in that mode and mouse movement or scrolling arrives as text.

To restore it:

```
reset
```

If `reset` is unavailable or the terminal does not respond, `stty sane` followed by **Ctrl+L** usually works.

To reduce how often this happens:

- Add the keepalive above, so sessions drop less often in the first place.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Turn off mouse reporting in tools that enable it. In `tmux`, remove or comment `set -g mouse on` in `~/.tmux.conf`.
- Detach from a multiplexer session with its own key binding rather than closing the terminal window.

Related information

- Troubleshooting
- Create a workspace
- Manage workspaces
- Workspaces page

What persists in a SecurSpaces workspace

Overview

Citrix SecurSpaces™ workspaces are containerized and ephemeral. Each time a workspace starts, the container is recreated from its base image —so anything you install or change outside the persistent volume does not persist across restarts.

There is exactly one location whose contents survive across workspace restarts, image updates, and infrastructure changes:

```
/home/developer/
```

Everything else —system packages installed with `apt`, files under `/etc`, `/usr`, `/opt`, `/var`, and the rest of the root filesystem —lives on a temporary filesystem and is discarded when the workspace container is recreated.

This page describes the three ways to make changes survive across restarts, when to use each, and the trade-offs.

1. Save files under `/home/developer`

The simplest and most common case. Anything you write under `/home/developer/` is on a persistent volume and will be there next time you open the workspace.

This covers:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Source code, including any cloned Git repositories
- Shell configuration: `~/.bashrc`, `~/.profile`, `~/.zshrc`, `~/.inputrc`
- Editor and IDE settings: `~/.config/`, `~/.vscode/`, `~/.config/JetBrains/`
- Tool-specific user data: `~/.aws/`, `~/.kube/`, `~/.docker/`, `~/.npm/`, `~/.cargo/`

User-scoped tool installations also work here, because they install into your home directory by default —`nvm`, `pyenv`, `rustup`, `uv`, `pipx`, `cargo install`, `npm install -g` (with a prefix under `~/`), Go modules under `~/go`, and others.

```
# Example: Install Node.js global tools into ~/.local (persists across
    restarts)
mkdir -p ~/.local
npm config set prefix ~/.local
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc
npm i -g yarn pnpm typescript
```

```
# Example: Install Python CLI tools with pipx (installs into ~/.local
    by default)
pipx install ruff
pipx install pre-commit
```

After the initial setup, these tools remain available across restarts without any startup script.

If a tool only offers a system-wide installer (typically `apt install` or anything that writes to `/usr/local/`), the installation resets on the next restart. Use one of the next two options instead.

2. Startup scripts

Each workspace can run a **Pre-Startup Script** (before the IDE comes up) and a **Post-Startup Script** (after the workspace is ready). Both execute on every workspace start.

This is the right tool for:

- Install system packages that must be present before you start working —`sudo apt-get update && sudo apt-get install -y <packages>`
- Start background services your project needs —a local database, a message broker, a build daemon
- Set environment variables or write configuration files into locations outside `/home/developer/` that your tooling expects
- Run login-time setup that depends on secrets or runtime values

Where to configure startup scripts

Startup scripts can be defined at three levels:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. **Workspace level** —Set during workspace creation or in the workspace configuration. Applies only to that workspace.
2. **Template level** —Defined in a workspace template. Applies to all workspaces created from the template.
3. **Profile level** —Set in Profile → Configuration. Applies to all workspaces you own, unless a workspace-level script overrides it.

> **Note:**
>
> A workspace-level startup script overrides the profile-level script.

Example: Install system packages (pre-startup)

```
#!/bin/bash
sudo apt-get update -qq && sudo apt-get install -y -qq gh
```

Example: Start a background service (post-startup)

```
#!/bin/bash
sudo service postgresql start
```

Important considerations

- Scripts run on **every** start. Keep them idempotent and as fast as possible —they directly add to your workspace startup time.
- If you reinstall the same `apt` packages on every launch, this adds unnecessary startup time. Move those installations into a custom image (next section) and keep the script for tasks that genuinely need to run at start time.
- Errors in a pre-startup script can prevent the workspace from coming up. Test changes carefully.
- Do not store secrets in scripts. Use SecurSpaces platform-managed secrets instead.

3. Custom container images

For anything persistent across sessions —system packages, language runtimes, command-line tools, baseline configuration —the recommended approach is to include it in a container image.

Why this is preferred for a shared, persistent setup

- **Build once, reuse everywhere.** Every workspace launched from the template starts in seconds with everything already installed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Reproducible.** The image is a pinned, versioned artifact. Every developer gets the same environment, which eliminates configuration drift.
- **Shareable.** A team or whole organization can standardize on a single image instead of each developer maintaining their own startup script.
- **No startup-time penalty.** Layers are pulled and cached on the cluster nodes, not reinstalled per workspace.

The workflow

1. Write a `Dockerfile` starting `FROM` the SecurSpaces base image of your choice.
2. Add your `RUN apt-get install …`, language toolchains, global command-line tools, baseline `/etc` configuration, and any setup that should be the same for everyone using this template.
3. Build and push the image to your container registry (Docker Hub, GHCR, ECR, ACR, GAR, JFrog, Harbor —anything reachable from the SecurSpaces cluster).
4. Reference the image in your workspace template or individual workspace settings.

Example Dockerfile

```
FROM your-base-image:latest

# Install system-level dependencies
RUN apt-get update && apt-get install -y --no-install-recommends \
    build-essential \
    cmake \
    libssl-dev \
    && rm -rf /var/lib/apt/lists/*

# Install a specific Go version
RUN wget -q https://go.dev/dl/go1.22.0.linux-amd64.tar.gz \
    && tar -C /usr/local -xzf go1.22.0.linux-amd64.tar.gz \
    && rm go1.22.0.linux-amd64.tar.gz

ENV PATH="/usr/local/go/bin:${
 PATH }
 "
```

> **Important:**
>
> Custom images must meet the SecurSpaces container image requirements: an SSH client, Git and Git LFS, and a
> user named `developer` with UID 1000. For the full list, see
> Container image requirements.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How to import the image

After building and pushing the image to your registry, import it into SecurSpaces under **Resources → Container Images**. If your registry is private, make sure SecurSpaces has credentials for it —see the Container Images resource configuration in your project or organization settings.

For detailed steps, see How to Update an Existing Container Image. For sample Dockerfiles and scripts, see the Citrix sample images repository.

Docker inside a workspace

Workspaces can build and run containers. For what the workspace is allowed to do and how to get the Docker
client into your image, see
Use Docker inside a workspace.

What persists is the part that belongs here:

What	Survives a restart
Files under `/home/developer`, including anything you mount into a container	Yes
Images you pull or build	Only with Persistent Docker
Containers, volumes, and the build cache	No

Persistent Docker is a licensed feature and is off by default, so re-pulling or rebuilding images after a restart is the default behavior rather than a fault.

Data written inside a running container is never kept when the container is removed, with or without Persistent Docker —it caches images, not container state. To keep what a container produces, mount a
directory from `/home/developer` into it. See
How to persist container data.

Choosing between the three

You want to persist	Use
Source code, dot files, IDE settings, user-installed tools	`/home/developer/`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

You want to persist	Use
Per-launch setup that depends on runtime state, or a quick one-off package	Startup script
System packages, runtimes, baseline tooling shared across a team	Custom image

A common pattern is to combine all three: a custom image provides the baseline environment, a short post-startup script handles anything dynamic, and your personal configuration lives in `/home/developer/`.

References

- Create a Workspace —Startup Scripts
- Workspace Templates
- Profile —Configuration —Startup Script
- Container Images
- How to Update an Existing Container Image
- Sample images repository on GitHub
- Prebuilt images on Docker Hub

Connect to external services over SSH

August 18, 2026

You often need to reach systems outside your workspace over SSH —to push to a Git repository hosted on Azure
DevOps, deploy to a staging server, or open a session on a remote database host.

By default the SecurSpaces proxy service blocks outbound SSH from workspaces. One of two options must be
enabled first, and both are enabled by a project owner or security officer:

Option	What you do	What your administrator does
Personal SSH Identity	Generate your own key in your profile and register the public key on the target service	Switches on **Enable Personal SSH Identity** for the workspace or template

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Option	What you do	What your administrator does
Connected SSH Services	Nothing. Run `ssh <service-name>`.	Creates the service with its credentials and attaches it to the workspace

If neither is enabled, see Outbound SSH is blocked.

Administrators should see
Enable outbound SSH.

Option 1: Personal SSH Identity

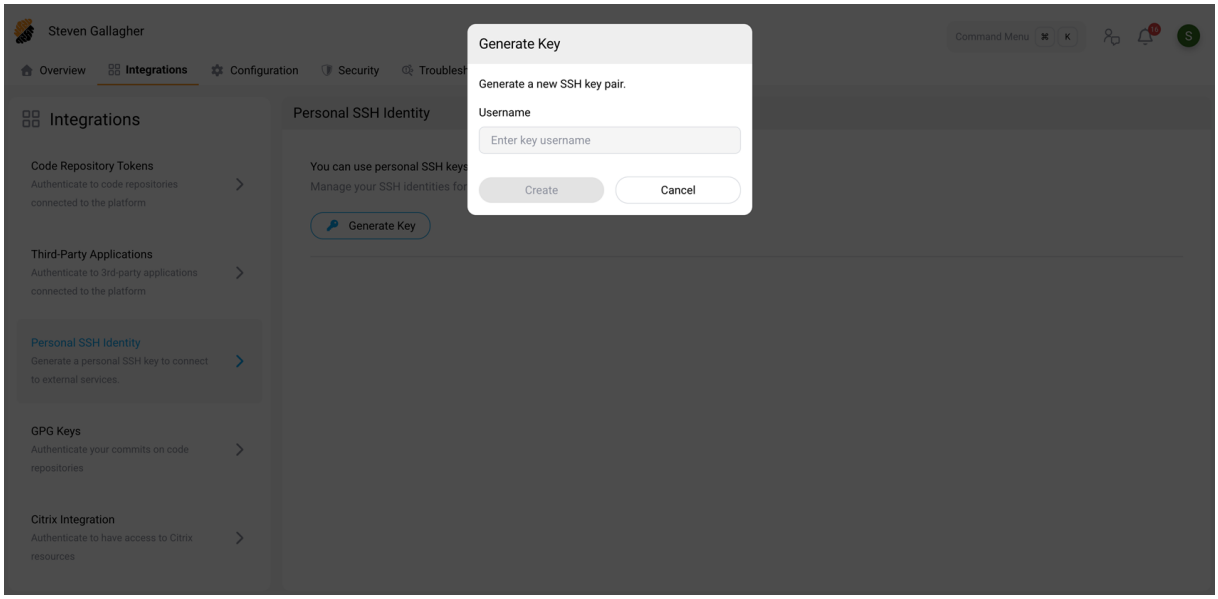
The platform generates and holds an SSH key pair for you. The private key is applied automatically when you
connect and is never exposed inside the workspace.

Prerequisite: a project owner or security officer has switched on **Enable Personal SSH Identity** for your workspace or its template.

Generate your key

1. Select your profile picture in the top-right corner to open your profile.
2. Go to **Integrations** > **Personal SSH Identity**.
3. Select **Generate Key**.
4. Enter a default user name, for example `git`. You can override it per service when you connect.
5. Select **Create**.

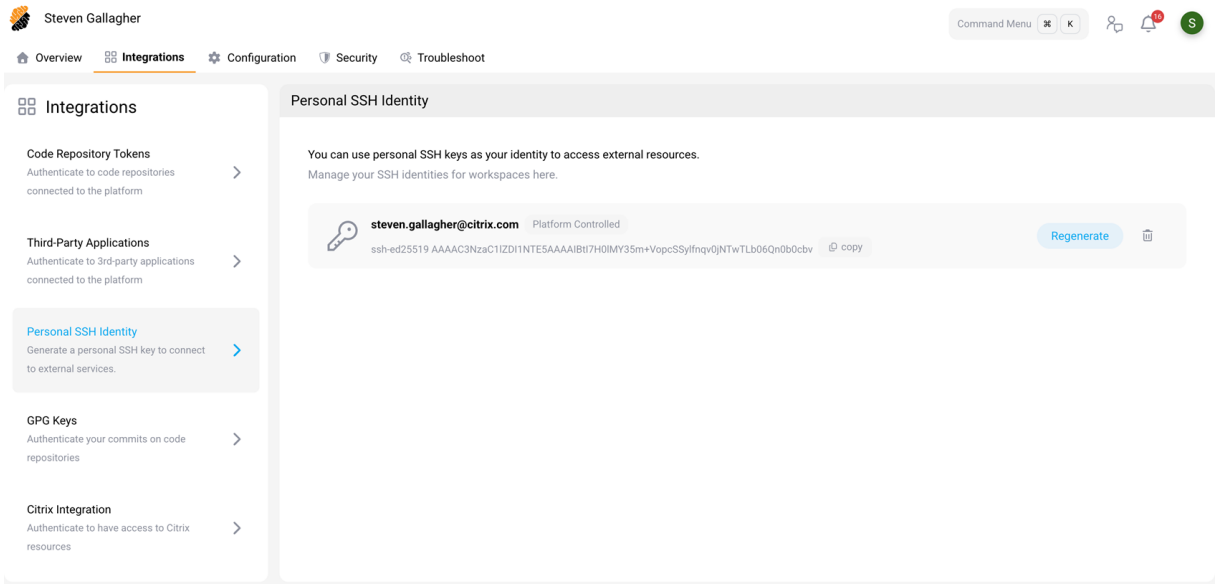
© 1997–2026 Citrix Systems, Inc. All rights reserved.



The platform generates the pair and shows the public key. The key applies to new and existing workspaces.

Register the public key

1. On the **Personal SSH Identity** page, select **Copy**.
2. Add the public key to the target service, for example by appending it to `~/.ssh/authorized_keys` on the host. The exact steps depend on the service.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Connect

Open a terminal in any workspace where Personal SSH Identity is enabled and connect as usual:

```
ssh <username>@<hostname>
```

SecurSpaces applies your private key. There is no key to manage inside the workspace.

Option 2: Connected SSH Services

Your administrator defines the target host and its credentials once, then attaches it to the workspace. You do
not handle a key or a password.

Open a terminal in the workspace and use the service name as the host:

```
ssh gcp-vm
```

SecurSpaces authenticates you automatically.

Outbound SSH is blocked

If no SSH method is configured for the host you are reaching, the proxy closes the session and the terminal
shows this before it does. Depending on the client, you might also see `no signers found for request` or
`ssh: no auth passed yet`, which mean the same thing—the workspace has no SSH credential to present.

```
SSH access to external hosts is disabled by default in this workspace.
To connect, an administrator must enable one of:
  - Personal SSH Identity (per-user key) - in the workspace/template
     Security Settings
  - a Connected SSH Service - then run: ssh <service-name>
See: https://docs.citrix.com/en-us/securspaces/workspace/ssh-
    connectivity
```

Ask a project owner or security officer to enable one of the two options. Personal SSH Identity suits access
to your own accounts; a Connected SSH Service suits a host the whole team uses.

> **Note:**
>
> This message appears only when the SSH connection itself is blocked for lack of a configured method. A

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> connection that fails because the destination is not on an allow list is a network policy matter. See
> Monitor and troubleshoot.

Related information

- Enable outbound SSH
- Connected SSH Services
- Integration

Share a running application

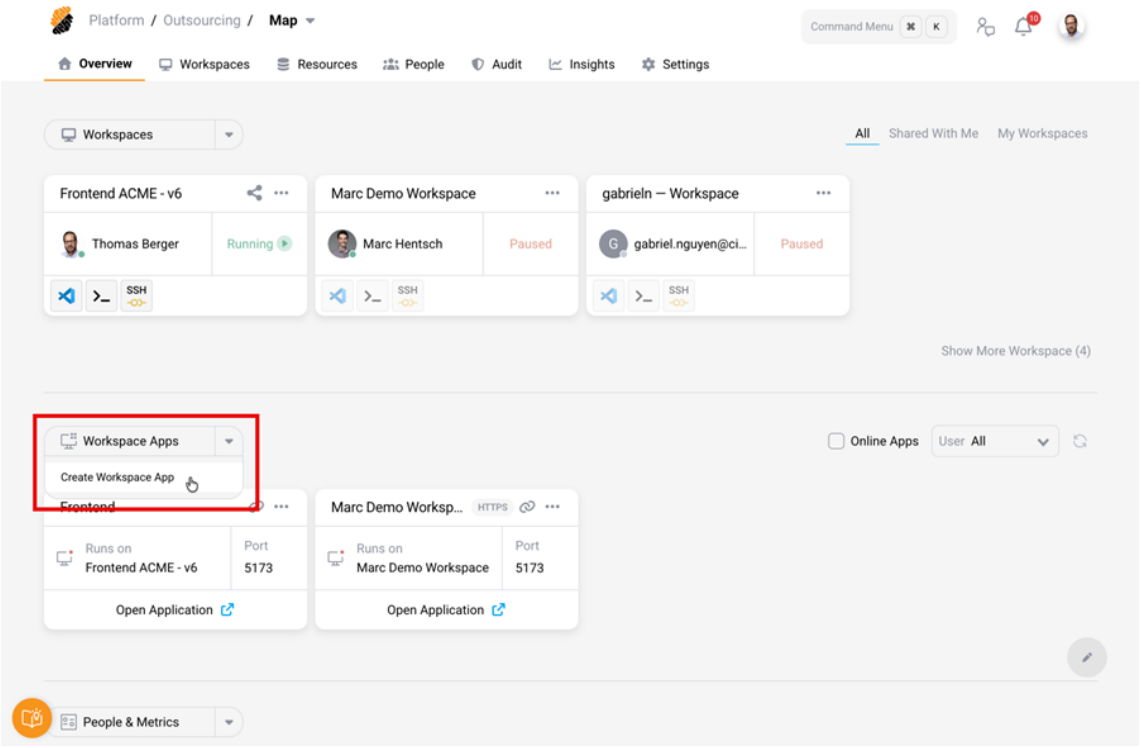
A **Workspace App** lets you access an application or process through HTTP or HTTPS running on a port of your Workspace. You can have multiple Workspace Apps attached to a single workspace, each accessing an application running on a different port. Common use cases include testing web applications, webhooks and callbacks, and exposing the web interfaces of development tooling.

Create a Workspace App

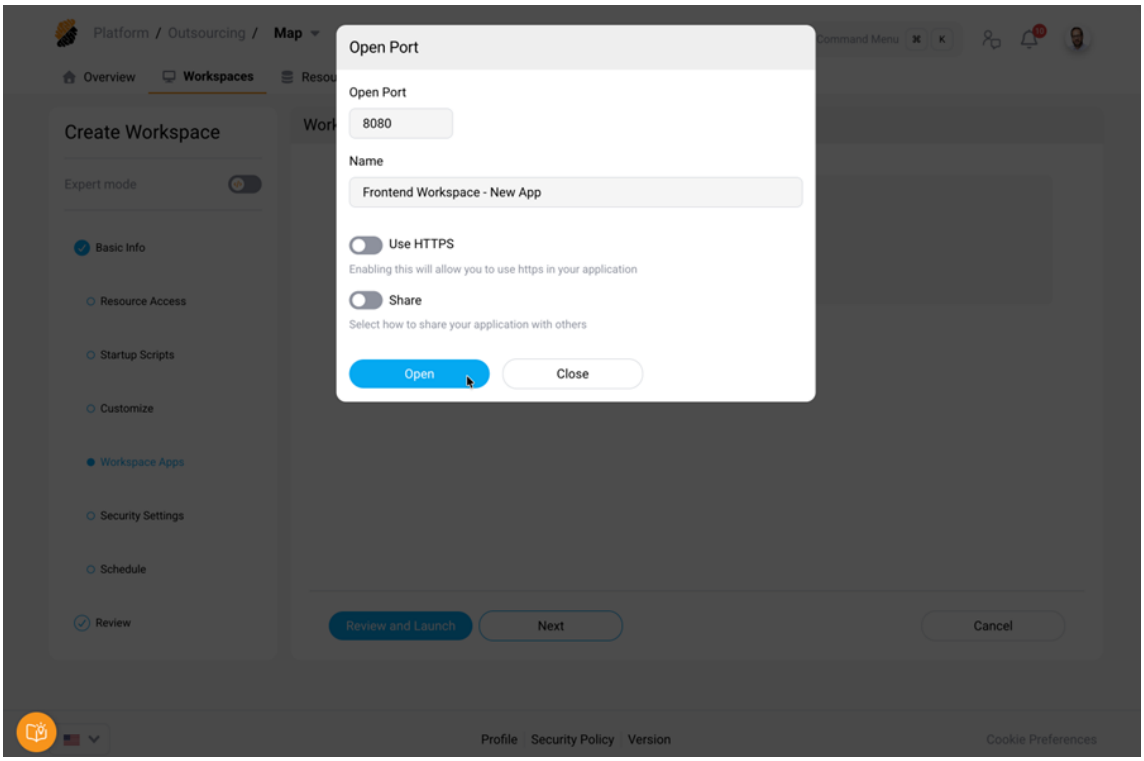
Workspace apps can be created in different ways.

- From the Project Overview page, select the **Create Workspace App** button from the **Workspace Apps** drop-down menu and follow the onscreen instructions.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- When creating or editing a workspace or workspace template, select the **Workspace Apps** menu in the wizard and follow the onscreen instructions. When adding a workspace app to a template, all child workspaces inherit this setting automatically.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Click the "…" icon on a workspace and select **Edit Ports**.

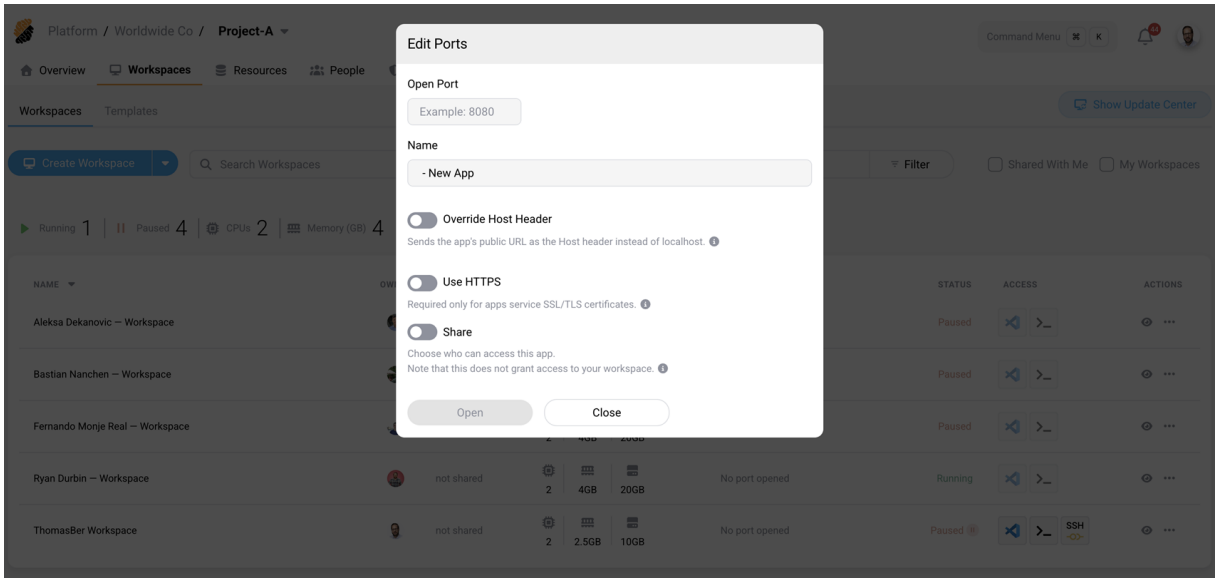
In either case, you need to enter the following information:

1. **Port** —The port where the app or process is running in your workspace.
2. **Name** —A display name for the workspace app. Citrix SecurSpaces™ uses this name to reference the app.
3. **Override Host Header** —Enable this option when your application rejects requests because of a Host header mismatch. See Override Host Header for details.
4. **Use HTTPS** —Enable this option only if your application serves its own TLS certificates. For most scenarios, leave this off. Regardless of this setting, the platform is always exposed over HTTPS, so the browser connection appears as HTTPS. The toggle controls only whether the platform calls the app within the workspace using HTTP or HTTPS.
5. **Share** —Control who can access the application. See Share a Workspace App for options.

> **Tip**
>
> When you create a workspace app for a Node.js project, make sure the port number matches the one opened in the localhost of the workspace.

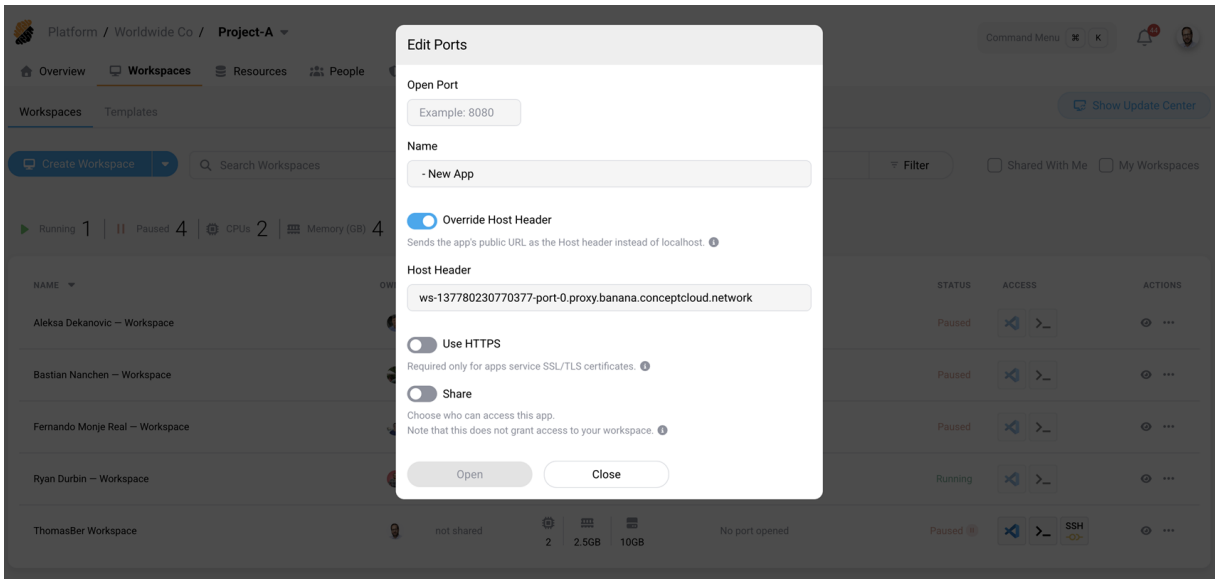
Override Host Header

When a workspace app runs behind the SecurSpaces proxy service, some applications (such as Jupyter Notebook or pgAdmin) may block requests because of a mismatch between the Host header and the Origin header. By default, the proxy forwards `localhost` as the Host header, while the Origin header contains the workspace app's public URL. Applications that do not check `X-Forwarded-Host` interpret this as a cross-origin request and reject it.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable **Override Host Header** to have the proxy send the workspace app's public URL in the Host header instead of `localhost`. This aligns the Host, Origin, and X-Forwarded-Host headers so that the application works without per-tool proxy configuration.



When enabled, a **Host Header** field appears, pre-filled with the workspace app's fully qualified domain name (FQDN). You can accept the default or enter a custom value.

> **Important**
>
> Leave **Override Host Header** disabled for local development workflows where the server expects `localhost` as the Host header. Enabling it in those cases can break the application.

Share a Workspace App

You can share a workspace app when creating it or by editing an existing one.

To update the properties of a workspace app, either:

- On the Project Overview page, click the "…" icon on the workspace app and select **Edit**.
- Click the "…" icon on the workspace that hosts the app and select **Edit Ports**.

There are three sharing options:

- **Public** —Everyone with the link who can connect to the SecurSpaces environment over the network can access the workspace app.
- **Project Sharing** —Only members of the SecurSpaces project can access the workspace app.
- **Share With Project members** —Only the selected members of the SecurSpaces project can access the workspace app.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Granting access to a workspace app does not provide access to the workspace that runs the app. To share a workspace with another user, use the **Share Workspace** functionality.

> **Note**
>
> The **Public** sharing option is governed by a platform-level policy. A platform administrator must enable **Allow Creation of Public Workspace Apps** in Workspace Apps Settings for public sharing to take effect. The two names refer to the same capability: **Public** is the sharing mode you select on an individual app, while **Allow Creation of Public Workspace Apps** is the platform policy that permits it. If the policy is disabled, a workspace app set to **Public** still requires viewers to authenticate to the platform.

Access a Workspace App Using an API Key

You can access a running workspace app programmatically with an API key instead of authenticating interactively in a browser. This is useful for scripts, CI pipelines, webhooks, and other automated callers.

This method works only when a platform administrator enables **Allow Access to Workspace Apps Using API Keys** in Workspace Apps Settings. When the policy is disabled, the platform ignores API keys and the request falls back to standard authentication.

To authenticate a request, include your API key in the `Strong-Network-Authorization` header:

```
Strong-Network-Authorization: <API_KEY>
```

The API key grants access only when its scope matches the workspace app and the user who created the key is allowed to access the app —that is, the user owns the workspace, the app is shared with the project, or the app is shared directly with that user.

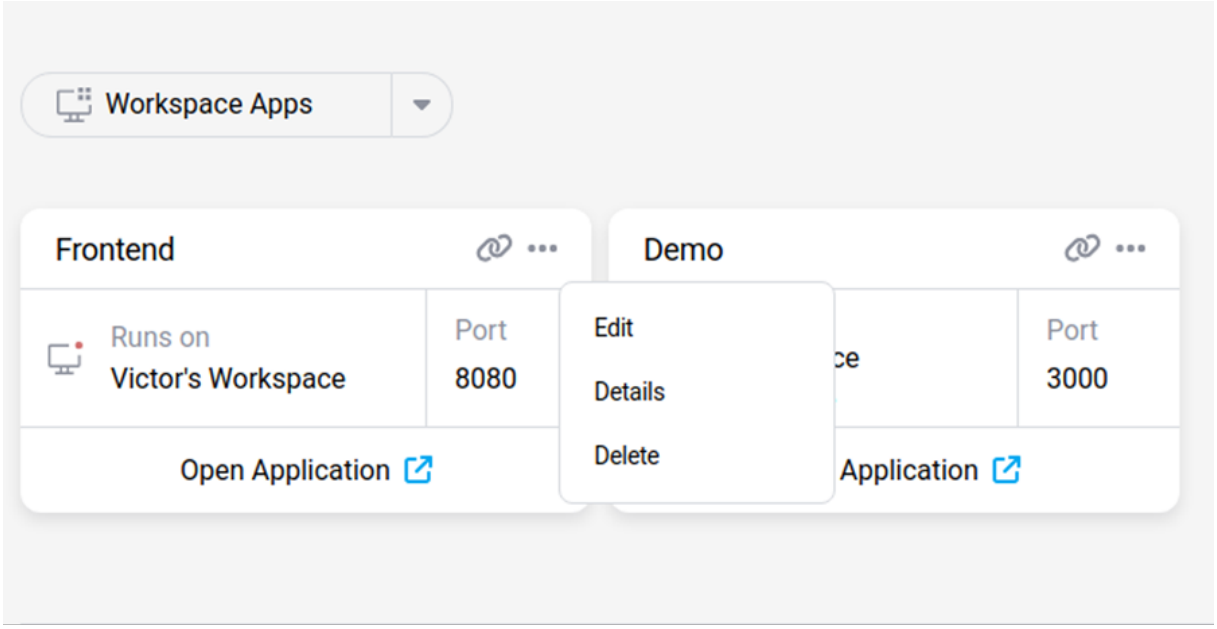
For example, to call a workspace app running on port 8010:

```
curl -v \
  -H "Strong-Network-Authorization: <API_KEY>" \
  https://<workspace-app-public-url>/your/app/path
```

Delete a Workspace App

You can delete a workspace app from the Overview page by pressing the "…" icon and selecting **Delete**. You can also delete a workspace app by clicking the "…" icon on a workspace and selecting **Edit Ports**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Your profile

> **Setting up for the first time?**
>
> Set up your account walks through these four pages in order, as a guided first-run task. Use the pages below when you want the detail on a particular setting.

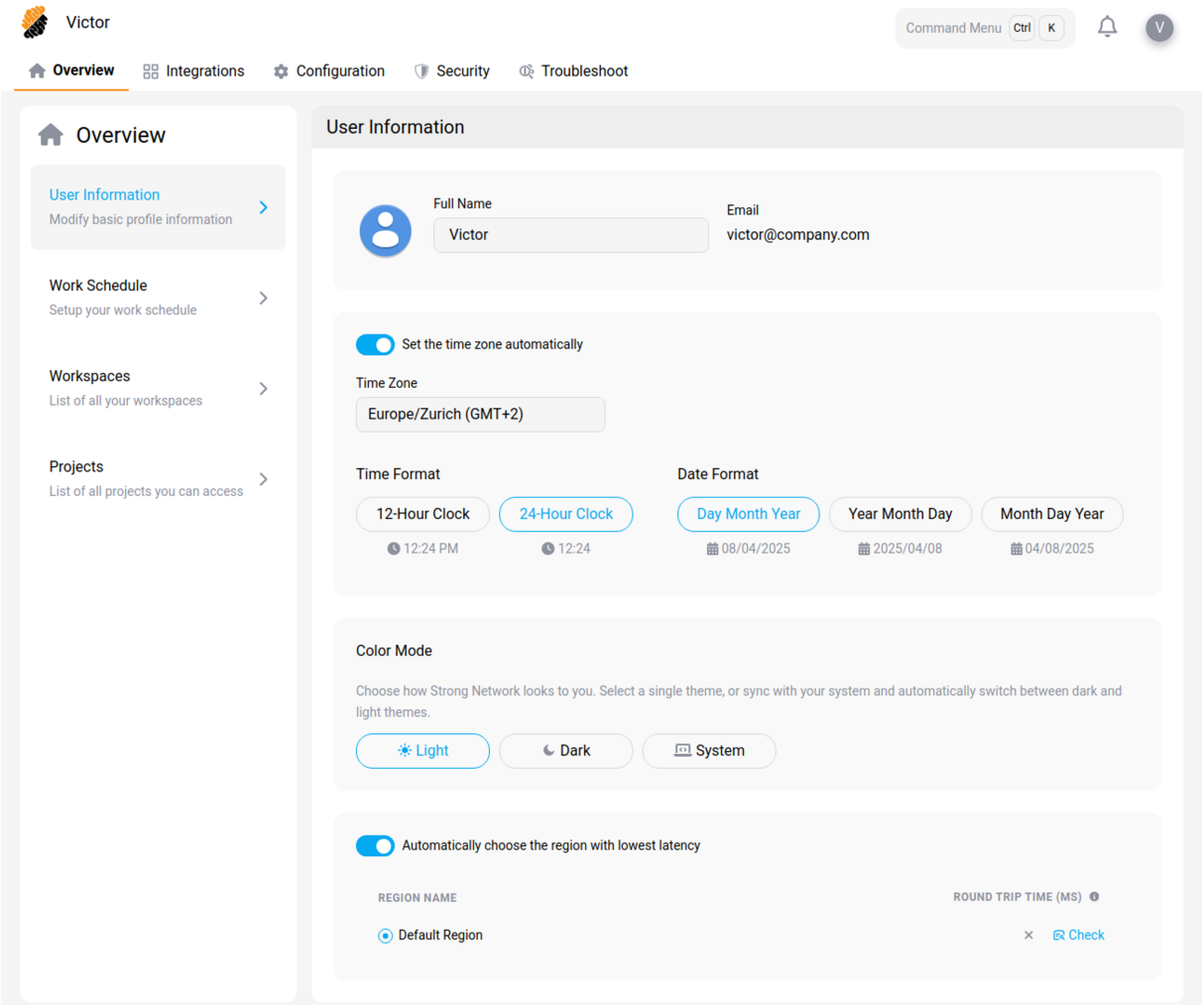
The **Profile and Account Settings** pages let you manage personal data and set preferences around your work habits. For example, you can set-up a work schedule such that your workspace is automatically deployed at pre-set hours.

The profile is used also to store any personal configuration files such as .bashrc, etc needed to customize your workspaces.

In addition, you can use the profile to record IDE configurations, including installed plug-ins, and replicate them across workspaces. Finally, the profile is the place to manage the different authentication tokens and access keys to authenticate to GIT applications attached to the platforms and accessible from the workspaces.

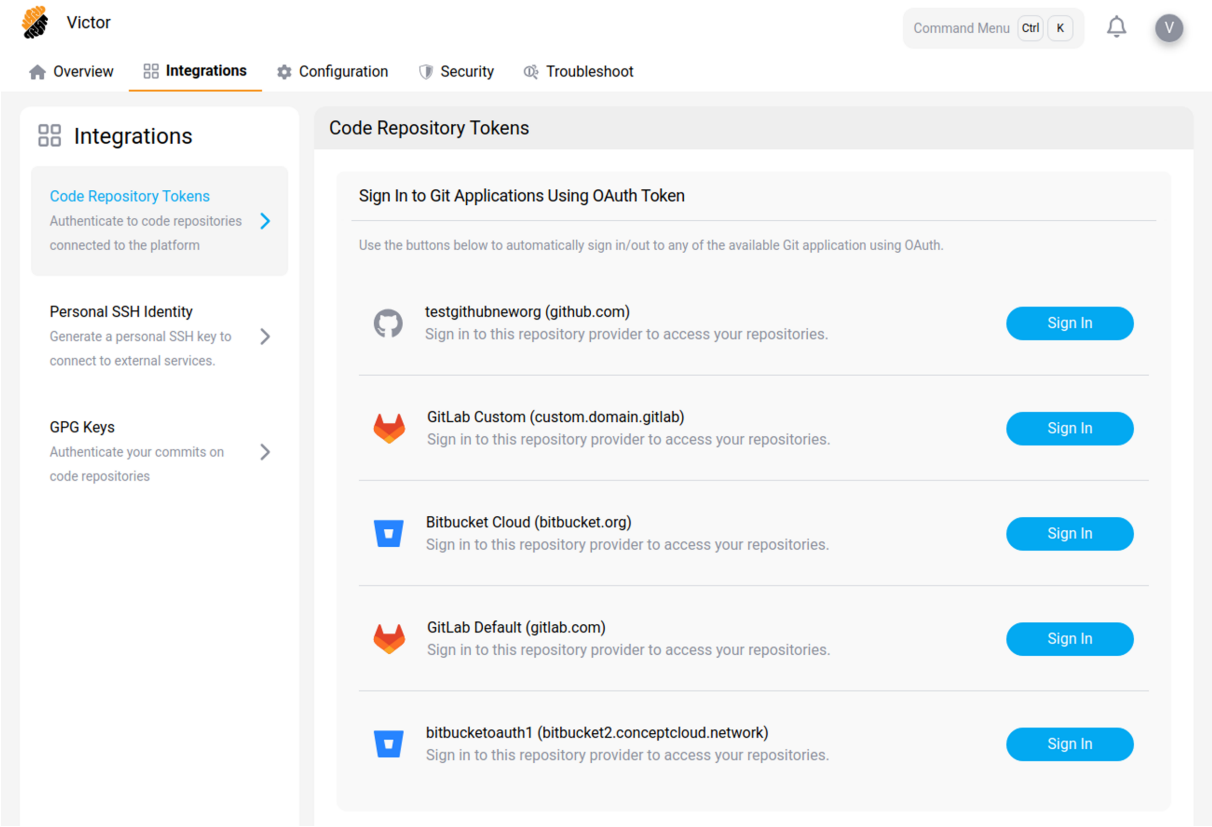
The **Overview Page** allows you to edit personal information, define a work schedule, view owned workspaces and project membership.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



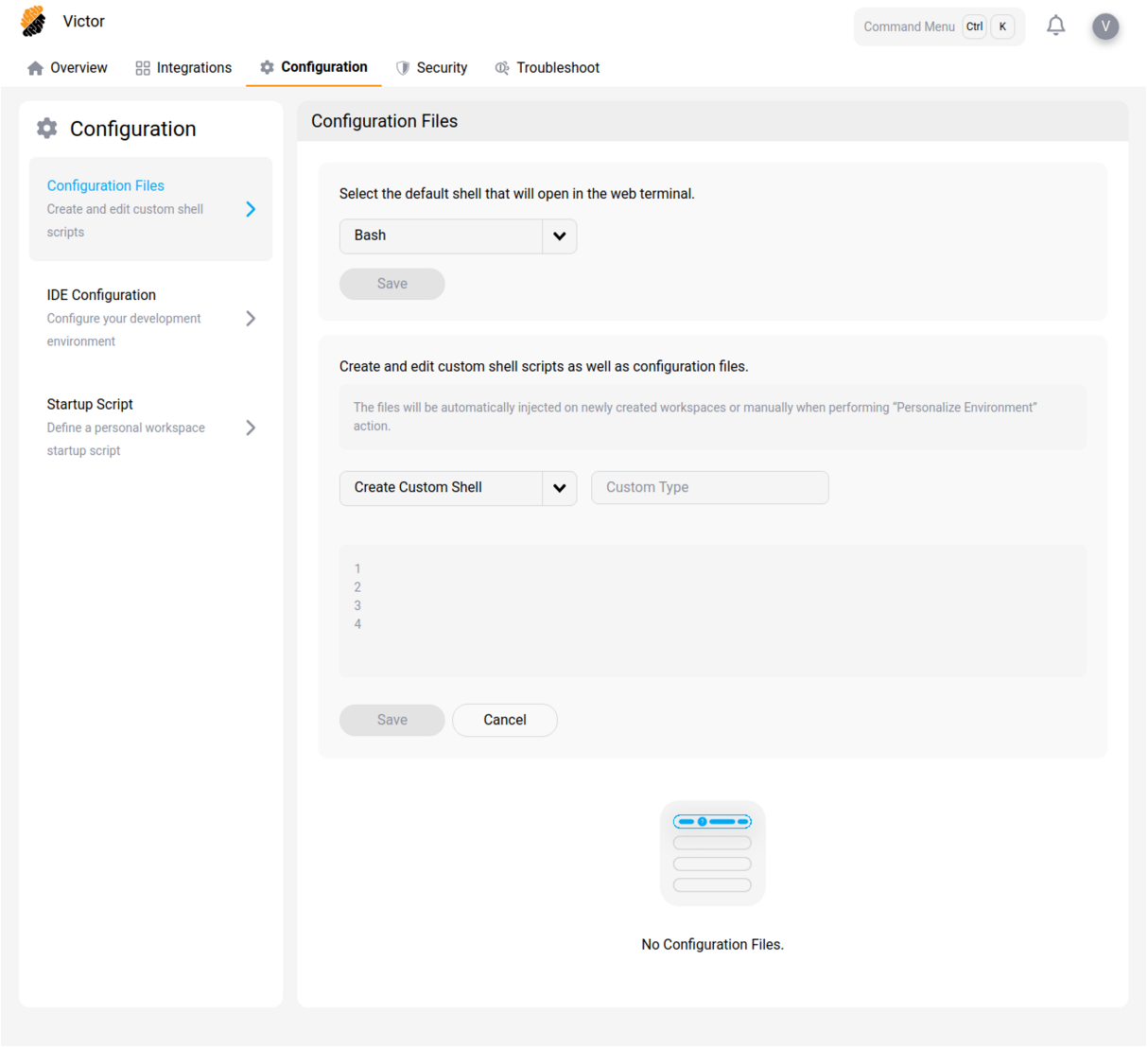
The **Integration Page** allows you to create and edit different authentication tokens, personal SSH identity, and GPG keys.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



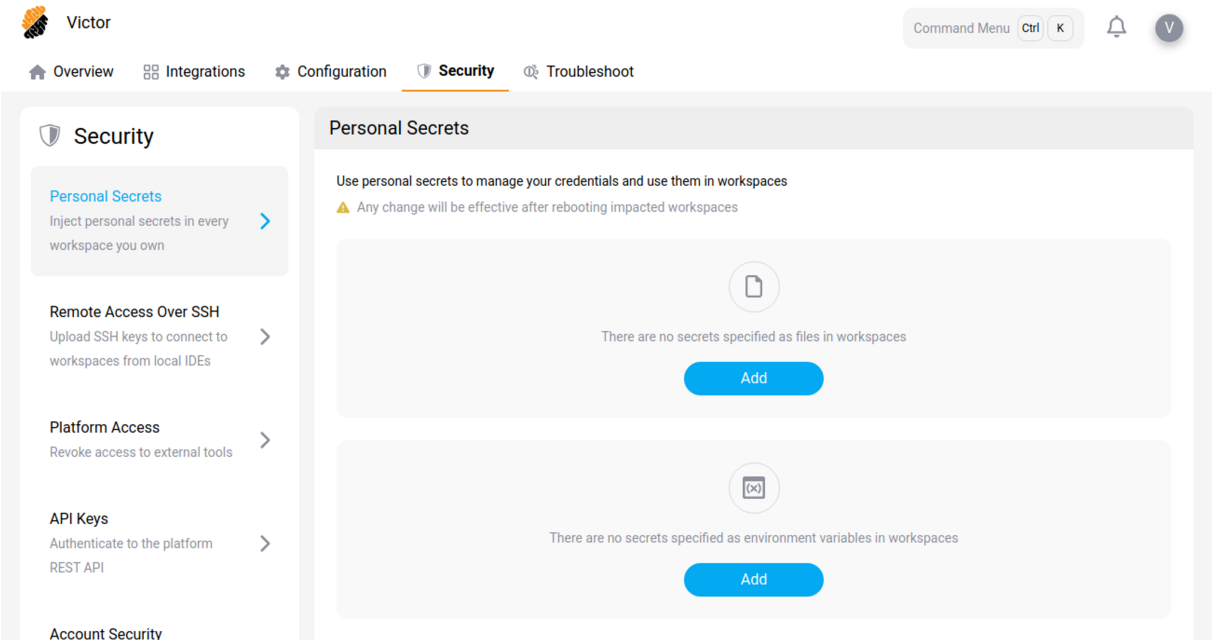
The **Configuration Page** allows you to create and edit custom configuration files, IDE configurations and workspace startup scripts.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



The **Security Page** allows you to create and edit API keys, SSH keys and personal secrets.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Account menu

Select your profile picture at the top right of the screen to open the account menu. Alongside the profile
pages and the theme setting, the menu
includes:

- **REST API** —the API specification for your deployment. See
  REST API.
- **Documentation** —this documentation set.
- **What's New** —the SecurSpaces release notes, in a new browser tab.
- **Your Organizations** and **Your Projects** —the organizations and projects you belong to.

Content

- **Overview Page**
- **Integration Page**
- **Configuration Page**
- **Security Page**
- **Troubleshoot** —four diagnostic tools for
  the IDE connection, region latency, DNS, and network protocols

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Profile overview

August 13, 2026

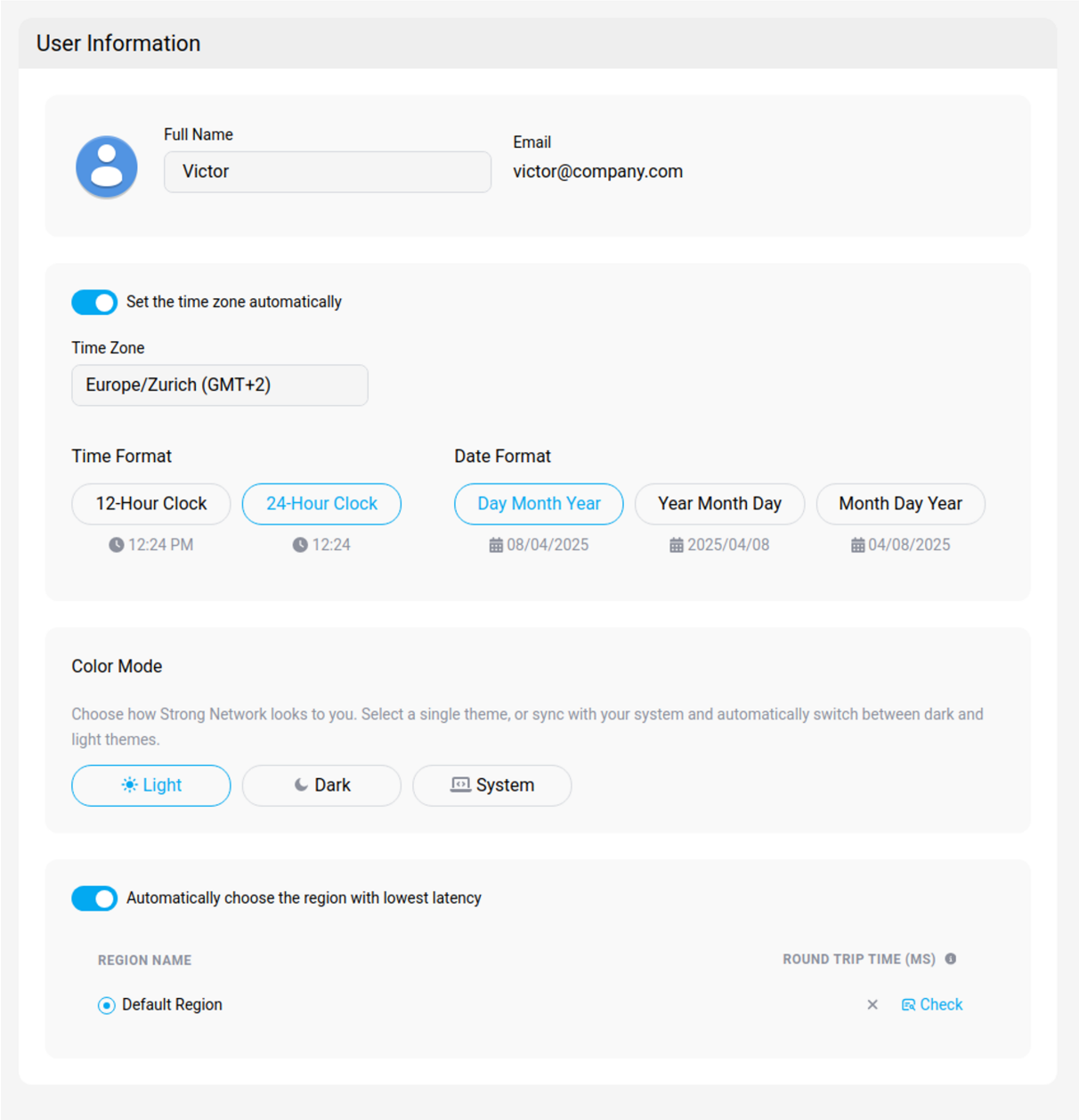
The **Profile Overview Page** serves as a comprehensive summary of the user's information, their workspace ownership and project membership.

- User Information
- Work Schedule
- Workspaces
- Projects

User Information

In the **User Information** section you can modify your user's name and time zone.
The email linked to your profile cannot be modified.
The profile picture is retrieved from your identity provider when available.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Display preferences

The same tab holds the settings that control how the console looks to you. They apply to your account only,
follow you between devices, and affect nothing that other people see.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	Options
Time Zone	Set manually, or turn on **Set the time zone automatically**
Time Format	12-hour or 24-hour
Date Format	Day/Month/Year, Year/Month/Day, or Month/Day/Year
Color Mode	**Light**, **Dark**, or **System** to follow your operating system
Interface Language	**English (United States)** or **Français (Suisse)**

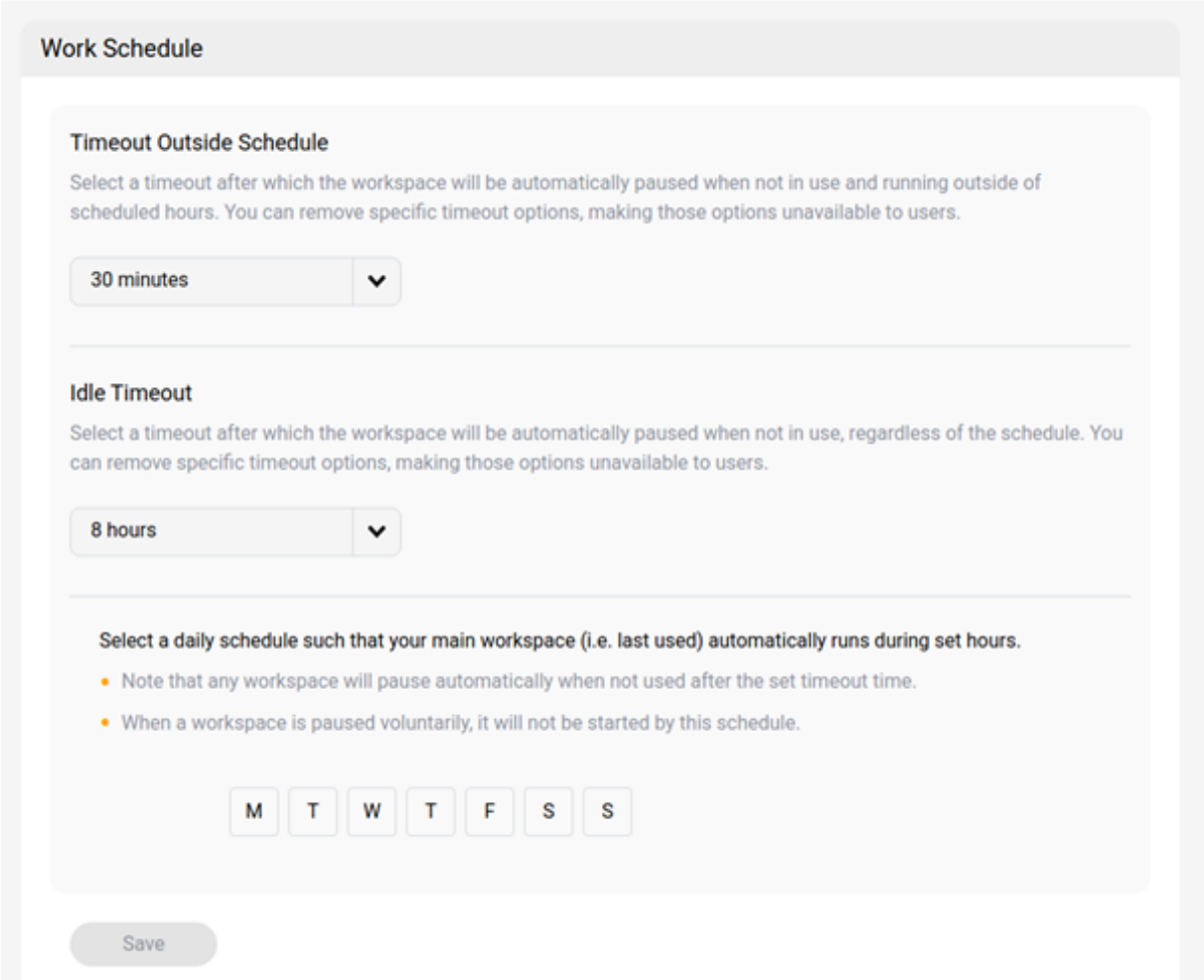
Interface Language changes menus, labels, and system messages in the console. It does not translate
content that you or your administrators created, such as project names, workspace names, or template
descriptions, and it does not change the language inside a workspace or its IDE.

Below these settings, **Automatically choose the region with lowest latency** lets the platform pick a region
for you. The table beneath it lists each region with its round-trip time, and **Check** measures it again.

Work Schedule

In the **Work Schedule** section, you can configure your profile's work schedule. During set hours your main workspace (i.e. last used) is automatically deployed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

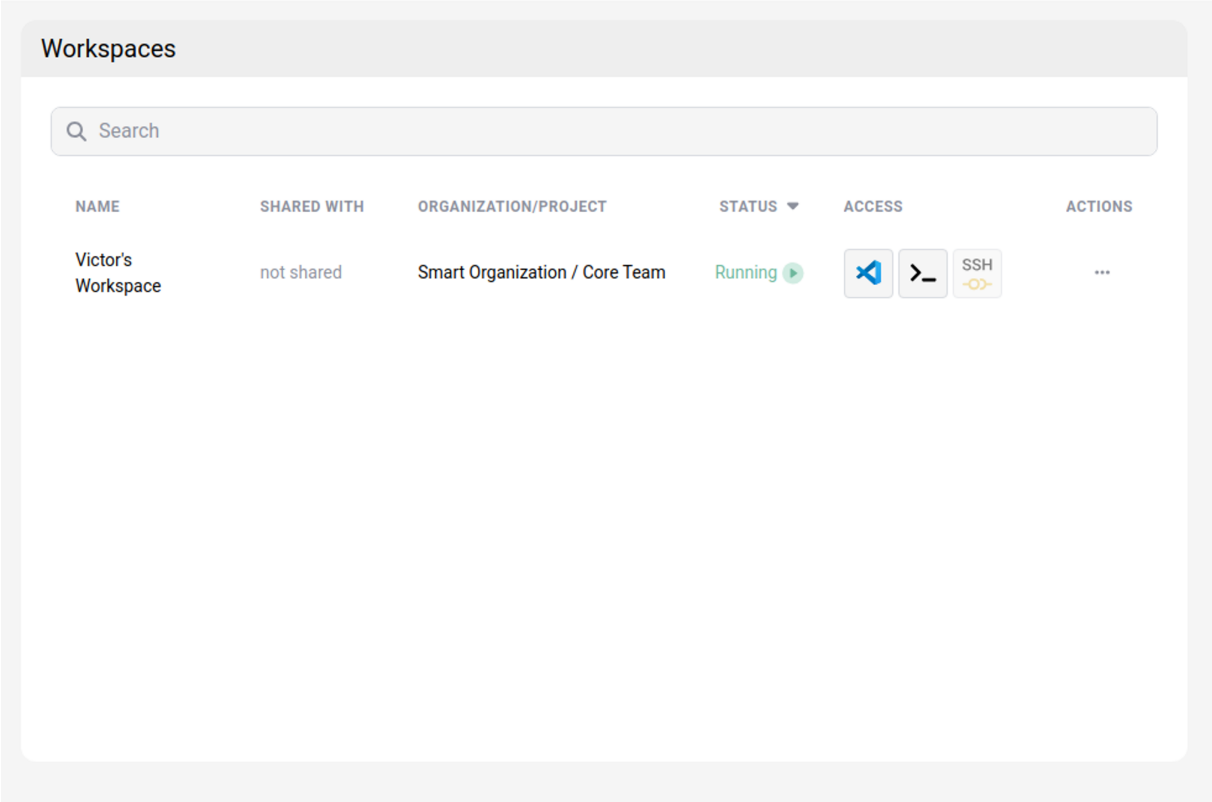


> Tip:
>
> Workspaces will pause automatically when not used for over a pre-set time, typically 60 minutes, depending on the setup of your platform.
>
> When a workspace is paused voluntarily, it will not be impacted by the schedule.

Workspaces

In the **Workspaces** section, you can find details about your individual workspaces across all projects that you are a part of. By selecting the "**...**"option on a specific workspace, you can directly perform actions such as running, pausing, editing, viewing details, or deleting the workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Workspaces pending deletion

Your workspaces are listed in two sections. **Active workspaces** lists the workspaces you can use now. **Workspaces pending deletion** lists your own workspaces that have been moved to the recycling bin and can
still be recovered.

A workspace usually reaches this section because your administrator configured a
custom action that removes workspaces
after a period of inactivity. Deleting a workspace yourself also places it here.

Each entry shows the date and time at which the workspace is permanently deleted, in UTC.

> **Note**
>
> The date shown is the deadline for permanent deletion, not the date the workspace was moved to the recycling
> bin. After that time the workspace cannot be recovered.

To recover a workspace, select **Restore** and confirm. The workspace returns to **Active workspaces**.

Restoring counts against the maximum number of workspaces you are allowed to have. If you are already at your

© 1997–2026 Citrix Systems, Inc. All rights reserved.

limit, delete or release another workspace first.

To remove a workspace immediately rather than waiting for the deadline, select **Delete** and confirm. Permanent deletion cannot be undone.

Email notifications about pending deletion

When a custom action moves one of your workspaces to the recycling bin, SecurSpaces emails you. You receive:

- A message when the workspace is moved to the recycling bin.
- A reminder 24 hours before the workspace is permanently deleted.

Both messages name the workspace and its project, give the permanent deletion time in UTC, and link to this
page. If you restore the workspace, the reminder is cancelled.

Restore a workspace as an administrator

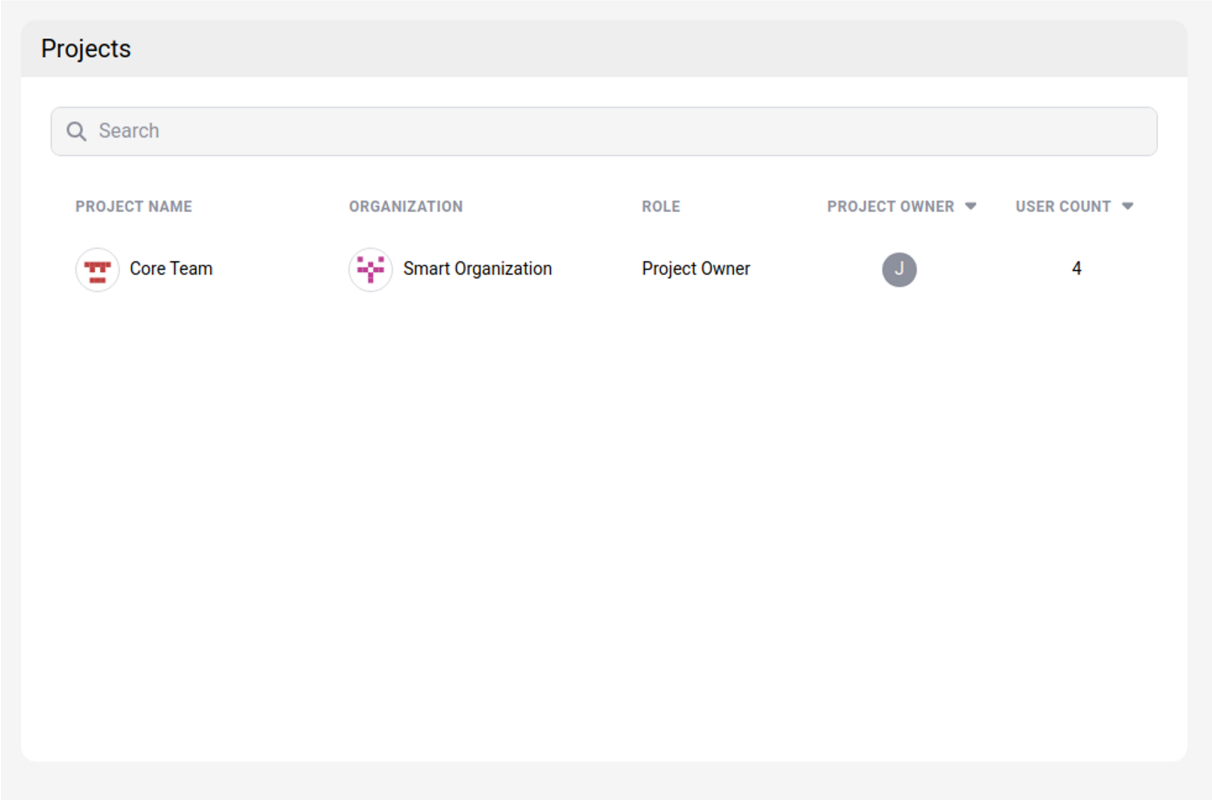
Project owners, platform administrators, and Security Officers can restore any workspace in their scope, rather
than only their own. Open **Deleted Workspaces** from the project settings or the platform workspace settings,
find the workspace, and select **Restore**.

For the retention period that controls how long a deleted workspace stays recoverable, see Workspace Settings.

Projects

The **Projects** section displays information about every project that you are a member of, within the organizations to which you belong. This includes details such as the project name, the organization hosting the project, your role within the project, the project owner, and the number of users involved in the project. By clicking on a project's name, you can access its dashboard for more information.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Related information

- Configuration
- Integrations
- Security
- Your profile

Integrations

In the **Integration Page** you can manage the different access keys, secrets and tokens that are linked to the user's profile.
This includes **Code Repository Tokens**, **Personal SSH Identity** and **GPG Keys**. The keys and tokens are used to authenticate and authorize access to different services, such as remote repository applications. By managing their keys, tokens and secrets in one location, users can easily keep track of which ones are being used, for what purpose and can revoke or add new ones as needed. The page also allows the user to view, create, and remove them, to manage access levels and to have an overview of their expiration date. This helps to ensure that only authorized users have access to the necessary resources and services, and that access is revoked when necessary.

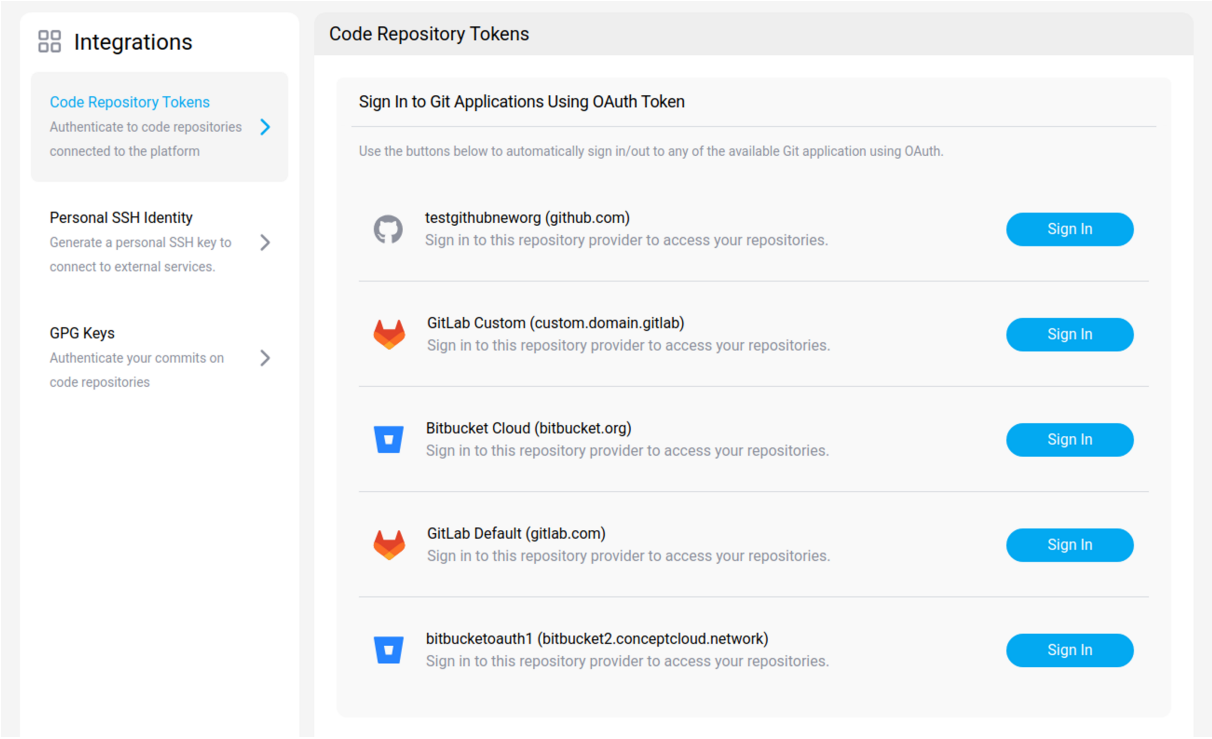
© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Code Repository Tokens
- Personal SSH Identity
- GPG Keys
- Citrix Integration

Code Repository Tokens

Under **Code Repository Tokens**, you can configure authentication, using **OAuth Authentication Tokens** or **Personal SSH Keys**, to the following git providers:

- **GitHub**,
- **GitLab**,
- and **Bitbucket**.
  For certain of these git providers, you have the option to choose between the 'Default'or 'Internal'options. An 'Internal Service'is self-hosted, whereas a 'Default Service'is hosted on the cloud.



For how repositories are added as project resources, see
Code repositories.

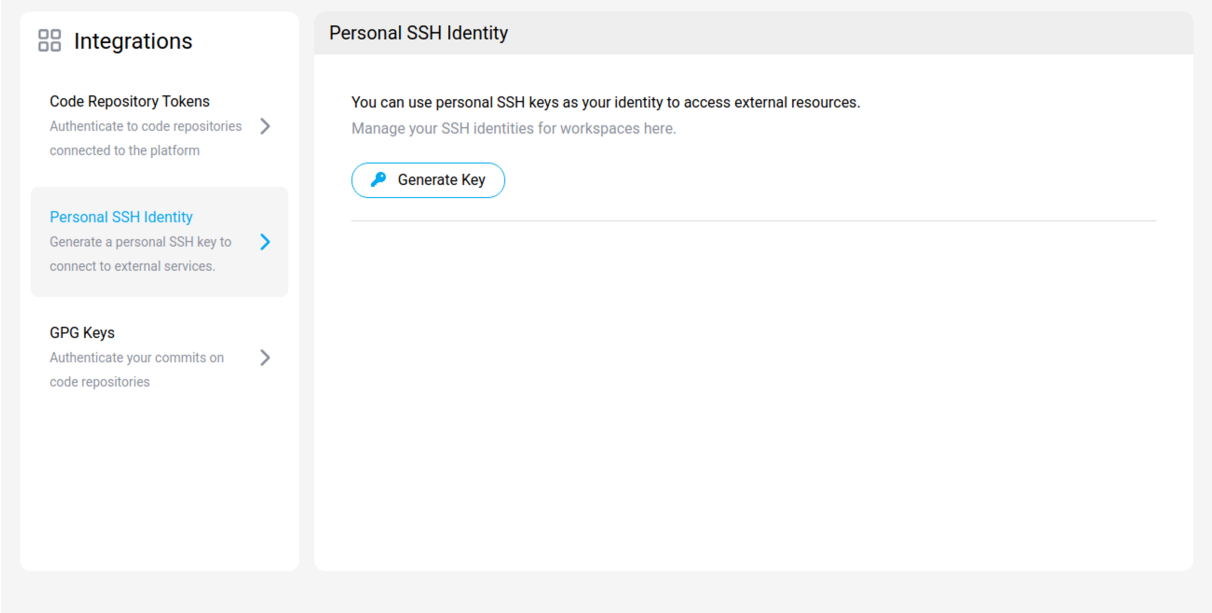
Personal SSH Identity

Use the Personal SSH identity option to authenticate with external resources over SSH.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

To set this up, select Generate Key. The system creates a public/private key pair and displays the public key. Add this public key to your external SSH service as an authentication option.
When you access the external service, Citrix SecurSpaces™ automatically applies the private key to your session. This eliminates the need for manual key management and prevents malicious code in the workspace from extracting the key. **Note:** The SSH key applies to both new and existing workspaces.

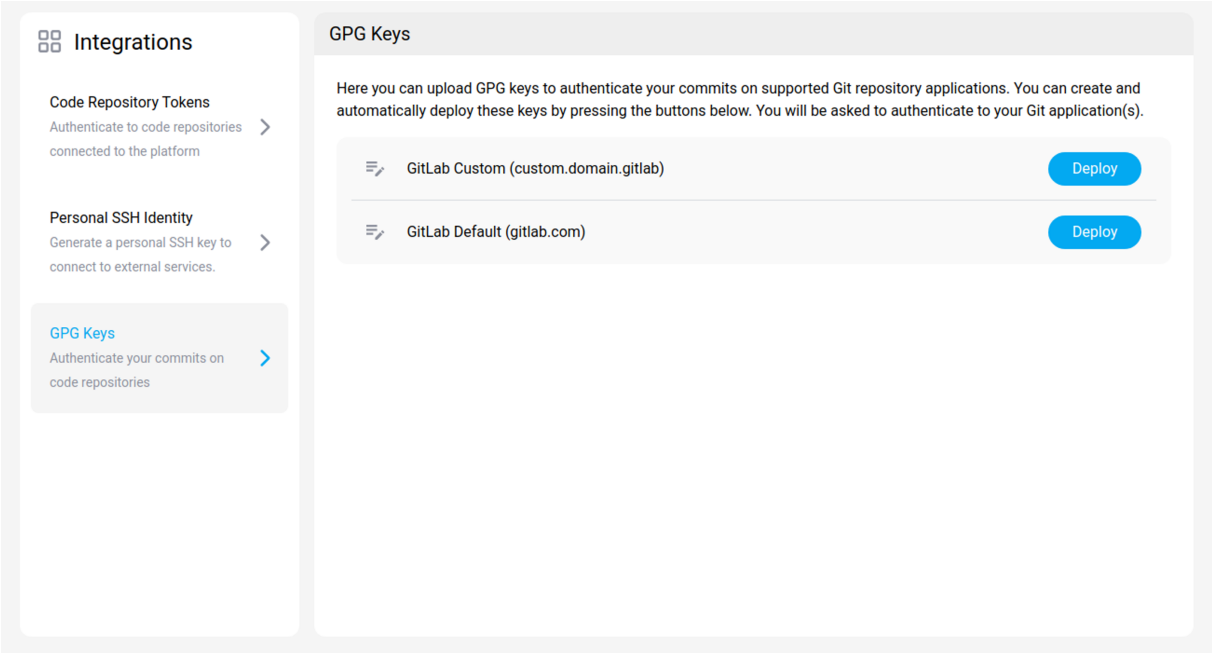
> **Important:**
>
> To connect to SSH services, you must either configure a personal SSH identity or set up the SSH service as a project resource.



GPG Keys

You can generate and automatically deploy GPG keys to authenticate your commits on supported Git repository applications (i.e. GitHub).

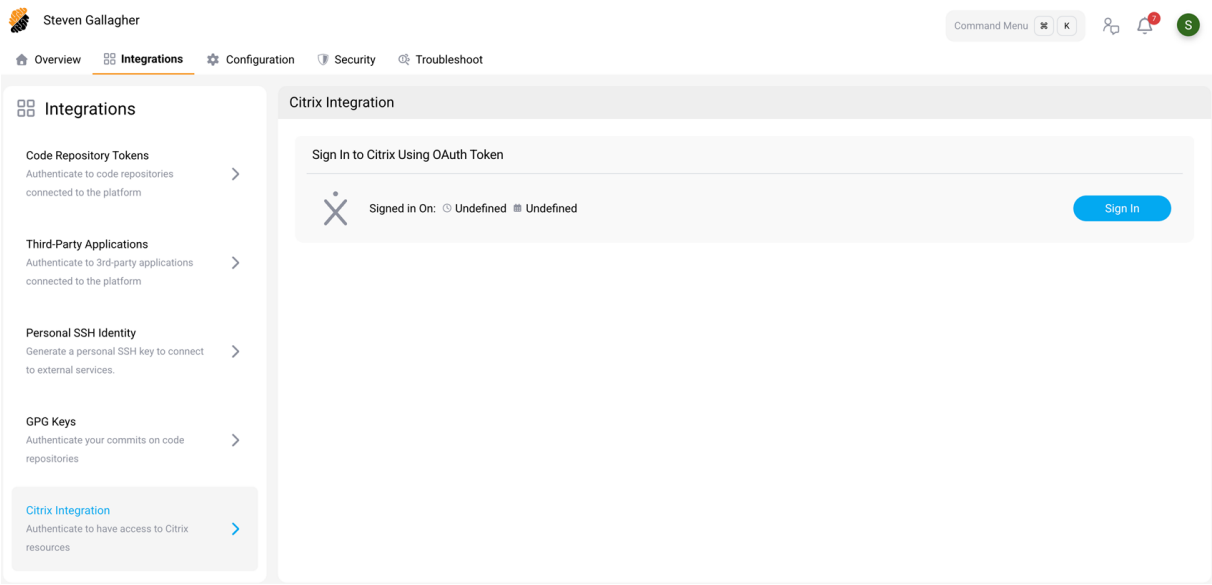
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Citrix Integration

The **Citrix Integration** section on the Integrations page lets you sign in to your Citrix account to access Citrix DaaS virtual apps and desktops directly from the SecurSpaces console. After you sign in, your assigned Citrix resources appear on the project Overview page.

This section is available at **Profile > Integrations > Citrix Integration**.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

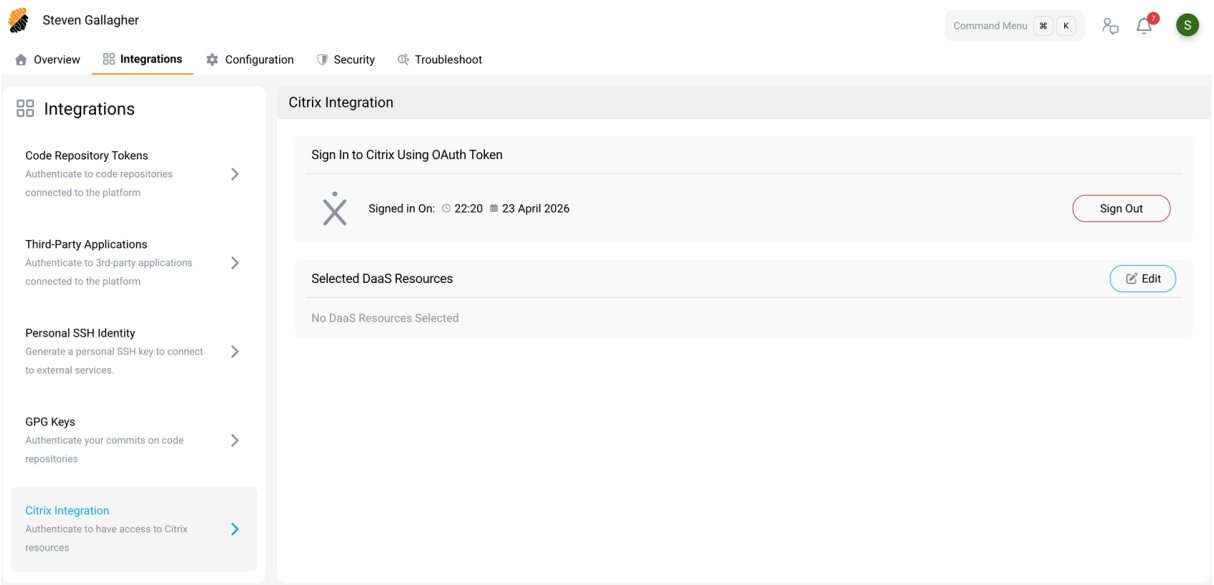
Prerequisites

- Your SecurSpaces platform administrator must first configure the Citrix DaaS integration under **System Configuration > Integrations > Citrix DaaS Integration**. See Citrix DaaS Integration.
- You need a Citrix Workspace account with virtual apps or desktops assigned to you.

Sign in to Citrix

1. Go to **Profile > Integrations**.
2. In the left sidebar, select **Citrix Integration**.
3. Under **Sign In to Citrix Using OAuth Token**, select **Sign In**.
4. Complete the authentication in the Citrix Workspace sign-in dialog that opens.

After you sign in, the page displays your sign-in timestamp and a **Sign Out** button.



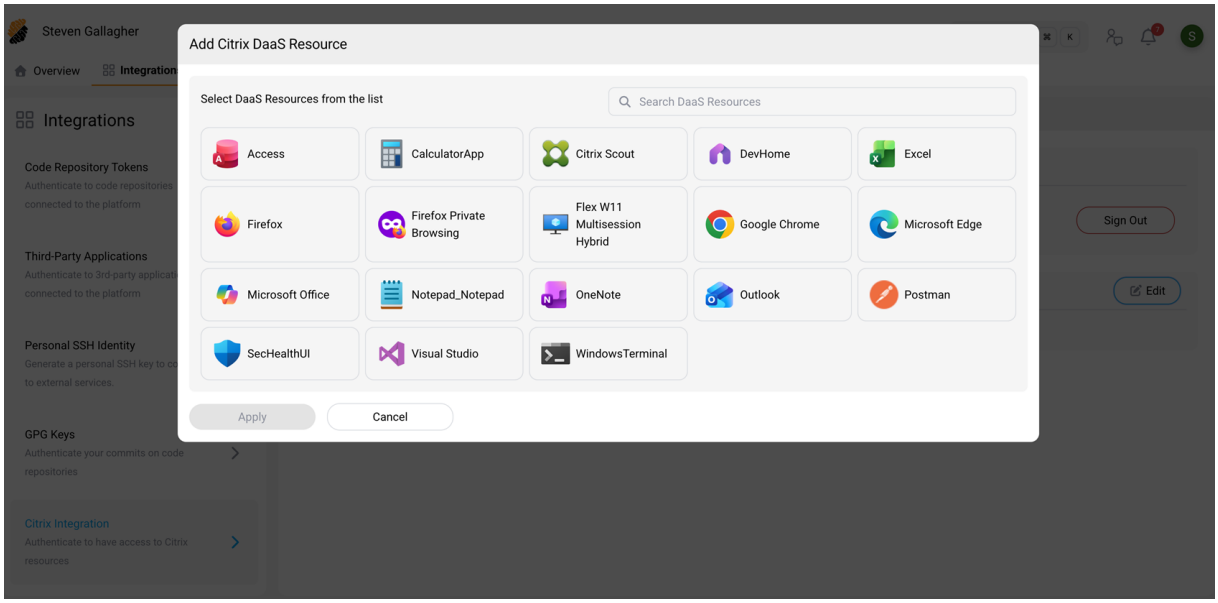
Select DaaS resources

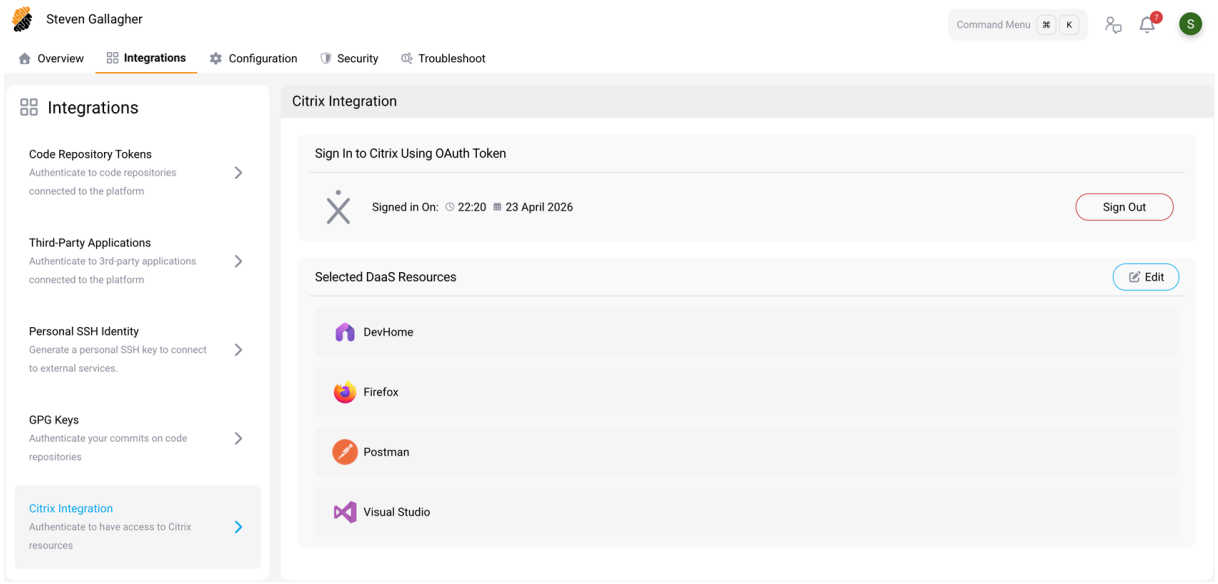
After you sign in, a **Selected DaaS Resources** section appears below the sign-in area. Use this section to choose which Citrix virtual apps and desktops you want to access from the SecurSpaces console.

1. Select **Edit** next to **Selected DaaS Resources**.
2. In the **Add Citrix DaaS Resource** dialog, browse or search for available resources.
3. Select one or more resources from the list.
4. Select **Apply**.

The selected resources now appear under **Selected DaaS Resources** on this page and in the **Citrix DaaS Resources** section on the project Overview page.

© 1997–2026 Citrix Systems, Inc. All rights reserved.





Launch Citrix resources

After you select resources, they appear on the project Overview page under **Citrix DaaS Resources**. Select a resource to launch it. See Citrix DaaS Resources.

Sign out

To disconnect your Citrix account, return to **Profile > Integrations > Citrix Integration** and select **Sign Out**. Your selected Citrix DaaS resources will no longer appear on the project Overview page.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Configuration
- Overview
- Security
- Your profile

Configuration

August 13, 2026

The **Configuration Page** is used to create and edit custom shell scripts and configuration files, configure your IDE and define personal workspace startup scripts. You can also configure additional settings (e.g. theme) by clicking on the profile picture on the top right of the screen.

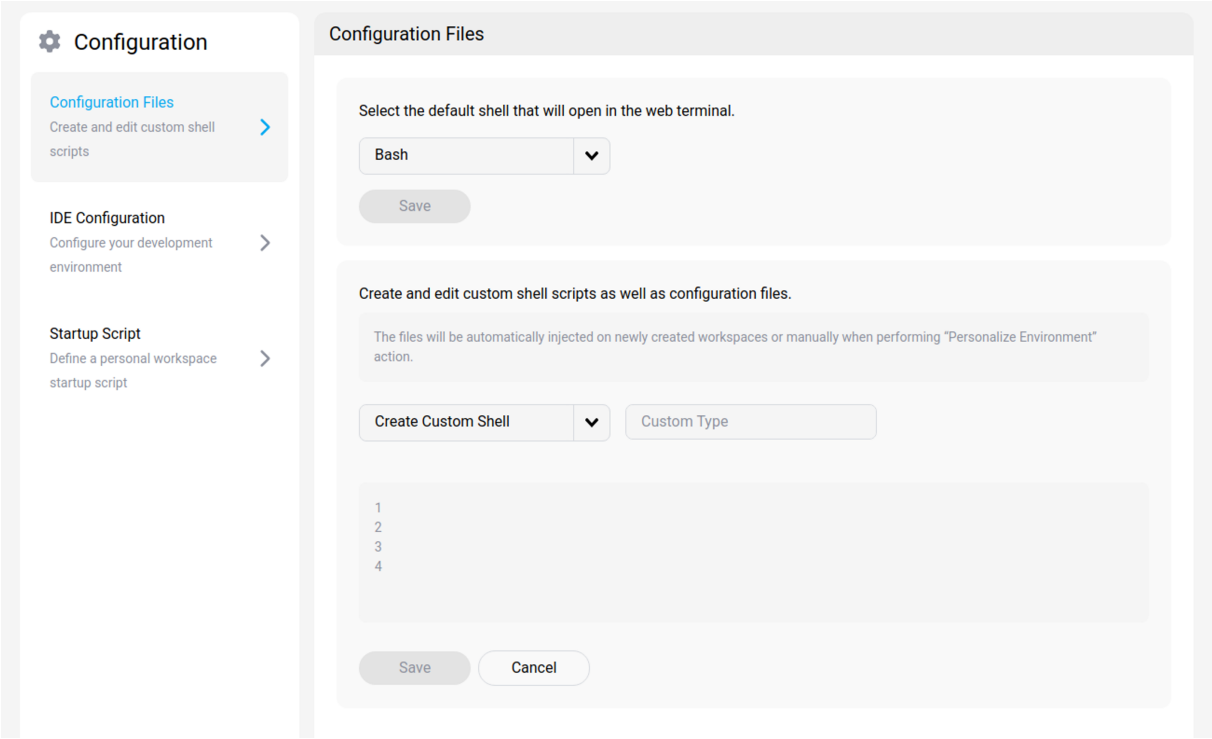
- Manage Configuration Files
- Dot files
- IDE Configuration
- Startup Script
- Theme
- Language

Manage Configuration Files

Use the **Configuration Files** section to personalize your development environment. You can specify your preferred default shell and create custom configuration scripts (such as .bashrc or .zshrc) that automatically apply to your workspaces.

This ensures that every new workspace you create includes your preferred aliases, environment variables, and tool settings without requiring manual setup each time.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Set the default shell

You can define which shell launches automatically when you open a web terminal in your workspace.

1. Go to **Configuration > Configuration Files**.
2. Under **Select the default shell that will open in the web terminal**, select the dropdown menu.
3. Choose your preferred shell (for example, **Bash**).
4. Select **Save**.

Create and edit custom configuration files

You can create custom shell scripts and configuration files to persist your environment settings. These files are automatically injected into:

- **Newly created workspaces**: The files are applied during the workspace creation process.
- **Existing workspaces**: You can manually apply updates by selecting the **Personalize Environment** action within a workspace.

To create a new configuration file

1. Go to the **Configuration Files** section.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. In the **Create and edit custom shell scripts** area, select the**Create Custom Shell** dropdown menu.
3. Select the type of file you want to configure (e.g. **Bash Shell** or **Zsh Shell**).
    - If you select **Custom Type**, enter the specific filename in the adjacent field.
4. Enter your script or configuration code in the editor pane.
5. Select **Save**.

Sample configurations

You can copy and paste the following examples into your **.bashrc** file to improve productivity and safety.

Improve directory navigation Use these aliases to list files with more detail (including hidden files and file sizes) and to navigate folders quickly.

```
# List all files including hidden ones
alias ll='ls -alF'
alias la='ls -A'

# Go back one or two directories
alias ..='cd ..'
alias ...='cd ../..'
```

Prevent accidental deletions Add safety prompts to critical commands to ensure you confirm before deleting or overwriting files.

```
# Ask for confirmation before executing
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Configure Git shortcuts Reduce repetitive typing for common Git operations.

```
# Check status
alias gs='git status'

# Add all changes
alias ga='git add .'

# Commit with a message
alias gc='git commit -m'

# Push changes
alias gp='git push'
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Dot files

Use **dot files** to apply your personal shell, editor, and tool configuration to Citrix SecurSpaces™ workspaces. You connect a Git repository once, and SecurSpaces clones it to `~/.dotfiles` in each selected
workspace.

Dot files is a per-user setting. It uses your connected Git integration and affects only workspaces that you
own.

Before you begin

Before you configure dot files, make sure that:

- Your dot files repository is available through GitHub, GitLab, Bitbucket, Azure DevOps, or Azure DevOps
  with Entra ID.
- You connected the matching Git provider in **Profile > Integration**.
- Your repository includes one of the supported install scripts.

SecurSpaces looks for the first install script that exists in this order:

1. `install.sh`
2. `install`
3. `bootstrap.sh`
4. `bootstrap`
5. `setup.sh`
6. `setup`
7. `script/bootstrap`
8. `script/setup`

If the file is executable, SecurSpaces runs it directly. Otherwise, SecurSpaces runs it with `bash`.

Configure dot files

1. Go to **Profile > Configuration > Dot Files**.
2. Select the **Code repository integration** that can access your dot files repository.
3. Enter the **Dot-files repository URL**, or select **Browse repositories** and choose the repository.
4. Select the **Install-script timeout**. The timeout can be from 1 to 10 minutes.
5. Under **Apply dot files to**, select one of the following options:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **All my workspaces** —Apply dot files to every workspace you own.
- **Only selected workspaces** —Apply dot files only to the workspaces you select.

6. Select **Save Changes**.

Expected result: The next time an included workspace starts or resumes, SecurSpaces clones the repository
to ~/`.dotfiles` and runs the install script.

Apply dot files to an existing workspace

To reapply dot files without waiting for a workspace restart:

1. Open the workspace actions menu.
2. Select **Personalize Environment**.

Expected result: SecurSpaces runs the dot files install script again. Use this action after changing your
Dot files repository or after updating the dot files configuration.

Review dot files output

Dot files output appears in the workspace startup logs under **Applying dot files**. The log shows clone output, script output, and the script exit code.

Dot files does not block workspace startup. If the repository cannot be cloned, no install script is found, the
script exits with an error, or the timeout expires, SecurSpaces records the result in the logs and continues
starting the workspace.

Change or remove dot files

You can update or remove your dot files configuration at any time:

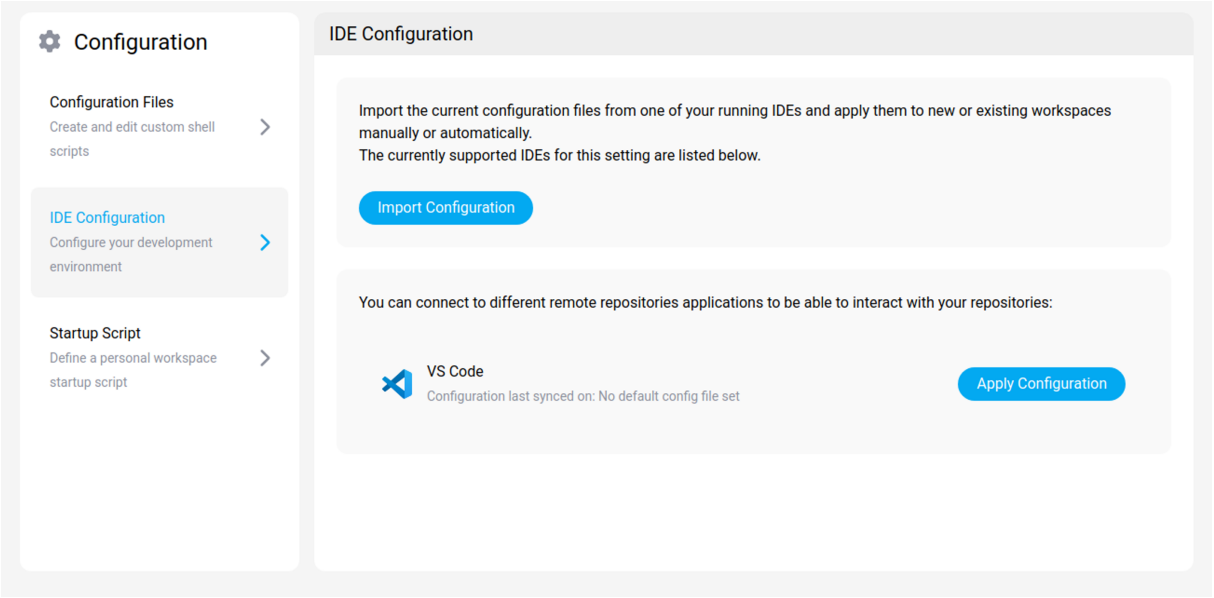
- To use a different repository, update the **Dot files repository URL** and select **Save Changes**.
- To stop applying dot files, select **Remove dot files configuration**.
- To limit where dot files run, choose **Only selected workspaces** and select the workspaces to include.

Removing the configuration stops future dot files runs. It does not delete an existing ~/`.dotfiles` directory
from any workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

IDE Configuration

IDE configuration files can be managed from the **profile settings**. A configuration must be initially imported from a **running** workspace.
Then, it can be applied to new or existing workspaces manually or automatically.



Currently supported IDEs are:

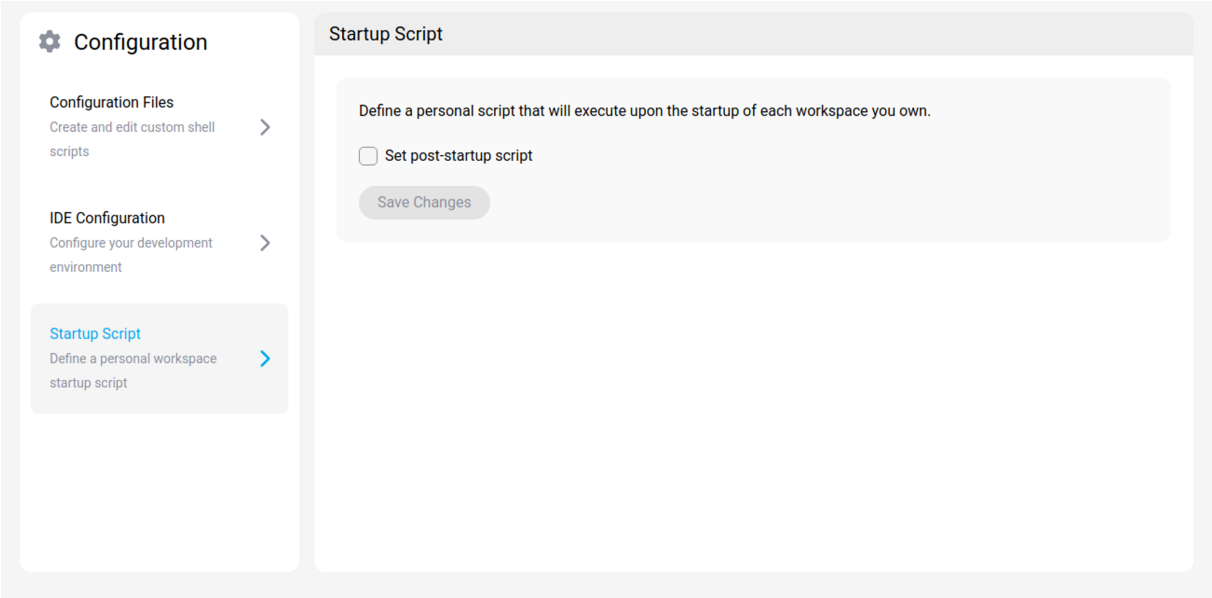
- **VSCode**,
- **any IDEs from Jetbrains**.

Startup Script

You can define a personal script that will be executed upon each startup of the workspaces that you own.
For scripts set on an individual workspace, see
Startup Scripts.

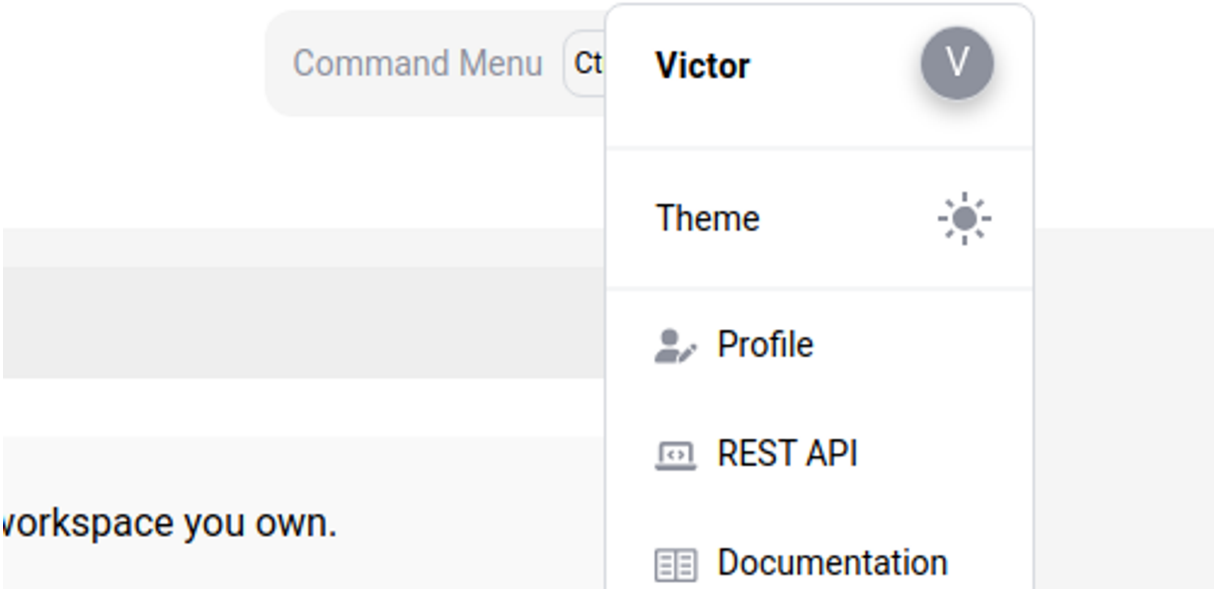
> Tip:
>
> Note that if you defined a startup script for a given workspace, then it will override this one

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Theme

Two color themes for dashboards are available in the **Profile Menu**. You can switch between a **light** and **dark** theme for the User Interface (UI) display.



Language

A language for the UI can be selected from the footer. Supported languages for the platform UI are:

- **English**,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **French**.

Related information

- Integrations
- Overview
- Security
- Your profile

Security

The **Security** page of your profile is where you manage the keys, secrets, and tokens tied to your user account, and where you review what has access to your account.

Managing them in one place makes it easy to see what exists, what it is for, and to revoke or replace it.

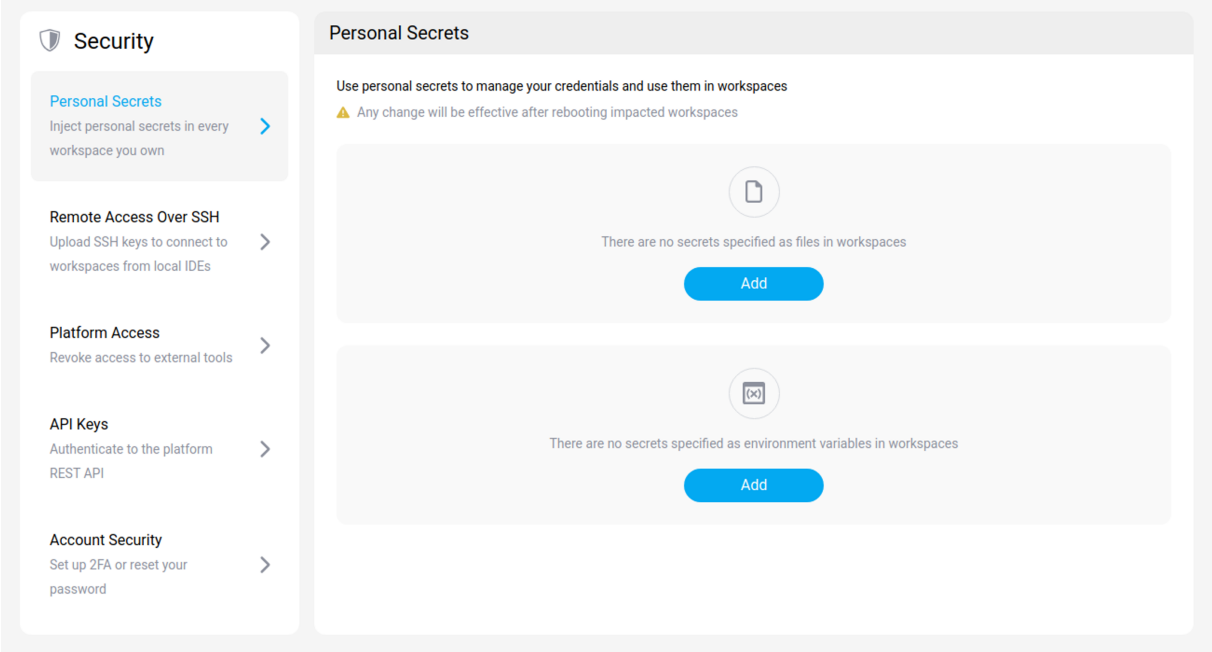
Section	What it is for	Shown when
Personal Secrets	Inject personal secrets into every workspace you own	Always
Remote Access Over SSH	Upload SSH keys to connect to workspaces from local IDEs	SSH to workspaces is enabled for you and allowed at platform level
Platform Access	Revoke access granted to external tools	Always
API Keys	Authenticate to the SecurSpaces REST API	The REST API is enabled
Account Security	Set up two-factor authentication or reset your password	You have two-factor authentication enabled, or you sign in with a local account rather than an identity provider

> **Note:**
>
> GPG keys are not on this page. You manage them under Integration, alongside your repository tokens and personal SSH identity, because they are used to sign commits to your Git provider.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Personal Secrets

Under **Personal Secrets**, you manage the secrets that are injected into every workspace you own. You can add
a secret as a file in the workspace, or as an environment variable.



Remote Access Over SSH

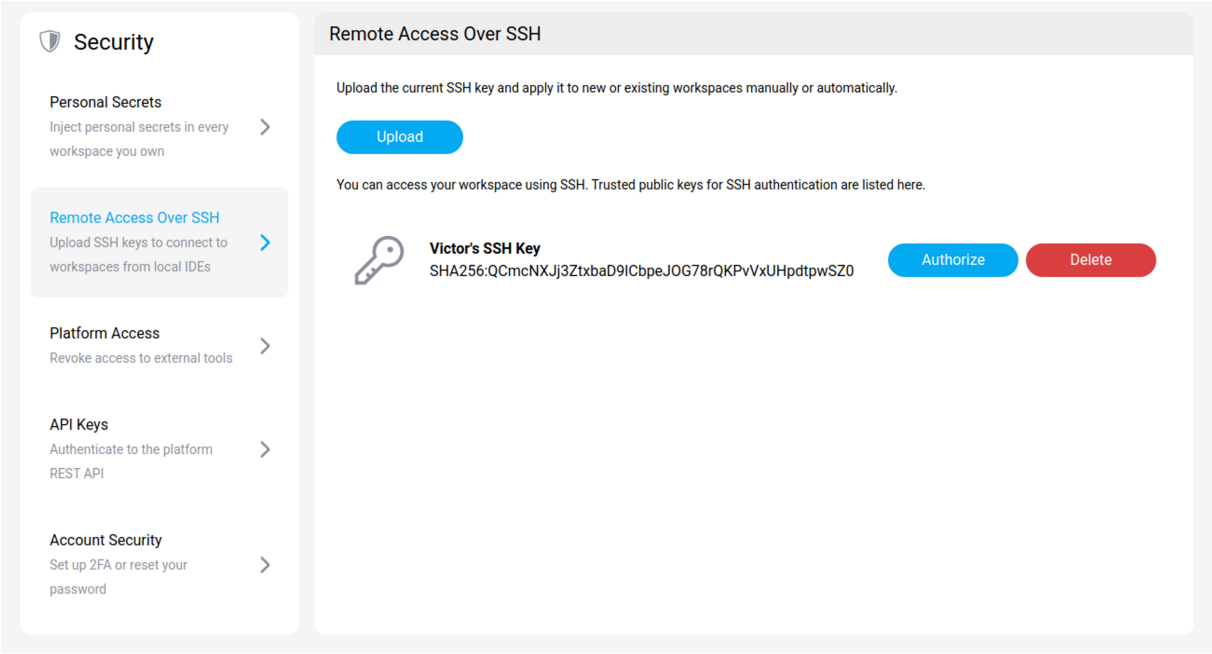
Trusted public keys for SSH authentication are listed here, each linked to your profile. Upload a key to connect to a workspace from a local IDE.

A registration expires after a period your platform administrator sets, 7 days by default. The SecurSpaces
IDE integrations renew it for you; with any other SSH client you re-register the key yourself. See Key expiry and renewal.

Running your IDE locally over SSH lets you keep your familiar local setup while the code and the compute stay
on the remote workspace. To set it up, see
Connect over SSH.

This section appears only when SSH to workspaces is enabled for your account and allowed at the platform
level. If you cannot see it, ask a platform administrator whether remote development over SSH is turned on.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Platform Access

Platform Access lists the external tools you have granted access to your SecurSpaces account, and lets you
revoke that access.

The IDE extension appears here after you sign in from a local IDE. Revoking an entry ends that tool's access
and it must be authorized again before it can be used.

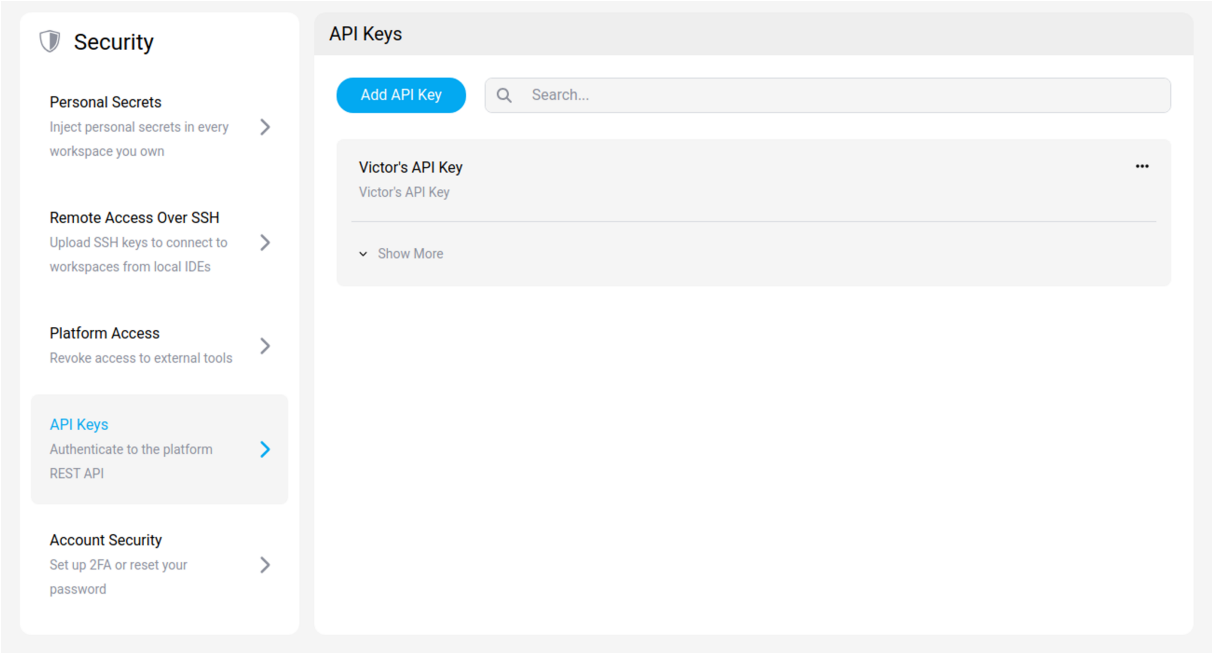
API Keys

An **API key** authenticates you to the SecurSpaces REST API. Use one when a script or an external system
needs to call the API on your behalf.

An API key carries your identity, so treat it like a password: keep it secret, and revoke it here if it is exposed or no longer needed.

This section appears only when the REST API is enabled for your account.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Account Security

Account Security is where you set up two-factor authentication or reset your password.

It appears only if you already have two-factor authentication enabled, or if you sign in with a local SecurSpaces account. If your organization uses an identity provider such as SAML or OIDC, your password and
multi-factor authentication are managed there instead, and this section is not shown.

Related information

- Integration —repository tokens, personal SSH identity, and GPG keys
- Connect over SSH
- REST API

Persistent Docker

Overview

Persistent Docker lets developers keep pulled Docker images on their workspace's persistent volume. With the feature enabled, images that a developer downloads inside a workspace are cached under `/home/developer/` and remain available after the workspace is paused, restarted, or updated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Without Persistent Docker, the Docker data root lives on the workspace's ephemeral root filesystem. Every time the workspace container is recreated —on pause/resume, restart, or image update —the Docker image cache is discarded and any image referenced by the developer's tooling has to be downloaded and unpacked again.

To check which applies to a workspace, run `docker info` and look at **Docker Root Dir**. A path under `/home/developer` means Persistent Docker is active for you.

What problem it solves

Pulling images on every workspace start can dominate ramp-up time, especially when:

- A developer's workflow depends on several large images (databases, browsers for end-to-end tests, language toolchains, CI runners).
- Images have many layers and the workspace is sized with only one or two CPUs, so layers can only be unpacked sequentially.
- The registry rate-limits or throttles pulls (for example, public Docker Hub limits per IP or per organization).
- The cluster has limited or metered egress to the registry.

In environments where multiple images are pulled per session, the cost can be significant —a workspace that takes seconds to resume can still leave the developer waiting 20+ minutes before their tooling is usable.

Persistent Docker eliminates the redundant pull-and-unpack work after the first time an image is used. The workspace itself still starts in the same amount of time; what improves is the **time-to-productive** —how long until the developer's Docker-based tooling is actually ready.

> **Note:**
>
> Persistent Docker caches **images**, not container state. Data written inside a running container is still discarded when the container is removed. To persist data produced by a container (a database file, application state, generated artifacts), mount a directory under `/home/developer/` into the container as a Docker volume. See What persists in a SecurSpaces Workspace.

How to get access

Persistent Docker is gated by a license feature flag. By default, the flag is not enabled on customer licenses.

To request access:

1. Contact Citrix Support or your account team and ask for **Persistent Docker** to be enabled on your Citrix SecurSpaces™ license.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Citrix enables the feature flag on the license record for your domain.
3. A platform administrator in your SecurSpaces environment refreshes the license so the new flag is picked up:

    - **Online license** —Open the SecurSpaces admin console, go to **System Configuration → License**, and select **Upload New License → Online License**. The refreshed license includes the Persistent Docker feature flag.
    - **Air-gapped (file) license** —Citrix provides a new license file. Upload it under **System Configuration → License → Upload New License**.

After the license refresh, **Persistent Docker** appears as a new entry in the platform feature flags list, alongside other gated features such as **GPU Support**.

Configure Persistent Docker in SecurSpaces

Once the feature flag is visible in your environment, a platform administrator controls who can use it.

Enable the feature flag

1. Sign in to the SecurSpaces admin console as a **platform administrator**.
2. Go to **System Configuration → Platform Features**.
3. Locate **Persistent Docker** in the list of feature flags.
4. Choose the rollout scope:

    - **Enabled** —All developers on the platform get Persistent Docker.
    - **Enabled for selected users** —Only the users, domains, or groups you specify get Persistent Docker. Add entries under **Users**, **Domains**, and **Groups** as needed.

5. Save the configuration.

Apply the change to running workspaces

The feature is applied at workspace start. A developer with a workspace that was already running when the flag was enabled must do one of the following to start caching images:

- Pause and resume the workspace, **or**
- Restart the workspace, **or**
- Update the workspace.

New workspaces created after the flag is enabled automatically use Persistent Docker.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Plan for additional persistent-volume usage

Persistent Docker stores image layers on the workspace's persistent volume. Because developers commonly depend on multiple images —and often multiple versions of the same image —the persistent volume fills up considerably faster than without the feature.

Before enabling Persistent Docker broadly, review your workspace specifications:

1. Go to **System Configuration → Workspace Specifications**.
2. Either:
    - Increase the default disk size on the specifications used by developers who will get Persistent Docker, **or**
    - Enable **Allow users to increase storage** on those specifications so developers can self-size their persistent volume as their image cache grows.

If a developer's persistent volume fills up, image pulls and other workspace operations will start to fail. Plan headroom accordingly.

Verify Persistent Docker is active

A developer can confirm the cache is persisting across restarts:

1. Open a workspace terminal.
2. Pull or run an image, for example:

```
1 docker pull postgres:16
2 docker images
```

3. Confirm the image is listed by `docker images`.
4. Pause and resume the workspace.
5. Open a terminal again and run:

```
1 docker images
```

   Expected result: The `postgres:16` image is still listed without being re-pulled.

If the cache is empty after a restart, Persistent Docker is not active for that developer. Confirm with a platform administrator that the feature flag is enabled and that the developer is in scope (user, domain, or group).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How to persist container data (Docker volumes)

Persistent Docker only caches images. Anything a container writes to its own filesystem is still lost when the container is removed. This is standard Docker behavior; SecurSpaces does not change it.

To persist data produced by a container, mount a folder from `/home/developer/` into the container as a volume. For example, to run Postgres with its data directory backed by your persistent volume:

```
mkdir -p ~/postgres-data
docker run -d \
  --name pg \
  -e POSTGRES_PASSWORD=secret \
  -v ~/postgres-data:/var/lib/postgresql/data \
  postgres:16
```

The database files live under `~/postgres-data`, which is on the persistent volume, so they survive workspace restarts. Use the same pattern for any other stateful container: identify the path the application writes to, and mount a directory from `/home/developer/` to that path.

For a complete overview of what survives across workspace restarts, see What persists in a SecurSpaces Workspace.

References

- What persists in a SecurSpaces Workspace
- Develop in a workspace
- Workspace Specifications
- Custom Workspace Image

Run a graphical desktop in a workspace with the Debian GUI image

Some workflows need more than a browser-based IDE and terminal. When you have to run GUI-based tooling, design applications, or test an application's graphical interface, you can use the sample Debian GUI container image provided by Citrix Citrix SecurSpaces™. This image ships with a preconfigured XFCE desktop that you reach directly in your browser through noVNC.

This guide explains how to make the Debian GUI image available on your platform and how to open its desktop from a workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How it works

The Debian GUI image is a standard SecurSpaces CDE image (container image) that adds:

- An **XFCE** desktop environment.
- A **TigerVNC** server and the **noVNC** client, which streams the desktop to your browser over port **8081**.
- A startup script, `/usr/bin/strong_network_startup/xstartup/initialize_workspace.sh`, that starts the VNC server and the desktop session.

You expose port 8081 to your browser by using either a **Custom Access Item** or a **Workspace App**, and then run the startup script to launch the desktop.

Prerequisites

Before you begin, make sure you have:

- The sample Debian GUI image available to your team. The reference image and `Dockerfile` are published in the Citrix images repository on GitHub. Build and push it to a registry your platform can reach, or use a prebuilt image if your administrator provides one.
- The image added as a CDE image in your platform, organization, or project resources. This action requires the Security permission set to **Manage**.
- Permission to create or edit a workspace or workspace template that uses the image.

Open the desktop with a Custom Access Item

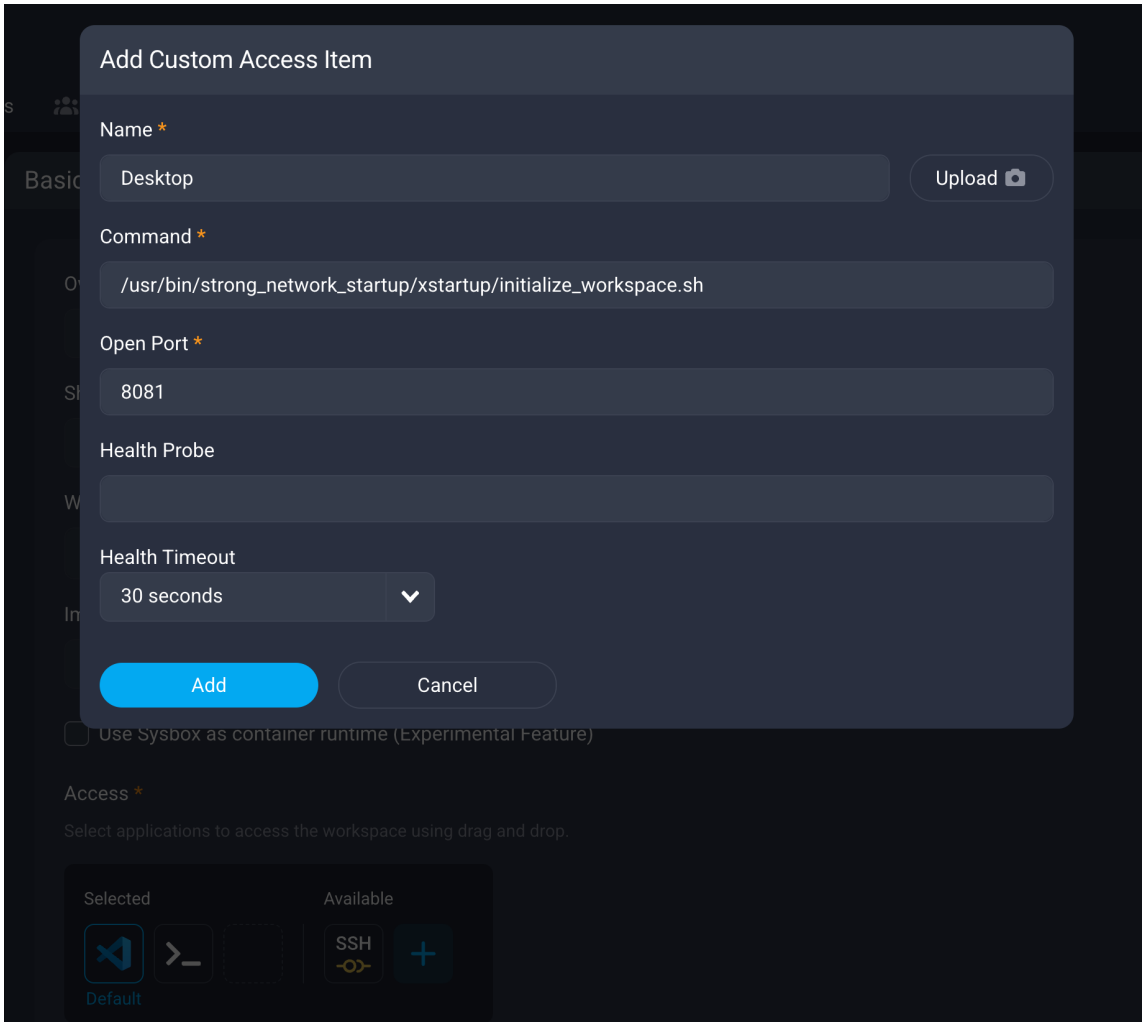
Use a Custom Access Item when you want the desktop to appear as a one-click connection option on the workspace, alongside VS Code and Terminal.

1. Create or edit a workspace that uses the Debian GUI image.
2. In the workspace wizard, go to **Basic Info** and open the **Access** menu.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

3. Click **Add** to open the **Add Custom Access Item** dialog and enter the following:

   - **Name** —A display name of your choice, for example `Desktop`.
   - **Port** —`8081`. This is the port where noVNC runs.
   - **Command** — `/usr/bin/strong_network_startup/xstartup/initialize_workspace.sh`



4. Save the wizard and launch the workspace.
5. Open the desktop by clicking the access item you created.

Expected result: The XFCE desktop opens in your browser through noVNC.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Open the desktop with a Workspace App

Use a Workspace App when you
prefer to start the desktop manually from the terminal, or when you want to
reuse an existing app-based workflow.

1. Create a workspace that uses the Debian GUI image.
2. Expose port **8081** as a workspace app. For details, see Create a Workspace App.
3. Start the workspace, open the terminal, and run the startup script:

```
1 /usr/bin/strong_network_startup/xstartup/initialize_workspace.sh
```

4. Open the workspace app.

Expected result: The XFCE desktop opens in the workspace app.

Start the desktop automatically

To avoid running the command manually each time, automate the steps above:

1. Add the startup command to the workspace startup
   script
   so the desktop launches every time the workspace starts:

```
1 /usr/bin/strong_network_startup/xstartup/initialize_workspace.sh
```

2. Create a workspace
   template that uses
   the Debian GUI image and already exposes port 8081 as a workspace app.

Workspaces created from this template start with the desktop ready to use, with
no manual setup.

Troubleshooting

- **The desktop does not load** —Confirm that port **8081** is exposed (as a Custom Access Item or a Workspace App) and that the startup script has run. With a Custom Access Item, the script runs automatically; with a Workspace App, run it manually unless it is part of the startup script.
- **Clipboard copy and paste does not work** —The image installs `autocutsel` for clipboard support. Verify that any clipboard security

© 1997–2026 Citrix Systems, Inc. All rights reserved.

policy
on the workspace allows the action you are attempting.

- **The image fails validation when added** —A CDE image must include an SSH client and the Git and Git LFS clients, and it must have a `developer` user with UID 1000. The sample Debian GUI image already meets these requirements. See Add a CDE Image.

Install SecurSpaces as a desktop app

Citrix Secure Access with Chrome Enterprise Premium and SecurSpaces provide a high-performance, zero-trust environment for cloud-based development. You can access a full-featured version of Visual Studio Code (VS Code) directly within your web browser.

To optimize your workflow, you can install the SecurSpaces instance as a Progressive Web App (PWA). This allows you to run your development environment in a dedicated, standalone window rather than a browser tab, providing a native desktop experience.

Architecture of SecurSpaces PWAs

When you use Google Chrome or Chrome Enterprise Premium, a Progressive Web App (PWA) acts as a bridge between the local operating system and the secure Citrix cloud container. Unlike a standard browser shortcut, a PWA-installed SecurSpaces instance functions as a standalone application in the OS taskbar and manages its own windowing logic.

Installing the SecurSpaces VS Code instance as a PWA provides the following benefits:

- **Reduced context switching**: Separates development workspaces from general web browsing.
- **Increased screen real estate**: Removes browser UI elements, such as the address bar and bookmarks, to maximize the coding area.
- **System integration**: Enables native OS shortcuts (such as **Alt+Tab** or **Cmd+Tab**) and keeps the IDE active even if you close the main browser window.

Install SecurSpaces as a PWA

Follow these steps to create a dedicated desktop shortcut for your SecurSpaces environment using Google Chrome or Chrome Enterprise Premium.

1. **Access your workspace**. Navigate to your organization's SecurSpaces URL and launch your VS Code web workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. **Install the application**. In the Chrome address bar (Omnibox), select the **Install** icon (the monitor icon with a down arrow) to the right of the URL.

   > **Note:**
   >
   > Alternatively, select the **More** menu (three vertical dots), select **Save and Share**, and then select **Install page as app**.

3. **Configure the app name**. In the dialog box, enter a name for the application (for example, Engineering-Prod-SecurSpaces) to help distinguish between multiple project environments.
4. **Launch and pin the app**. The SecurSpaces instance opens in a new, borderless window. To ensure permanent access, right-click the icon in your Windows taskbar or macOS Dock, and then select **Pin**.

Security for Chrome Enterprise Premium

For organizations using **Chrome Enterprise Premium**, PWA functionality remains governed by corporate policy. IT administrators can perform the following tasks:

- Force-install the SecurSpaces PWA to developer machines using Chrome Policy.
- Apply Data Loss Prevention (DLP) rules that persist within the standalone PWA window.
- Maintain URL filtering to prevent the PWA from navigating to unauthorized external domains.

Using a PWA provides a local-application experience while maintaining the security standards of the Citrix containerized environment.

Work from a virtual desktop

August 19, 2026

`Developer`

Citrix SecurSpaces™ supports **proxy mode**, which lets you keep working on an existing virtual desktop instead of in a workspace. You install the Strong Agent on the VDA, and all source code access is routed through it.

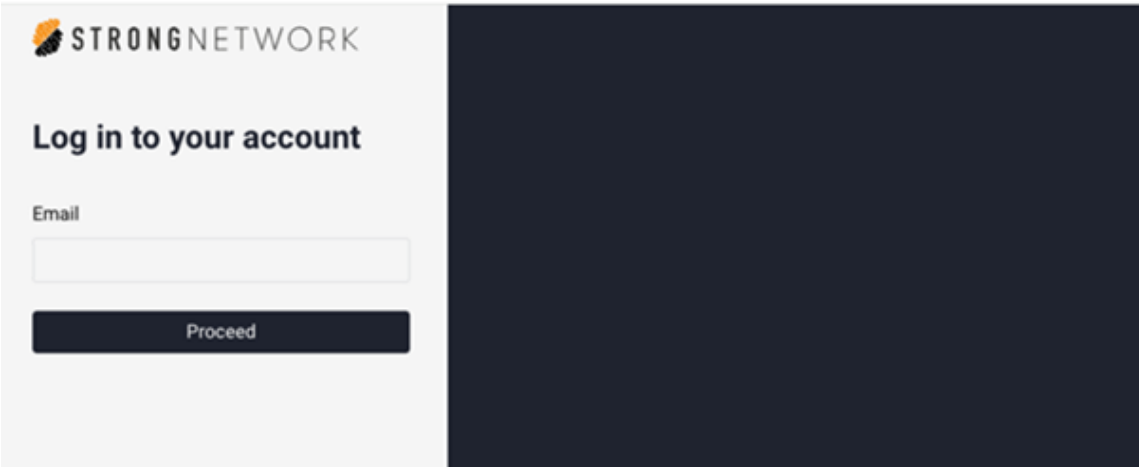
The point of this is credentials. The agent uses the source credentials and keys already held by the platform, so you never create or hold a personal SSH key to reach the repository, and the same access rules
apply as they would inside a workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

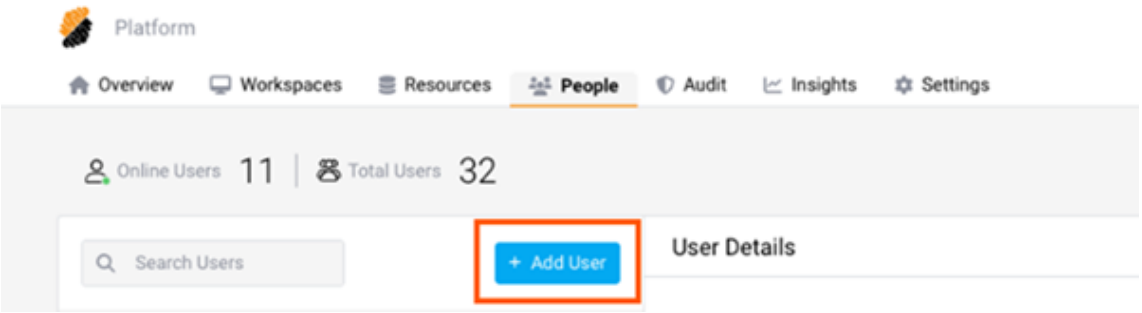
> **Note:**
>
> This procedure applies to Windows VDA, Linux VDA, and the Citrix VDA for macOS.

Set up your profile

1. If you have not used the cloud-native environment before, activate your SecurSpaces account with your
   organization's SecurSpaces instance URL.

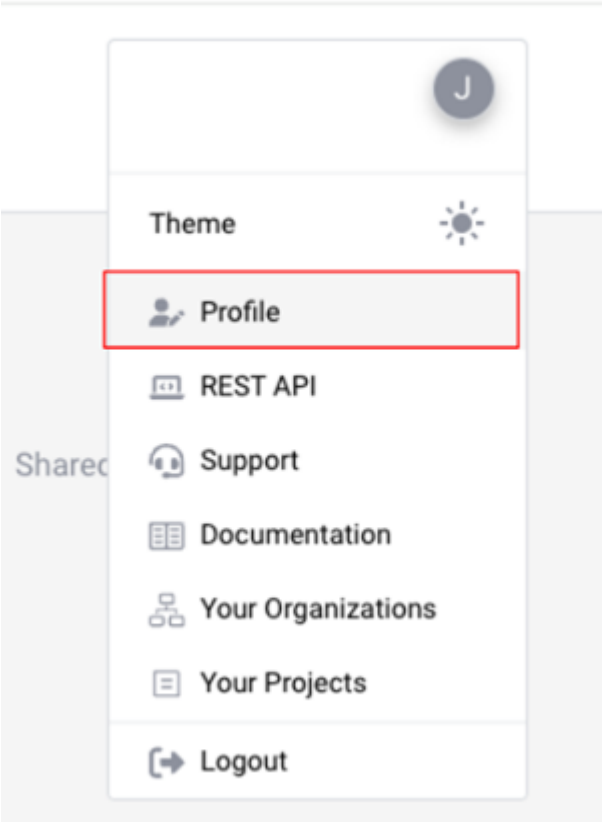


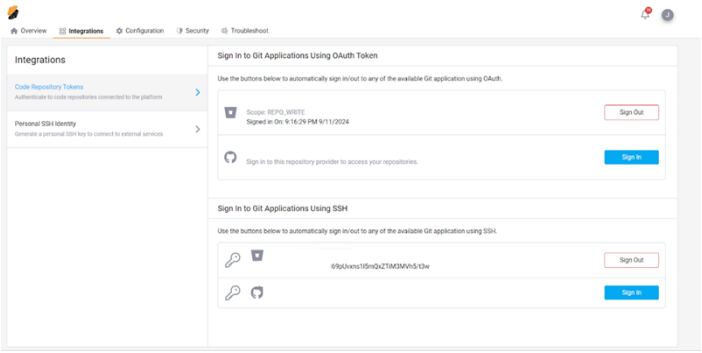
2. Ask your project owner to add you to a project.



3. Configure your profile to set up source code access permission.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

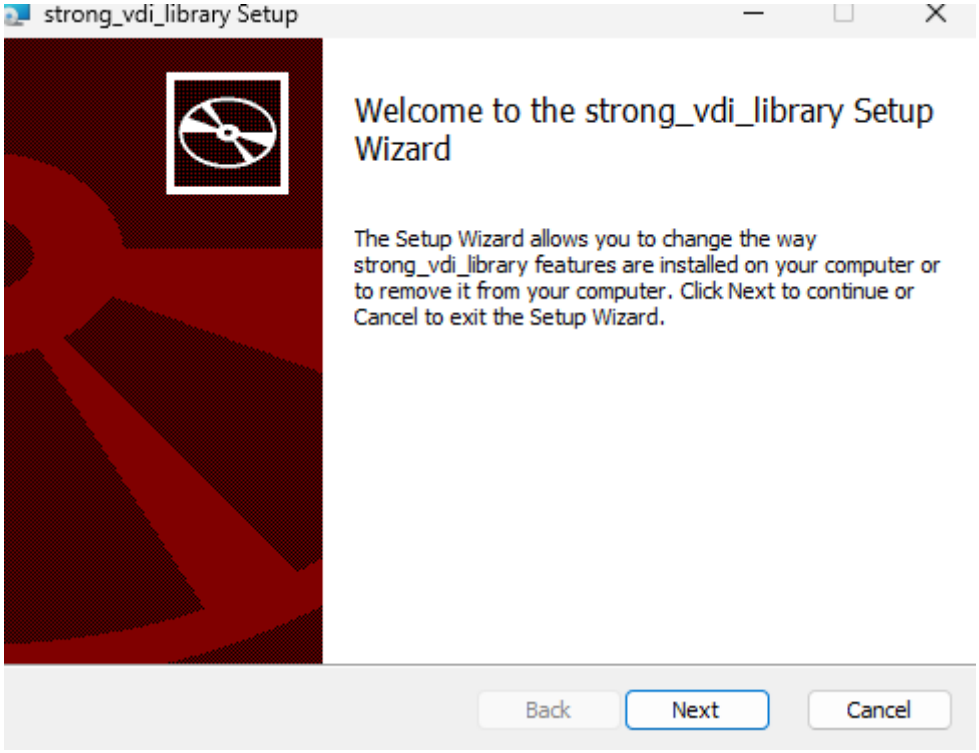


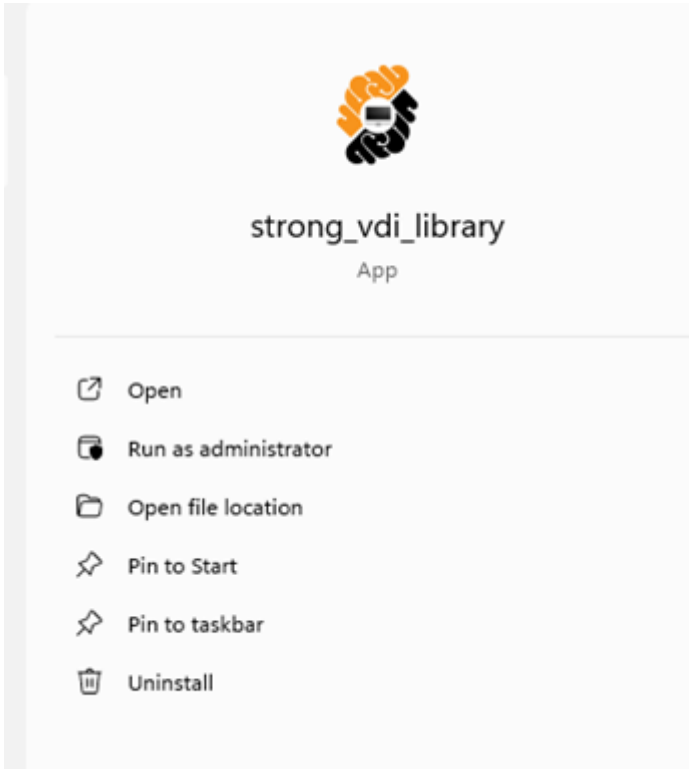


Install the agent

1. Install the Strong Agent, then run the Strong VDI library.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

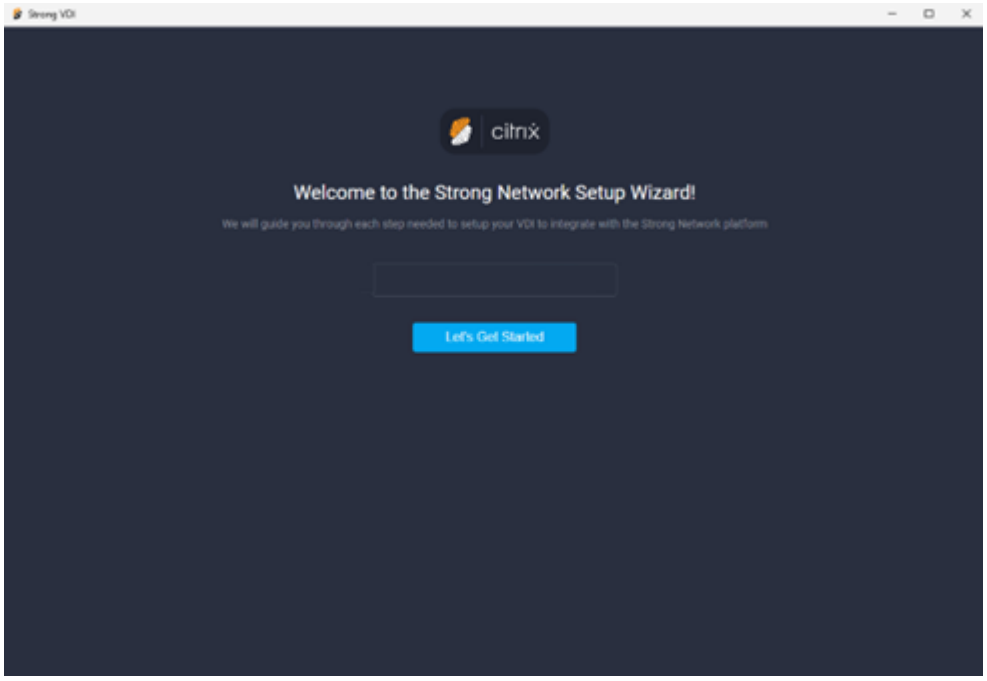




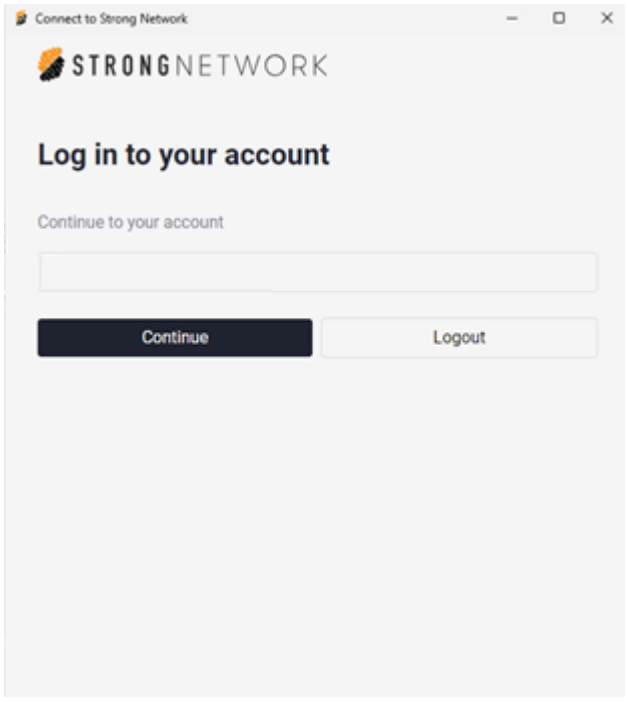
Configure the agent

1. Run the binary and enter your SecurSpaces workspace URL.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

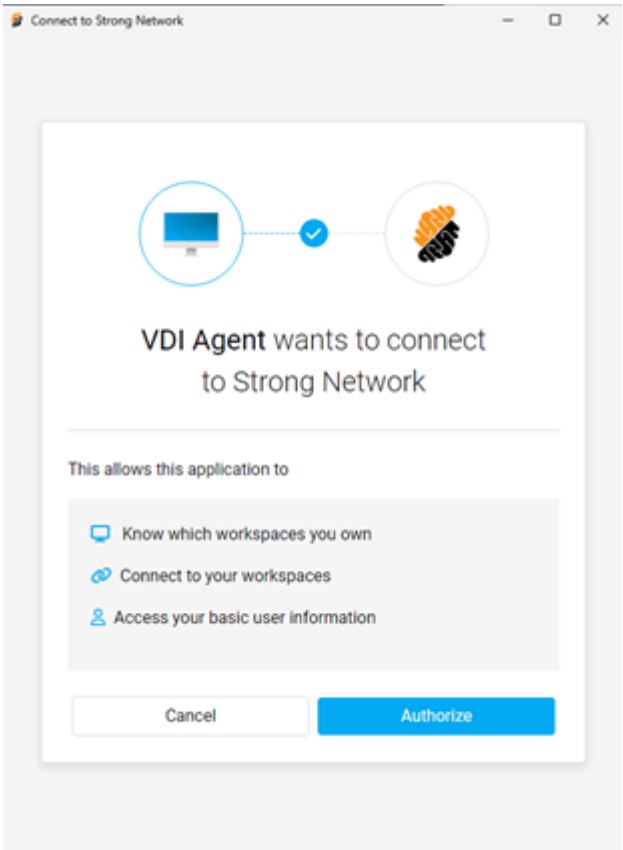


2. Select **Let's Get Started** and sign in to your account.

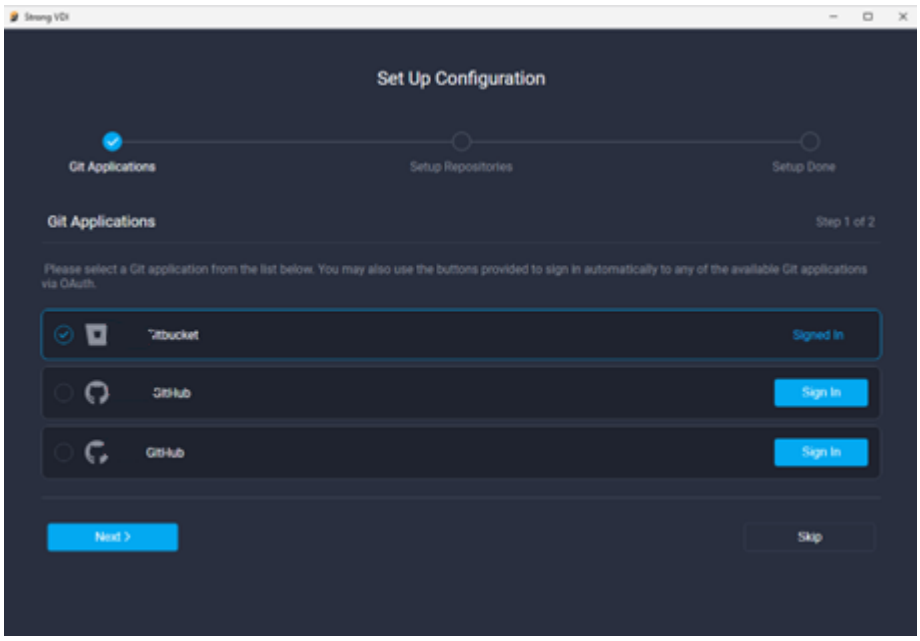


3. Authorize the connection to SecurSpaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



4. Configure the environment and complete the setup.



You can now use any terminal to clone your source code, and a local IDE such as Visual Studio or Xcode for
day-to-day work. SecurSpaces manages the credentials centrally in the background.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Your profile
- Connect over SSH
- Citrix DaaS integration

Give your AI assistant access to the documentation

August 24, 2026

If you use an AI coding assistant —GitHub Copilot, Claude Code, Cursor, Windsurf, or a self-hosted LLM —you can give it access to the full Citrix SecurSpaces™ documentation so it can answer SecurSpaces-specific questions accurately and help you with SecurSpaces tasks.

The documentation is served directly from your SecurSpaces instance, so this works the same way in air-gapped and internet-connected environments. All documentation fetches stay on your network, and the documentation always matches your deployed SecurSpaces version.

Overview

The documentation is published in an AI-readable format based on the llms.txt standard. Two files make this possible:

- **`llms.txt`** —a structured index of every documentation page, so your AI assistant can discover what is available and fetch the right page.
- **`llms-full.txt`** —the complete documentation corpus in a single file, for ingestion into a context window or a retrieval-augmented generation (RAG) system.

Which source to use

Both files are available from two places. Prefer the first if you have a SecurSpaces deployment.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Source	URL	Use it when
Your SecurSpaces instance	`https://<your-securspaces-url>/docs/llms.txt`	You have access to a deployment. The documentation matches your deployed version, fetches stay on your network, and it works in air-gapped environments.
The public copy	`https://assets.sds.citrix.com/llms.txt`	You have no deployment to point at, or your tool cannot reach one. Always the latest published documentation, which may be ahead of your version.

The public copy indexes the same pages on `docs.citrix.com`, so it needs outbound internet access. Use the
instance-hosted copy where you can: it cannot drift from the platform you are actually running.

This guide walks through:

1. Verifying the documentation is available from your SecurSpaces instance.
2. Adding SecurSpaces context to your AI assistant.
3. Testing that your assistant uses the documentation.

Before you begin

- Know your **SecurSpaces instance URL** —the URL you use to log in to the SecurSpaces console (for example, `https://sds.mycompany.internal`).
- Use an AI assistant that supports system prompts or custom instructions.

Step 1: Verify the documentation is available

In your browser, navigate to:

```
https://<your-securspaces-url>/docs/llms.txt
```

You should see a structured list of SecurSpaces documentation pages. If this page loads, you are ready to continue.

If it does not load, contact your platform administrator. If you have no SecurSpaces instance to point at,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

use `https://assets.sds.citrix.com/llms.txt` instead and substitute that URL wherever this guide refers to
your instance.

Step 2: Add SecurSpaces context to your AI assistant

Add the following snippet to your AI assistant's system prompt or custom instructions. Replace <`your-securspaces-url`> with your actual SecurSpaces instance URL.

```
You have access to documentation for Citrix SecurSpaces, the cloud development environment I work in. In SecurSpaces, my code, IDE, and developer tools run in a remote, containerized workspace that I access from a browser or over SSH — not on my local machine.
This workspace IS a SecurSpaces workspace. Treat any reference to "my/this workspace," "my environment," "this machine/container," or my dev tooling as referring to SecurSpaces — even when I don't say "SecurSpaces" explicitly.

Documentation sources:
- Documentation index: https://<your-securspaces-url>/docs/llms.txt
- Full documentation corpus: https://<your-securspaces-url>/docs/llms-full.txt

How to answer SecurSpaces questions:
- Fetch the index first to find relevant pages, then fetch the specific page(s) for detailed answers. For a clearly SecurSpaces-related question, prefer this documentation over general knowledge.
- Cite the specific doc page(s) you used (title or URL) so I can verify the source. If you did not consult the docs for a SecurSpaces answer, say so.
- SecurSpaces workspaces are ephemeral. When advising on installing tools or changing the environment, check what persists and recommend an approach that survives workspace restarts.

Accuracy and uncertainty:
- Base SecurSpaces-specific answers on the fetched documentation. Clearly separate what the docs confirm from your own general knowledge or inference.
- If the documentation does not cover the question, or covers it only partially, say so explicitly rather than filling the gap with assumptions.
- Do not invent SecurSpaces features, configuration options, URLs, CLI commands, API fields, or version numbers. If you are not certain a detail is in the docs, state your uncertainty and point me to the docs index or my platform administrator instead of guessing.
- This documentation matches my deployed SecurSpaces version. If your general
  knowledge conflicts with it, the documentation wins. Flag any behavior that may depend on version, configuration, or admin-controlled settings.
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
20 - If you cannot reach the documentation URLs, tell me the fetch failed
       and do not answer SecurSpaces-specific questions from memory alone.
```

Where to add this snippet

AI Tool	Where to add
GitHub Copilot (VS Code)	Settings → Copilot → Custom Instructions, or `.github/copilot-instructions.md` in your repo
Claude Code	Add to `CLAUDE.md` in your project root, or `~/.claude/CLAUDE.md` for global
Cursor	Settings → Rules for AI, or `.cursorrules` file in your repo
Windsurf	Settings → Cascade → Rules
Self-hosted LLM (Ollama, etc.)	Modelfile `SYSTEM` directive, or pass as a system message in your client

Step 3: Test it

Ask your AI assistant a clear SecurSpaces-specific question, for example:

> "How do I configure a workspace template in SecurSpaces?"

The assistant should fetch the documentation and provide a specific answer with SecurSpaces terminology and concrete steps.

Troubleshooting

Symptom	Possible cause	What to do
`https://<your-securspaces-url>/docs/llms.txt` does not load	Wrong instance URL, or docs endpoint not reachable from your network	Confirm the URL you use to log in to the SecurSpaces console. If it still fails, contact your platform administrator.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Symptom	Possible cause	What to do
The assistant answers from general knowledge instead of the docs	Snippet not applied, or the tool does not auto-fetch URLs	Confirm the snippet is saved in the correct location for your tool. Ask the assistant to fetch the documentation index first.
The assistant reports it cannot reach the documentation URLs	Air-gapped tool without network access to the SecurSpaces instance	Ensure the AI tool runs where it can reach your SecurSpaces instance URL, or use `llms-full.txt` as pasted context.
You have no SecurSpaces instance to point the assistant at	Evaluating, or working outside a deployment	Use the public copy at `https://assets.sds.citrix.com/llms.txt` and `https://assets.sds.citrix.com/llms-full.txt`.

Related information

- Run AI coding agents in workspaces —how agents are governed when they run inside a workspace, and what your administrator sets up for them

Troubleshooting

September 21, 2026

`Developer`

Start here when something is wrong and you are not sure why. Each symptom below points at the page that
explains it. If none of them matches, the diagnostic tools in your profile will tell you whether the problem
is your workspace, the network, or the platform.

Many problems in Citrix SecurSpaces™ are not faults at all: they are policies working as configured. A blocked domain, a missing repository, or a workspace that pauses overnight are all deliberate, and all fixed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

by a project owner or platform administrator rather than by you. Knowing which is which saves the most time.

Find your symptom

Starting and running a workspace

Symptom	Likely cause	Where to look
The workspace takes minutes to start	The cluster is provisioning a node, or pulling a large image	How long a workspace start takes
The workspace does not start, or stalls partway	A stage of the start sequence is failing	What happens when a workspace starts
The workspace stopped while I was away	Idle detection paused it	Workspace idle detection
A long job was killed overnight	Idle detection again —ask for **Bypass Schedule**	Bypass Schedule
I cannot create a workspace	A project or licence limit, or your role	Roles and permissions
`No space left on device`	The workspace disk is full	Develop in a workspace
The workspace error mentions `OOMKilled`	A task used more memory than the workspace allows	Workspace runs out of memory

Losing work

Symptom	Likely cause	Where to look
Files are gone after a restart	Only some paths survive a restart	What persists in a workspace
Installed packages disappear	They were installed outside the persisted path	What persists in a workspace
Docker images have to be pulled or rebuilt after every restart	Persistent Docker is not enabled, so the image cache is not on the persistent volume	Persistent Docker

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Symptom	Likely cause	Where to look
The workspace is running a different image than the console shows	An image change applies when the workspace restarts	Create a workspace
A shared file changed under me	Mount points are live, shared storage	Mount point limits

Reaching things outside the workspace

Symptom	Likely cause	Where to look
A download or package install is blocked	Network policy restricts outbound traffic	Network policies
I cannot clone or push to a repository	The repository is not attached, or your Git identity is not set up	Integration
A database or internal service is unreachable	The service is not attached, or DNS is failing	Connect to external services, then DNS Lookup
`no signers found` or `no auth passed yet` when using SSH or `git`	No SSH method is configured for that host	Connect to external services
My application is not reachable from the browser	The port is not published	Share a running application

Connecting

Symptom	Likely cause	Where to look
The IDE does not load in the browser	The websocket connection is failing	WebSocket Troubleshooting Tool
Everything feels slow	You may be working against a distant region	Region Latency Checker
SSH from my local IDE fails	Key, port, or platform setting	Connect over SSH

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Symptom	Likely cause	Where to look
SSH that used to work is refused, and nothing changed	The registered public key expired. Only the SecurSpaces IDE integrations renew it automatically	Key expiry and renewal
An SSH or Remote-SSH session connects, then drops	The session was idle and an intermediary closed it	Keeping a session alive
The terminal shows stray characters after reconnecting	A full-screen program was running when the connection dropped	Recovering a corrupted terminal
Copy and paste does not work	A clipboard policy applies to the workspace	Create a workspace
Dragging a file into the IDE does nothing	File transfer is restricted	Ask your platform administrator

For problems specific to a task, the page that covers the task also covers its failures —see Develop in a workspace, Run a graphical desktop, and Give your AI assistant access.

How long a workspace start takes

A start is not instant, and a slow one is usually not a fault. Two things dominate:

Cost	Why	When it applies
Waiting for a node	The cluster has no room and must provision one. The log shows repeated `0/4 nodes are available` scheduling messages while it waits	First workspace of the day, or after the cluster scaled down
Pulling the image	A workspace image is often over a gigabyte. The log names each image, its size, and how long the pull took	First use of an image on that node

Together these can take several minutes. Once the image is on the node and capacity exists, later

© 1997–2026 Citrix Systems, Inc. All rights reserved.

starts are
much quicker, and the preparation that follows —configuration, startup scripts, cloning —is usually a
matter of seconds.

Both costs can be reduced by a platform administrator, so consistently slow starts are worth raising rather
than living with:

Mitigation	What it removes
Warm nodes —a pool of nodes kept provisioned and idle	The wait for the cluster to scale out. See Warm Nodes Configuration
Cached images —an image version pre-pulled onto workspace nodes	The image pull. An image marked as cached shows a **Cached Image** indicator in the image list

Show your platform administrator the log described below: it tells them which of the two is actually costing
you the time.

Read the startup log

The log is not only visible while a workspace starts. You can open it at any time:

1. On the Workspaces page, select **More Actions** (**…**) on the workspace.
2. Select **Logs**.

It is split into two phases, each showing its elapsed time, so you can see where the time actually went rather than guessing:

- **Setting up container** —scheduling, volume attachment, and image pulls. This is where a long start
  nearly always is.
- **Setting up the workspace** —environment, configuration, startup scripts, IDE setup, and repository
  cloning, with a time against each step.

If a step failed rather than merely being slow, the log is also where the error appears.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What happens when a workspace starts

A workspace that seems stuck is usually working through one of these stages. The progress panel and the
startup log name the current one, which tells you where to look:

Stage	What it means	If it stalls here
Checking image compatibility	The image is being validated against the workspace infrastructure	The image may not meet the requirements. See Container image requirements
Building image	The container image is being built or pulled	A large image takes longer on first use. Repeated failures are a registry or image problem — tell your project owner
Setting up container	The container is starting and storage is being attached	Often a resource or mount point issue
Setting up environment	Repositories, secrets, and services are being attached	A resource you cannot access will fail here
Cloning workspace repository	Attached repositories are being cloned	Check your Git identity under Integration
Applying dot files	Your personal dot files are being applied	See Configuration
Applying custom configuration files	Configuration files from your profile are being applied	See Configuration
Executing system startup script	The startup script defined on the template runs	A failing script is a template problem —tell your project owner
Workspace startup scripts	Your own startup script runs	Check your script
Setting up IDE configuration	IDE settings and extensions are being applied	See Configuration
Preparing IDE for launch	The IDE is starting	If it never completes, use the WebSocket Troubleshooting Tool

Workspace runs out of memory

If the workspace error details include `OOMKilled`, a process in the workspace used more memory than the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

workspace specification allows:

```
Workspace failure: CrashLoopBackOff. Reason: OOMKilled
```

This usually follows a build, test run, package install, or AI-assisted tooling task that briefly needed more
memory than expected.

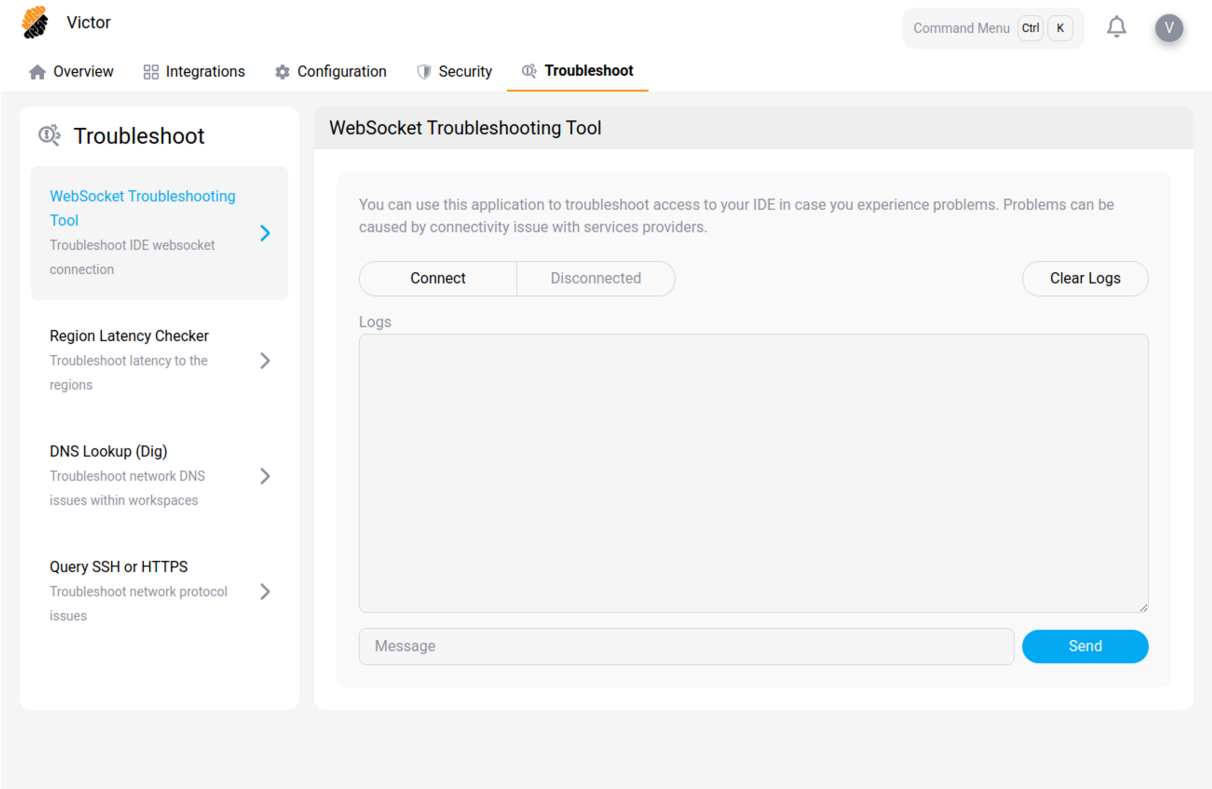
To recover:

1. Start the workspace again, then review what was running when the error appeared.
2. Reduce the memory the task uses, or run it in smaller steps.
3. If the task legitimately needs more memory, ask your project owner for a specification with a higher
   memory limit. If the workspace came from a template, the template has to change instead.
4. Restart the workspace so the new limit applies.

Expected result: The workspace starts and the task completes within its memory limit.

Diagnostic tools

Your profile carries four tools for narrowing a problem down. Select your profile picture, select **Profile**, then open the **Troubleshoot** tab.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

WebSocket Troubleshooting Tool

Tests the websocket connection the Cloud IDE depends on. Select **Connect**, and the tool reports whether the
connection is established and logs what happens. You can send a message to confirm traffic passes in both
directions.

Use it when the IDE will not load, or drops repeatedly, in the browser. A websocket that never connects usually means something between you and the platform is closing the connection —a corporate proxy, a VPN, or
a TLS-inspecting gateway. That is information your platform administrator needs.

Region Latency Checker

Reports round trip time to each available region. Use it when the IDE feels sluggish rather than broken.

If a distant region is markedly faster than the one you are using, your workspace may be running in the wrong
place. Which regions you can use is set by your administrator.

DNS Lookup (Dig)

Resolves a domain name from inside a workspace. Use it when something is unreachable and you do not know
whether the name is failing to resolve or the connection is being refused.

If the name resolves but the connection still fails, the cause is usually
network policy rather than DNS.

Query SSH or HTTPS

Tests whether a host is reachable over SSH or HTTPS from the workspace. Use it after DNS Lookup, to separate
a name that does not resolve from a port that is blocked.

When to escalate, and what to include

Some problems you cannot fix yourself. Anything involving network policy, workspace resources, templates,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

attached repositories, privileged mode, or Bypass Schedule is set by a project owner or platform administrator.

Include the following, or the first reply will only ask for it:

- The workspace name, and the project it belongs to.
- The region, if your deployment has more than one.
- What you were doing, and the exact message you saw.
- When it happened, with the time zone.
- Which stage the start sequence reached, if the workspace did not start.
- The output of the relevant diagnostic tool, if you ran one.

Related information

- What persists in a workspace
- Workspace idle detection
- Network policies
- Roles and permissions

Manage

August 19, 2026

This section is for **project owners** and anyone else running a project day to day. It covers getting a team
working, giving them what they need, and setting the rules their workspaces follow.

Most tasks here need the **Members**, **Resources**, or **Security** permission at a level your role carries. See Roles and permissions.

Getting a team working

Task	Page
Bring developers onto the platform	Onboard developers
Add and remove people, and set their roles	People
Understand roles and what each permission allows	Roles and permissions

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Giving them what they need

Task	Page
Give everyone the same starting environment	Templates
Provide container images and toolchains	Workspace images
Attach repositories, secrets, storage, and services	Project resources
Set a project up for AI coding agents	Run AI coding agents in workspaces

Setting the rules

Task	Page
Rename a project, or recover a deleted one	Project and organization settings
Control clipboard, limits, SSH, and network traffic	Workspace policy
Let workspaces reach external services over SSH	Enable outbound SSH

Related information

- Administer —platform-wide configuration
- Concepts

Onboard developers

August 19, 2026

`Project Owner`

Onboarding someone into a project is one decision followed by a short task. The decision is whether they
create their own workspaces or you create one for them. Everything else follows from that.

This page is written for the person doing the onboarding. For what the developer does once they are in, see
Get started.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Two ways to onboard someone

The difference is the **Workspaces** permission carried by the role you assign.

	Self-served	Assigned
Role	Developer, Manager, or Project Owner	Guest
Workspaces permission	Manage Personal or higher	Access
Who creates the workspace	The developer, from your template	You, before they arrive
Can they change it	Yes —settings and resource access	No —the configuration is fixed
Suits	Employees and long-term team members	Contractors, external collaborators, anyone temporary
Your effort	Set up a template once	One workspace per person

Self-served is the normal case. Choose assigned when you need the environment to be exactly what you decided
and to stay that way, which is usually about limiting what an outsider can reach.

Both models can coexist in one project. The role is per person, so you can have ten self-served developers
and two contractors on fixed workspaces.

Before you onboard anyone

Have the project ready first, or your first developer arrives to an empty environment. See
Set up a project for your team, which
covers images, repositories, secrets, and the template both models depend on.

You need the **Members** permission set to **Manage** to add people. The default Manager and Project Owner
roles both carry it.

Onboard a self-served developer

1. Open **People** and select **Add New User**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Find the user, and leave the role as **Developer**, or choose Manager or Project Owner if they will run
   part of the project.
3. Select **Apply**.

They can now create their own workspace from your template. Nothing else is required from you.

Onboard a guest or contractor

1. Open **People** and select **Add New User**.
2. Find the user and set the role to **Guest**.
3. Select **Create a new workspace from a template or an existing one**, and choose the template that
   matches the work.
4. Select **Set Expiration date** and choose when their access ends.
5. Select **Apply**.

The workspace is provisioned as part of adding them, so there is no separate assignment step. When they sign
in, the workspace is waiting and its settings are fixed.

Set the expiration date even when you expect the engagement to be extended. An expiry that has to be renewed
deliberately is safer than access that quietly persists after someone has moved on.

See Add and remove users for the
full dialog, including inviting someone who is not on the platform yet.

What the platform sends for you

Adding a user to a project triggers a **You've joined a new project** email containing a link straight into the project. Someone new to the platform entirely also receives an invitation email.

This only works if a platform administrator has configured an email gateway. **Until one is configured, the
platform sends nothing at all**, and your new developer is waiting for a message that will never arrive. If
you are not sure, check with your platform administrator before you rely on it, or tell people directly that
they have been added.

See
Onboarding email notifications.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What the developer does next

Point them at Get started. A self-served developer wants Your first workspace; a guest already has one, so Set up your account is the better starting point.

If you would rather hand over something written for your own team, adapt the reusable onboarding template, which you can fill in with your repositories, tools, and policies.

Related information

- Roles and permissions
- People
- Templates
- Set up a project for your team

Reusable template for SecurSpaces workspace onboarding

Overview

Use this template to onboard developers to Citrix SecurSpaces™. This framework provides a concise, step-by-step guide to move from initial access to a running application. You can customize and extend this document with your specific tools, repositories, and policies to deliver a consistent, personalized onboarding experience for your team.

Before you begin

Before starting, ensure that one of the following is available:

- A workspace template shared by your IT team or team lead via link or accessible via the SecurSpaces UI.
- Access to the default SecurSpaces template from the workspace.

If neither is available, contact your team lead before proceeding.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Step 1: Configure your profile and launch a workspace

Administrator Note: In this phase, users will configure their profiles in the SecurSpaces UI to quickly access, build, and run an application. The environment uses a standard Linux terminal and Visual Studio Code. The objective is to run the existing application in the new environment as soon as possible to establish familiar practices.

1. Set up your profile

1. Log in to the **SecurSpaces Application UI**.
2. In your profile,
    - Set your **time zone** and preferred **work schedule**.
    - Authenticate with your **Git** provider (GitHub, GitLab, or Bitbucket).

2. Launch your workspace

Choose one of the following options:

- **Option A:** From the SecurSpaces UI, select the arrow next to **Create Workspace**, and then select **From Template**. Select a **workspace template**, and then select **Launch**.
- **Option B:** If your Git repository has a file with a **Workspace** link (typically readme.md), click **Open Workspace** directly from the project's README.md.

3. Clone and build your project

1. Open the workspace using the browser-based VS Code interface or terminal.
2. Clone your repository using your standard **git clone** command.
3. Run your usual build commands.
4. Check that the VS Code extension for SecurSpaces is installed and active. This extension allows you to open and manage applications running on your workspace.
5. Use the extension to run your application and preview the application using the URL.

You are done with Step 1 when your application is running and accessible via a live URL.

Step 2: Set up quick access

Administrator Note: In this phase, users set up their preferred method for accessing a workspace: either from a desktop icon, via a progressive Web app, or using a URL embedded in a web or markdown page. The latter is typically used to start a workspace from a Git repository.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Create a desktop shortcut to your workspace (open from the IDE or Terminal)

Create a desktop or taskbar shortcut to:

- The SecurSpaces UI, or
- A specific workspace.

This allows instant access without navigating bookmarks or the SecurSpaces UI.

2. Enable Git repository start-up

On the Workspace page, under **Template**, copy the **Quick Start** template link to add an **Open Workspace** icon to your repository's README.md, making it easy to launch SecurSpaces directly from the repo.

Step 3: Customize your profile settings

Administrator Note: In this phase, users explore workspace customization options. They can configure environments in the SecurSpaces UI via the user profile menu or by editing the workspace configuration.

1. Persist shell configuration

Using the profile menu, add your **.bashrc** or **.zshrc** files so that they will be automatically added to your home directory when a new workspace is created. You can update an existing workspace using the update option from the workspace menu.

2. Add secrets and keys

Your profile also lets you define variables (e.g., with secrets) and generate an SSH key to be available in your workspace:

- Personal environment variables
- SSH keys –allow you to ssh to a host using a private key stored in the credentials proxy.

These settings can be updated at any time.

3. Preload VS Code extensions

Using the profile menu, configure your preferred VS Code extensions to be automatically available when the workspace launches.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Step 4: (Optional) Customize your Workspace settings

Administrator Note: In this phase, users modify their workspace configuration to optimize performance and automate repetitive tasks.

1. Adjust hardware resources

Update CPU and memory settings based on project needs, such as large builds or resource-intensive tooling. These settings can be changed at any time.

2. Automate workspace startup

Using the workspace editor, add pre- or post-startup scripts to automatically install tools or dependencies when the workspace starts. This option adds tools to an existing base image that cannot be installed in the home folder (`./home/developer`). To learn more about how data is persisted across reboots, see Maintain persistent changes in your workspace.

3. Configure autorun

Using the workspace editor, you can set application running on your workspace to start automatically at boot time.

Getting help

If your workspace fails to start, your build does not complete, or your application does not run as expected, contact your team lead or the SecurSpaces support channel.

People page

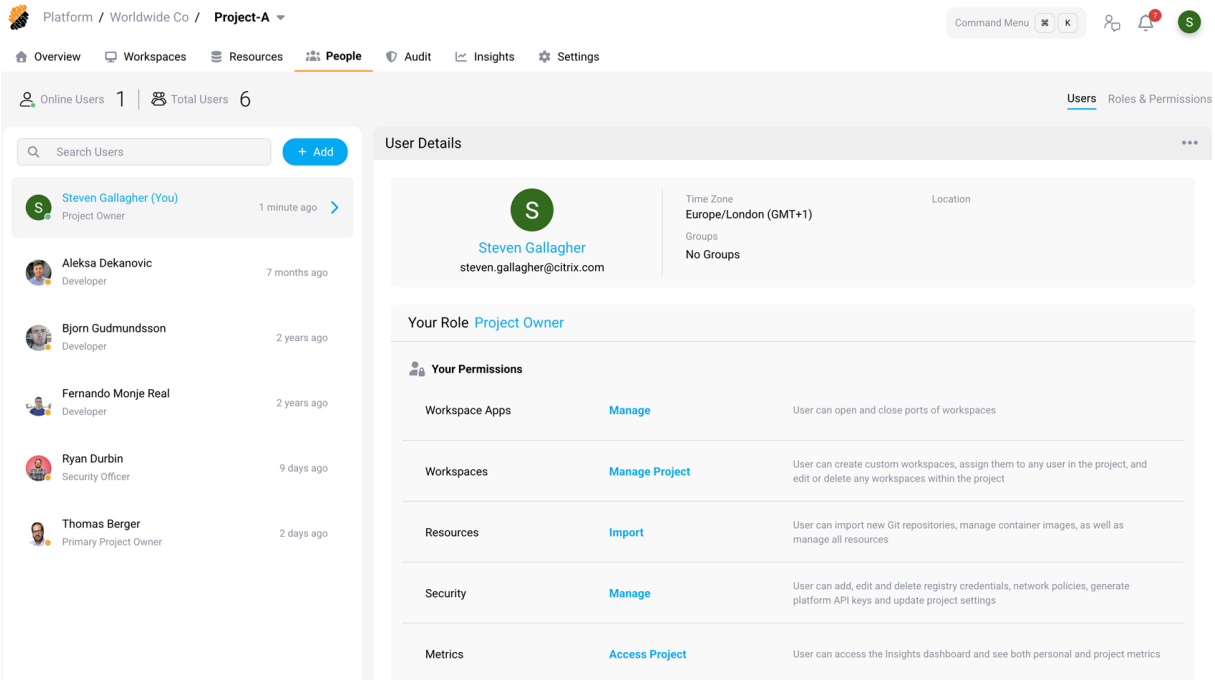
The People page contains information about users onboarded to a project, an organization or the entire platform. Switching projects or organizations therefore updates the membership in the table.

You can see the role, permissions and public details for each user in the View User panel.

This page provided typical team management functions to users with the appropriate permissions. The *project owner* has permissions to update the roles of the users in the project.

In addition, the project owner can create new roles or update existing ones from the Permission Management panel.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



> **Info:**
>
> A regular user can view all of the roles in the project and their permissions, even without the Members permission set to **Manage**.

Content

- **View Users** panel.
- **Access Control** panel.

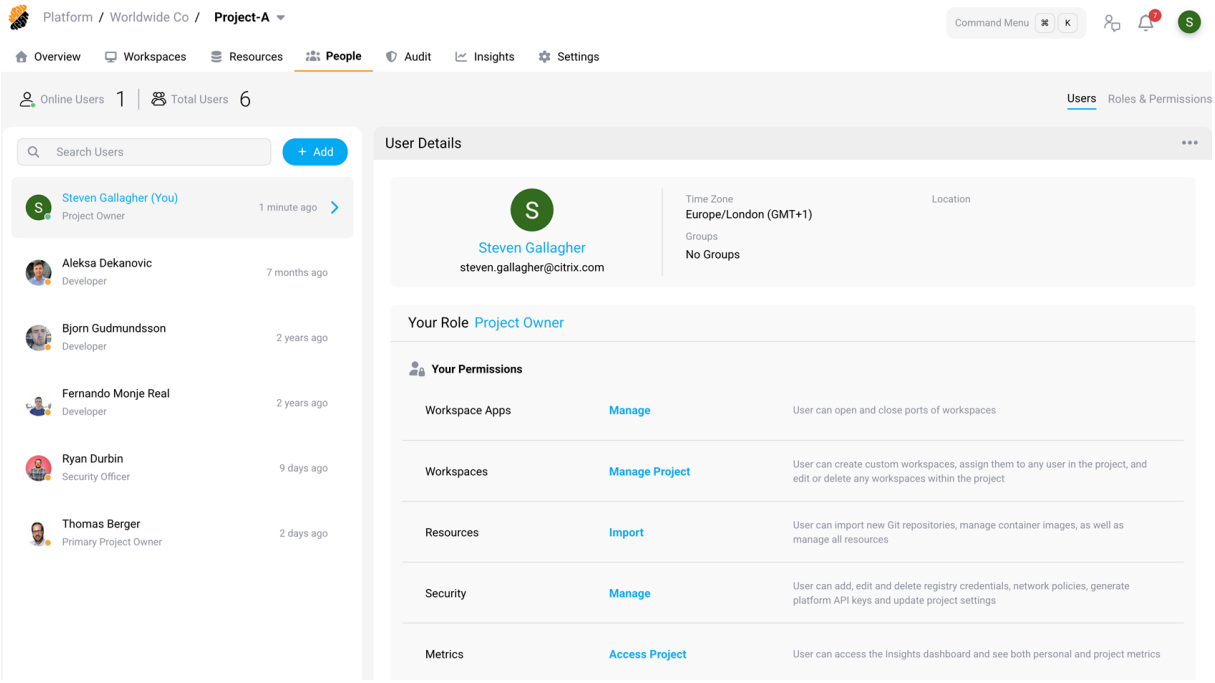
Users

The **People** page displays users at two levels: project and platform. The information shown and the available actions vary by level.

Project-level view

At the project level, the **People** page uses a split-panel layout. The left panel lists project members, and the right panel shows details for the selected user.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



User list

The left panel displays all project members, sorted by most recent activity. Each entry shows:

- User name and avatar
- Role in the current project (for example, Primary project owner, Developer, security officer)
- Last activity timestamp
- Online status indicator (for currently connected users)

Summary counts for **Online Users** and **Total Users** appear above the list.

Use the **Search Users** field above the user list to filter by name. This narrows the displayed list without changing the actual membership.

User details panel

Select a user in the list to view their details in the right panel.

Profile information:

- Name and email address
- Time zone
- Location (if set by the user)
- Group memberships

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Role and permissions:

The panel displays the user's current role in the project and the associated permissions. Each permission category shows the category name, the permission level, and a description of what the level allows.

To change a user's role, select **Change Role**. This requires the Members permission set to **Manage**.

More Actions menu:

Select the **...** icon at the top right of the User Details panel to access:

- **More Details** —Opens the full user details page (same page as the platform-level user details described below).
- **Remove user from project** —Removes the user from the current project. See Add and remove users and groups for details.

Tabs

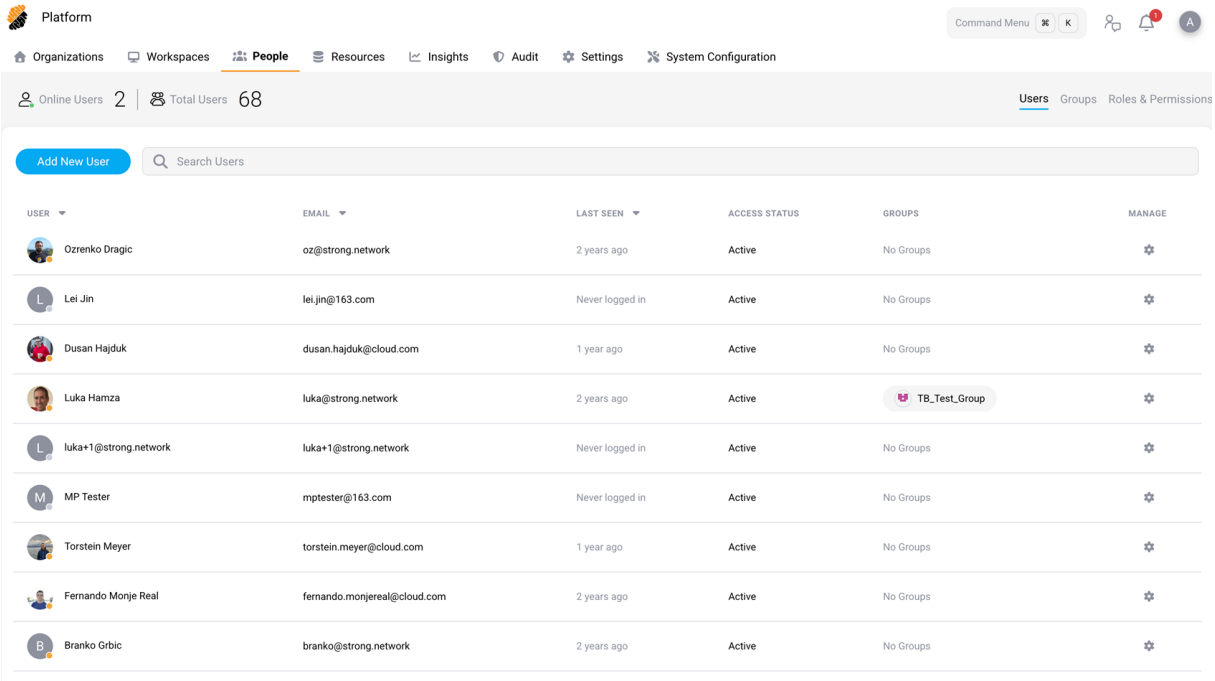
The project-level People page has two tabs in the top-right corner:

- **Users** —The user list described above.
- **Roles & Permissions** —Manage project roles. See Access Control for details.

Platform-level view

At the platform level, the **People** page is available to **security officers** and **platform administrators**. It displays all registered users on the platform in a table.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



User table

The table includes the following columns:

Column	Description
User	Name and avatar. Sortable.
Email	Email address. Sortable.
Last Seen	Timestamp of the user's most recent activity. Sortable.
Access Status	Whether the user's account is active.
Groups	Groups to which the user belongs.
Manage	Settings gear icon for user management actions.

Summary counts for **Online Users** and **Total Users** appear above the table. Use the **Search Users** field to filter the list.

Tabs

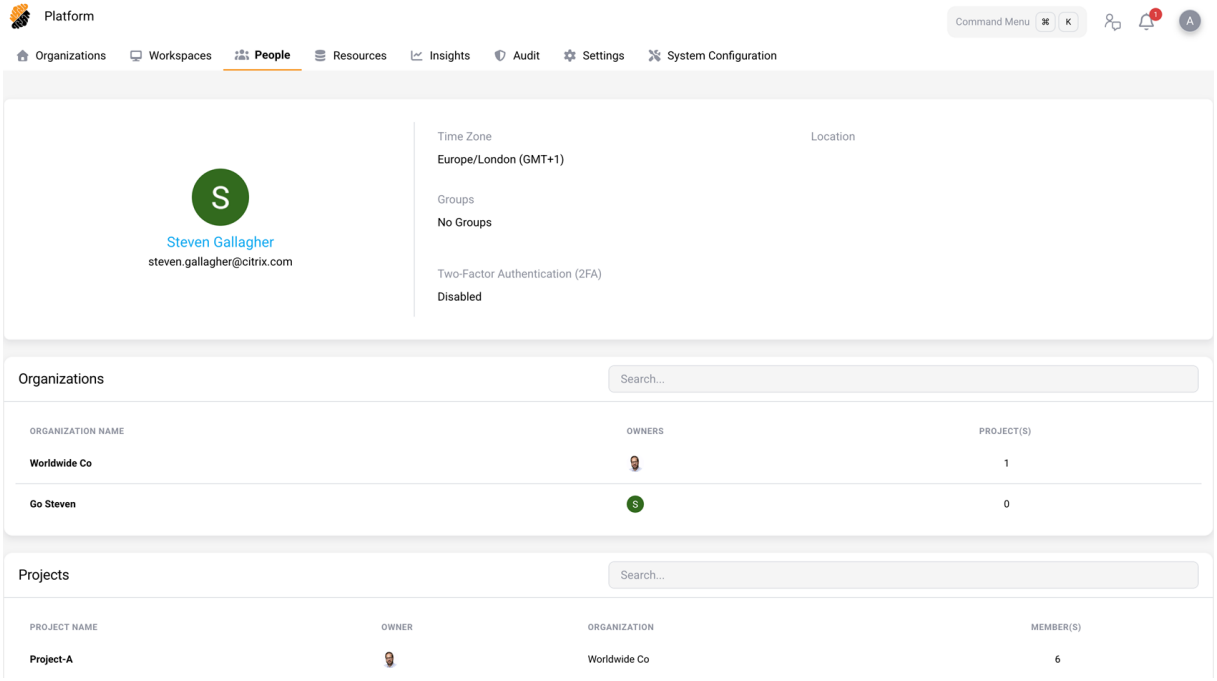
The platform-level People page has three tabs:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Users** —The user table described above.
- **Groups** —Manage platform-level groups.
- **Roles & Permissions** —Manage platform-level roles and permissions.

User details page

Select a user row to open the full user details page. This page provides a comprehensive view of the user's access and activity across the platform.



Profile:

- Name, email address, and avatar
- Time zone
- Location
- Group memberships
- Two-Factor Authentication (2FA) status

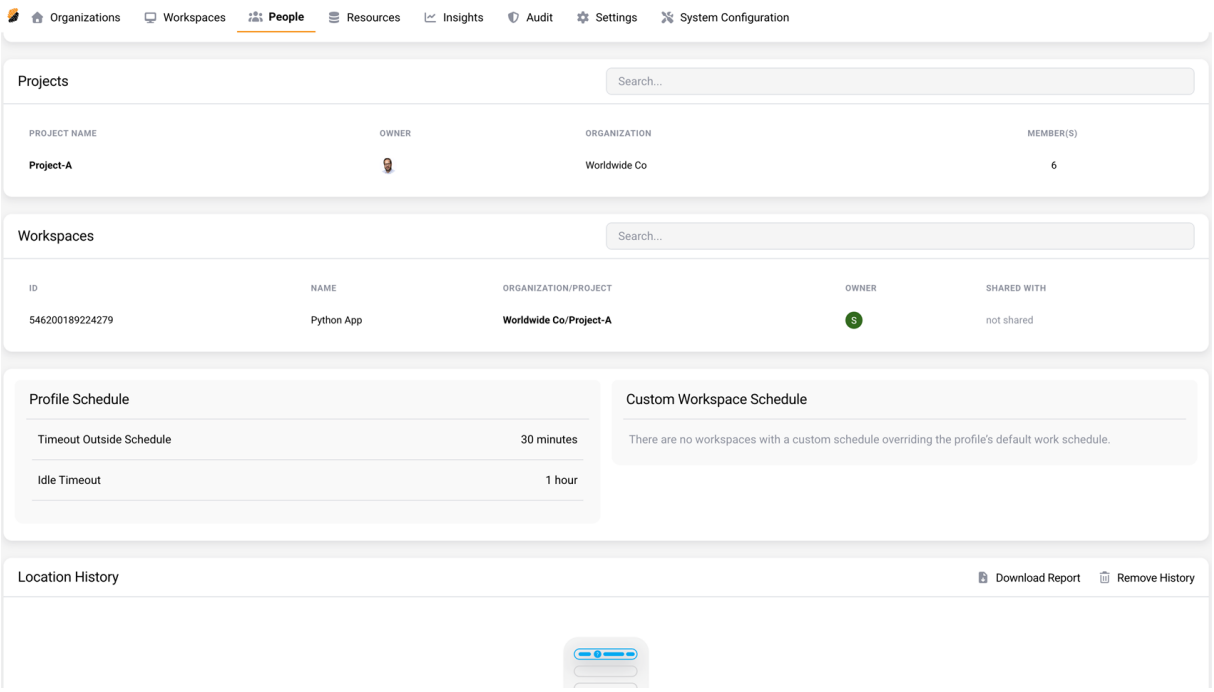
Organizations:

A searchable table listing all organizations the user belongs to, with columns for Organization Name, Owners, and number of Projects.

Projects:

A searchable table listing all projects the user can access, with columns for Project Name, Owner, Organization, and number of Members.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Workspaces:

A searchable table listing all workspaces owned by or shared with the user, with columns for ID, Name, Organization/Project, Owner, and Shared With.

Profile Schedule:

Displays the user's schedule settings:

- Timeout Outside Schedule —how long a workspace stays running outside the user's work schedule before pausing.
- Idle Timeout —how long a workspace stays running without activity before pausing.

Custom Workspace Schedule:

Lists any workspaces that have a custom schedule overriding the user's default work schedule.

Location History:

Displays the user's login location history. Includes **Download Report** and **Remove History** actions.

Related information

- Add and remove users
- Roles and permissions
- People

© 1997–2026 Citrix Systems, Inc. All rights reserved.

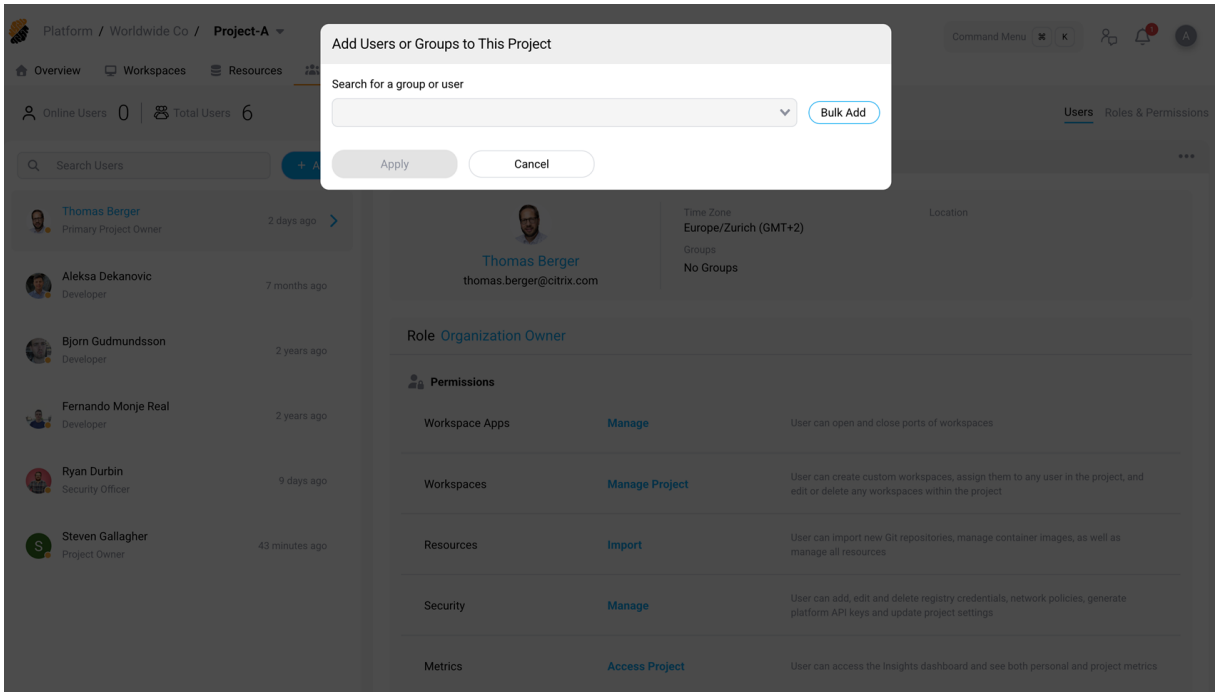
Add and remove users and groups

This page describes how to add and remove users and groups in a project. These actions are available on the project-level **People** page and require the Members permission set to **Manage**.

Add users or groups to a project

1. Navigate to your project and select **People** in the navigation bar.
2. Select **+ Add**.

The **Add users or groups to this project** modal opens.

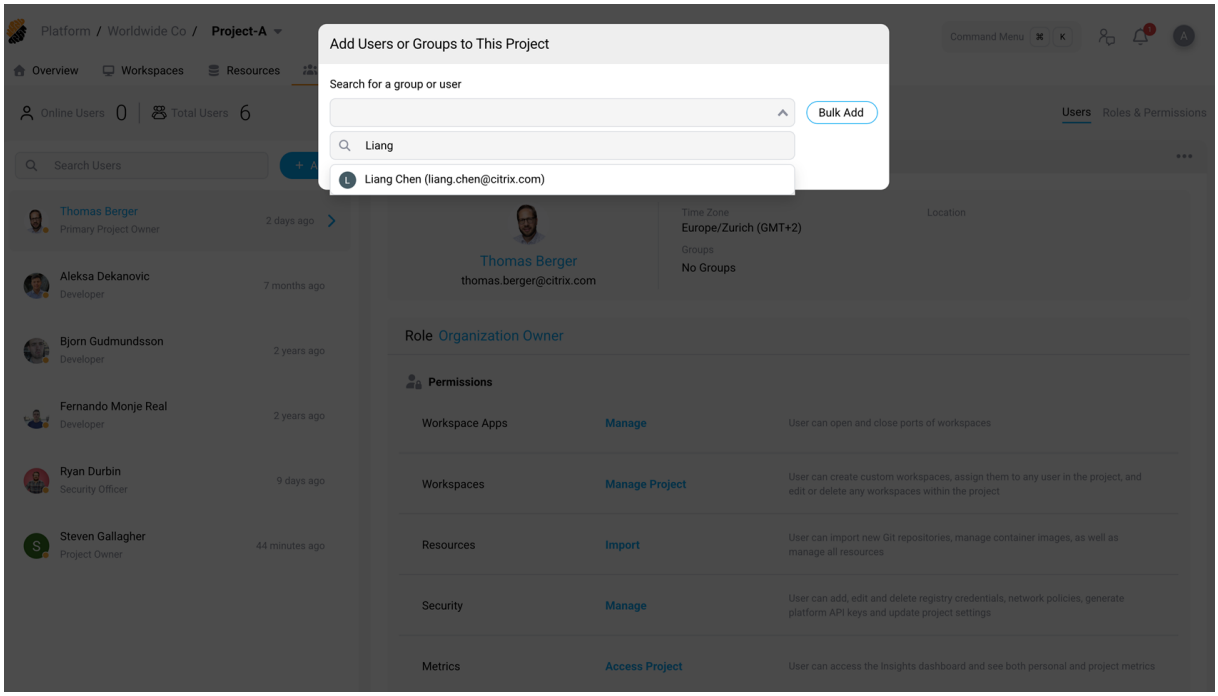


Add a user by search

The search dropdown lists platform users who are not yet members of the current project. This is a single-select field —select one user at a time to avoid confusion.

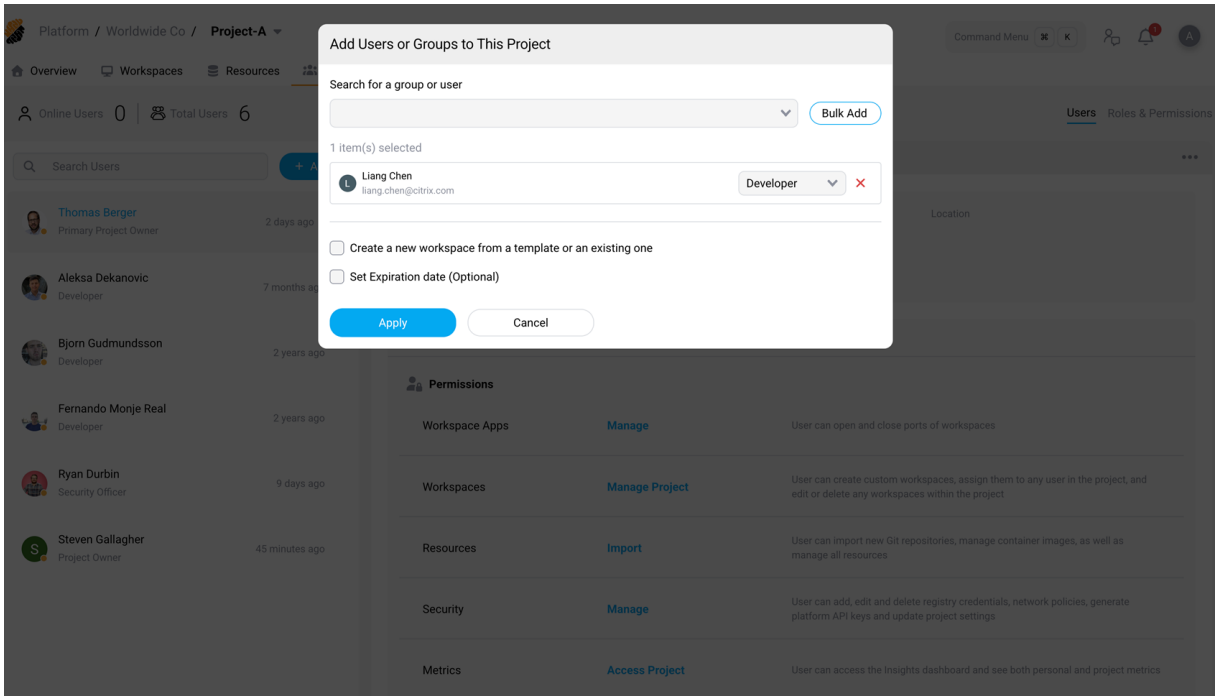
1. In the **User or group** field, start typing the user's name or email address.
2. Select the user from the filtered results.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



The user appears in the selection list. You can:

- Change their **role** using the dropdown next to the user's name (defaults to Developer).
- Check **Create a new workspace from a template or an existing one** to provision a workspace during onboarding.
- Check **Set Expiration date (Optional)** to set a time limit for the user's access to the project.
- Remove the user from the list by selecting the **X** button.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Select **Apply** to add the user.

Invite a user who is not on the platform yet

The **User or group** field is not limited to people who already have an account. You can also use it to invite
someone new.

1. In the **User or group** field, type the person's full email address.
2. Select the add (+) icon.

While you are typing, and before the address is complete, the field reports that no matches were found and
explains that entering a full email address lets you invite someone. Once the address is valid, the field confirms that the user is not on the platform yet and prompts you to select the add (+) icon.

The invited user appears in the selection list, where you can assign a role and the same options as an existing
user. Select **Apply** to send the invitation.

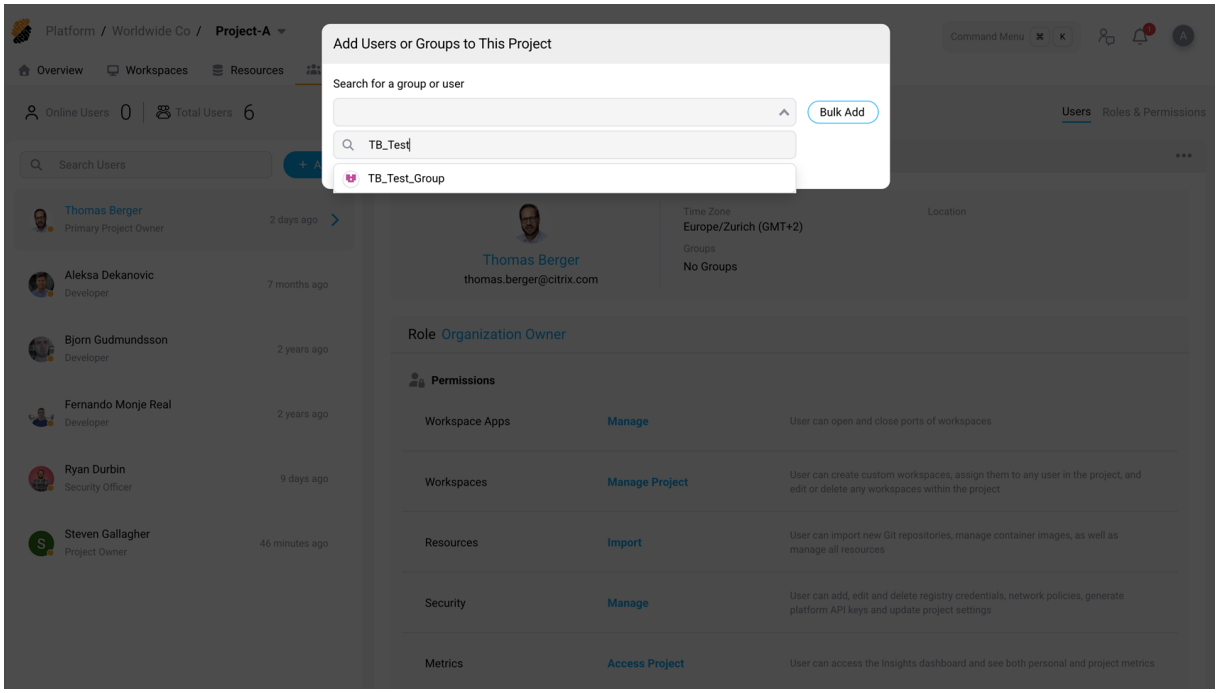
> **Note**
>
> Inviting a user whose email domain is not associated with a configured identity provider requires local
> account registration to be enabled. See
> Add external users.

Add a group

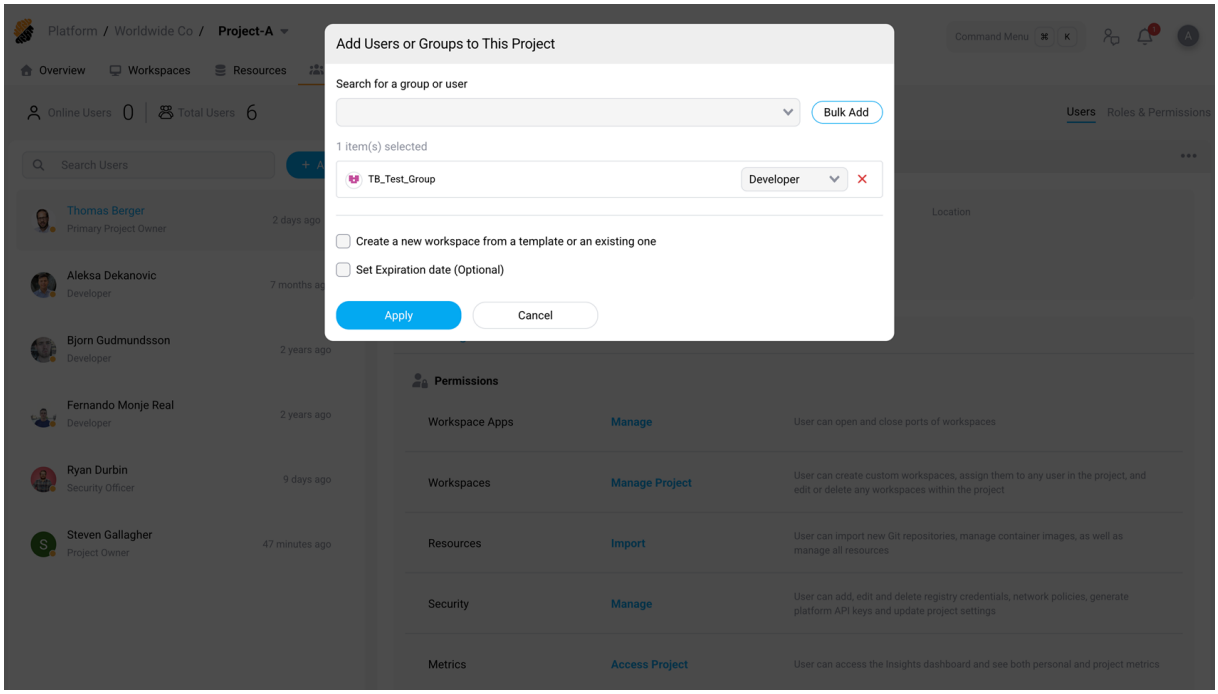
Groups that exist on the platform can be added to a project in the same modal.

1. In the **User or group** field, type the group name.
2. Select the group from the filtered results. Groups are shown with a group icon.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



The group appears in the selection list with the same role assignment and options as individual users.



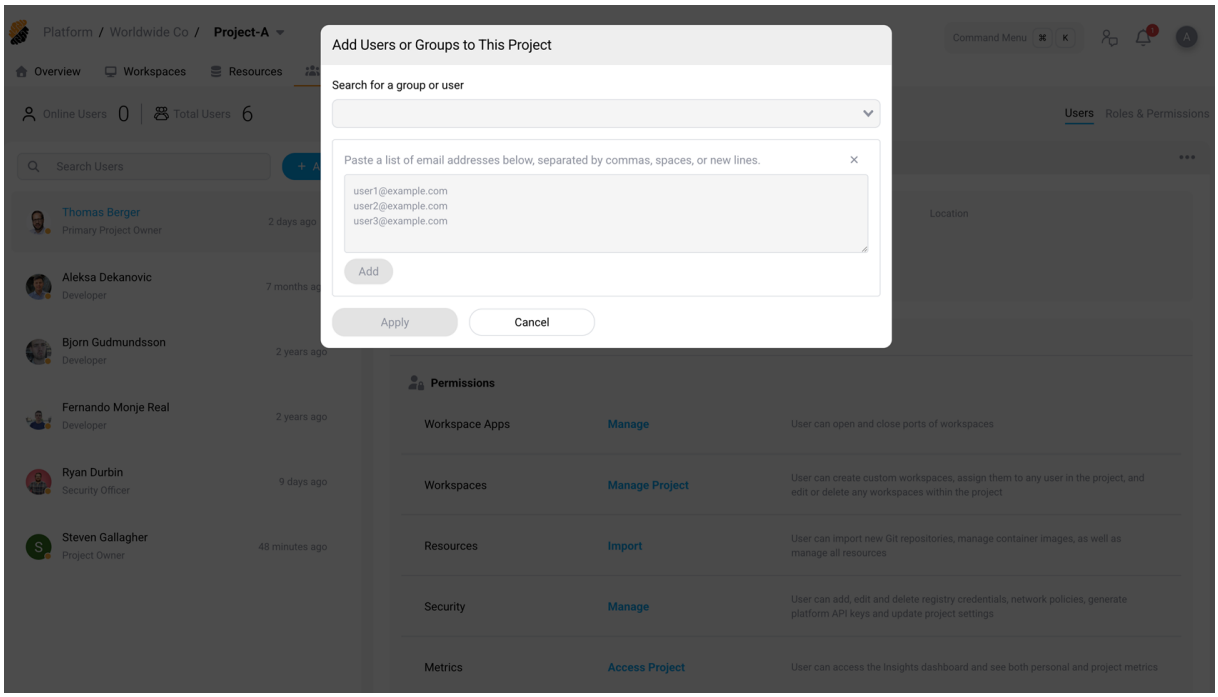
1. Select **Apply** to add the group.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Bulk add users

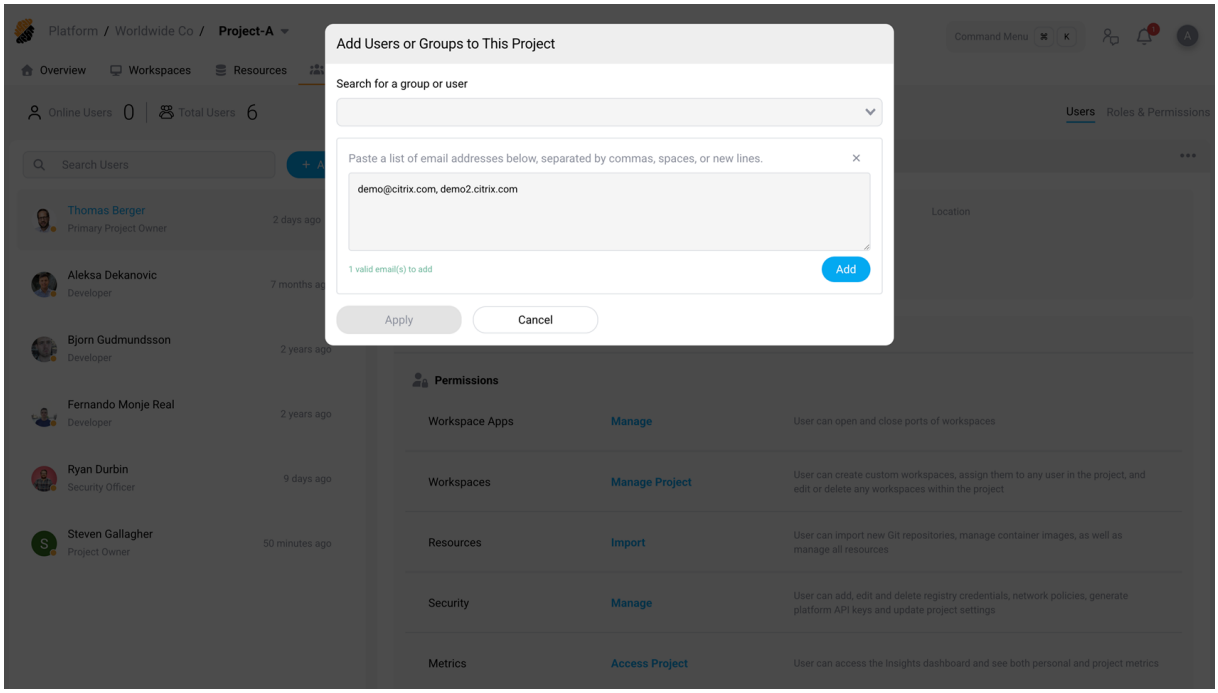
Use bulk add to onboard multiple users at once by email address.

1. In the **Add users or groups to this project** modal, select **Bulk add**.
2. Paste a list of email addresses into the text area. Commas, spaces, or new lines can separate addresses.



The system validates addresses as you type and shows the number of valid emails.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



1. Select **Add**. The system validates the addresses and adds valid users to the selection list. If any addresses are invalid, an error message appears below the input.
2. Assign roles and configure options for the added users.
3. Select **Apply** to complete the operation.

Add external users

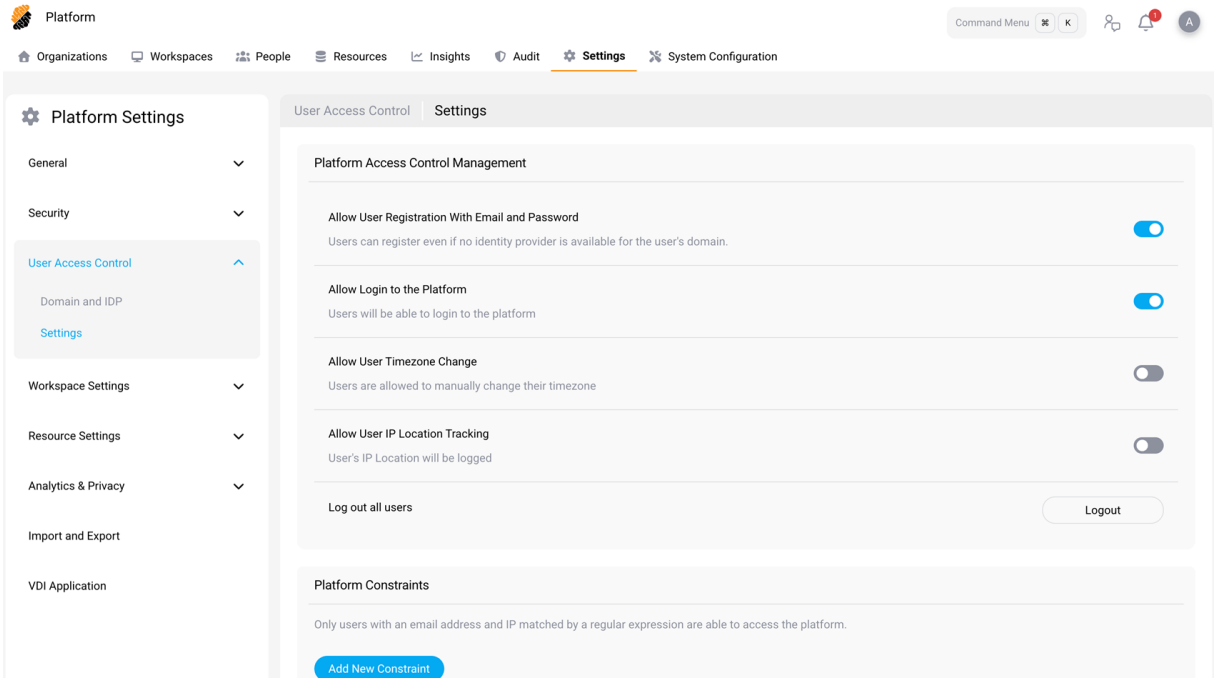
External users are users whose email domain is not associated with a configured identity provider (IdP). Adding external users requires a platform-level setting and follows a different flow from adding existing platform users.

Enable local account registration

Before external users can be added, a platform administrator or security officer must enable **Allow User Registration With Email and Password** under **Platform > Settings > User Access Control > Settings**.

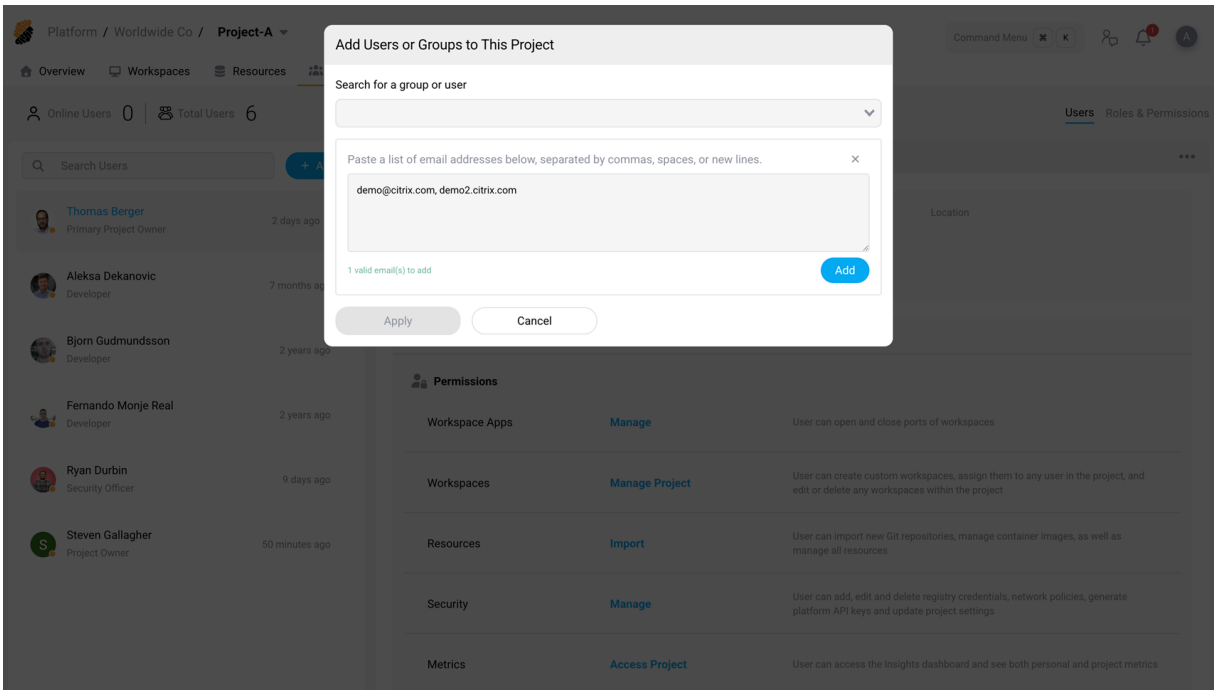
When this setting is enabled, Citrix SecurSpaces™ creates a local account for external users. The user can then sign in with email and password instead of the organization's IdP.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Add external users to a project

1. Open the **Add Users or Groups to This Project** modal (see Add users or groups to a project).
2. Select **Bulk Add** and paste the external email addresses.



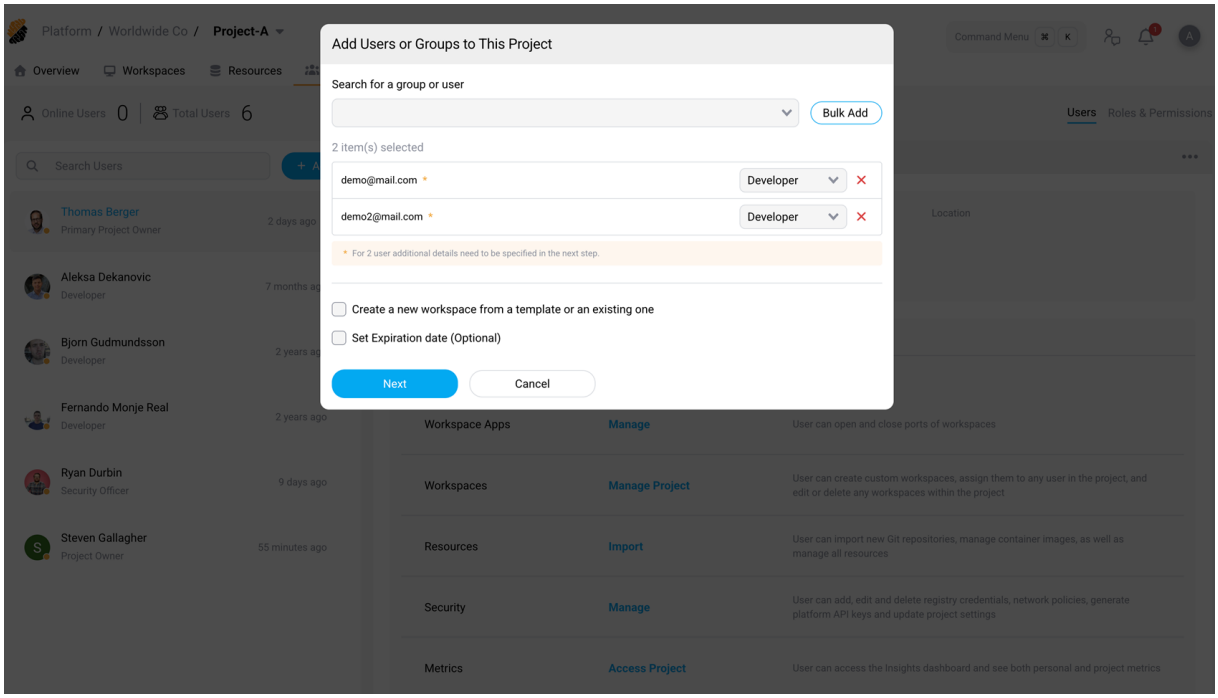
1. Select **Add**. External users appear in the selection list marked with an asterisk (*) next to their

© 1997–2026 Citrix Systems, Inc. All rights reserved.

email address. A notice reads: “For *N* users, additional details need to be specified in the next step.”

2. Assign roles and configure options as needed.

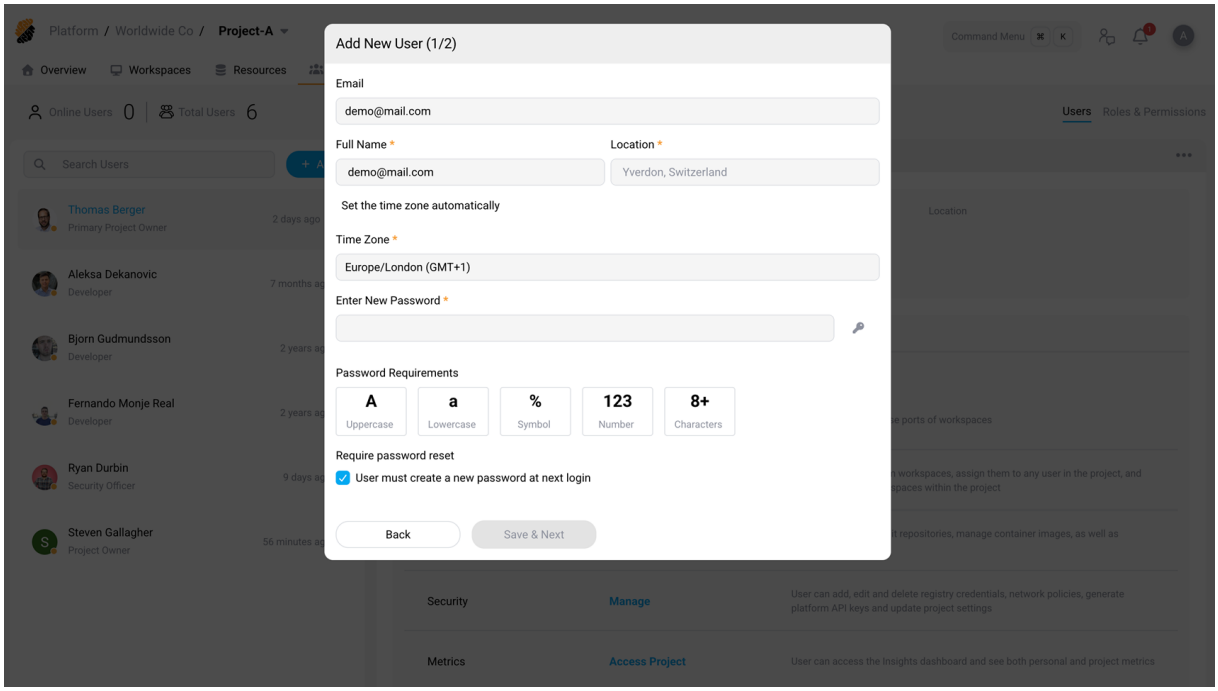
The **Apply** button changes to **Next** when external users are present.



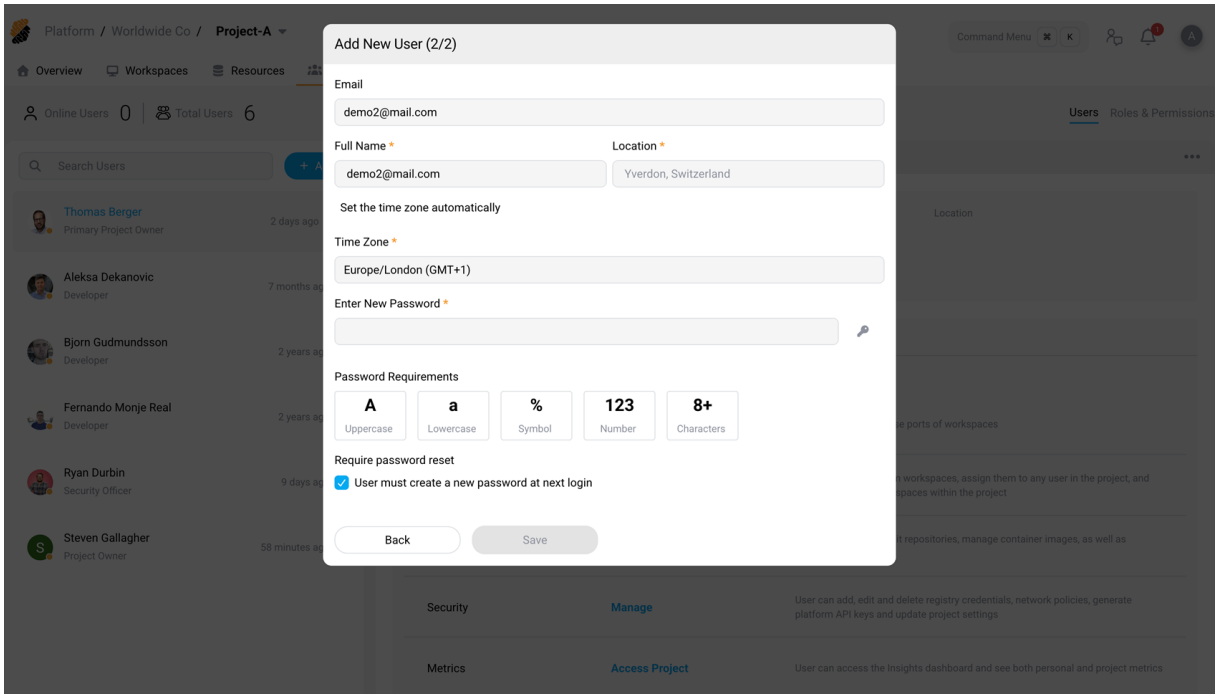
1. Select **Next**. SecurSpaces displays an **Add New User** form for each external user. The form shows a step counter (for example, “1/2”) and requires:

- **Full Name** —pre-filled with the email address; editable.
- **Location** —required.
- **Time Zone** —auto-populated based on the location.
- **Password** —must meet the displayed requirements (uppercase, lowercase, symbol, number, 8+ characters).
- **Repeat New Password** —must match the password.
- **Require password reset** —checked by default. When enabled, the user must create a new password at the next sign-in.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

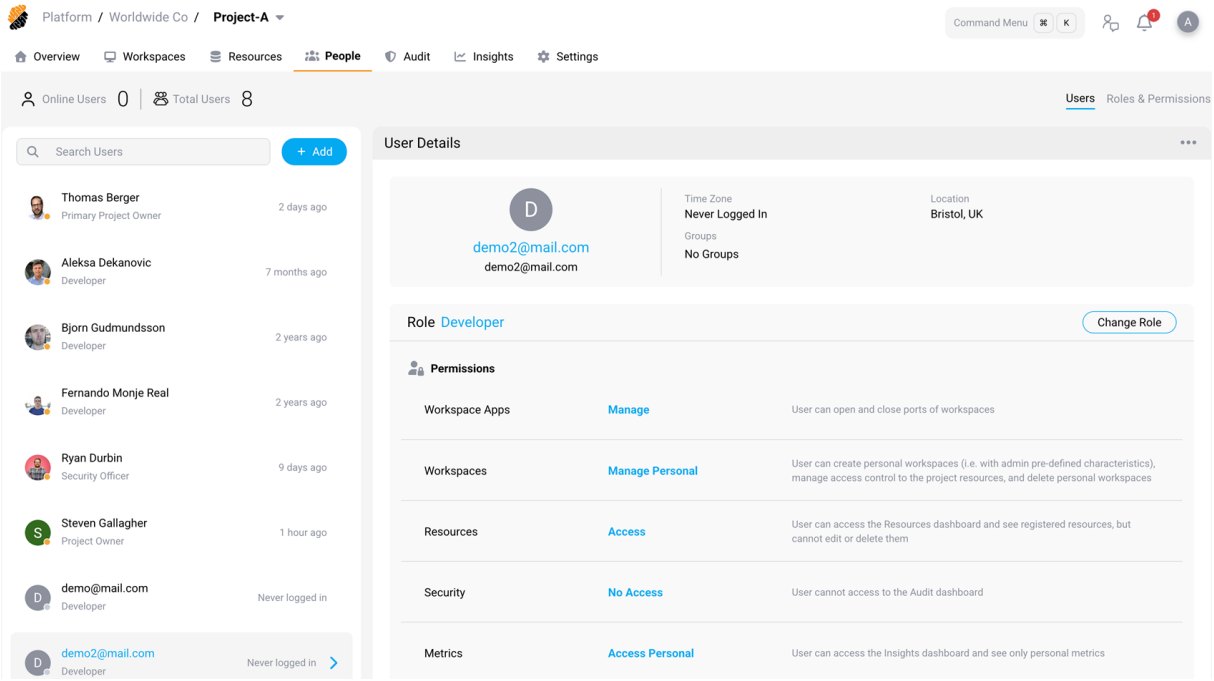


1. Select **Save & Next** to proceed to the next external user. For the last user in the batch, the button changes to **Save**.



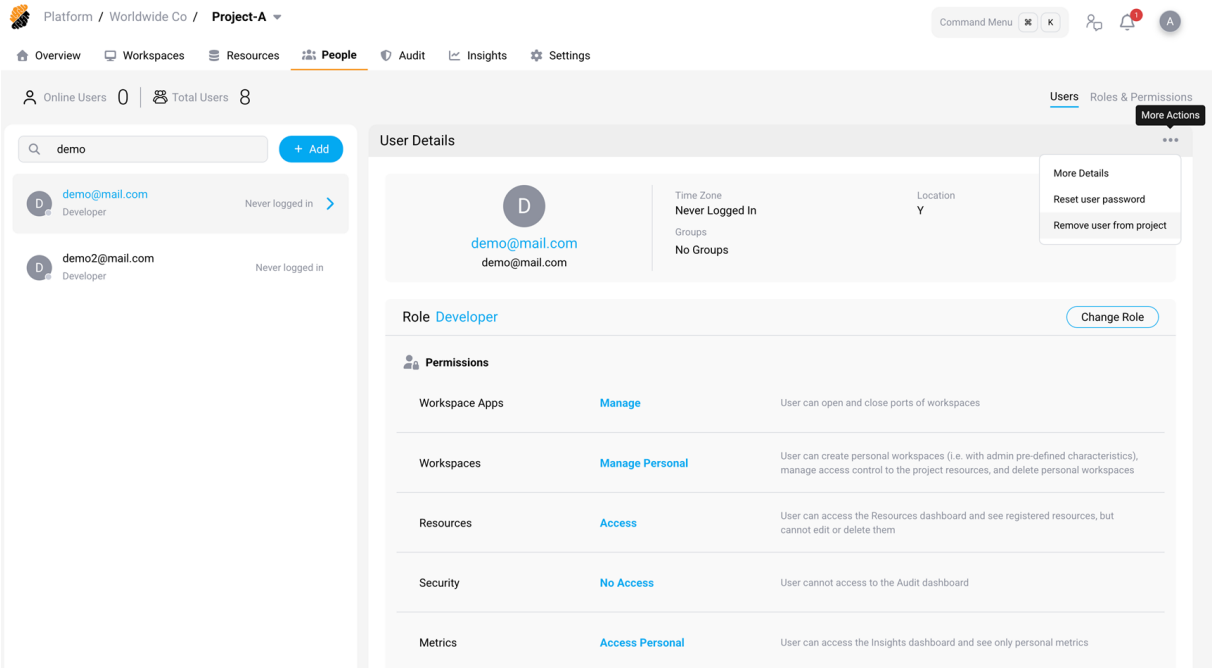
After saving all external users, SecurSpaces creates local accounts and adds the users to the project. Until an external user signs in for the first time, the user appears in the project member list with the status “Never logged in”.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



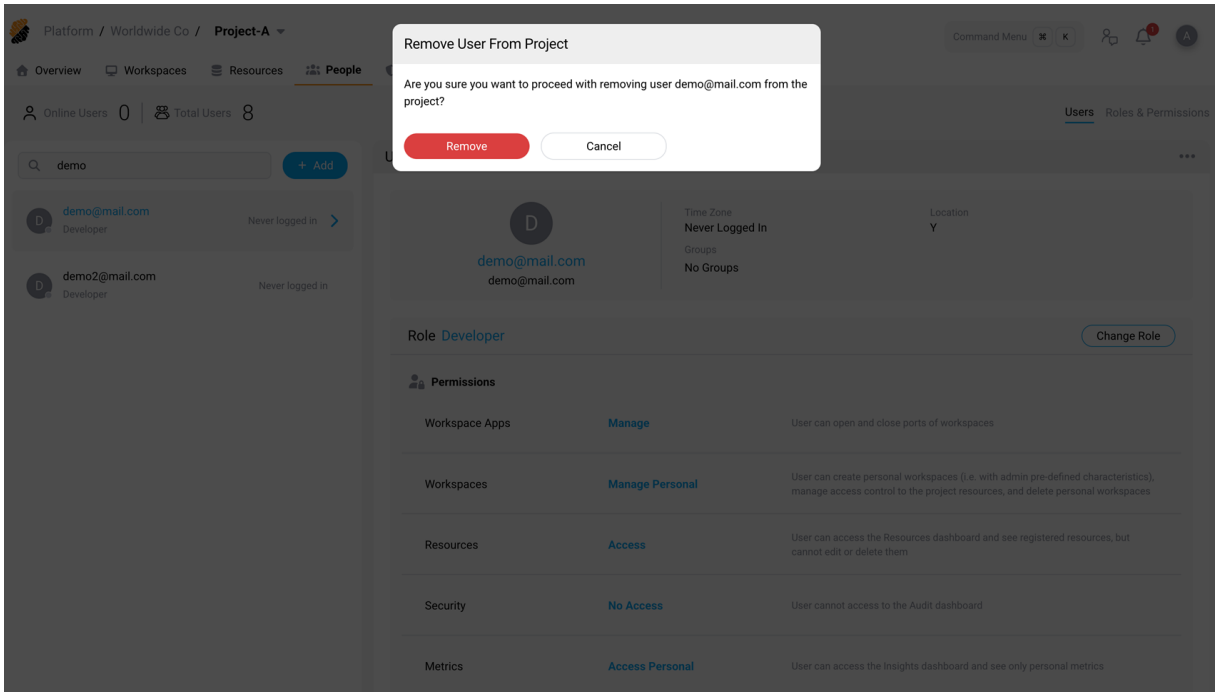
Remove a user from a project

1. On the **People** page, select a user in the user list.
2. In the **User Details** panel, select the **...** (More Actions) menu at the top right.
3. Select **Remove user from project**.



A confirmation dialog asks you to confirm the removal.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



If the user owns workspaces in the project, the dialog warns that removing the user will delete those workspaces. You can deselect individual workspaces to keep them before confirming.

1. Select **Remove** to confirm.

The user loses access to the project and all associated resources. The user remains in the platform database. To fully remove a user from the platform, a platform administrator or security officer must remove the user from the People page at the platform level.

Permissions

Action	Required permission
Add users or groups to a project	Members –Manage
Remove a user from a project	Members –Manage
Change a user's role	Members –Manage
View project members	Members –Access

Related information

- Roles and permissions
- Users

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- People

Roles and permissions

September 21, 2026

Access in Citrix SecurSpaces™ is granted through **roles**. A role is a named set of permissions, and every member of a project holds one. Roles are project bound, so the same person can hold different roles in
different projects.

This page is the reference for what each role and permission allows. Other articles link here rather than
repeating it.

Roles and permissions are shown on the People page.
A project owner can create roles or change existing ones from the access control panel.

> **Warning:**
>
> Roles decide who can reach your source code, secrets, and security settings. Assign them on a least
> privilege basis.

Why a feature is hidden

SecurSpaces hides what your role cannot use rather than showing it disabled. If a page, button, or menu entry
described in this documentation is not there, the usual reason is that your role does not carry the permission
for it.

Some features are also hidden for reasons unrelated to your role, such as a platform setting being off or an
optional integration not being configured. The article for each feature says which applies.

To see your own permissions, open the People page
and look at your role.

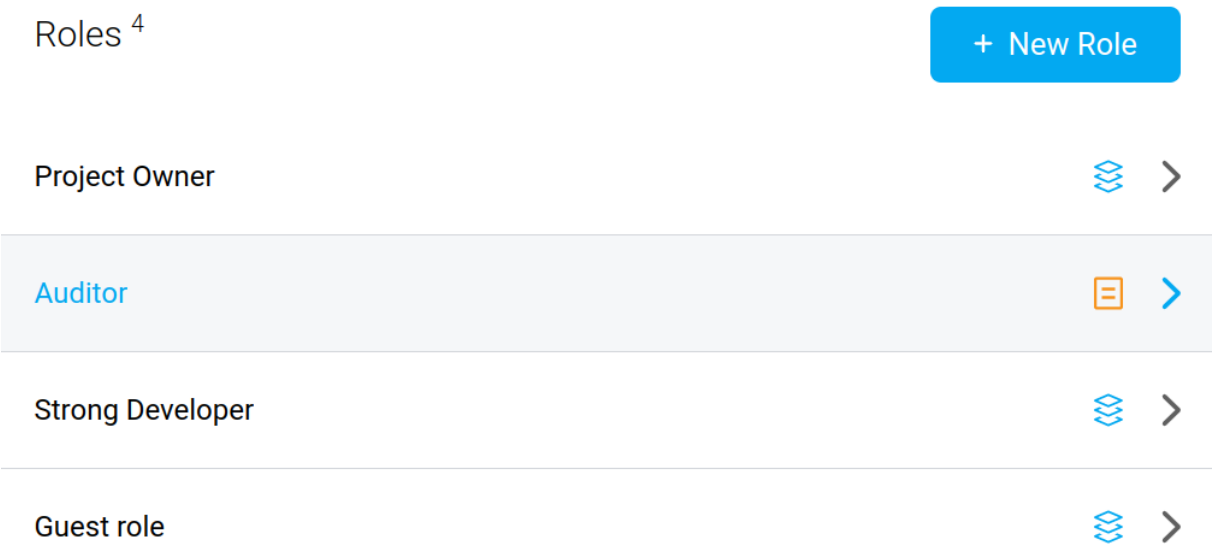
Default roles

A new project has four roles.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Role	Intended for
Guest	Someone who needs to look at a project without changing anything
Developer	Day-to-day development in their own workspaces
Manager	Running the project: its workspaces, resources, and members
Project Owner	Running the project and its security settings

A project owner differs from a manager in two ways. A manager holds no **Security** permission, so network
policies, registry credentials, platform API keys, and project settings are out of reach. A manager also cannot
manage roles, so they cannot create or edit roles, or transfer ownership. A manager can still manage workspaces, resources, and members.



What each default role holds

Permission	Guest	Developer	Manager	Project Owner
Workspace Apps	Manage	Manage	Manage	Manage
Workspaces	Access	Manage Personal	Manage Project	Manage Project

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Permission	Guest	Developer	Manager	Project Owner
Resources	Access	Access	Import	Import
Regulated resources	No	Yes	Yes	Yes
Confidential resources	No	Yes	Yes	Yes
Metrics	No Access	Access Personal	Access Project	Access Project
Members	No Access	Access	Manage	Manage
Security	No Access	No Access	No Access	Manage
Manage roles and permissions	No	No	No	Yes

> **Note:**
>
> A platform administrator can disable the predefined **Project Owner** role and nominate a replacement, so a
> deployment may differ from this table. A disabled role cannot be assigned to anyone, although members who
> already hold it keep it. Check the access control panel for your project.

Permissions

A role sets a level for each of six permissions, plus three separate settings and one checkbox that allows role management.

Workspace Apps

Controls access to applications running inside workspaces.

Level	The user can
No Access	Not open workspace ports to view apps, or see apps shared by others
Access	View apps shared with them by other users
Manage	Open and close ports on workspaces

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Workspaces

Controls what the user can do with workspaces.

Level	The user can
No Access	Not access workspaces
Access	Use workspaces assigned to them, but not edit properties, change resource access, or delete them
Manage Personal	Create personal workspaces with characteristics an administrator has predefined, manage access to project resources, and delete their own workspaces
Manage Project	Create custom workspaces, assign them to anyone in the project, and edit or delete any workspace in the project

Resources

Controls the Resources dashboard: repositories, secrets, connected services, data buckets, and mount points.

Level	The user can
No Access	Not open the Resources dashboard
Access	See registered resources, but not edit or delete them
Manage	See, edit, and delete project repositories, secrets, external services, and data buckets
Import	Everything Manage allows, plus import Git repositories, container images, and SAML connected apps

Security

Controls the Audit dashboard and the security configuration of the project.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Level	The user can
No Access	Not open the Audit dashboard
Access	Open the Audit dashboard and see network policies, but not add, edit, or delete them
Manage	Add, edit, and delete registry credentials and network policies, generate platform API keys, and update project settings

Metrics

Controls the Insights dashboard.

Level	The user can
No Access	Not open the Insights dashboard
Access Personal	See their own metrics
Access Project	See their own and project-level metrics

Members

Controls the People dashboard.

Level	The user can
No Access	Not open the People dashboard
Access	See project members
Manage	Add and remove project members, and assign or change their roles, up to the level of the role the user holds themselves

Manage roles and permissions

A checkbox at the foot of the role editor rather than a level on a scale, labelled **User can manage roles and permission of other users (enable all permissions)**. It allows project administration: creating, editing, and deleting roles, transferring project ownership, and restoring deleted workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	The user can
User can manage roles and permission of other users	Create, edit, and delete roles, transfer project ownership, and restore deleted workspaces

Only the **Project Owner** role has it selected by default. It is separate from the **Members** permission: a manager can add members and set their roles, but cannot create or edit the roles themselves.

> **Important:**
>
> Selecting this checkbox also raises every other permission on the role to its highest level, which is what
> "enable all permissions"in the label means. Workspace Apps, Workspaces, Resources, Security, Metrics, and
> Members all move to their maximum, and the regulated and confidential settings are switched on. Clearing
> the checkbox again restores the levels the role had before. Treat it as granting everything in the project,
> not only role management.

Separate settings

These are switches on the role rather than levels on a scale.

Setting	Effect
Regulated resources	The role can access resources marked as regulated, meaning they fall under a regulation
Confidential resources	The role can access resources marked as confidential, such as intellectual property
Require templates for workspace creation	The role can create workspaces only from a template. Creating a custom workspace, or copying an existing one, is blocked.

Limits on delegating access

A user with the **Members** permission set to **Manage** can bring people into the project and set their roles, but not use that to escalate. Four rules apply, and the platform enforces all of them.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **You cannot grant a role above your own.** When you assign a role, it is compared against your own role
  in that project, permission by permission. If the role you are assigning is higher on any single permission —including role management and the regulated and confidential settings —the assignment is
  refused, even if it is lower on every other one.
- **The limit applies to the role you grant, not the person you act on.** It stops you creating someone more
  powerful than yourself. It does not stop you acting on a member who already holds more than you: a member
  manager can still change a manager's role to a lower one, or remove them from the project.
- **You cannot change your own role,** and you cannot remove yourself from a project.
- **You cannot remove the project owner.** Ownership has to be transferred first, which requires the
  role-management checkbox.

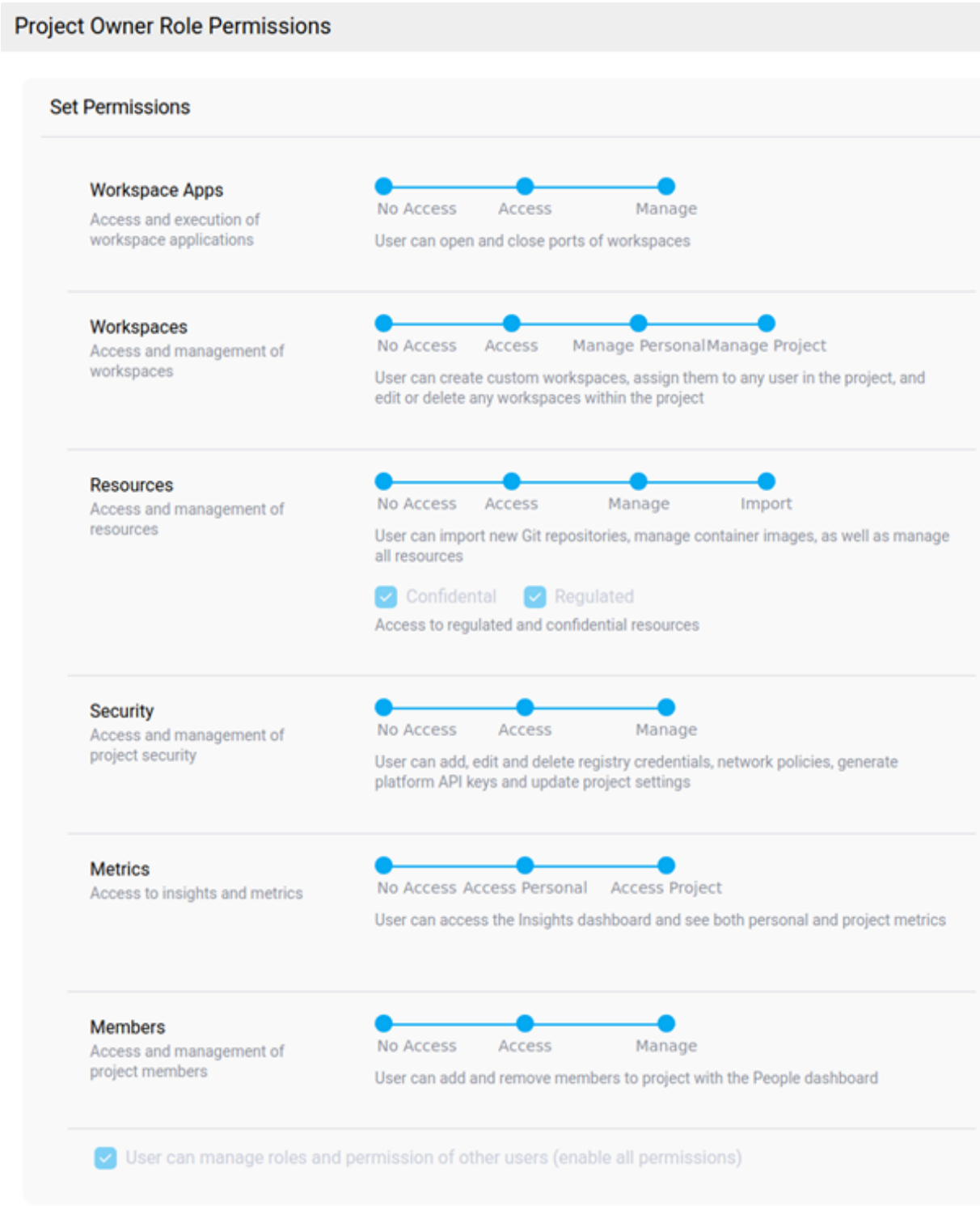
> **Note:**
>
> These limits apply to project roles. A platform administrator, a security officer, or the owner of the
> project's organization is not bound by them.

Create a role

Requires a role with **User can manage roles and permission of other users** selected, which the **Project Owner** role has. A manager cannot create or edit roles: role management is separate from the
Members permission that governs adding and removing members.

Open the access control panel from the People page,
add a role, name it, and set each permission. A role created at project level is available only in that project.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Related information

- People

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Platform roles
- Audit dashboard
- Network policies
- Project resources

Projects and organizations

August 18, 2026

General settings for a project and for the organization that contains it, and the recovery windows that apply when either is deleted.

For what projects and organizations are and how they relate, see
Projects and
Organizations.

Where each setting lives

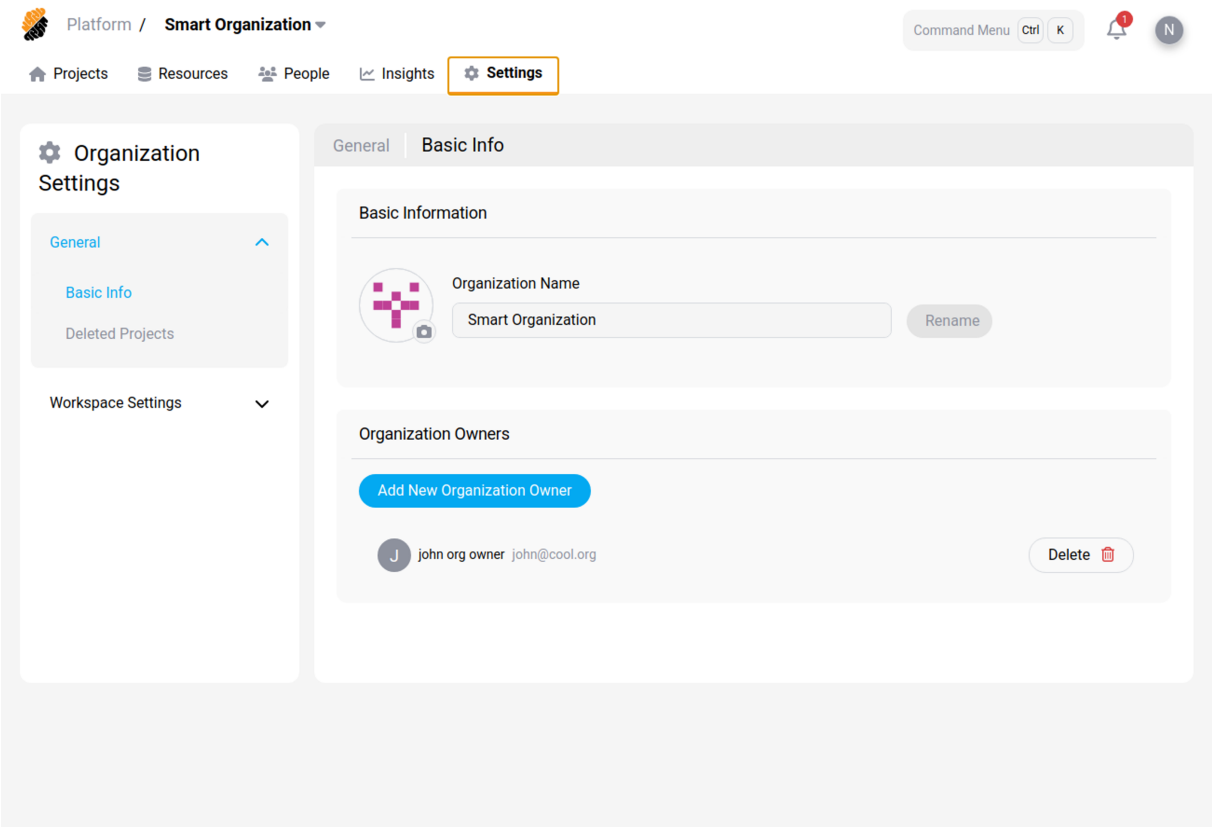
Setting	Organization	Project
Rename	yes	yes
Recover a deleted project	yes	—
Recover a deleted workspace	—	yes
Workspace policy, including security and network settings	Organization workspace policy	Workspace policy

Organization settings apply to every project inside the organization, so they are the place to standardize
configuration rather than repeating it project by project.

Organization settings

Organization settings are the control point for every project in the organization. Defining a setting here
gives a consistent baseline that each new and existing project picks up.

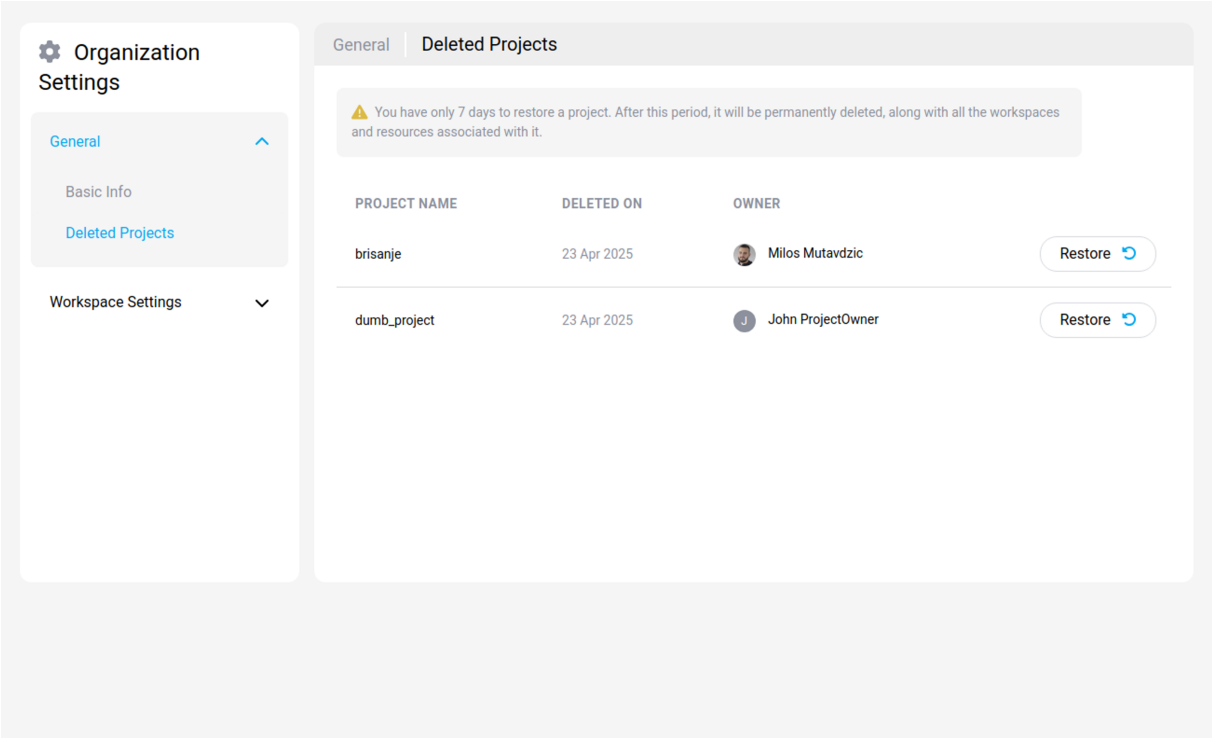
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Recover a deleted project

A deleted project can be restored for **7 days**. After that it is removed permanently, along with its workspaces, secrets, images, repositories, and other resources.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

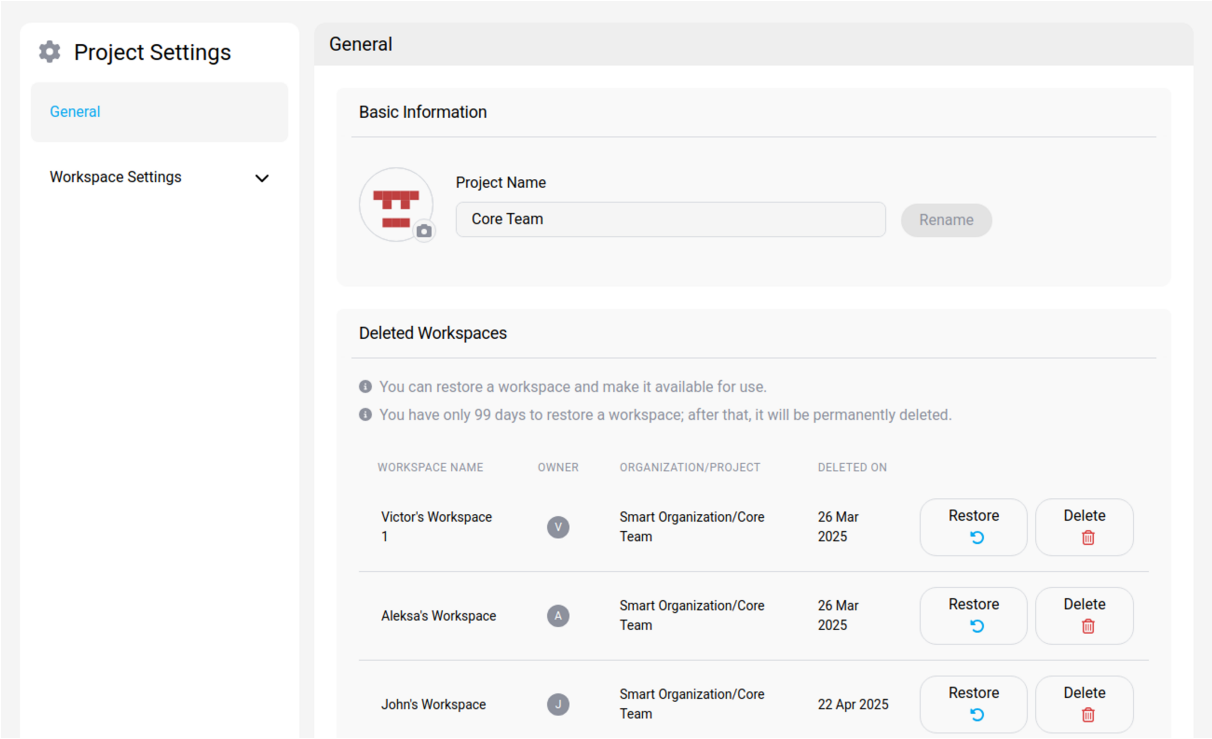


> **Note:**
>
> Cleanup runs on a schedule rather than at the exact moment the window closes, so a project may remain
> visible slightly past 7 days. Do not rely on the extra time.

Project settings

Under **Basic Information** you can change the project's name.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Recover a deleted workspace

A deleted workspace can be restored for a period your platform administrator sets, from **1 to 99 days**. After that it is removed permanently.

The retention period is configured at platform level under
Workspace policy,
not per project.

> **Note:**
>
> A mount point cannot be deleted while a deleted workspace in the recycling bin still uses it, because that
> workspace can still be restored. See
> Mount point limits and behavior.

Related information

- Projects
- Organizations
- Workspace policy
- Onboard developers

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Templates

September 29, 2026

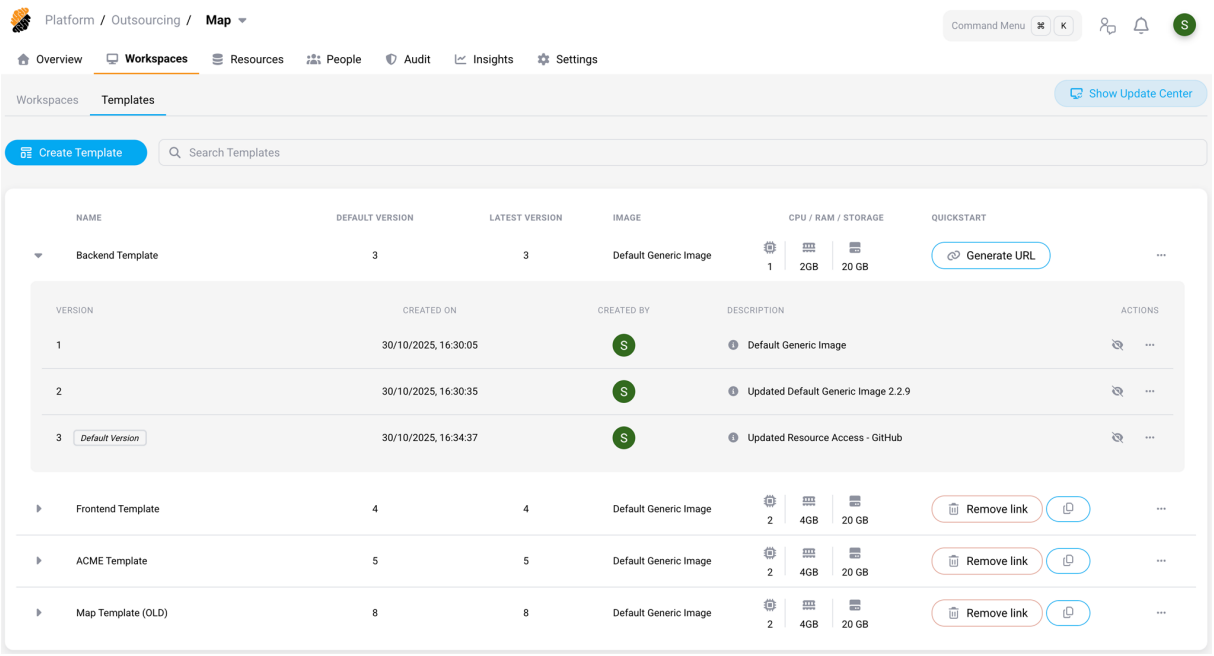
Workspace **Templates** help streamline project onboarding by eliminating the need for manual workspace setup. Each template defines all required configuration parameters including Workspace settings, repositories, secrets, startup scripts, and security policies, ensuring consistency across all Workspaces within a project.
Use the Quickstart feature to create a new Workspace with a single click from an external source, such as a code repository or engineering portal.

- View Templates
- Built-in template
- Create a Template
- Create a new version of a Template
- Track which template version a Workspace uses
- Update a Workspace to the default template version
- Archive a template
- Quickstart
- Duplicate a template

View Templates

Templates are displayed in the **Templates** section of the Workspaces Page. Each template can have multiple versions, which are visible when expanding the chevron on the left-hand side of the screen.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



A template is defined by the following characteristics:

- **Basic Information**: Name, container image, CPU/RAM/Storage settings, and description.
- **Class Level**: Confidential or regulated.
- **Workspace Configuration**: All the other elements describing a workspace.

Built-in template

Every new project includes one built-in template. It is provided as a working starting point rather than a
complete set, so most teams create their own templates from it or from a custom image.

Name	Image	CPU / RAM / Storage	Description
Monitored VSCode Template	Default generic image	2 CPU / 4 GB / 20 GB	Creates a fully updated Ubuntu container with monitored traffic and monitored clipboard.

The template is created with the **Monitor Traffic (default)** network policy applied. For the other policies
available to a project, see
Network policies.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

More images to build templates from

Citrix publishes example Dockerfiles and prebuilt images you can use or adapt:

- strong-network/images on GitHub —Dockerfiles and build scripts, including the generic base image and language-specific and desktop variants.
- strongnetwork on Docker Hub —prebuilt images you can reference directly.

To make one available to your project, see Container images.

Create a Template

Requires the Workspaces permission set to **Manage Project**.

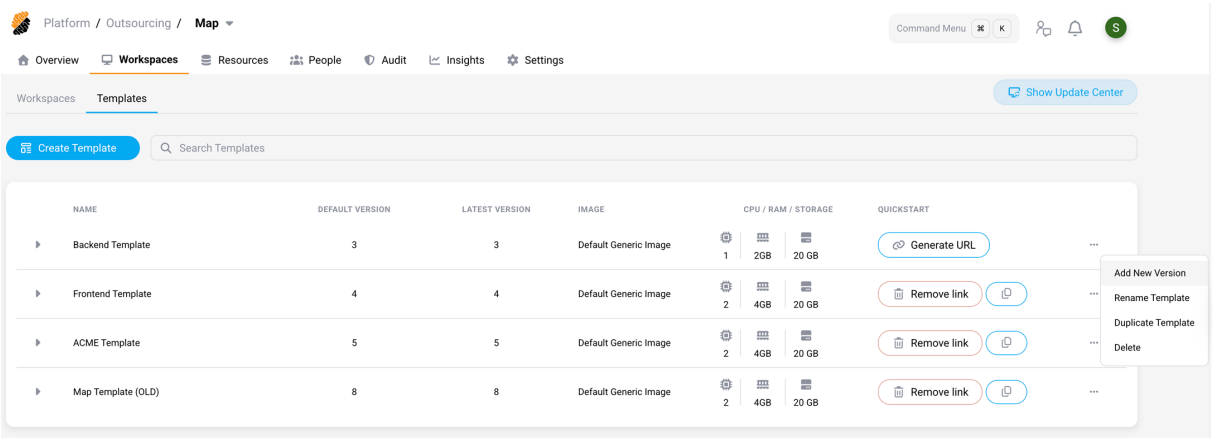
On the Workspaces Page, in the **Templates** section, you can create a template by clicking on the **Create Template** button.
You would follow the same steps as during the initial setup of a Workspace.

> **Note**
>
> Mount points are supported only in the primary region, but the template editor lets you attach one
> regardless of the region the project works in. A template that carries a mount point is therefore usable only in the primary region: Workspaces launched from it in any other region cannot use that
> storage. When a developer creates a Workspace from such a template in the project, the region selector is locked to the primary region. See
> Mount points in workspace templates.

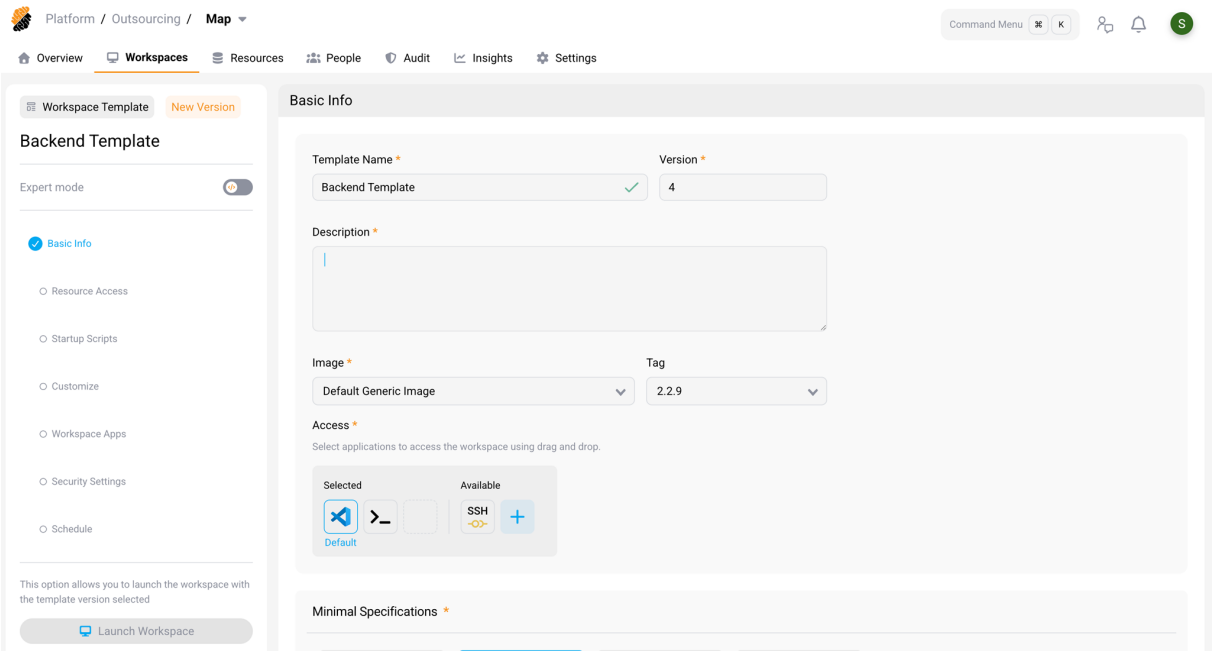
Create a new version of a Template

Template versions allow you to adjust the configuration of a template programmatically. A new version can be created by clicking on the "**...**"button on the right of a template and select **Add new version**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



This opens the same configuration UI as for creating a new Workspace or template, but with all current configurations, specified in the most recent version of the template, loaded.



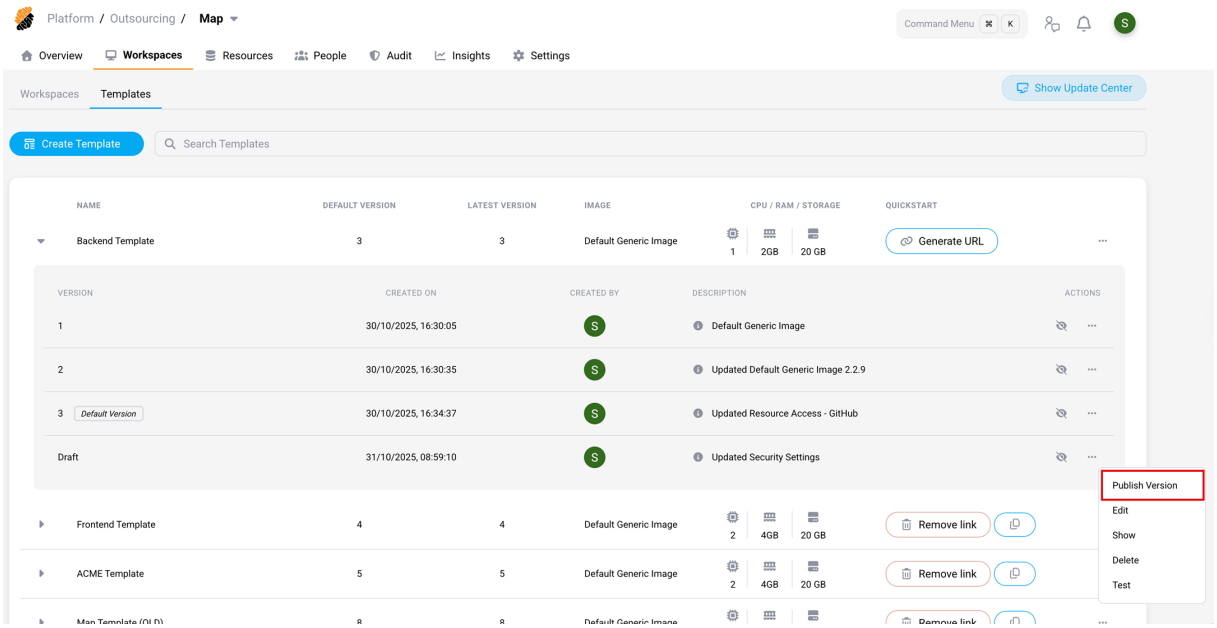
After making the necessary changes, you can either save the new version as a draft, which allows further modifications, or save it as a final template version right away, which cannot be changed afterwards.

A draft or new template version can be tested by either:

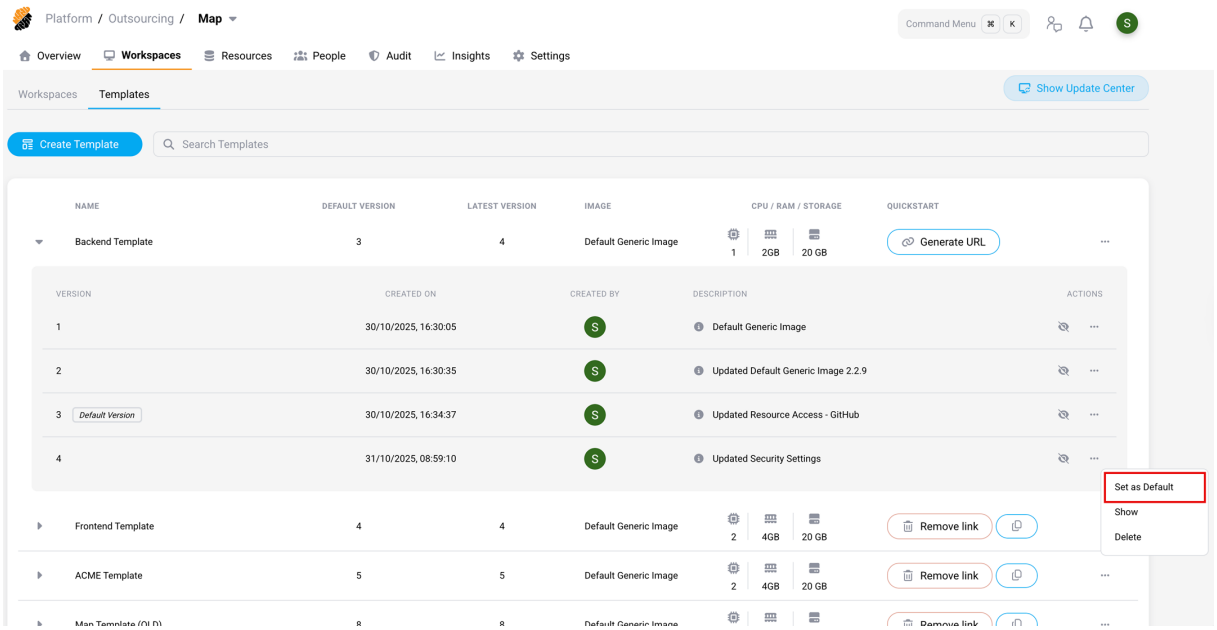
- Selecting **Launch Workspace** right within the template editor.
- Clicking on the "**...**"button on the right of a template and select **Test**.
- Manually selecting it from the list of version in the **Create Workspace from Template** wizard.

After finalizing a draft version, it can be published as a new template version by clicking on the "**...**" button on the right of a template and select **Publish Version**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



To ensure the new version of the template is automatically selected for newly created workspaces, click on the "**...**"button on the right of a template and select **Set as Default**.



> **Note**
>
> A template version cannot be deleted while any Workspace still uses it, and a template cannot be
> deleted while any of its versions are still in use. To retire a template that is still in use, see Archive a template.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Track which template version a Workspace uses

Every Workspace records the template and the template version it was created from. This lets you see which Workspaces still run an older configuration, so you can bring them back to your current standard.

> **Note**
>
> Version tracking applies to Workspaces created from a template. Workspaces created before this feature was introduced, and Workspaces created with the plain **Create Workspace** wizard rather than from a template, have no recorded version and are listed as **unversioned**.

Version status indicators

In the **Templates** section of the Workspaces Page,
each template name shows a counter for each group of Workspaces that is not aligned with the default version:

- **In testing** —Workspaces created from a draft version.
- **Outdated** —Workspaces created from a published version that is no longer the default version.
- **Unversioned** —Workspaces that have no recorded template version.

Workspaces that already use the default version are compliant and are not counted.

Select any counter to expand the template and open the **Associated Workspaces** window.

Review and export associated Workspaces

The **Associated Workspaces** window lists every Workspace that uses the template, grouped by the template version it was created from. Open it by selecting a status counter, by selecting **Associated Workspaces** in the expanded template, or by selecting the associated count on a version row.

1. Expand a version row to list the individual Workspaces and their owners.
2. Select individual Workspaces, or use the version-level checkbox to select every Workspace on that
   version.
3. Select **Export as CSV**.

Expected result: A CSV file downloads with the columns **Workspace Name**, **Owner**, **Owner Email**, and **Template Version**. Draft versions are exported as `Draft` and Workspaces with no recorded version are exported as `Unversioned`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Reviewing and exporting associated Workspaces requires permission to manage Workspace templates.
> Developers can create Workspaces from a template, but cannot open this window.

Update a Workspace to the default template version

When you set a new default version, existing Workspaces continue to run the version they were created from. Each affected Workspace shows an update indicator in the platform, and the update is applied to
one Workspace at a time. The workspace owner chooses when to apply it.

An update can be applied by the workspace owner, or on their behalf by a project owner, platform administrator, organization owner, or security officer.

> **Warning**
>
> Applying an update restarts the Workspace and **replaces** the Workspace configuration with the configuration defined in the default template version. Persistent data, such as the home directory
> and source code, is preserved. Configuration changes made in the Workspace after it was created,
> such as an added repository or secret, are lost. This action cannot be undone.

An update never shrinks a Workspace's disk. If the Workspace disk is larger than the new template version
asks for —because it was expanded after the Workspace was created —the Workspace keeps its current size.
A larger disk in the new version is applied as usual.

Apply an update

1. On the Workspace card, select the update icon in the card header, or select **Update** in the "**…**"menu.
2. Review the update details. When the Workspace was created from a published version, the window
   shows the current and new version together with a **Configuration Change(s)** list of the repositories, secrets, data buckets, mount points, connected services, images, network policy, and startup script that are added or removed.
3. Select **Update**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Expected result: The Workspace restarts on the new version and the update indicator clears.

> **Note**
>
> For a Workspace with no recorded template version, the window confirms which version the Workspace
> is aligned to but does not list the individual configuration changes.

Update before editing

A Workspace that is behind the default template version must be updated before its configuration can be edited. When you edit such a Workspace, a **Please Update Your Workspace** window opens instead of
the edit page. This applies when editing from the Workspace card, from the **Workspaces** tab of your profile, from the project Workspaces list, and when importing an IDE configuration.

Select **Update** to apply the update and continue to the edit page.

Archive a template

Archive a template when it must no longer be used for new Workspaces but still has Workspaces associated with it. Archiving is the alternative to deleting a template that is still in use.

To archive a template, select the "**...**"button on the right of a template and select **Archive**.

Expected result: The template is marked with an **Archived** badge and can no longer be selected when creating a Workspace, including through a Quickstart link. Existing Workspaces created from the
template are unaffected and can still be updated to a version of the template.

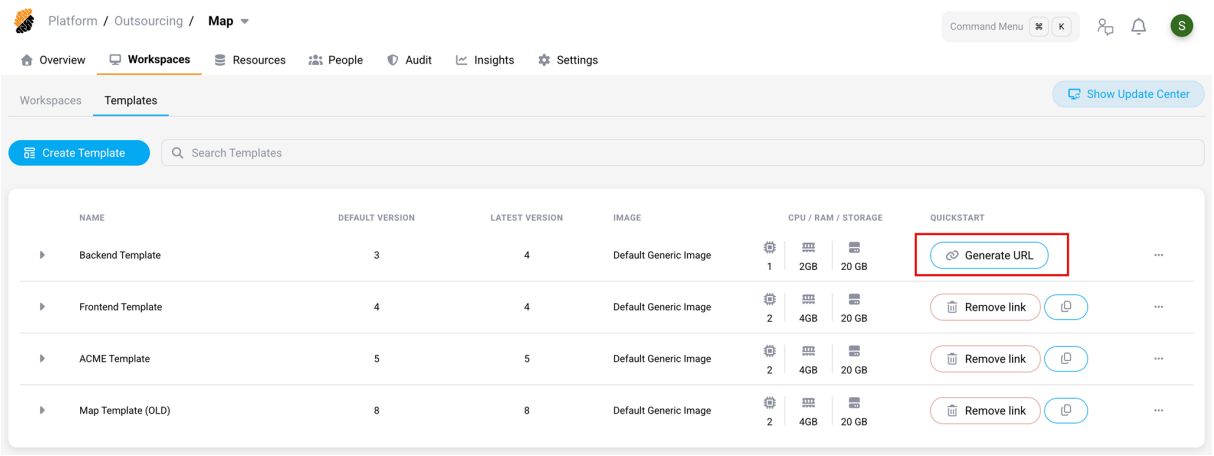
Archived templates are hidden by default. Use the archived filter on the **Templates** section to show or hide them. To make an archived template available again, select the "**...**"button and select **Restore**.

Quickstart

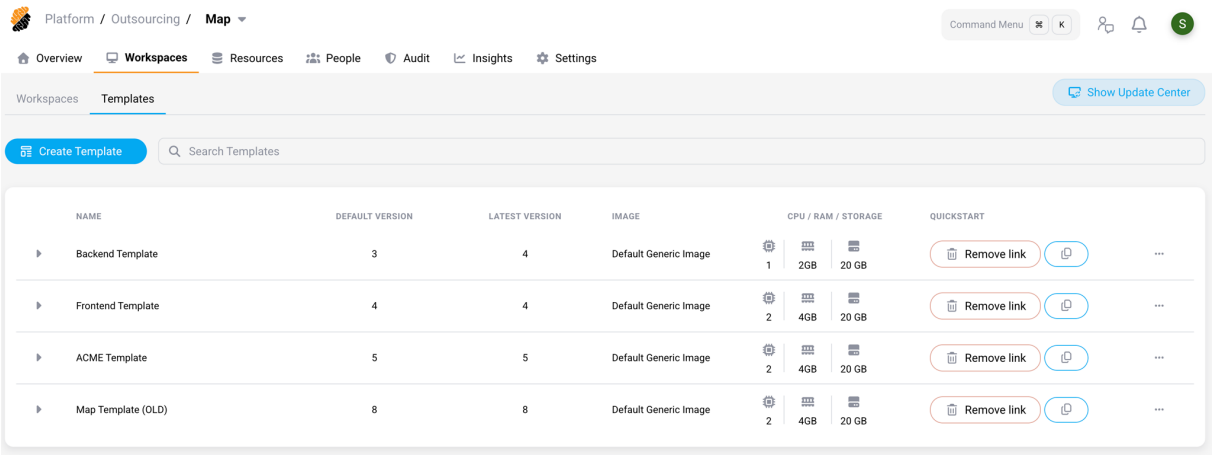
The Quickstart functionality allows developers to create a new workspace with a single click from a code repo, engineering portal or any other location outside of Citrix SecurSpaces™.

Create a Quickstart link by clicking the **Generate URL** button on the right of a template.

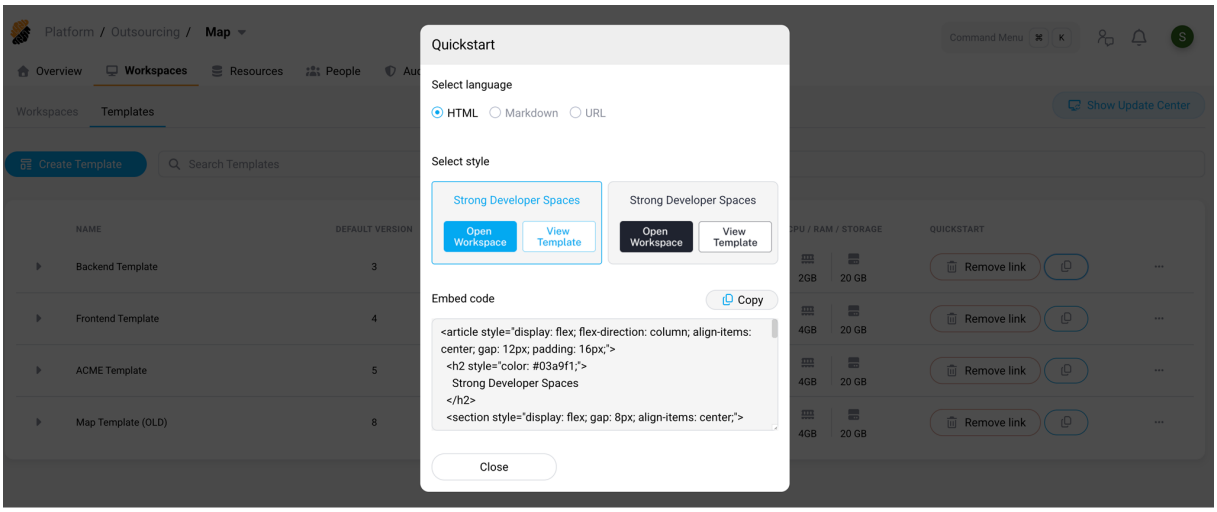
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Then click the **Copy** icon.



Select any of the available options.

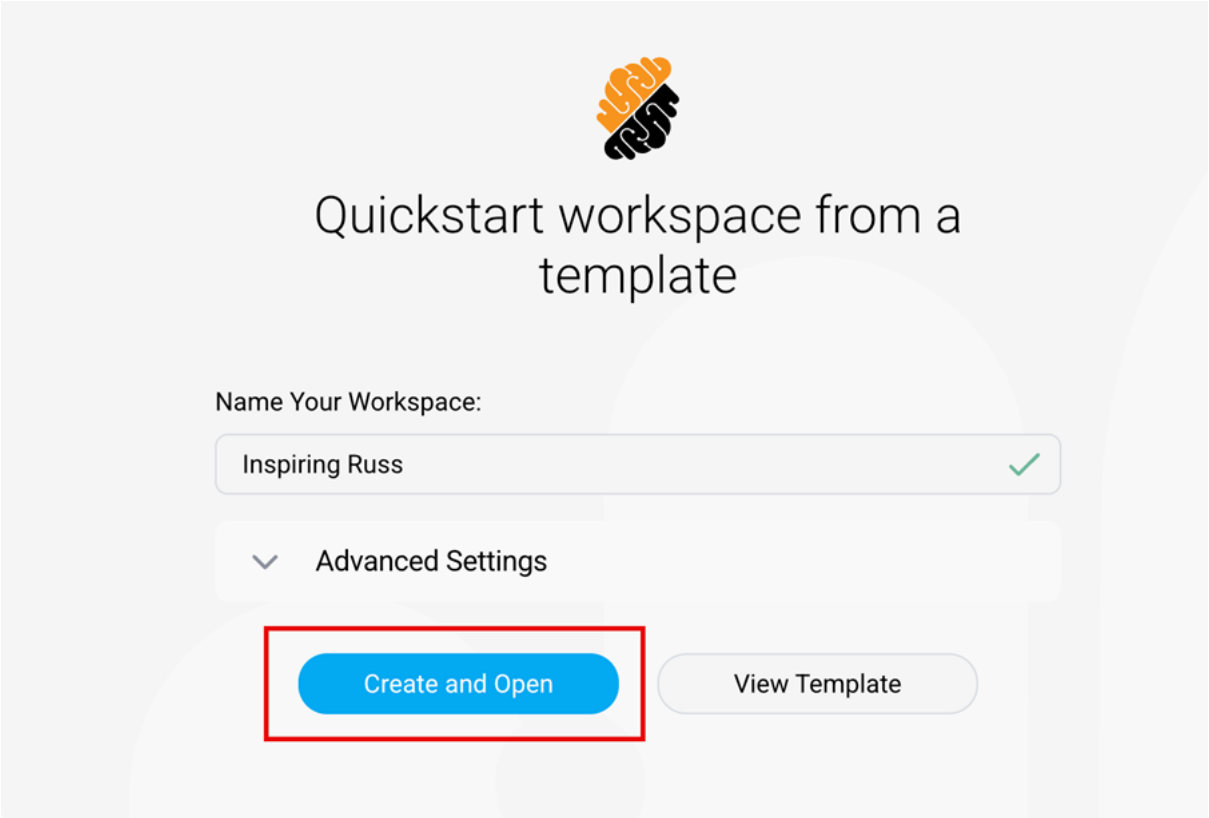


When a user accesses the Quickstart URL SecurSpaces initiates the creation of a new Workspace, unless the user already has a Workspace based on this particular template. In this case, the user will be

© 1997–2026 Citrix Systems, Inc. All rights reserved.

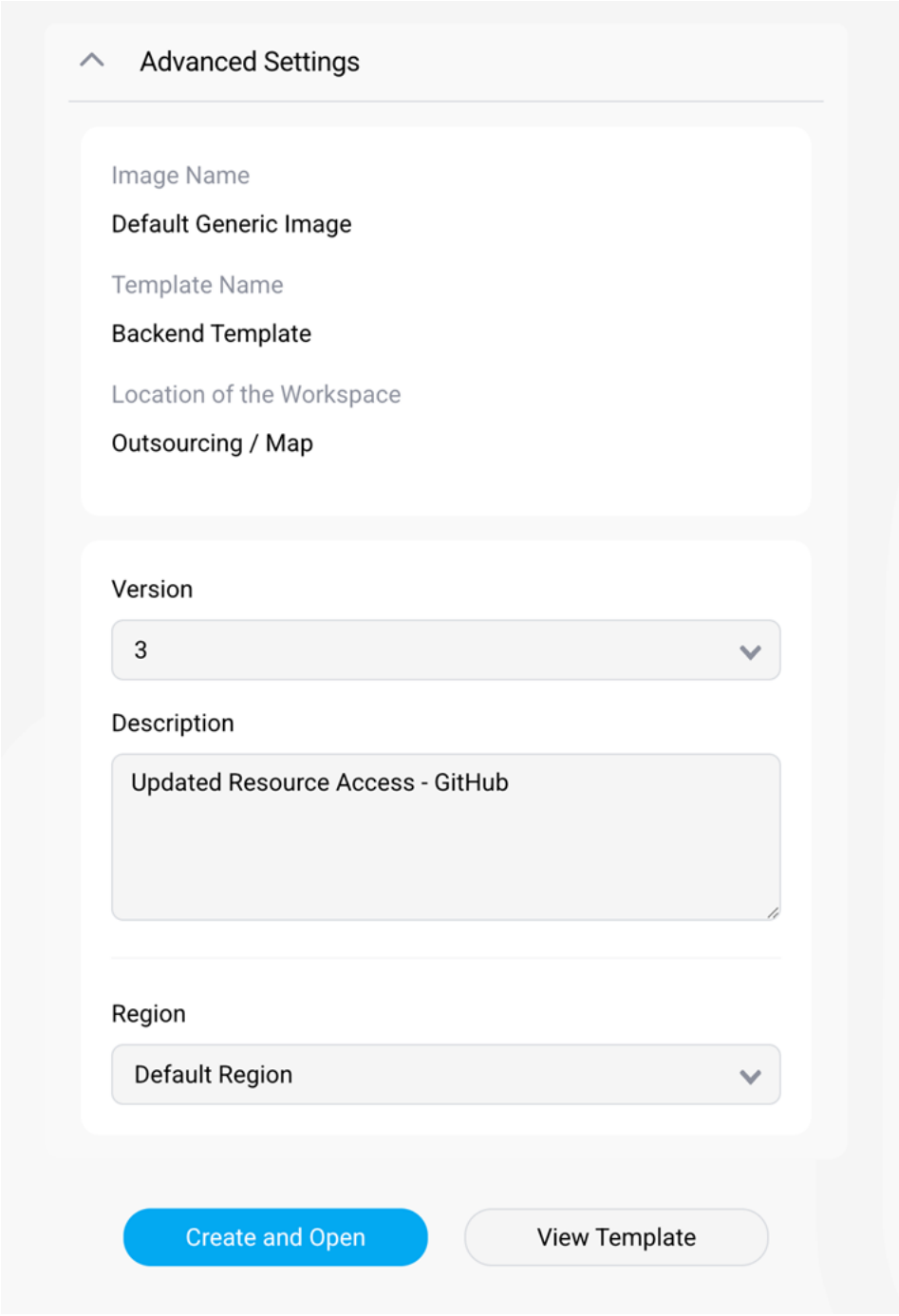
forwarded to the respective Workspace automatically.

When a new Workspace needs to be created the user can configure the name of the Workspace and finalize the creation flow, by selecting **Create and Open**.



Via the **Advanced Settings** menu, configuration details, such as base template or related SecurSpaces project, can be verified and template version as well as deployment region can be configured.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

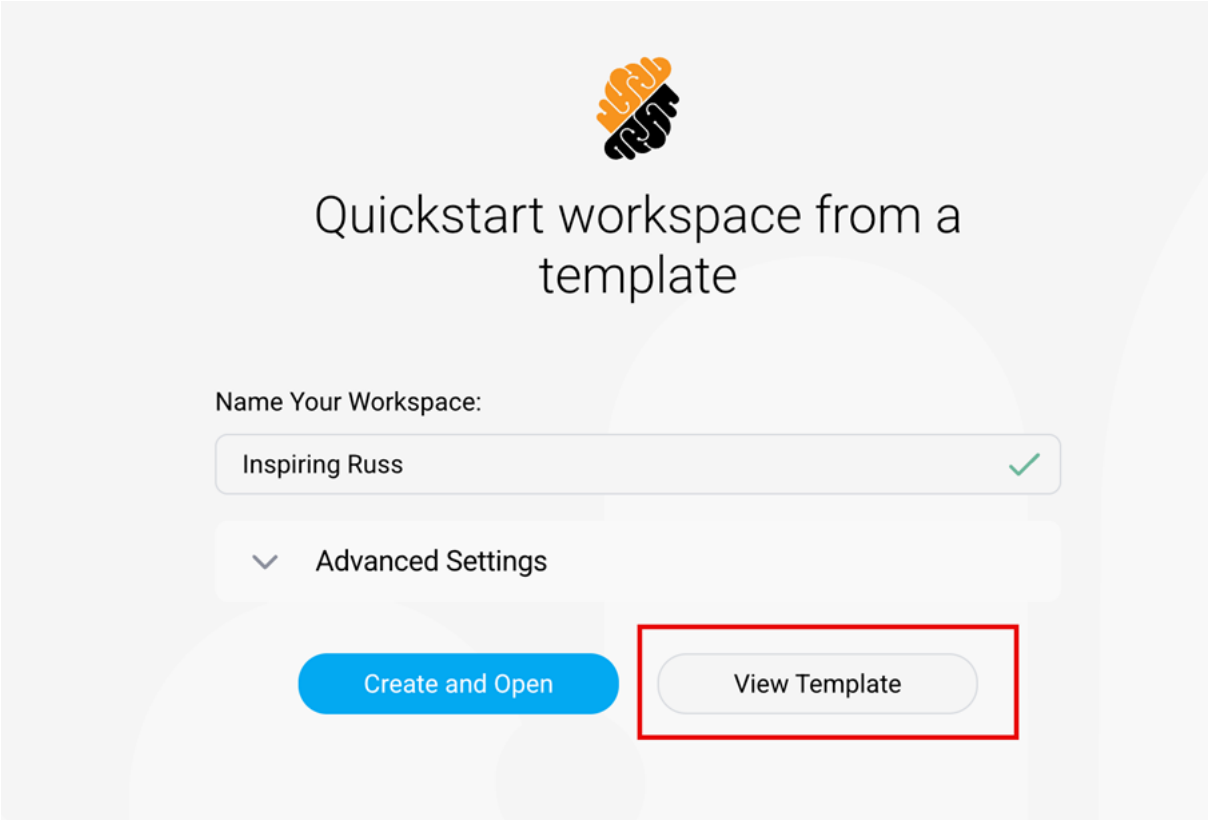


Warning

The deployment region offered here is not checked against the template's mount points. If the template carries a mount point and you select a region other than the primary one, the Workspace is
created in that region with the mount point attached, but it cannot use that storage. Leave the region set to the primary region when the template carries a mount point.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The **View Template** button opens the Workspace Template editor for the selected version, to verify further configuration details.



Automate Quickstart links with the REST API

You can also manage Quickstart links through the REST API. Use the API when you need to publish template launch links in
a developer portal, repository README file, or internal onboarding workflow.

The Quickstart link API can:

- Enable a Quickstart link for a workspace template.
- Return the raw URL, view-template URL, and generated Markdown or HTML embed snippets.
- Generate light or dark embed snippets.
- Revoke a Quickstart link when it must no longer be used.

For endpoint details, see REST API.

Include Custom Access Items in templates

Workspace templates can include Custom Access Items. Use Custom Access Items to provide one-click access to tools or

© 1997–2026 Citrix Systems, Inc. All rights reserved.

services that run inside workspaces created from the template.

Custom Access Items can be managed in the workspace template configuration or through the REST API. Each item defines a
display name, port, command, and optional health check settings.

For endpoint details, see REST API.

Duplicate a template

A Workspace Template can be duplicated by clicking on the "**...**"button on the right of a template and select **Duplicate**. This allows quickly creating new templatized configurations based on existing templates.

Related information

- Run AI coding agents in workspaces —what a template needs when agents run in it
- Workspace images

Build and manage custom workspace images for SecurSpaces

While standard images provide a quick start, using custom workspace images within Citrix SecurSpaces™ ensures your development environment is tailored to your organization's needs while maintaining strict compliance.

Benefits of custom images

- **Tailored developer experience:** Pre-install project-specific runtimes (Node, Python, Go) and CLI utilities (kubectl, terraform). You can pre-configure internal network settings, CA certificates, IDE extensions, and shell aliases so the workspace is ready immediately upon launch.
- **Standardized governance:** Ensure every container follows internal standards, including approved, hardened OS distributions and software provenance from trusted repositories.
- **Proactive security:** Integrate images into your vulnerability management lifecycle. Trigger automated builds to rotate images when a CVE is detected and reduce the attack surface by removing unnecessary packages.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

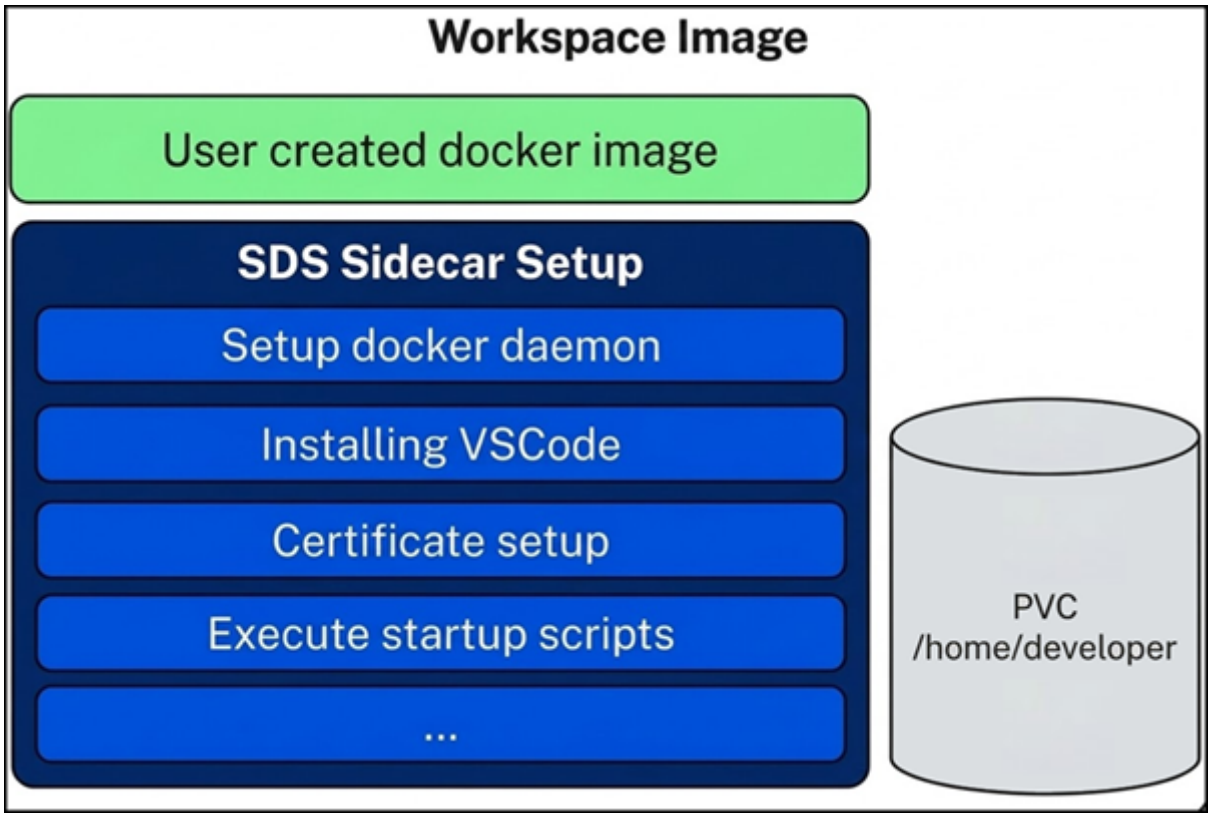
Architecture

Runtime architecture

In the SecurSpaces platform, a workspace runs inside a Kubernetes Pod. There is a one-to-one relationship between a workspace and a Pod to ensure container-level isolation.

The workspace runtime consists of three primary layers:

- **The workspace container:** The execution environment containing your compilers, debuggers, and tools.
- **Persistent storage:** A persistent volume (PV) mounted in the user home directory (`/home/developer`). This ensures that code, local configurations, and shell history persist if the Pod restarts.
- **Platform sidecars:** Auxiliary containers managed by the SecurSpaces platform that provide features like startup scripts and Docker-in-Docker support.



Layered image architecture

We recommend a multi-layered image architecture to balance centralized governance with developer autonomy:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Base Image:** A hardened OS provided by Corporate IT. Includes enterprise CA certificates and core security configurations.
- **Organization Image:** Owned by the SecurSpaces Platform team. Contains core SecurSpaces dependencies, common development tools, and global proxy settings.
- **Project Image:** Owned by application teams. Includes specific runtimes (e.g., JDK 17, Node 20) and project-specific dependencies.

Understanding persistence

In a SecurSpaces workspace, only the `/home/developer` directory is persistent. When a workspace restarts, any changes to the root filesystem (`/etc`, `/usr`, `/var`) are discarded.

> **Important:**
>
> The persistent volume is mounted at runtime. Any files placed in `/home/developer` during the `docker build` phase will be overwritten by the persistent volume once the workspace starts.

Persistence design patterns

- Binaries: Install binaries in system directories like `/usr/local/bin` during the Docker build.
- Local data and configuration: Configure tools to store logs, cache, and plugins in `/home/developer`.
- Global configuration: Use `/etc/` for global tool settings to serve as a fallback.

Non-standard persistence

Tools that dynamically install binaries to the home directory (e.g., `conda`, `nvs`, `gcloud CLI`) must be treated as Local Data. Use startup scripts to ensure these configurations are restored or initialized after the persistent volume is mounted.

Automated configuration with startup scripts

Platform-managed scripts

Managed via the SecurSpaces UI, these scripts are executed every time a workspace starts.

- **Pre-startup scripts:** Run during the initial container initialization phase.
- **Post-startup scripts:** Run after the workspace services are active.
- **Idempotency:** Because these run on every start, ensure scripts check for existing configurations:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
if [ ! -f "/home/developer/.my_tool_config" ]; then
    echo "Initializing configuration..."
fi
```

Image-embedded scripts

Best for strict DevOps practices, these scripts are stored in the Docker image and run only once during the initial workspace creation.

- **Target directory:** `/usr/bin/strong_network_startup/`
- **Permissions:** The `developer` user must have execution permissions.
- **Naming:** Scripts run in alphabetical order. Use numeric prefixes (e.g., `01_setup.sh`).

Files without an execute bit are skipped silently, so set the mode when you copy the script in. A failing script logs an error and the remaining scripts still run.

Pre-install VS Code extensions

The base image example creates `/usr/bin/strong_network_startup/vscode_extensions`. That directory is not
part of the startup-script mechanism: it is where the platform looks for extensions to install into the Cloud IDE.

Put `.vsix` files there and they are installed the first time a workspace starts from the image. This is how
you ship a standard extension set with a project image, rather than asking every developer to install the
same extensions by hand.

```
COPY --chown=developer:developer extensions/*.vsix /usr/bin/
    strong_network_startup/vscode_extensions/
```

To use a different directory, set the `VSCODE_EXTENSION_INSTALL_DIR` environment variable on the workspace.
When it is not set, the platform uses the path above.

Create an image on the platform

If your team only needs a few additions to an existing image, SecurSpaces can build the derived image for
you, without a Dockerfile or a pipeline. See
Create an image from an existing one.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Security best practices

Secret handling

Never embed credentials (SSH keys, API tokens) directly in a Docker image. SecurSpaces provides secure injection at runtime:

- **Environment variables:** Best for API keys and usernames.
- **File mounts:** Secrets are mounted in the `/secrets/` folder. Preferred for license files and cryptographic keys.

Third-party integrations

Always prefer native platform integrations (e.g., JFrog Artifactory, Git providers) over manual secret injection. SecurSpaces handles the authentication layer automatically, so you do not need to manually manage `.npmrc`, `.gitconfig`, or `.docker/config.json` files in your image.

System requirements

Custom images must meet the SecurSpaces container image requirements —core packages, per-distribution and
per-architecture packages, and the `developer` UID 1000 account. See
Container image requirements.

Build your first base image

Design philosophy

- **Completeness over minimality:** Unlike production images, workspace images should prioritize developer experience by including all necessary compilers, headers, and `-dev` libraries.
- **Skip multi-stage builds:** Keep tools installed during the build process available for the developer.
- **Orchestration overrides:** SecurSpaces overrides `ENTRYPOINT` and `CMD` to launch platform services. Use startup scripts for initialization logic.

Ubuntu 24.04 example

```
1  FROM ubuntu:24.04
2
3  # 1. Install SecurSpaces core requirements
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
4
5  RUN apt-get update && apt-get install -y --no-install-recommends \
6      curl bash tar git git-lfs openssh-client sudo \
7      && apt-get clean && rm -rf /var/lib/apt/lists/*
8
9  # 2. Create developer user (Repurposing default 'ubuntu' user for
       24.04)
10
11 RUN usermod -l developer -d /home/developer -m ubuntu && \
12     groupmod -n developer ubuntu
13
14 # 3. Setup startup directory
15
16 RUN mkdir -p /usr/bin/strong_network_startup/vscode_extensions && \
17     chown -R developer:developer /usr/bin/strong_network_startup && \
18     chmod u+rwx /usr/bin/strong_network_startup
19
20 # 4. Run as the developer user, in the developer home directory
21
22 USER 1000
23 WORKDIR /home/developer
```

When the image is added, SecurSpaces tests it before anyone can use it. See
How the platform checks an image.

Install a private CA certificate

If your organization runs its own certificate authority, add the root certificate to the system trust store at build time. Without it, `git`, `curl`, and package managers inside the workspace fail to verify internal
hosts.

```
1  COPY internal-ca.crt /usr/local/share/ca-certificates/internal-ca.crt
2  RUN update-ca-certificates
```

On Red Hat and CentOS, copy to `/etc/pki/ca-trust/source/anchors/` and run `update-ca-trust` instead.

This is separate from the certificate SecurSpaces injects when a network policy is attached to a workspace,
which the platform handles for you. See
Certificates when a network policy is attached.

Some tools ship their own CA bundle and ignore the system store. Those need pointing at your certificate
separately, usually through an environment variable.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enabling root access

To grant the developer user sudo privileges, add the following to your Dockerfile:

```
RUN apt-get update && apt-get install -y sudo && rm -rf /var/lib/apt/
    lists/*

# Grant passwordless sudo

RUN echo "developer ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers.d/developer
     && \
     chmod 0440 /etc/sudoers.d/developer
```

> **Note:**
>
> For increased security, you can allowlist specific tools instead of granting full access:
> `developer ALL=(ALL)NOPASSWD: /usr/bin/apt-get, /usr/bin/systemctl`

Use Docker inside a workspace

Workspaces can build and run containers. A Docker daemon runs as a sidecar in the workspace Pod and its
socket is exposed to the workspace at `/var/run/docker.sock`, so **you only need the Docker client in your
image, not the daemon**.

```
# Install the Docker CLI, Buildx, and Compose from the official static
    builds.
# Pin the versions your organization has approved rather than tracking
    latest.
ARG DOCKER_VERSION=27.3.1
RUN curl -fsSL "https://download.docker.com/linux/static/stable/x86_64/
    docker-${
 DOCKER_VERSION }
 .tgz" \
      -o /tmp/docker.tgz && \
    tar -C /tmp -xzf /tmp/docker.tgz && \
    mv /tmp/docker/docker /usr/local/bin/ && \
    rm -rf /tmp/docker /tmp/docker.tgz
```

Install the CLI into a system directory such as `/usr/local/bin`, not into `/home/developer`, so it survives
a restart. See Understanding persistence.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What the workspace is allowed to do

In the default rootless configuration the socket does not lead straight to the daemon. It is a proxy that inspects each request and rejects anything that would break workspace isolation. Ordinary builds and
`docker run` work normally; the following are refused:

Rejected	Examples
Privileged containers	`--privileged`
Host namespaces	`--pid=host`, `--network=host`, `--ipc=host`, `--uts=host`
Added capabilities beyond Docker's defaults	`--cap-add=SYS_ADMIN`, `--cap-add=ALL`
Relaxed security options	`--security-opt apparmor=unconfined`, `seccomp=unconfined`, `no-new-privileges=false`
Host device access	`--device`
Custom cgroup parent	`--cgroup-parent`
Bind mounts from outside the workspace	Only `/home/developer`, `/tmp`, and `/var/run/docker.sock` are permitted as sources

A rejected request fails with a message naming the restriction, for example
`privileged containers are not allowed`. Named volumes are unaffected.

> **Note:**
>
> **Root Docker in Docker** is a platform feature flag. When a platform administrator turns it on, the socket
> reaches the daemon directly and these restrictions no longer apply. The flag can be set for everyone, or
> for selected users and groups, so it may be on for some developers and not others. See
> Turn features on or off.

Examples published by Citrix

Rather than starting from scratch, you can adapt an image Citrix already builds:

- strong-network/images on GitHub —the Dockerfiles and build scripts behind the supplied images, including base, generic, language-specific, and GUI variants.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The
repository has a makefile, so `make base_image` or `make all` builds them locally.
- strongnetwork on Docker Hub —the same images, prebuilt, which
you can reference directly or use as a `FROM` line.

Related information

- Container image requirements —what an image must
provide, and the check it has to pass
- Run AI coding agents in workspaces —packaging agent tooling so it
is present at workspace start
- Templates
- Image caching

Container images

Container images or also Cloud Development Environments (CDEs) are used to define the configuration of a development environment. Typically, CDEs define all the software dependencies necessary for building the intended application once implemented. Users create workspaces with such an image as "blueprint", and begin contributing code to the project within this context.

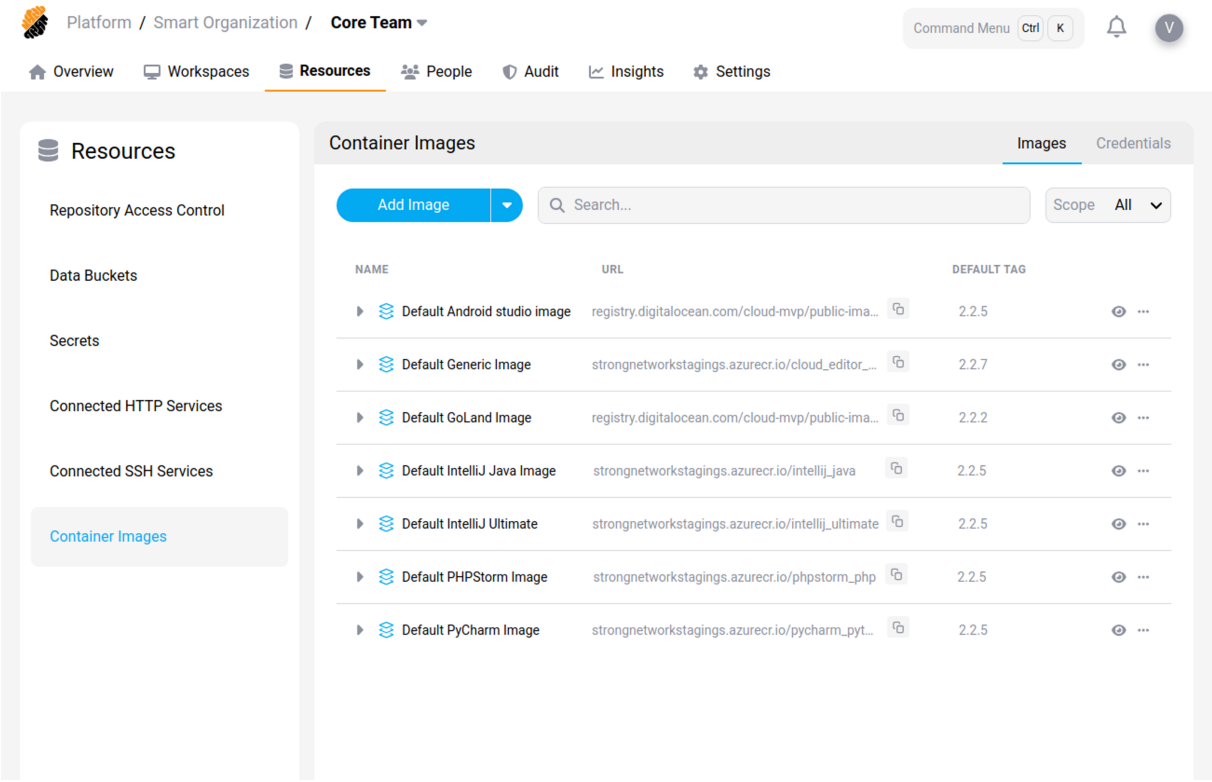
CDE images are imported from a registry as part of the resources available to users on the platform. Registries are either public or private.
For private registries, you need to provide credentials to authenticate properly before importing the image. Public registries, by definition, do not need credentials.

- View CDE Images
- Add a CDE Image
- View Registry Credentials
- Add a Registry Credential
- Update a Registry Credential

View CDE Images

The panel displays the available CDE images in the project. You may search for one or filter those used in workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



A CDE image is defined by the following characteristics:

- **Basic information**: Name, scope of use (platform, organization or project), URL and tag.
- **Class Level**: This option defines the visibility for the CDE image based on the user's permissions.
- **Asset Information**: This option allows for providing a description of the CDE.

By clicking on a CDE image, you can see a list of the CDE's versions followed by basic details such as imported date, status.

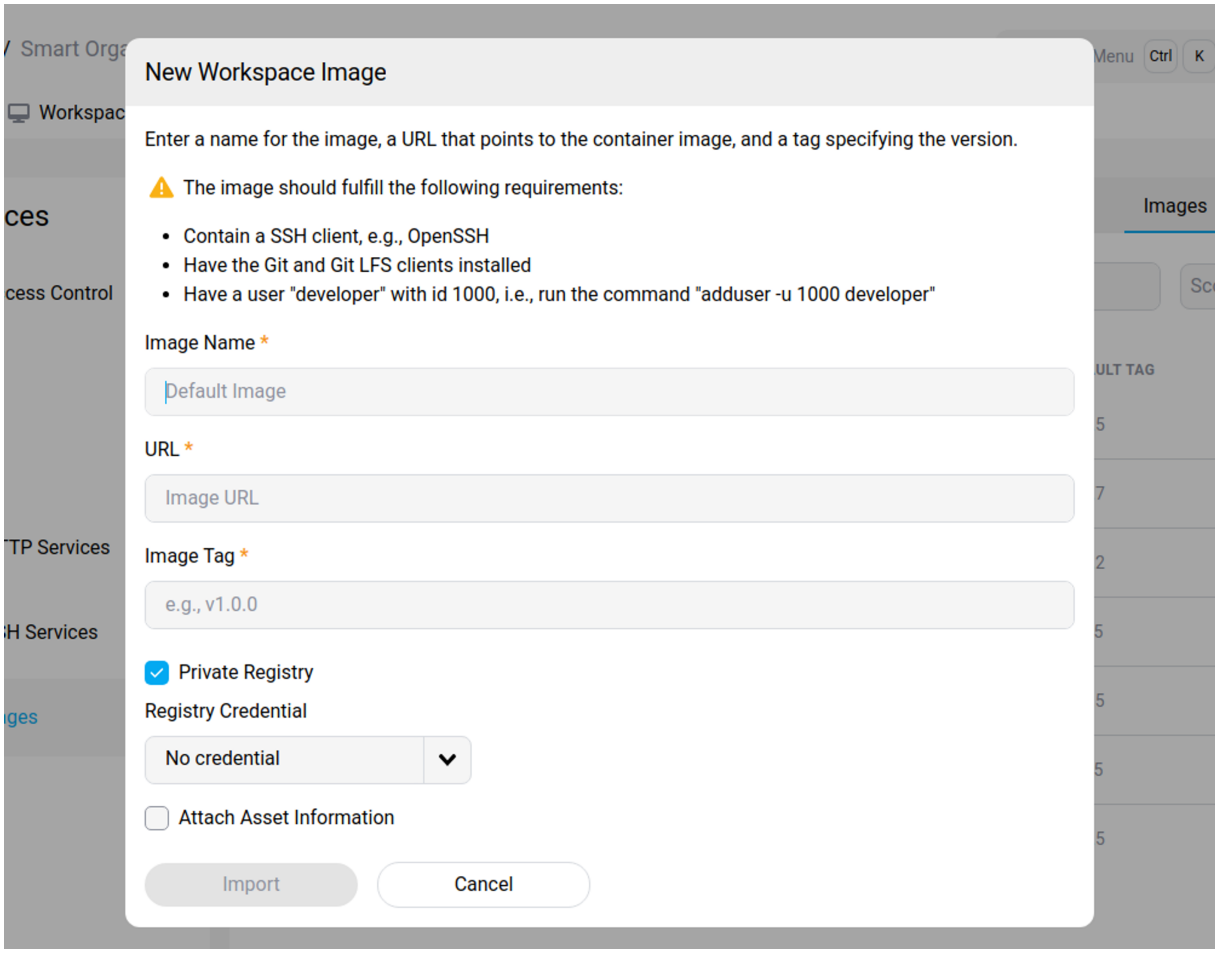
Add a CDE Image

Requires the Resources permission set to **Import**.

Select **Add Image**. The form opens as **New Workspace Image** and asks for:

1. **URL**, the full path to the image, in the form `<registry>/<namespace>/<repository>:<tag>`.
2. **Image Name**, a name to identify the image.
3. **Image Tag**, for example `v1.0.0`.
4. **Private Registry** (optional), the credential used to reach a private registry.
5. **Attach Asset Information** (optional), a description of the image.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



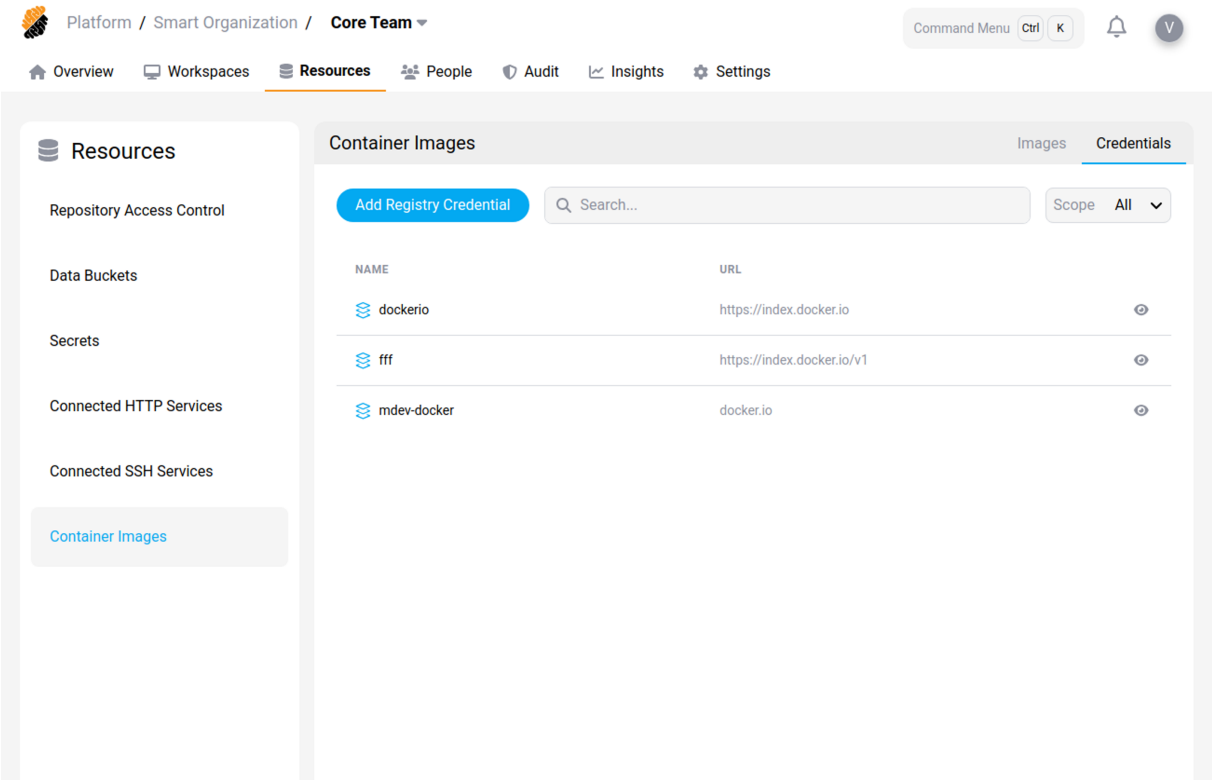
> **Warning**
>
> The image must meet the SecurSpaces container image requirements before it can be used for a workspace. At
> minimum it needs an SSH client, Git and Git LFS, and a user named `developer` with UID 1000. For the full
> list, see
> Container image requirements.

You can edit or delete a CDE image by clicking on the "**...**"icon next to its class level.

View Registry Credentials

To display credentials used in the project click on the "**Credentials**"button on the top right of the panel. You may search for one or filter those used in workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



A Registry Credential is defined by the following characteristics:

1. **Name**,
2. **Scope** and
3. a **URL**.

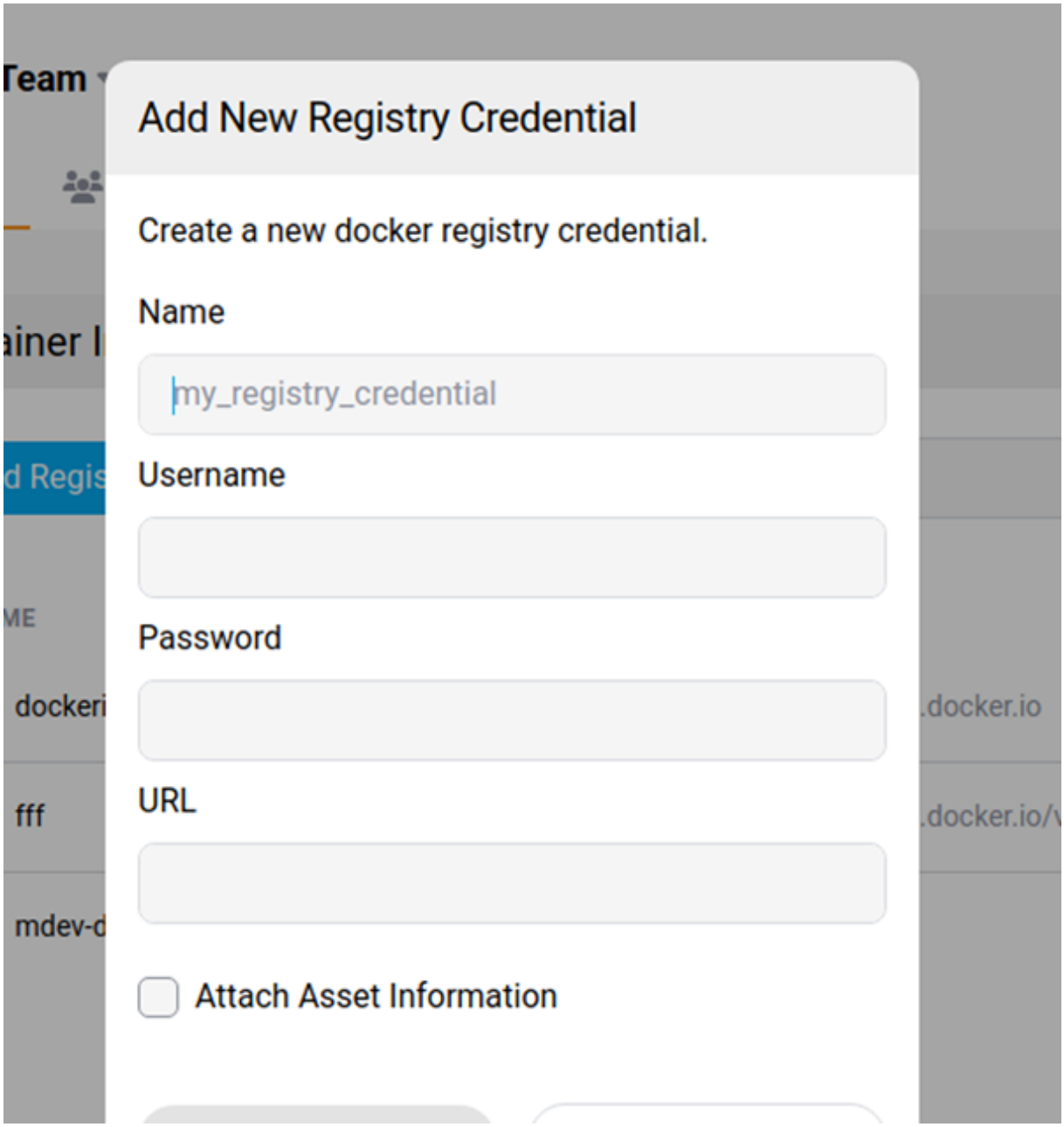
For security purpose, no credentials are directly exposed or available for consultation.

Add a Registry Credential

Requires the **Security** permission set to **Manage**.

You can add a Registry Credential by pressing the "**Add Registry Credential**"button.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



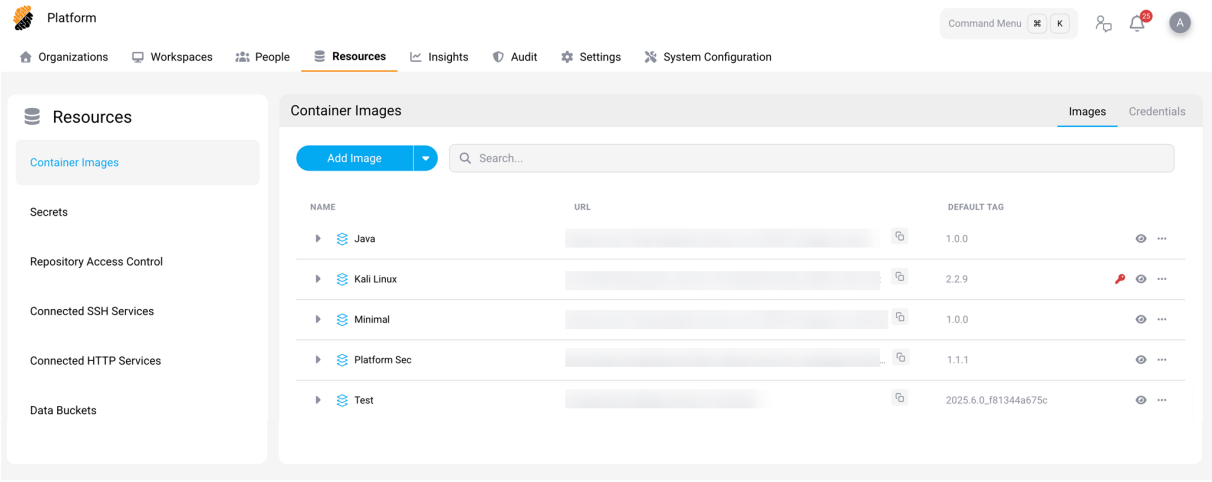
You will need to enter the following information:

1. **Name**, to identify the credentials when needed during the registration of a CDE image,
2. **Username**:, and
3. **Password**:, as credential values, and an
4. **URL**: where the authentication is performed.
5. **Asset information**, a description of the registry credential.

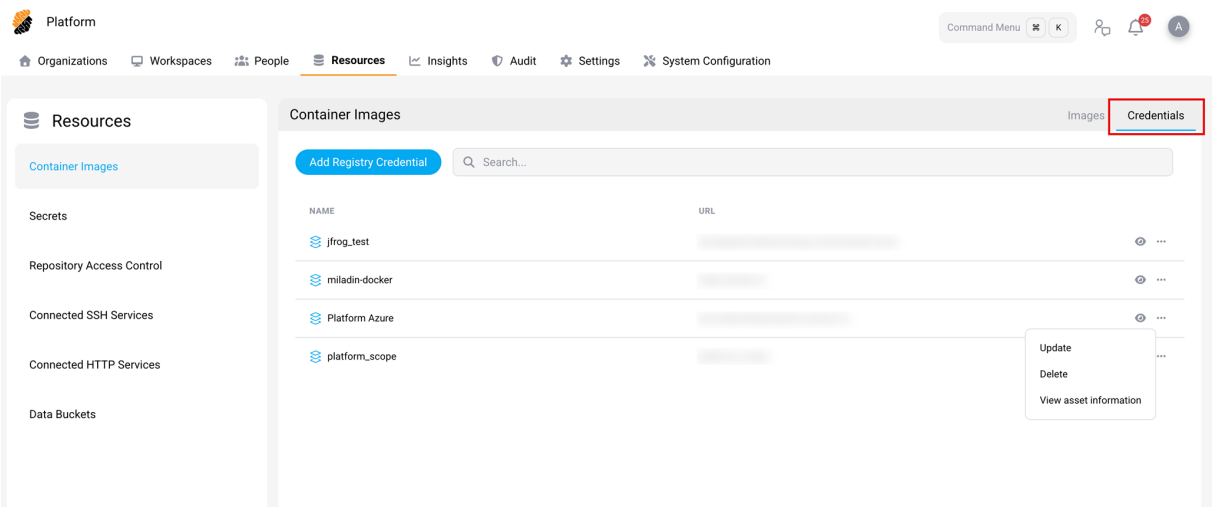
© 1997–2026 Citrix Systems, Inc. All rights reserved.

Update a Registry Credential

When a registry credential becomes invalid, a red key icon will be displayed next to the related container image, as shown in the screenshot below.



Hover over the icon to reveal the name of the credential, then switch to the **Credentials** view by clicking on the respective tab in the top right corner.
Find the credential that was identified before in the list of stored credentials and click on the "**...**" button on the right. Select **Update** to update the credential information.



Related information

- Create an image from an existing one
- Manage toolchains with Mise
- Manage toolchains with Nix
- Manage workspace toolchains

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Update an existing container image
- Workspace images

Update an existing container image

Overview

This article focuses on the platform workflow to update an existing container image so teams can reuse a consistent, preconfigured toolchain across workspaces. For day-to-day software persistence management, refer to the How to Maintain Persistent Changes in Your Workspace.

Why persistency matters

Container images define the baseline configuration of a development environment. Users create workspaces from an image as a *blueprint*, which helps prevent environment drift and speeds up onboarding.

Context: persistence and startup scripts

In SecurSpaces, only data in `/home/developer` is persistent by default. Changes outside that directory (for example OS-level package installations) do not persist across workspace restarts.

The best practice is to keep shared tooling (e.g. across a team, a project, etc.) in a workspace base image, so teams get a consistent toolchain across workspaces. Use startup scripts only for individual customization, e.g. user-specific.

For details about workspaces' persistence model and how to make subsequent installations persistent, refer to How to Maintain Persistent Changes in Your Workspace.

Manage container images (Resources → Container images)

> **Permission:**
>
> To add a workspace image, you need the Resources permission set to **Import**.

Images are imported from a registry (public or private). Private registries require credentials; public registries do not.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

View container images

The **Container images** panel (under **Resources → Container images**) displays the images available in the selected scope (project, organization, or platform). You can search for images or filter those currently used in workspaces.

Add a container image

To add an image:

1. Go to **Resources → Container Images** for your project.
2. You can add container images at the project, organization, or platform scope (depending on your permissions)
3. Select **Add Image**.
4. Provide **URL**, **Image Name**, and **Image Tag**. Optionally, set **Private Registry** and **Attach Asset Information**.

> **Warning:**
>
> For image requirements, see Container image requirements.

(Optional) Sample images

If you want examples of workspace images, Citrix publishes both Dockerfiles and prebuilt images:

- strong-network/images on GitHub —Dockerfiles and build scripts you can adapt.
- strongnetwork on Docker Hub —prebuilt images you can reference directly.

Make an environment persistent for your team

There are two common ways to standardize tooling for a team:

Option 1: Create a new workspace image (where supported)

In product deployments, Citrix SecurSpaces™ can transform a startup script into a new workspace image automatically whenever needed. Feature availability can differ between environments.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Option 2: Start from an existing image and standardize configuration with startup scripts

Use this option when you want a reliable baseline and only need to apply configuration or smaller installations at workspace startup. Startup scripts can run pre-start or post-start depending on whether the workspace must wait for them.

Points of consideration

- Only `/home/developer` is persistent between workspace restarts. Plan installations accordingly (image, user-space tooling, or startup script).
- Image prerequisites: verify SSH, GIT, GIT LFS, and developer UID 1000 before sharing an image broadly.
- Governance: consider restricting allowed registries and configuring private registries centrally (Image URL Constraints and Create Image Registry).

References

- Maintain persistent changes in your workspace
- Container Images
- Create a Workspace
- Resource Settings (Image URL Constraints, Create Image Registry)
- Sample images repository on GitHub
- Prebuilt images on Docker Hub

FAQ

- Why did my installed tools disappear after a restart?

  Only `/home/developer` is persistent. Anything installed elsewhere must be baked into the image or re-applied at startup.

- What is the fastest way to onboard a team?

  Provide a workspace image that includes the shared toolchain and keep only dynamic configuration in startup scripts.

- When should I use post-start?

  Use post-start for heavy work so the workspace becomes available quickly while setup continues in the background.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Container images
- Create an image from an existing one
- Manage toolchains with Mise
- Manage toolchains with Nix
- Manage workspace toolchains
- Workspace images

Create an image from an existing one

August 19, 2026

Requires the Resources permission set to
Import.

Most teams need a standard image with a few additions: a language runtime, an internal CLI, a certificate.
Maintaining a Dockerfile and a build pipeline for that is disproportionate work.

Create Image From Existing does it on the platform instead. You pick a base image, supply a bash script,
and Citrix SecurSpaces™ builds a new image with your script as the final RUN layer of the Dockerfile, then
pushes it to your registry. The result appears as an ordinary container image that any workspace or template
can use.

Before you start

The option only appears when a platform administrator has configured a **destination registry** —the registry that images built on the platform are pushed to. Until then, **Add Image** is a plain button with no
dropdown.

If you do not see **Create Image**, that setting is missing. See
Registry access.

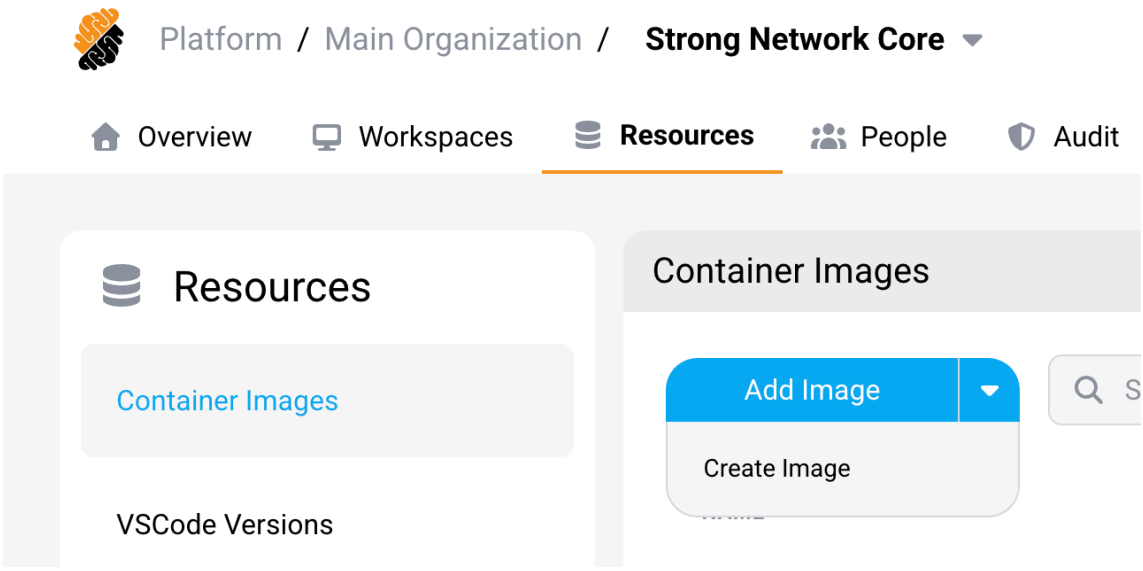
You also need a **registry credential whose URL matches the destination registry**. The build pushes with it,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

and workspaces pull the finished image with it. If no credential matches, the modal says so and **Confirm**
stays disabled.

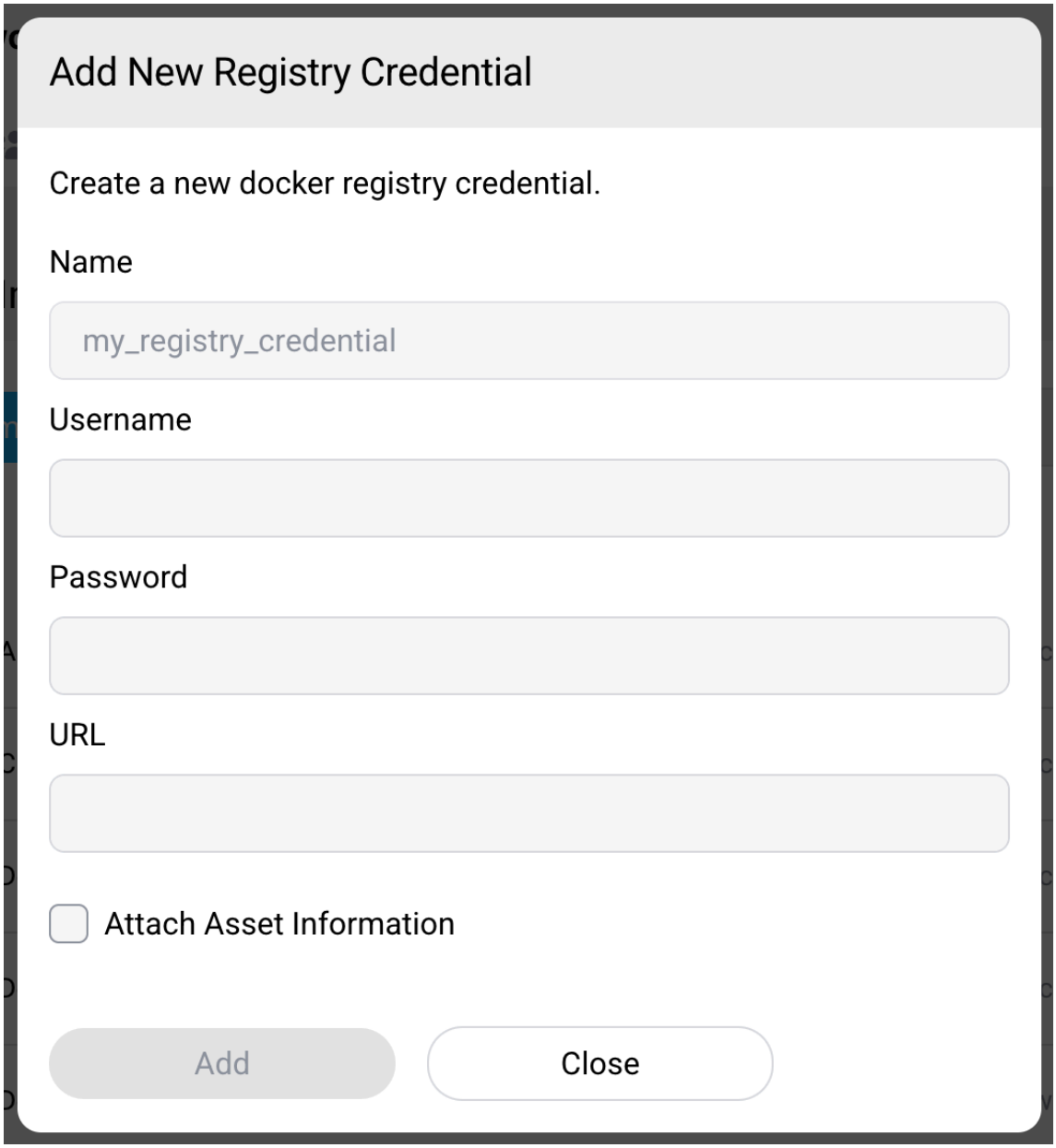
Create the image

1. Open **Resources > Container Images**.
2. Select the arrow next to **Add Image**, then select **Create Image**.

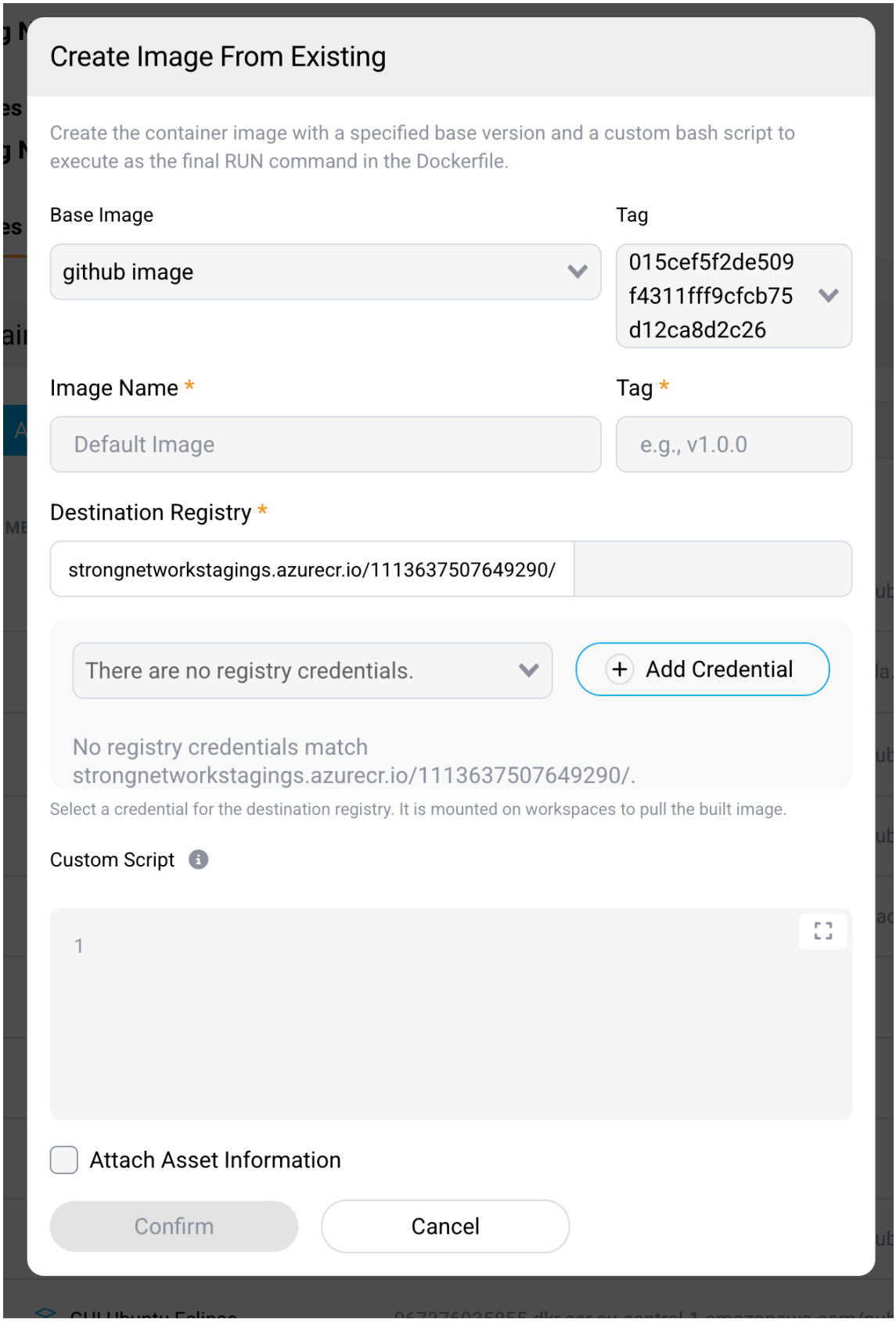


3. Choose the **Base Image** and the **Tag** to build on.
4. Enter the **Image Name** and **Tag** for the image you are creating, for example `v1.0.0`.
5. Check the **Destination Registry**. It is filled in from the platform setting.
6. Select a **registry credential** for that registry. If the list is empty, select **Add Credential** and provide a name, username, password, and the registry URL.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



© 1997–2026 Citrix Systems, Inc. All rights reserved.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

7. Enter your **Custom Script**. It runs as the last RUN command in the generated Dockerfile.
8. Select **Confirm**.

The build runs on the cluster. When it finishes, the image is listed alongside your other container images.

Writing the script

The script is a single RUN layer appended to the base image, so ordinary Dockerfile rules apply:

- It runs as part of the build, not at workspace start. Anything it installs is baked into the image.
- It runs non-interactively. Commands that prompt will hang until the build deadline expires.
- Only the filesystem changes persist. Environment variables set with `export` do not survive the layer.
- The base image must still satisfy the container image requirements after your script has run —in particular the `developer` user with UID 1000, and the SSH, Git, and Git LFS clients.

Keep the script to what is genuinely shared by the team. A workspace-specific step belongs in a startup script, which runs at start and can be changed without rebuilding.

Build limits

Builds run as a Kubernetes Job with fixed bounds. A build that exceeds them fails rather than degrading the cluster:

Limit	Default
CPU	500m requested, 2 cores maximum
Memory	1 GiB requested, 4 GiB maximum
Wall clock	30 minutes
Retries	None —a failed build is reported, never retried

A large compile or a script that downloads a great deal of data can exceed the memory limit or the deadline.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If that happens, move the heavy work into a base image built by your own pipeline, and use this feature for
the thin layer on top.

A platform administrator can raise these bounds with `imageBuilderSpecification` in the Helm values, which
accepts `cpuRequest`, `cpuLimit`, `memoryRequest`, `memoryLimit`, and `deadlineSeconds`. Limits are raised to
match requests if they are configured lower.

How the build is isolated

Your script is arbitrary code, and it runs on the cluster. SecurSpaces builds it in a sandbox rather than on
a Docker daemon:

- The build runs **rootless BuildKit** inside a user namespace, as a non-root user.
- The container is not privileged, drops all Linux capabilities, and the build process itself holds none.
- CPU and memory ceilings mean a runaway script is stopped rather than exhausting the node.
- The Job is never retried, so a hung script cannot restart indefinitely, and it is cleaned up after it finishes.

This is why the feature is gated on a platform-level setting rather than being available by default: enabling
it lets project members run build-time code on your cluster, within that sandbox.

Create an image from a workspace

The same build is available while editing a workspace, under **Startup Scripts**, as **create a custom image based
on the current workspace image, extended with selected startup scripts**. Use it to promote startup scripts
that have settled into the image, so they no longer run on every start.

That route additionally requires the workspace image to have a registry credential attached.

Related information

- Container images
- Workspace images

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Container image requirements
- Registry access

Manage workspace toolchains

August 18, 2026

A toolchain manager lets each project declare the language runtimes and tools it needs in a file in its own
repository, so one base container image serves many teams. Citrix SecurSpaces™ documents two:

- Mise
- Nix

Neither is built into the platform. Both are ordinary tools you install in a base image, and both work because
of the same platform behavior, described below.

Choose between Mise and Nix

	Mise	Nix
Configuration file	`mise.toml`	`flake.nix`
Typical first start	10–60 seconds	30–120 seconds
What persists by default	Tools install under `/home/developer`, which persists	The Nix store is at `/nix`, which does **not** persist unless you add a volume
Reproducibility	Pinned tool versions	Fully pinned dependency closures through `flake.lock`
Learning curve	Low. Familiar to anyone who has used `asdf` or `nvm`.	Higher. Requires learning the Nix language.

Choose **Mise** when you want per-project tool versions with minimal setup, and your teams are comfortable
pinning versions in a simple file.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Choose **Nix** when you need bit-for-bit reproducible environments and are prepared to configure a persistent
volume for `/nix` and to invest in the Nix language.

Platform behavior both rely on

Only `/home/developer` persists

The workspace container is recreated from its image on every start. `/home/developer` is a persistent volume
and survives; everything else does not. See
What persists in a SecurSpaces workspace.

This is what makes a toolchain manager practical: tools download once and remain on subsequent starts.

Files written during the image build are hidden

A persistent volume is empty when it is first created, and mounting it hides anything the image build wrote to
that path. Installing tools into `/home/developer` in a Dockerfile therefore has no effect at runtime.

Install to a system path such as `/usr/local`, and use a first-start script for anything that must live in the
home directory.

First-start scripts

Scripts in `/usr/bin/strong_network_startup/` in the image run once, on first start, after `/home/developer`
exists.

The platform records that they have run by creating `/home/developer/.strong_network_init`. Because that
marker lives on the persistent volume, the scripts run again for any workspace with a new volume, and again if
the marker is deleted.

These are distinct from the startup scripts configured in the console, which run on **every** start. See
Startup scripts.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Certificates when a network policy is attached

When a workspace has a network policy attached, its outbound traffic is routed through the SecurSpaces egress
proxy, and the proxy's certificate is appended to the system trust store at `/etc/ssl/certs/ca-certificates.crt`.

The platform also configures several ecosystems that keep their own trust store:

Ecosystem	Configured
npm	`cafile`
pip	`global.cert`
yarn	`strict-ssl` disabled
Node.js	`NODE_EXTRA_CA_CERTS`
Python `requests`	`REQUESTS_CA_BUNDLE`
Java	The certificate is imported into the JDK keystore
VS Code	`http.proxyStrictSSL` disabled

Tools that read the system trust store, including Mise and Nix, work without further configuration. A tool
that ships its own CA bundle and ignores the system store needs to be pointed at the proxy certificate explicitly.

See Network policies.

Sample scenario

Both worked examples use the same project, a web application requiring:

- Node.js 22, the application runtime
- Python 3.13, for build scripts and tooling
- Terraform 1.x, for infrastructure management
- jq, for JSON processing

Related information

- Manage workspace toolchains with Mise
- Manage workspace toolchains with Nix

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- What persists in a SecurSpaces workspace
- Workspace images
- Workspace templates

Manage workspace toolchains with Mise

Instead of building and maintaining separate container images for every team, you can use Mise-en-Place (Mise) to define project-specific toolchains on top of a single base image. Mise is a polyglot developer tool version manager that lets each team declare their required languages, runtimes, and CLI tools in a `mise.toml` configuration file that lives in the repository alongside the code.

Why use Mise?

Mise simplifies how your organization manages development toolchains in Citrix SecurSpaces™. It offers several benefits over maintaining multiple container images:

- **Fewer images to maintain.** Use one base image across your organization and let teams define their tooling needs per project.
- **Faster onboarding.** Developers clone a repository and the tools install automatically —no manual setup, no outdated documentation.
- **Version-controlled environments.** Toolchain definitions live in Git, so changes go through pull requests and code review just like any other code change.
- **Faster workspace restarts.** Because workspaces persist the `/home/developer` directory, installed tools survive between sessions. After the first start, subsequent starts activate in under a second.
- **Self-service for developers.** Teams update their own toolchain without requesting a new container image from the platform team.
- **Consistency across the team.** Every developer on a project gets the same tool versions, reducing "works on my machine"issues.
- **No root required for runtime changes.** Developers install or upgrade language runtimes into their own home directory, so common friction points like `npm install -g` permission errors or `pip` polluting the system Python disappear. The base image stays immutable.
- **Less image sprawl.** Instead of maintaining a separate image for every combination of runtimes (Python 3.10 + Node 18, Python 3.12 + Node 20, and so on), a single polyglot image lets developers pick versions per project, which reduces the number of images you build, scan, and store.

How it works in SecurSpaces

The overall approach follows this pattern:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Your platform team creates a single base container image with Mise pre-installed.
2. Each team commits a `mise.toml` configuration file to their repository.
3. When a developer starts a workspace, Mise reads the configuration and installs the declared tools.
4. Because `/home/developer` persists across workspace restarts, installed tools remain available on subsequent starts without re-downloading.

What happens on first start versus subsequent starts

First workspace start	Subsequent starts
Downloads and installs declared tools (typically 10–60 seconds depending on the number of tools)	Detects tools already present; activates instantly (under 1 second)

Sample scenario

The examples in this guide use a web application project that requires the following toolchain:

- Node.js 22 (application runtime)
- Python 3.13 (build scripts and tooling)
- Terraform 1.x (infrastructure management)
- jq (JSON processing)

Configure a base container image

Start with the generic image from the SecurSpaces image repository. This image is based on Ubuntu 24.04 and includes common development utilities like Git, Docker, kubectl, and cloud CLIs.
Extend this image by installing Mise at build time. This ensures Mise is available immediately when a workspace starts, without requiring developers to install it themselves.
Create a Dockerfile that extends the generic image:

```
FROM strongnetwork/generic:latest

USER root

# Install Mise for all users
RUN curl https://mise.run | MISE_INSTALL_PATH=/usr/local/bin/mise sh

# Add a first-start script that configures Mise for the developer user.
# Scripts in /usr/bin/strong_network_startup/ run automatically on
    first
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
# workspace start, after /home/developer has been created.
COPY setup-mise.sh /usr/bin/strong_network_startup/setup-mise.sh
RUN chmod +x /usr/bin/strong_network_startup/setup-mise.sh

USER 1000
WORKDIR /home/developer
```

Create `setup-mise.sh` alongside your Dockerfile:

```
#!/bin/bash
# setup-mise.sh — runs once on first workspace start
# At this point /home/developer exists and is writable.

if ! grep -q 'mise activate' /home/developer/.bashrc 2>/dev/null; then
  echo 'eval "$(mise activate bash)"' >> /home/developer/.bashrc
fi
```

Why not write to `.bashrc` in the Dockerfile? The `/home/developer` directory doesn't exist during the initial startup phases of a new workspace. It's created later in the startup flow. Scripts placed in `/usr/bin/strong_network_startup/` run after `/home/developer` is available, so they can safely write to it.
Mise installs tools into `~/.local/share/mise/installs/` by default. Because this path is inside `/home/developer`, all installed tools persist across workspace restarts.

Harden the Mise installation for production

The `curl https://mise.run | sh` pattern above is convenient for getting started, but most production environments will want stronger supply-chain controls. Two changes turn the base image into a production-ready Mise install:

1. Replace the remote installer with a pinned download from your own artifact registry.
2. Enforce system-level Mise settings (registry mirrors, proxy, allowed plugins) in `/etc/mise/config.toml` so they apply to every developer.

Install Mise from an internal artifact registry

Avoid executing remote installer scripts at image build time. Instead, mirror the Mise release into your internal artifact registry (for example, JFrog Artifactory) and download it during the build with an authenticated, version-pinned request. Use BuildKit build secrets so the registry token never lands in an image layer.

Create a hardened install script alongside your Dockerfile:

```
#!/bin/bash
# install-mise.sh — downloads a pinned Mise release from JFrog
    Artifactory
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
set -euo pipefail

MISE_URL="https://artifactory.example.com/artifactory/generic-local/mise/v${ MISE_VERSION }
"
INSTALL_PATH="/usr/local/bin/mise"

# Resolve the architecture-specific asset name
case "$(uname -m)" in
  x86_64)  MISE_ARCH="x64" ;;
  aarch64) MISE_ARCH="arm64" ;;
  *) echo "Unsupported architecture: $(uname -m)"; exit 1 ;;
esac
FILENAME="mise-v${ MISE_VERSION }
-linux-${ MISE_ARCH }
.tar.gz"

# Download and verify
curl -fsSL -H "Authorization: Bearer ${ JF_TOKEN }
" \
  "${ MISE_URL }
/${ FILENAME }
" -o /tmp/mise.tar.gz
curl -fsSL -H "Authorization: Bearer ${ JF_TOKEN }
" \
  "${ MISE_URL }
/${ FILENAME }
.sha256" -o /tmp/mise.tar.gz.sha256
(cd /tmp && sha256sum -c mise.tar.gz.sha256)

tar -xzf /tmp/mise.tar.gz -C /usr/local/bin --strip-components=1 bin/mise
chmod +x "${ INSTALL_PATH }
"
rm /tmp/mise.tar.gz /tmp/mise.tar.gz.sha256
```

Then call the script from the Dockerfile, passing the registry token as a build secret:

```
ARG MISE_VERSION=2026.2.20

USER root

COPY install-mise.sh /tmp/install-mise.sh
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
COPY mise-config.toml /etc/mise/config.toml

RUN --mount=type=secret,id=jfrog_token \
    JF_TOKEN="$(cat /run/secrets/jfrog_token)" \
    MISE_VERSION="${
 MISE_VERSION }
 " \
    bash /tmp/install-mise.sh && \
    rm /tmp/install-mise.sh

RUN echo 'eval "$(mise activate bash)"' >> /etc/bash.bashrc
```

Build the image with the secret bound from your CI environment, for example:

```
DOCKER_BUILDKIT=1 docker build \
  --secret id=jfrog_token,env=JFROG_TOKEN \
  -t your-registry/sds-base:latest .
```

The token is available only to the `RUN` step that mounts it and is never written to an image layer.

Enforce system-level Mise configuration

Settings that should apply to every developer —registry mirrors, an internal proxy, an allowlist of plugins —belong in `/etc/mise/config.toml`. Because the file lives in the container layer, it cannot be overridden by changes inside `/home/developer`, and it is reapplied on every workspace start.

A minimal configuration pointing Mise at internal mirrors:

```
[settings]
# Rewrite outbound download URLs to your internal mirror
url_replacements = {
  "github.com" = "artifactory.example.com/github-mirror"  }


[settings.node]
mirror_url = "https://artifactory.example.com/artifactory/nodejs-dist"

[settings.python]
# Resolve Python builds from your internal mirror of python-build-
    standalone
mirror_url = "https://artifactory.example.com/artifactory/python-build-
    standalone"
```

Developers can still add their own `mise.toml` per project, but they cannot bypass the system-level mirror, proxy, or plugin restrictions defined here. For the full list of available settings, see the Mise settings reference.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable Mise on existing workspaces

The container image approach above works for new workspaces created from the updated image. For existing workspaces that are already running, use a workspace startup script instead.
Configure a startup script in the SecurSpaces console (under workspace properties or as part of the workspace template). The script checks whether Mise is already set up and only runs the installation if needed:

```
#!/bin/bash
# Workspace startup script — configure in the SecurSpaces console or
    workspace template

# Install Mise if not already present
if ! command -v mise &> /dev/null; then
  curl https://mise.run | MISE_INSTALL_PATH=/home/developer/.local/bin/
      mise sh
fi

# Activate Mise in the shell if not already configured
if ! grep -q 'mise activate' /home/developer/.bashrc 2>/dev/null; then
  echo 'eval "$(mise activate bash)"' >> /home/developer/.bashrc
fi

# Source Mise for the current script session
eval "$(mise activate bash)"

# Install project tools if a mise.toml exists
if [ -f /home/developer/project/mise.toml ]; then
  cd /home/developer/project
  mise install
fi
```

Because this script checks for existing installations before acting, it's safe to run on every workspace start. On an already-configured workspace, it completes in under a second.

Define your project toolchain

Once your base image includes Mise, each team defines their toolchain in a `mise.toml` file at the root of their repository:

```
[tools]
node = "22"
python = "3.13"
terraform = "1"
jq = "latest"
```

When a developer opens a workspace and navigates to the project directory, Mise automatically installs the specified tool versions and adds them to the shell's `PATH`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Install tools on workspace start

If you want tools pre-installed before the developer opens a terminal, add `mise install` to your workspace startup script (see Enable Mise on existing workspaces for the full script pattern). The essential command is:

```
cd /home/developer/project
mise install
```

On first start, this downloads and installs all declared tools. On subsequent starts, Mise detects the tools are already present and completes in under a second.

Use environment variables and tasks

Mise also manages environment variables and project tasks. Add them to the same `mise.toml`:

```
[tools]
node = "22"
python = "3.13"
terraform = "1"
jq = "latest"

[env]
NODE_ENV = "development"
AWS_REGION = "eu-west-1"

[tasks]
dev = "npm run dev"
test = "npm test"
lint = "npm run lint && terraform validate"
```

Developers can then run `mise run dev` or `mise run test` and the correct tools and environment variables load automatically.
For full details on Mise configuration, see the Mise documentation.

Pre-install tools at image build time

The default Mise workflow installs tools into `/home/developer/.local/share/mise/installs/` the first time a workspace starts. That keeps the base image small, but it means developers wait for the download on first start and the workspace needs network access to a registry (or an internal mirror) at that moment.

If you want a specific runtime to be available immediately on first start —for example, because your team always uses Node 22, or because your workspaces have restricted egress —bake the tool into the image at build time instead.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Install tools globally into a shared Mise data directory

Set `MISE_DATA_DIR` to a system path (such as `/usr/share/mise`) during the build and symlink the resulting binary into `/usr/bin`. This makes the runtime available to every developer immediately, without consuming space on the persistent volume:

```
USER root

# Install Node 22 globally into /usr/share/mise and expose it on PATH
RUN MISE_DATA_DIR=/usr/share/mise mise use -g node@22 && \
    ln -s "$(MISE_DATA_DIR=/usr/share/mise mise which node)" /usr/bin/
       node
```

The tool installed this way lives in the container layer, so it is available immediately on every workspace start with no download. Developers can still add their own per-project `mise.toml` to layer additional or different versions on top—Mise resolves project-scoped tools from `~/.local/share/mise/installs/` first and falls back to the shared directory.

This pattern is the right choice when:

- A specific runtime version is mandatory for the team and should not need to be redownloaded.
- Workspace egress is restricted and you cannot rely on `mise install` succeeding at runtime.
- You want predictable first-start performance regardless of registry latency.

Pre-install with a first-start script (lazy loading)

If you prefer to keep the base image small but still avoid asking developers to install tools manually, use a first-start script that runs `mise use --global` for the required runtime. The download happens once, on first start, and persists in `/home/developer` for subsequent starts.

Place the script in `/usr/bin/strong_network_startup/` in your image:

```
#!/bin/bash
# /usr/bin/strong_network_startup/install-node.sh — runs once, on
   first start
mise use --global node@22
```

```
COPY --chmod=755 install-node.sh /usr/bin/strong_network_startup/
   install-node.sh
```

This keeps the image small but still requires network access to a Mise registry (or your internal mirror) on first start.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Understand workspace persistence

Mise installs tools under `/home/developer`, which persists across restarts, so tools download once.

Path	Persists
`/home/developer/.local/share/mise/installs/`	Yes. Tool binaries.
`/home/developer/.local/share/mise/`	Yes. Tool metadata and shims.
`/home/developer/.cache/mise/`	Yes. Download cache.
`/usr/local/bin/mise`	Part of the image layer, always present
Packages installed with `apt-get` at runtime	No

The first start always downloads tools, because the persistent volume is empty and hides anything the image
build wrote to that path. To shorten it, host an internal binary mirror and pin versions rather than using
`latest`, which avoids a lookup for the newest release.

For the persistence model in full, see Manage workspace toolchains
and What persists in a SecurSpaces workspace.

Network and security considerations

Mise downloads binaries from external registries at runtime. If your SecurSpaces deployment enforces egress controls, allow access to the following domains:

- `mise.run` —installer
- `github.com` and `api.github.com` —tool downloads and version resolution
- `nodejs.org`, `python.org`, and similar upstream registries depending on the tools used

Alternatively, configure internal mirrors for these registries and point Mise to your internal URLs. See Harden the Mise installation for production for an example that mirrors Mise itself and enforces system-level mirror settings, and the Mise settings documentation for details on configuring custom registries.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note:**
>
> When a workspace has a network policy attached, its traffic is routed through the SecurSpaces proxy service
> and the proxy certificate is added to the system trust store. Both tools read that store, so no extra certificate configuration is needed. See
> Certificates when a network policy is attached.

Where to go next

- Mise documentation —full reference for tool management, tasks, and environment variables
- SecurSpaces workspace images repository —base and generic container images for SecurSpaces
- Workspace templates —configure startup scripts and resource settings in your workspace templates
- Manage workspace toolchains —choosing between Mise and Nix, and the platform behavior both rely on

Related information

- Container images
- Create an image from an existing one
- Manage toolchains with Nix
- Manage workspace toolchains
- Update an existing container image
- Workspace images

Manage workspace toolchains with Nix

Instead of building and maintaining separate container images for every team, you can use Nix Package Manager (Nix) to define project-specific toolchains on top of a single base image. Nix is a purely functional package manager that lets each team declare their required languages, runtimes, and system packages in a `flake.nix` configuration file that lives in the repository alongside the code.

Why use Nix?

Nix simplifies how your organization manages development toolchains in Citrix SecurSpaces™. It offers several benefits over maintaining multiple container images:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Fewer images to maintain.** Use one base image across your organization and let teams define their tooling needs per project.
- **Faster onboarding.** Developers clone a repository and the tools install automatically —no manual setup, no outdated documentation.
- **Version-controlled environments.** Toolchain definitions live in Git, so changes go through pull requests and code review just like any other code change.
- **Faster workspace restarts.** With a persistent Nix store, installed packages survive between sessions. After the first start, subsequent starts activate in under a second.
- **Self-service for developers.** Teams update their own toolchain without requesting a new container image from the platform team.
- **Consistency across the team.** Every developer on a project gets the same tool versions, reducing "works on my machine"issues.
- **Reproducible builds.** Nix guarantees that if a package builds on one machine, it builds identically on another —packages are isolated and identified by cryptographic hashes of their entire dependency graph.

How it works in SecurSpaces

The overall approach follows this pattern:

1. Your platform team creates a single base container image with Nix pre-installed.
2. Each team commits a `flake.nix` configuration file to their repository.
3. When a developer starts a workspace, Nix reads the configuration and downloads the declared packages.
4. With a persistent Nix store, installed packages remain available on subsequent starts without re-downloading.

What happens on first start versus subsequent starts

First workspace start	Subsequent starts
Downloads package closures from the binary cache (typically 30–120 seconds depending on the number of packages)	Packages already in store; activates instantly (under 1 second)

> **Important:**
>
> This assumes you've configured persistence for the Nix store. Without it, Nix re-downloads packages on every restart. See Configure persistence for Nix for setup options.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Sample scenario

The examples in this guide use a web application project that requires the following toolchain:

- Node.js 22 (application runtime)
- Python 3.13 (build scripts and tooling)
- Terraform 1.x (infrastructure management)
- jq (JSON processing)

Configure a base container image

Start with the generic image from the SecurSpaces image repository. This image is based on Ubuntu 24.04 and includes common development utilities like Git, Docker, kubectl, and cloud CLIs.

Extend this image by installing Nix at build time. This ensures Nix is available immediately when a workspace starts, without requiring developers to install it themselves.

Create a Dockerfile that extends the generic image:

```
FROM strongnetwork/generic:latest

USER root

# Install Nix in single-user mode (installs to /nix)
RUN curl -L https://nixos.org/nix/install | sh -s -- --no-daemon

# Add a first-start script that configures Nix for the developer user.
# Scripts in /usr/bin/strong_network_startup/ run automatically on first
# workspace start, after /home/developer has been created.
COPY setup-nix.sh /usr/bin/strong_network_startup/setup-nix.sh
RUN chmod +x /usr/bin/strong_network_startup/setup-nix.sh

USER 1000
WORKDIR /home/developer
```

Create `setup-nix.sh` alongside your Dockerfile:

```
#!/bin/bash
# setup-nix.sh — runs once on first workspace start
# At this point /home/developer exists and is writable.

if ! grep -q 'nix-profile' /home/developer/.bashrc 2>/dev/null; then
  echo '. /nix/var/nix/profiles/default/etc/profile.d/nix.sh' >> /home/developer/.bashrc
fi
```

Why not write to `.bashrc` in the Dockerfile? The `/home/developer` directory doesn't exist during the initial startup phases of a new workspace. It's created later in the startup flow. Scripts placed in `/`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

`usr/bin/strong_network_startup/` run after `/home/developer` is available, so they can safely write to it.

Enable Nix on existing workspaces

The container image approach above works for new workspaces created from the updated image. For existing workspaces that are already running, use a workspace startup script instead.
Configure a startup script in the SecurSpaces console (under workspace properties or as part of the workspace template). The script checks whether Nix is already set up and only runs the installation if needed:

```
#!/bin/bash
# Workspace startup script — configure in the SecurSpaces console or
    workspace template

# Install Nix if not already present
if [ ! -d /nix ]; then
  curl -L https://nixos.org/nix/install | sh -s -- --no-daemon
fi

# Activate Nix in the shell if not already configured
if ! grep -q 'nix-profile' /home/developer/.bashrc 2>/dev/null; then
  echo '. /nix/var/nix/profiles/default/etc/profile.d/nix.sh' >> /home/
      developer/.bashrc
fi

# Source Nix for the current script session
. /nix/var/nix/profiles/default/etc/profile.d/nix.sh

# Build the development environment if a flake.nix exists
if [ -f /home/developer/project/flake.nix ]; then
  cd /home/developer/project
  nix develop --command true
fi
```

Because this script checks for existing installations before acting, it's safe to run on every workspace start. On an already-configured workspace, it completes in under a second.

Define your project toolchain

Once your base image includes Nix, each team defines their environment in a `flake.nix` file at the root of their repository:

```
{

  description = "Web application development environment";

```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  inputs = {

    nixpkgs.url = "github:NixOS/nixpkgs/nixos-unstable";
   }
;

 outputs = {
 self, nixpkgs  }
:
   let
     system = "x86_64-linux";
     pkgs = nixpkgs.legacyPackages.${
system }
;
   in
   {

     devShells.${
system }
.default = pkgs.mkShell {

       packages = [
         pkgs.nodejs_22
         pkgs.python313
         pkgs.terraform
         pkgs.jq
       ];

       shellHook = ''
         export NODE_ENV="development"
         export AWS_REGION="eu-west-1"
       '';
      }
;
    }
;
}
```

Developers enter the environment by running `nix develop` from the project directory. This activates the declared packages and sets the specified environment variables.

Install packages on workspace start

If you want packages pre-downloaded before the developer opens a terminal, add `nix develop --command true` to your workspace startup script (see Enable Nix on existing workspaces for the full script pattern). The essential command is:

```
cd /home/developer/project
nix develop --command true
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This builds and caches the development environment without entering a new shell. Developers then run `nix develop` interactively when they're ready to work.

Alternatively, use direnv with the nix-direnv extension to activate the environment automatically when entering the project directory.

Configure persistence for Nix

By default, Nix installs packages into `/nix/store/`, which is outside the persistent `/home/developer` directory. Without additional configuration, Nix re-downloads all packages on every workspace restart.

You have three options to make Nix packages persist:

1. **Add a persistent volume at `/nix` (recommended).** In your workspace template, configure an additional persistent volume mounted at `/nix`. This is the simplest approach and works with standard Nix substituters like `cache.nixos.org`.
2. **Use a local store under the home directory.** Configure Nix to use a store path within the persistent home directory. This requires user namespace support in the container:

   ```
   1 nix develop --store /home/developer/.nix-store
   ```

   > **Note:**
   >
   > A relocated store can't use the default binary cache without additional configuration. See the Nix manual on local stores for details.

3. **Use nix-portable.** A single binary that operates entirely within `$HOME` without requiring a system-level `/nix` mount. See the nix-portable repository for setup instructions.

Understand workspace persistence

Nix differs from most toolchain managers here: its store lives at `/nix`, outside the persistent home directory. Without a persistent volume at `/nix`, packages re-download on every restart. See Configure persistence for Nix.

Path	Persists
`/home/developer/.bashrc`	Yes. Shell activation.
`/home/developer/.nix-profile/`	Yes. Symlinks to the active packages.
`/nix/store/`	Only with a persistent volume at `/nix`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Path	Persists
`/nix/var/`	Only with a persistent volume at `/nix`
Packages installed with `apt-get` at runtime	No

The first start always downloads packages, because both volumes are empty on creation and hide anything the
image build wrote to those paths. To shorten it, run an internal
Nix binary cache as a
substituter, and lock `flake.lock` to a specific nixpkgs commit so an upstream channel update does not trigger
unexpected downloads.

For the persistence model in full, see Manage workspace toolchains
and What persists in a SecurSpaces workspace.

Network and security considerations

Nix downloads binaries from external registries at runtime. If your SecurSpaces deployment enforces egress controls, allow access to the following domains:

- `cache.nixos.org` —official binary cache
- `github.com` —flake inputs and source downloads
- `channels.nixos.org` —channel metadata

Alternatively, configure an internal binary cache and point Nix to your internal URLs. See the Nix configuration reference for details on configuring custom substituters.

> **Note:**
>
> When a workspace has a network policy attached, its traffic is routed through the SecurSpaces proxy service
> and the proxy certificate is added to the system trust store. Both tools read that store, so no extra certificate configuration is needed. See
> Certificates when a network policy is attached.

Where to go next

- Nix documentation —guides and reference for Nix flakes, packages, and development shells
- Nixpkgs search —search the 120,000+ available Nix packages

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- SecurSpaces workspace images repository —base and generic container images for SecurSpaces
- Workspace templates —configure startup scripts and resource settings in your workspace templates
- Manage workspace toolchains —choosing between Mise and Nix, and the platform behavior both rely on

Related information

- Container images
- Create an image from an existing one
- Manage toolchains with Mise
- Manage workspace toolchains
- Update an existing container image
- Workspace images

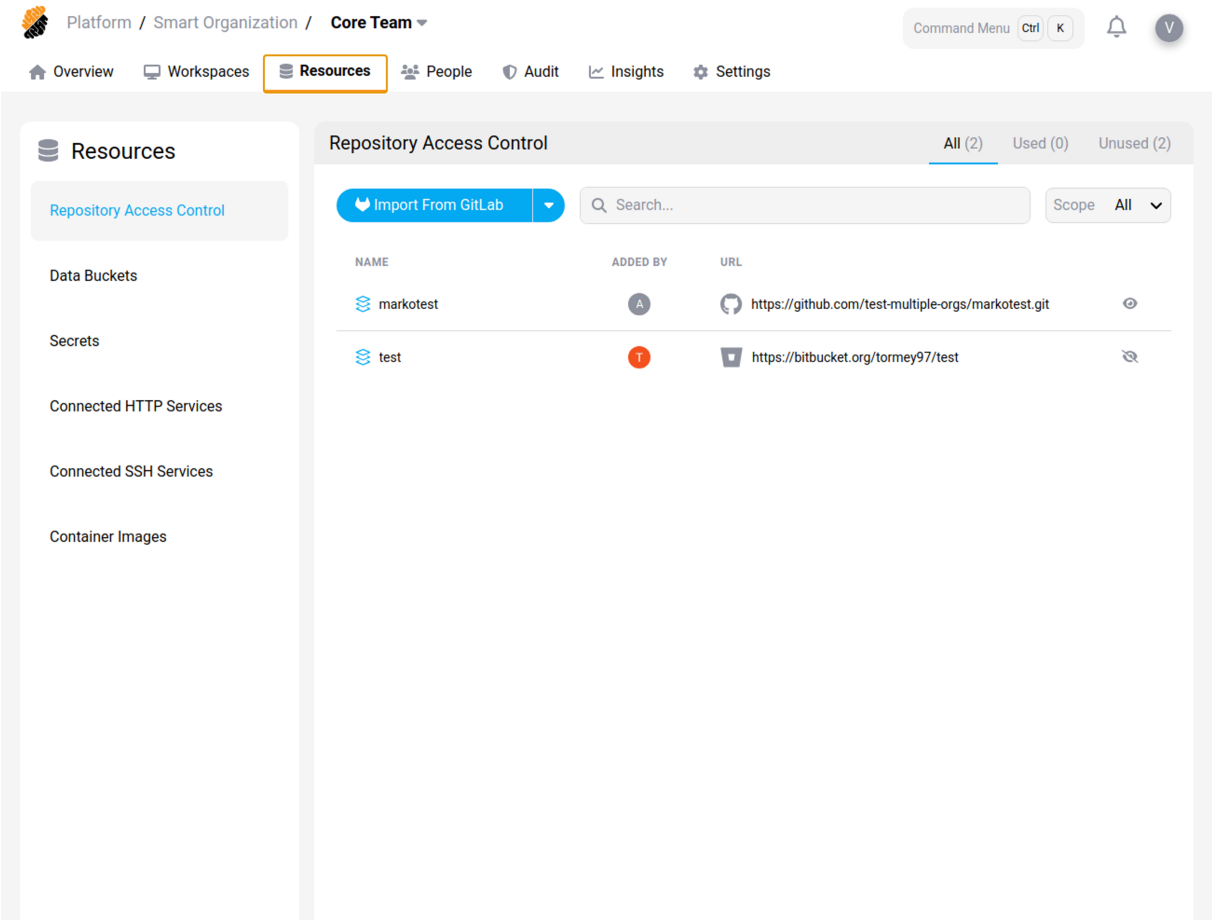
Project resources

On the resources page, you can view and manage the different resources used in the project.

Resources are used to define workspace properties such as container configuration and network policies, or the information available to users for development
such as code repositories, data buckets, secrets and services. Resources are managed at three levels of granularity depending on the intended scope of use: platform, organization and project.

Resources are attached to a workspace during the setup and update process. When resources are accessible to users, this process is a means
to define a fine-grain access control policy on an individual workspace basis.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Content

- Repository access control
- Data buckets
- Secrets
- Connected HTTP services
- Connected SSH services
- Container images

Code repositories

August 19, 2026

Importing a repository into a project is what lets a developer clone and push without holding any credentials

© 1997–2026 Citrix Systems, Inc. All rights reserved.

of their own. Citrix SecurSpaces™ injects the credentials at runtime, through the proxy, so the developer's
device never needs Git access to your source.

Repositories are Git based. They are imported from a provider your platform administrator has already
connected, or added manually by URL.

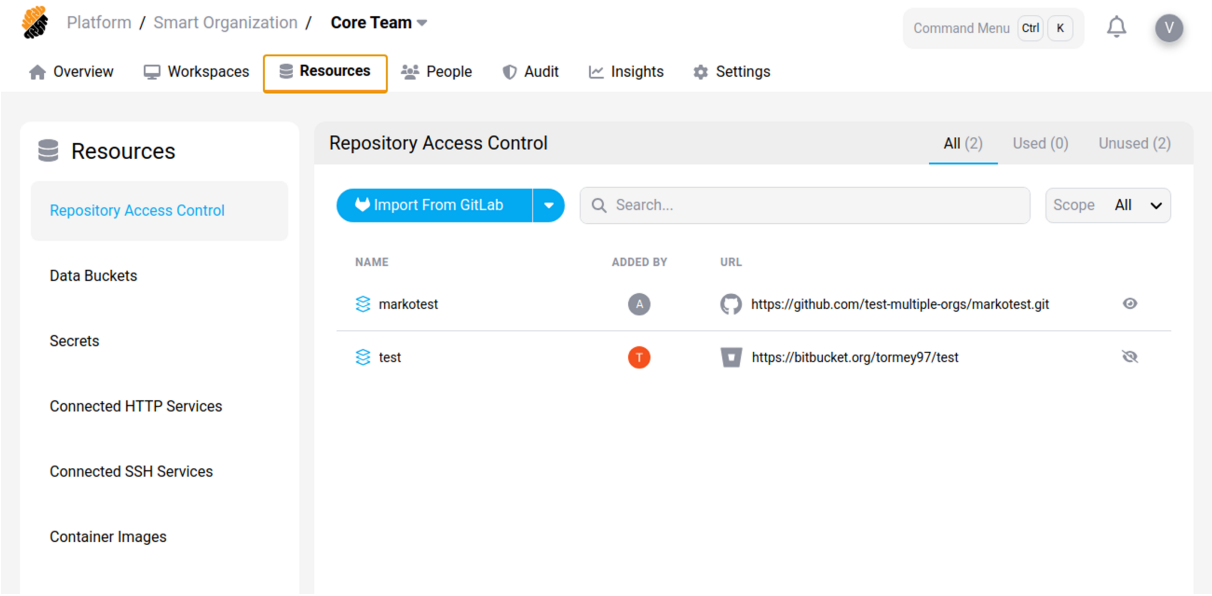
Provider	Supported
GitHub	Managed and self-hosted
GitLab	Managed and self-hosted
Bitbucket	Cloud and Server
Azure DevOps	Managed
Any other Git host	By URL, entered manually

If the provider you need is not offered when you import, it has not been connected at the platform level yet.
See Code repositories
for what that involves, and ask your platform administrator.

View repositories

Repositories imported into the project are listed in a table. You can search, or filter by the workspaces using them.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



A repository record holds:

- **Basic information** —name, scope (platform, organization, or project), who added it, the Git provider,
  and the URL.
- **Class level** —the visibility of the repository, evaluated against each user's permissions.
- **Asset information** —a description.

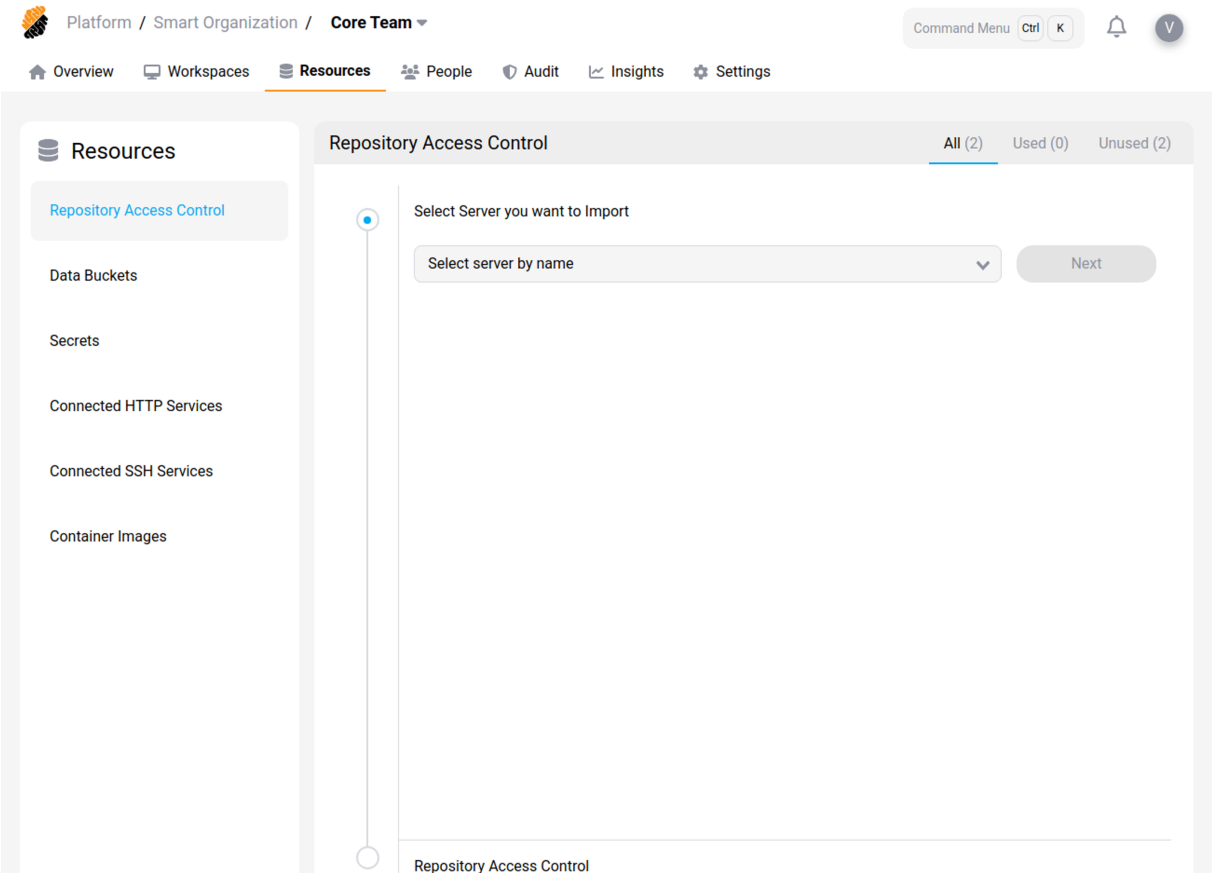
Scope matters here. A repository imported at platform or organization level is available to the projects beneath it, so check what you already have before importing your own copy.

Import a repository

Requires the Resources permission set to
Import.

1. Select **Import Repository**.
2. Select the provider.
3. The connected Git application is scanned for repositories. Select the one you want to import.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Importing records the repository with the project. It does not copy any code: the repository stays where it
is, and workspaces reach it through the platform.

Related information

- Project resources
- Code repositories (platform setup)
- Set up a project for your team
- Roles and permissions

Secrets

Secret management allows developers to securely store sensitive data such as passwords, keys, and tokens, in a protected environment with access controls capabilities.

Generally, the term “secret”points to any necessary credentials (e.g. cryptographic keys, tokens and password) necessary to authenticate with a service during the development process. The storage of

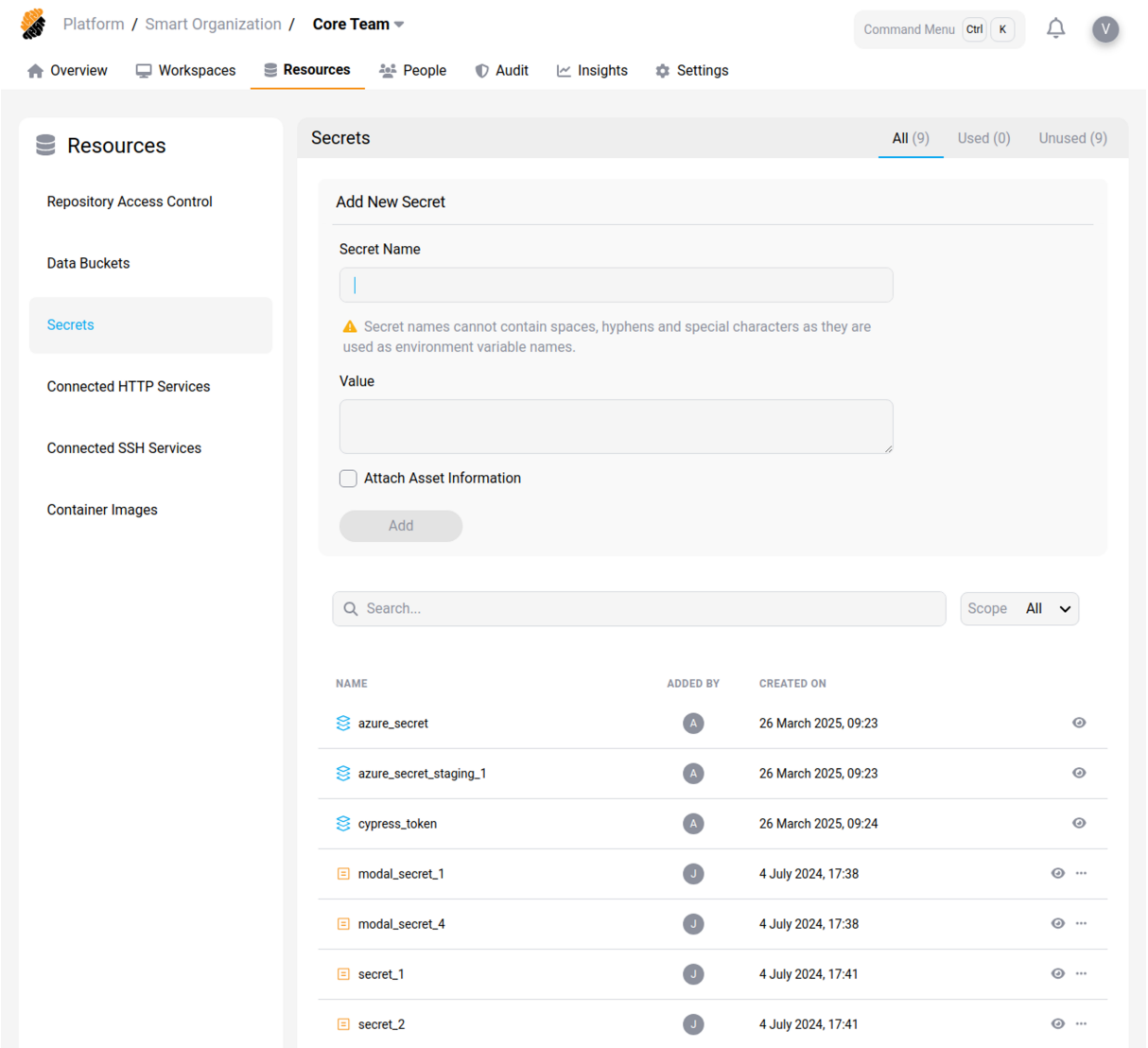
© 1997–2026 Citrix Systems, Inc. All rights reserved.

secrets is a service that can be provided by the platform or by an external mechanism. Once registered on the platform, secrets attached to workspaces are available in the container's filesystem as environment variables or files. This section explains how secrets are managed by the platform, but note that your platform might use an external service for that purpose.

- View Secrets
- Add a New Secret

View Secrets

Secrets used in the organization or project are displayed in a table. You may search for one or filter those used in workspaces.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

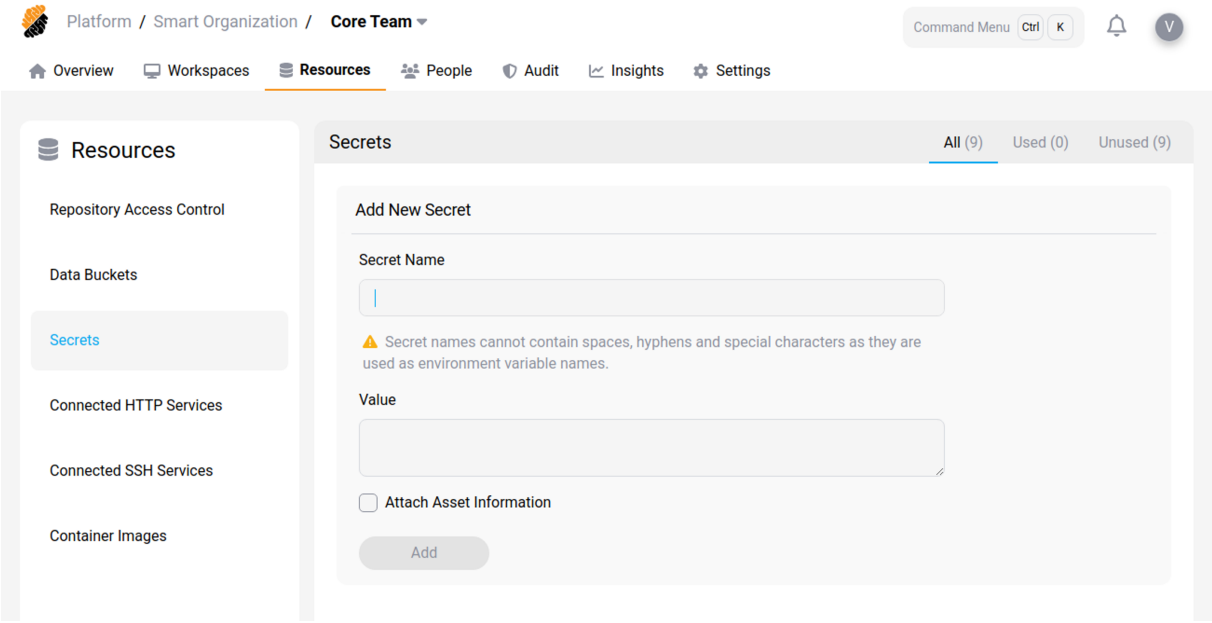
A Secret is defined by the following characteristics:

- **Basic information**: Information such as name, the user who added it, scope of use (platform, organization or project).
- **Class Level**: This option defines the visibility for the secret based on the user's permissions.
- **Asset Information**: This option allows for providing a description of the secret.

Add a New Secret

Requires the Resources permission set to **Manage**.

You can create a secret at the top of the **Secret Page**.



You will need to enter the following information:

1. **Name**, a name to identify the secret,
2. **Value**:, i.e. the secret's value, and an
3. **Asset information**, a description of the secret.

Related information

- Code repositories
- Connected HTTP services
- Connected SSH services
- Data buckets
- Choose between data buckets and mount points

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Project resources

Data buckets

August 13, 2026

A **Data Bucket** is used for general, unstructured storage of data online. This is basically a folder in S3 format that is commonly used to store and access
large datasets. Most cloud vendors offer S3 data buckets as a general storage data mechanism. The platform supports buckets from vendors such as Azure, Google
and Amazon Web Services. They are particularly popular for Data Science applications.

Data Buckets allow you to use your external datasets inside a workspace. A data bucket attached to a workspace is automatically mounted as a
folder to the container's filesystem.

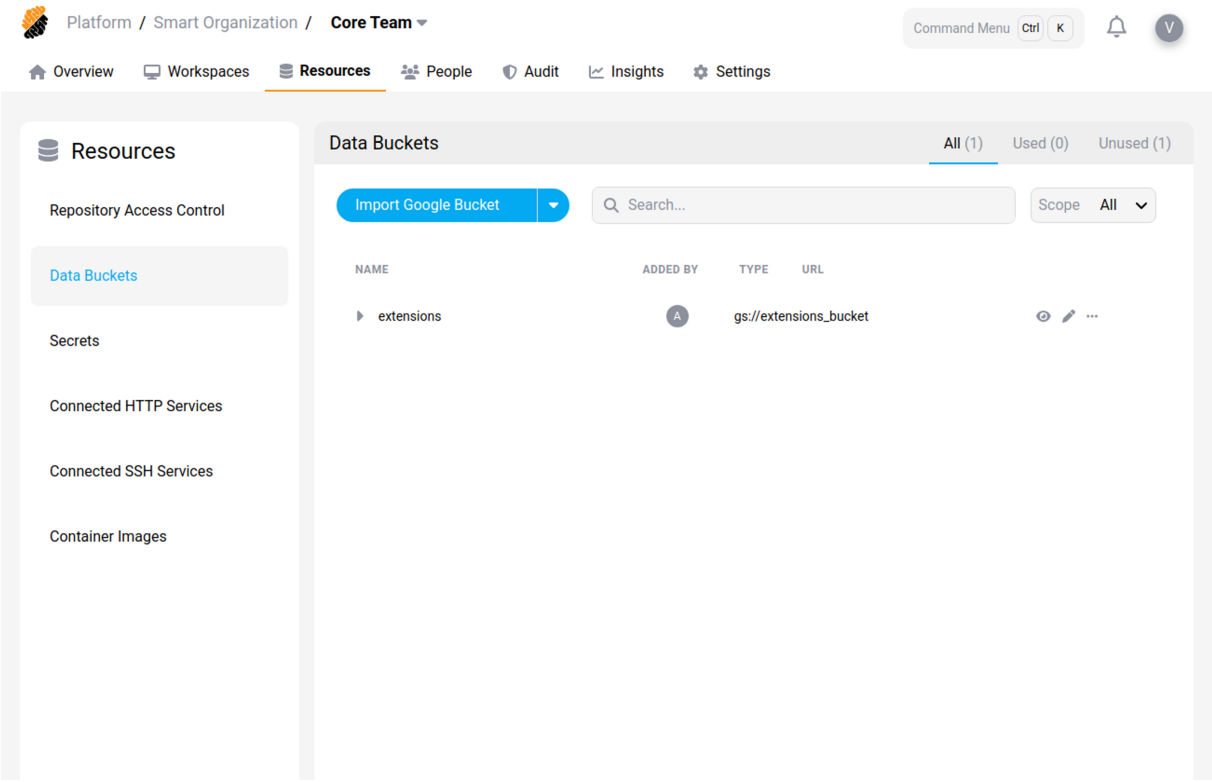
As for the other types of resources, data buckets are first imported to the platform such that they become available when creating or updating the configuration of a workspace.

- View Data Buckets
- Import a Data Bucket

View Data Buckets

Data Buckets used in the project are being displayed. You may filter those in use.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



A Data Bucket is defined by the following characteristics:

- **Basic information**: Information such as name, the user who added it, service provider (Google, Amazon or Microsoft) and URL.
- **Class Level**: This option defines the visibility for the container based on the user's permissions.
- **Permissions**: This option lets you define access to a data bucket as read or read and write.
- **Asset Information**: This option allows for providing a description of the data bucket.

The platform provides a mechanism to create versions of buckets. A new version is created when data is uploaded to a bucket from a workspace (with write access).
By clicking on a bucket you can see a list of versions followed by basic details (creation date, size, status, connections) as well as its content by clicking on the *book icon*.

To upload data from a workspace, the workspace must already be attached to the Data Bucket and must have
write access. If the workspace is not attached or has read-only access, choose another workspace or update the
workspace's resource access before uploading.

Import a Data Bucket

Requires the Resources permission set to **Manage**.

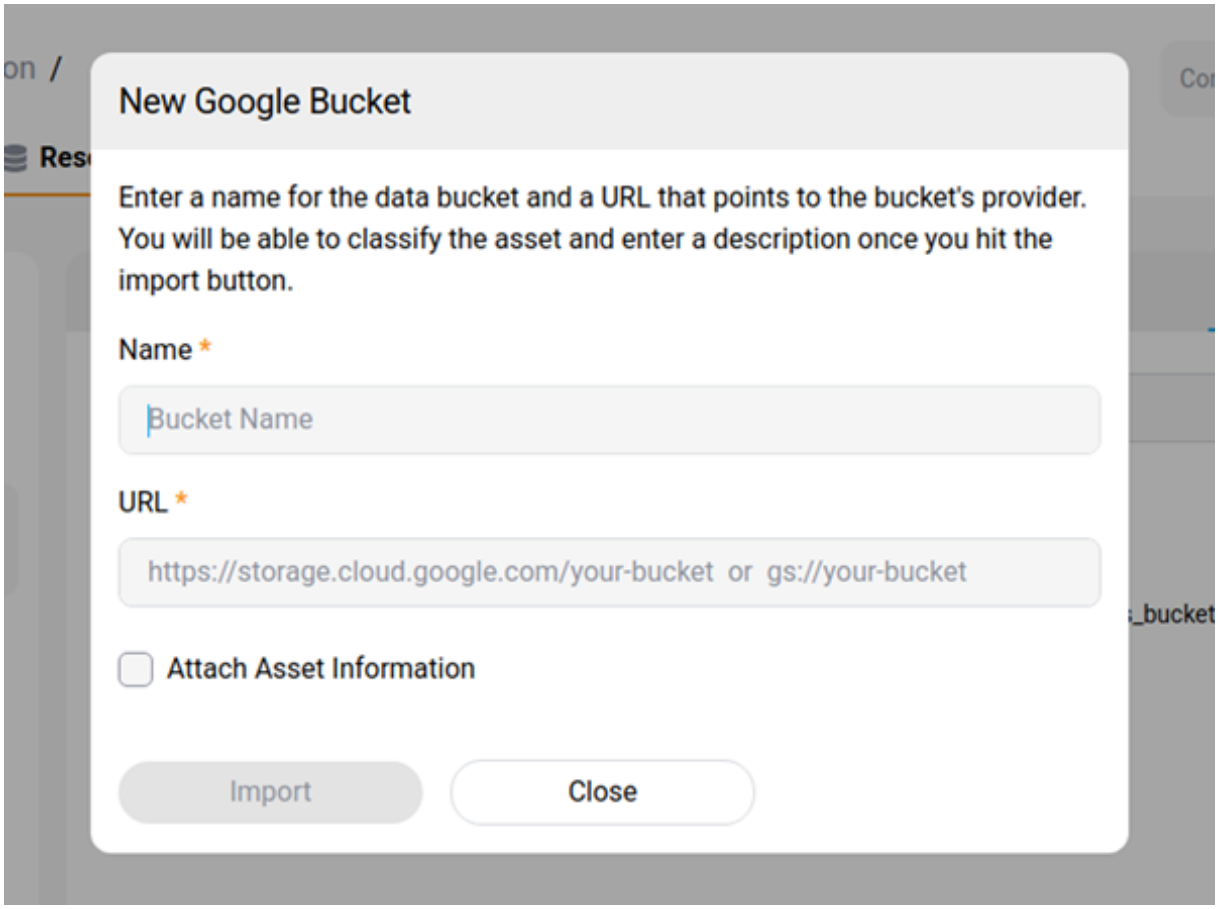
© 1997–2026 Citrix Systems, Inc. All rights reserved.

The import button names the provider it will use, so it reads **Import Google Bucket**, **Import Amazon Bucket**, or **Import Microsoft Bucket**. Select the provider that matches your bucket before importing.

The form opens as **New Google Bucket**, **New Amazon Bucket**, or **New Microsoft Bucket**, and asks for:

1. **Name**, a name to identify the data bucket.
2. **URL**, which points to the cloud provider's storage location.
3. **Attach Asset Information** (optional), a description of the bucket.

Select **Import** to save it.



> **Info**
>
> When importing Amazon buckets, you need to specify its region to optimize the data access performance.

Related information

- Code repositories

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Connected HTTP services
- Connected SSH services
- Choose between data buckets and mount points
- Mount points
- Project resources

Mount points

September 29, 2026

A **Mount Point** connects external file-based storage directly to a workspace file system. Unlike Data Buckets, which create point-in-time snapshots that are replicated to the cluster, Mount Points provide live access to the storage. Changes made inside the workspace are immediately visible to other workspaces sharing the same Mount Point, enabling real-time collaboration on shared files and datasets.

Mount Points are useful when teams need to:

- Share large datasets (for example, ML training data or model artifacts) without duplicating them across workspaces.
- Collaborate in real time on shared files without a manual snapshot-and-publish workflow.
- Access terabytes of data instantly, without waiting for replication.

> **Note**
>
> Mount Points are available only when a platform administrator has enabled the feature in **System Configuration > Integrations > Mount Point Storage**. See Enable Mount Points for details.

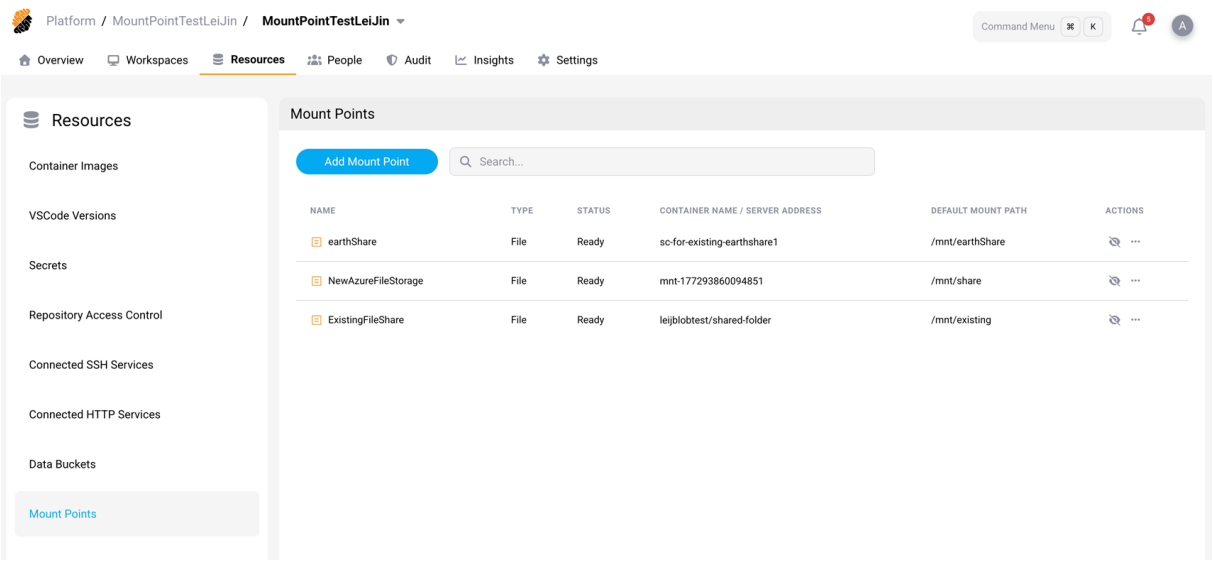
> **Note**
>
> Mount Points are not supported in multi-region configurations.

- View Mount Points
- Add a Mount Point
- Attach a Mount Point to a workspace

View Mount Points

Mount Points configured for the current project are displayed on the **Resources > Mount Points** page.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Each Mount Point displays the following information:

- **Name** —A descriptive label for the Mount Point.
- **Type** —The storage type. An Azure Mount Point reads **File** or **Blob**. An AWS Mount Point reads the
  backing service it uses, **Amazon EFS** or **Amazon S3 Files**.
- **Status** —The current state of the Mount Point. See
  Mount Point statuses.
- **Storage Identifier** —The storage behind the Mount Point. For Azure this is the storage account and file
  share, or the container or volume name. For AWS it is the name of the volume a platform administrator
  published, or the file system and access point identifiers.
- **Default Mount Path** —The file system path where the share is mounted inside the workspace (for example, `/mnt/shared-data`).
- **Actions** —View details or manage the Mount Point.

Mount Point statuses

Status	What it means
Creating	The Mount Point has been accepted and setup has started.
Provisioning	Storage is being provisioned and the platform is waiting for the volume to bind.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Status	What it means
Ready	The volume is bound. The Mount Point can be attached to a workspace.
Updating	An edit is being applied.
Deleting	The Mount Point is being removed.
Error	Provisioning or attaching failed, or a timeout expired. Delete the Mount Point and create it again.

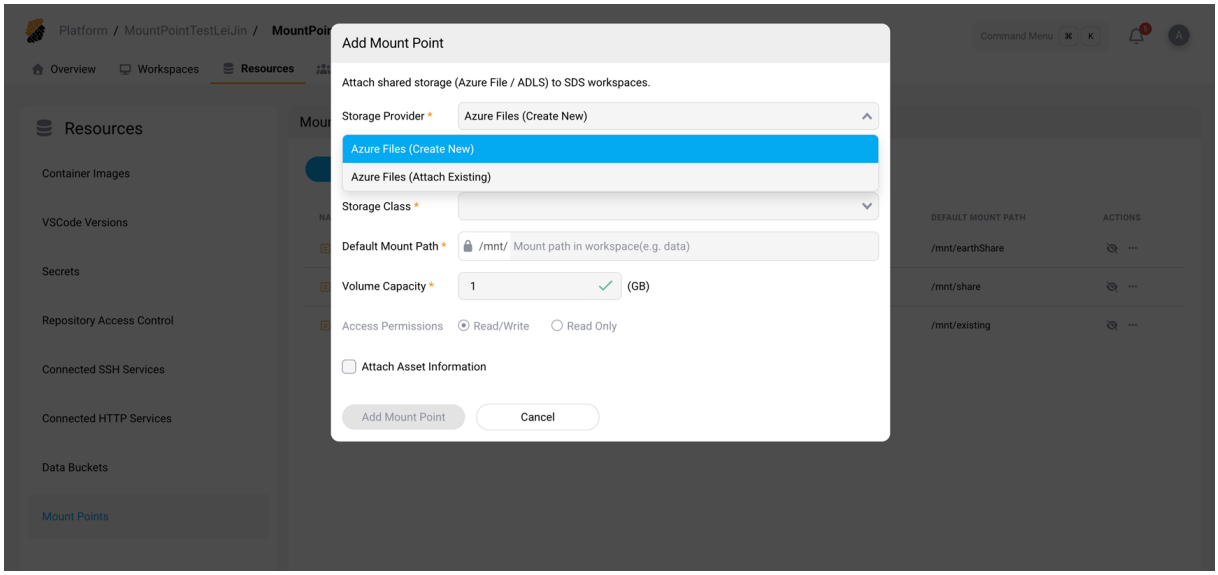
A Mount Point can be edited only while it is **Ready**. In every other status the **Edit** action is unavailable, and an edit submitted against it is refused with a message that the Mount Point is not ready.

> **Note**
>
> For an AWS Mount Point, **Ready** means the volume is bound. It does not mean the volume has been mounted. See
> Troubleshoot AWS Mount Points.

Add a Mount Point

Requires the Resources permission set to **Manage**.

To add a Mount Point, select **Add Mount Point** on the **Resources > Mount Points** page.



First, select a **Storage Provider** from the dropdown. The dropdown lists only the entries for the cloud your

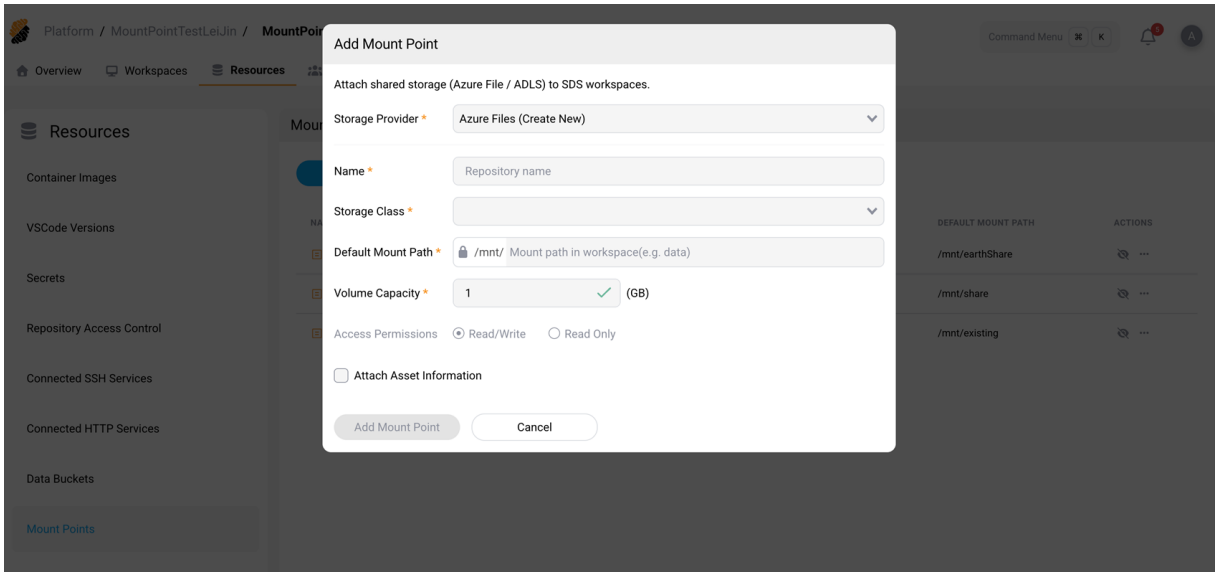
© 1997–2026 Citrix Systems, Inc. All rights reserved.

deployment runs in, so an AWS cluster offers the AWS entries and an Azure cluster offers the Azure entries.

- **Azure Files (Create New)** —Provisions a new Azure file share.
- **Azure Files (Attach Existing)** —Connects to an existing Azure file share.
- **AWS Files (Create New)** —Provisions new AWS-backed shared storage from an eligible storage class.
- **AWS Files (Attach Existing)** —Connects to existing Amazon Elastic File System (EFS) or Amazon S3 Files
  storage prepared by your infrastructure team.

Azure Files (Create New)

Provide the following information:

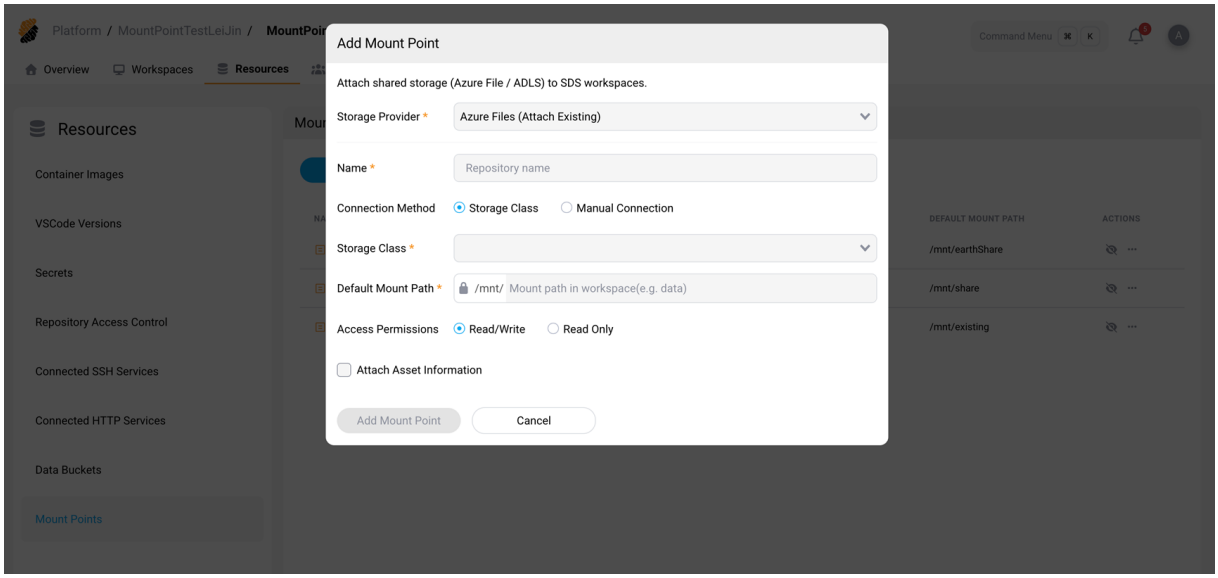


1. **Name** —A descriptive name to identify the Mount Point.
2. **Storage Class** —Select a storage class from the available options.
3. **Default Mount Path** —The path where the share is mounted inside the workspace file system. The path is prefixed with `/mnt/` (for example, entering `data` results in `/mnt/data`).
4. **Volume Capacity** —The size of the file share in GB.
5. **Access Permissions** —Fixed at **Read/Write** for a newly provisioned share. The control is shown but
   cannot be changed. Choose read-only access per workspace when you attach the Mount Point.
6. **Attach Asset Information** (optional) —Add metadata to classify the Mount Point.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Azure Files (Attach Existing)

When attaching an existing Azure file share, provide a **Name** and select a **Connection Method**. The two
methods are offered as a pair of options, and each one asks for a different set of fields.



- **Storage Class** —Select an eligible Azure file storage class. Use this method when a platform administrator has already prepared a storage class for the share.
- **Manual Connection** —Enter the storage details directly. All three fields are required:
  - **Storage Account** —The name of the Azure storage account that holds the share.
  - **Access Key** —The storage account access key.
  - **File Share Name** —The name of the file share to mount.

Then configure the **Default Mount Path**, and select **Read/Write** or **Read Only** under **Access Permissions**. The connection method and its fields are fixed once the Mount Point is created.

AWS Files (Create New)

When creating a new AWS-backed Mount Point, select a storage class that represents either Amazon EFS or
Amazon S3 Files. The storage class determines the backing service and must be configured by an administrator
before project owners can use it.

Provide the following information:

1. **Name** —A descriptive name to identify the Mount Point.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. **Storage Class** —Select an eligible Amazon EFS or Amazon S3 Files storage class.
3. **Default Mount Path** —The path where the storage is mounted inside the workspace file system. The path
   is prefixed with `/mnt/`.
4. **Access Permissions** —Fixed at **Read/Write** for a newly provisioned Mount Point. The control is shown
   but cannot be changed, and it cannot be changed later by editing the Mount Point. Choose read-only access
   per workspace when you attach the Mount Point, if users should not modify the shared data from that
   workspace.
5. **Attach Asset Information** (optional) —Add metadata to classify the Mount Point.

There is no size to set. Amazon EFS and Amazon S3 Files are elastic, so an AWS Mount Point has no provisioned
capacity and no per-Mount-Point size quota. See
Mount point limits and behavior.

> **Note**
>
> A new AWS Mount Point shows **Creating**, and then **Provisioning**, for up to three minutes while AWS
> creates the access point and the volume binds. That wait is normal and does not mean the Mount Point has
> failed. For what **Ready** does and does not guarantee, see
> Troubleshoot AWS Mount Points.

AWS Files (Attach Existing)

Use attach existing when an infrastructure team has already created the storage that the project should use.
There are two ways to identify that storage, and which of them you see depends on how the platform is
configured.

Select an existing volume. Every deployment offers this method, and unless a platform administrator has
enabled the second one, it is the only method available. The form shows a **Volume** list containing the volumes a platform administrator has published to your project, to your organization, or to the whole deployment. Provide the following information:

1. **Name** —A descriptive name to identify the Mount Point.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. **Volume** —Select a published volume by name.
3. **Default Mount Path** —The path where the storage is mounted inside the workspace file system.
4. **Access Permissions** —Select **Read/Write** or **Read Only**.

If the volume you expect is not listed, ask your platform administrator to check it. A volume is offered only
to the scope its label names, and a volume that a deleted Mount Point released must be reset before it can be
attached again. See
Author a PersistentVolume for Attach Existing.

Enter file system details. A platform administrator can additionally allow the platform to build the volume from AWS coordinates that you type. This method is turned off by default. Where it is enabled, an
Attach Method choice appears above the form, offering **Select an existing volume** and **Enter file system details**. Where it is not enabled, there is no choice to make and only the volume list is shown. Choosing **Enter file system details** asks for:

1. **Name** —A descriptive name to identify the Mount Point.
2. **Storage backend** —Select **Amazon EFS** or **Amazon S3 Files**.
3. **File System ID** —The AWS file system identifier, which starts with `fs-`.
4. **Access Point ID** —The access point identifier, which starts with `fsap-`. This field is required for
   both Amazon EFS and Amazon S3 Files.
5. **Subpath** (optional) —Mount a specific directory within the backing storage.
6. **Default Mount Path** —The path where the storage is mounted inside the workspace file system.
7. **Access Permissions** —Select **Read/Write** or **Read Only**.

Administrators and security officers see one further control on this form: a **Mount the entire file system
(no access point)** checkbox. Selecting it hides the **Access Point ID** field and mounts the root of the file
system, so a workspace sees every project's data on that file system and there is no per-project isolation. It
is refused on a file system that already backs Mount Points the platform created. In that case, clear the
checkbox and name a specific access point instead. Project owners do not see this checkbox.

> **Important**
>
> Amazon EFS and Amazon S3 Files are both mounted over NFS, and both need the `tls` mount option for that

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> traffic to be encrypted. A storage class or volume that does not carry it is not offered, and an attach to
> one is refused rather than mounted unencrypted. If a storage class, volume, or access point you expect is
> not available, ask your platform administrator to review the AWS Mount Point storage configuration.

AWS storage classes and existing file systems are prepared by your platform administrator outside of the
platform. If the AWS options are missing, or an existing file system you expect is not offered, see Prepare AWS storage for Mount Points.

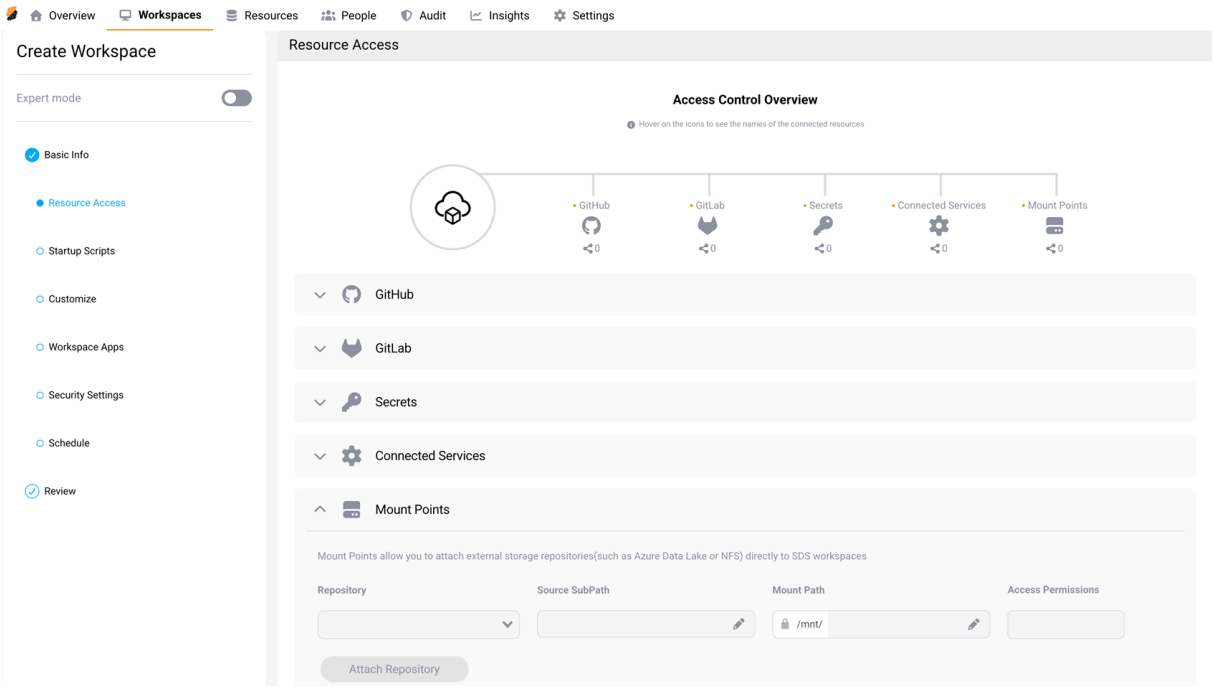
> **Note**
>
> Deleting a Mount Point is blocked while a live workspace, a workspace template, or a deleted workspace that
> can still be restored uses it. Detach it from each of those first.
>
> Editing is not affected by usage. A Mount Point can be edited whenever it is **Ready**, however many
> workspaces use it, and cannot be edited in any other status. An errored or still-provisioning Mount Point
> has to be deleted and created again rather than corrected.

Attach a Mount Point to a workspace

You can attach one or more Mount Points to a workspace during workspace creation or when creating a workspace template.

Mount Points appear in the **Resource Access** step alongside other resource types such as GitHub, GitLab, Secrets, and Connected Services.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



When attaching a Mount Point:

1. Expand the **Mount Points** section in the Resource Access step.
2. Select a Mount Point from the **Repository** dropdown.
3. Optionally, specify a **Source SubPath** to mount a specific subdirectory of the file share.
4. Review or customize the **Mount Path** —the location in the workspace file system where the share is accessible. The path is prefixed with `/mnt/`.
5. Select **Read/Write** or **Read Only** under **Access Permissions**. This is where you choose how the Mount Point behaves in this workspace. The dropdown starts on the Mount Point's own setting, which is
   Read/Write for every Mount Point created with **Create New**, and it is the only place to make an attach
   read-only.
6. Select **Attach Repository** to add the Mount Point to the workspace.

Attached Mount Points are listed under **Configured Repositories** below the attachment form.

> **Important**
>
> Each Mount Point in a workspace must use a unique mount path. If a path conflict is detected, the system displays an error and prevents the workspace from being created until the conflict is resolved.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mount Points can only be added in the primary region. If the workspace is configured for a non-primary region, the Mount Points section is disabled.

Mount Points in workspace templates

When you create or update a workspace template, you can pre-configure Mount Points in the same way as during workspace creation. Workspaces created from the template automatically inherit the configured Mount Points and their mount paths.

Developers creating a workspace from a template can still customize the mount path for each attached Mount Point before launching the workspace.

> **Note**
>
> The template editor is not covered by the primary-region restriction. It lets you attach a Mount Point
> whatever region the template targets, so a template can be saved carrying storage that work-spaces outside
> the primary region cannot use. Do not treat this as a supported way to use Mount Points in a secondary
> region. See Mount point limits and behavior.

Related information

- Code repositories
- Connected HTTP services
- Connected SSH services
- Data buckets
- Choose between data buckets and mount points
- Mount point limits and behavior
- Project resources

Connected HTTP services

Connected HTTP Services consist of services used for the implementation of software applications. These services are typically providing functions, data or host access via APIs over the HTTP network protocol.

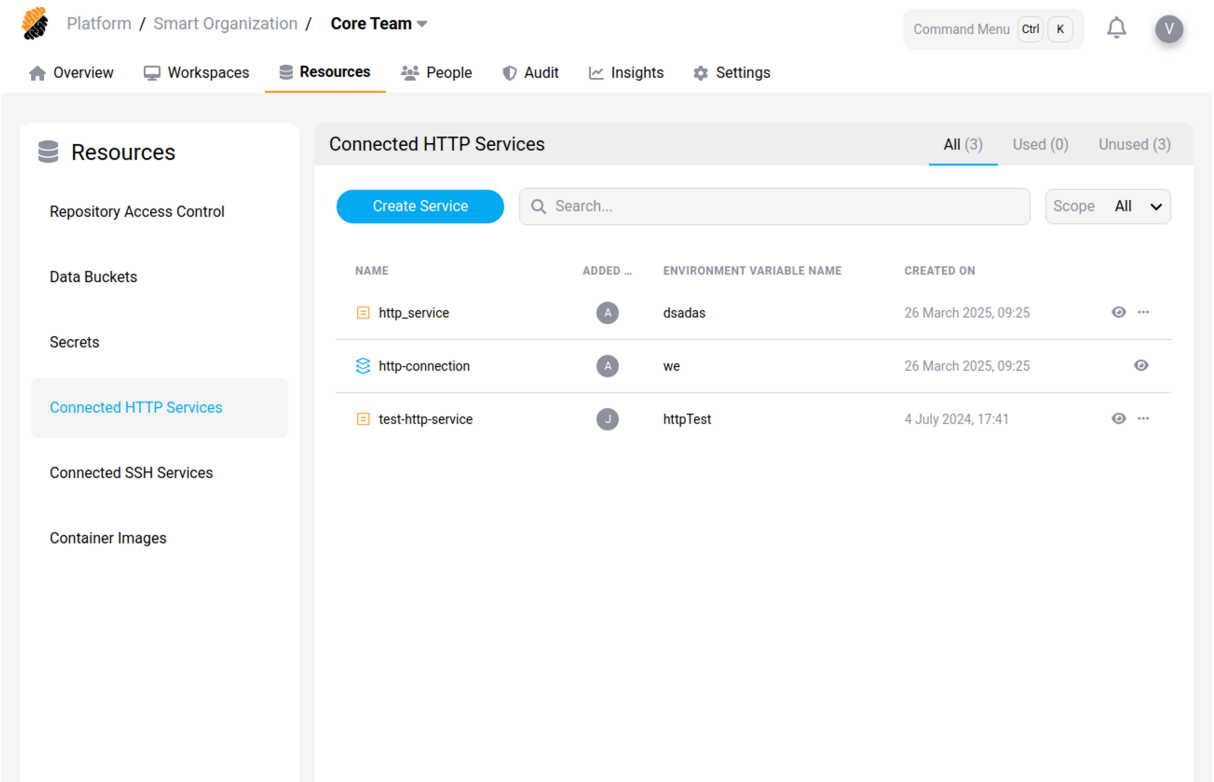
© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Tip**
>
> The nature and protocol of services that can be attached to workspaces depend on your platform's implementation.

As it is the case with other types of resources, HTTP services are attached to workspaces during the creation or the update of the workspace's settings.

- View Connected HTTP Services
- Add an HTTP Service

View Connected HTTP Services



Connected HTTP services are defined by the following characteristics:

- **Basic information**: Name, scope of use (platform, organization or project), URL and tag.
- **Class Level**: This option defines the visibility for the service based on the user's permissions.
- **Asset Information**: This option allows for providing a description of the container.
- **Environment Variable Name**: allows access to the service by naming an environment variable.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add an HTTP Service

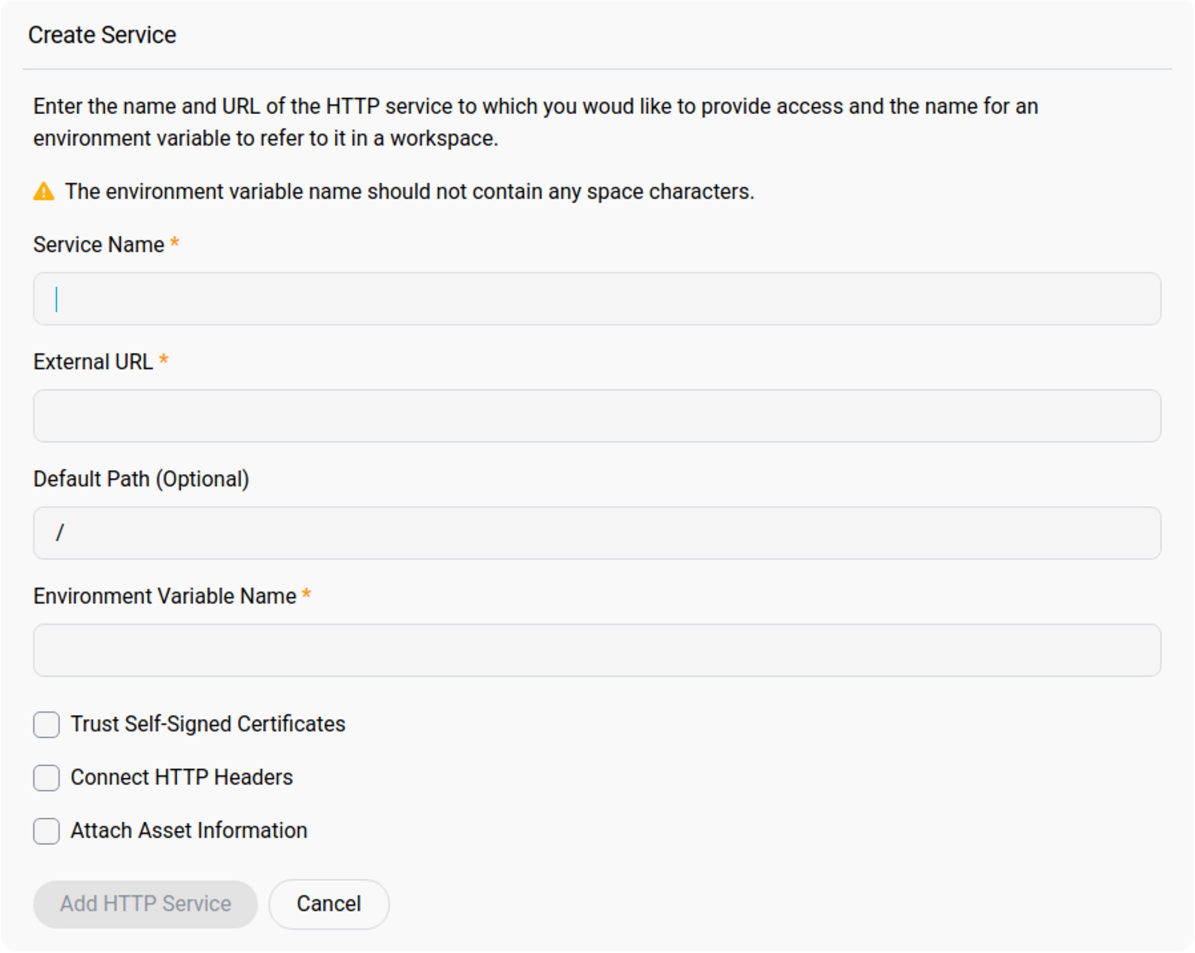
Requires the Resources permission set to **Manage**.

Select **Create Service**, then provide:

1. **Service Name**, a name to identify the service.
2. **External URL**, which points to the service location.
3. **Default Path (Optional)**, appended to the URL when the service is called.
4. **Environment Variable Name**, used to name the service inside the workspace environment. It must not
   contain spaces or special characters.
5. **Trust Self-Signed Certificates**, if the service presents a certificate that is not publicly trusted.
6. **Connect HTTP Headers**, used to pass authentication data when the service requires it. You can add
   basic access authentication or custom headers.
7. **Attach Asset Information**, used to provide a description of the service.

Select **Add HTTP Service** to save it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Related information

- Code repositories
- Connected SSH services
- Data buckets
- Choose between data buckets and mount points
- Mount points
- Project resources

Connected SSH services

Connected SSH Services consist of services used for the implementation of software applications. These services are typically providing functions, data or host access via APIs over the SSH network

© 1997–2026 Citrix Systems, Inc. All rights reserved.

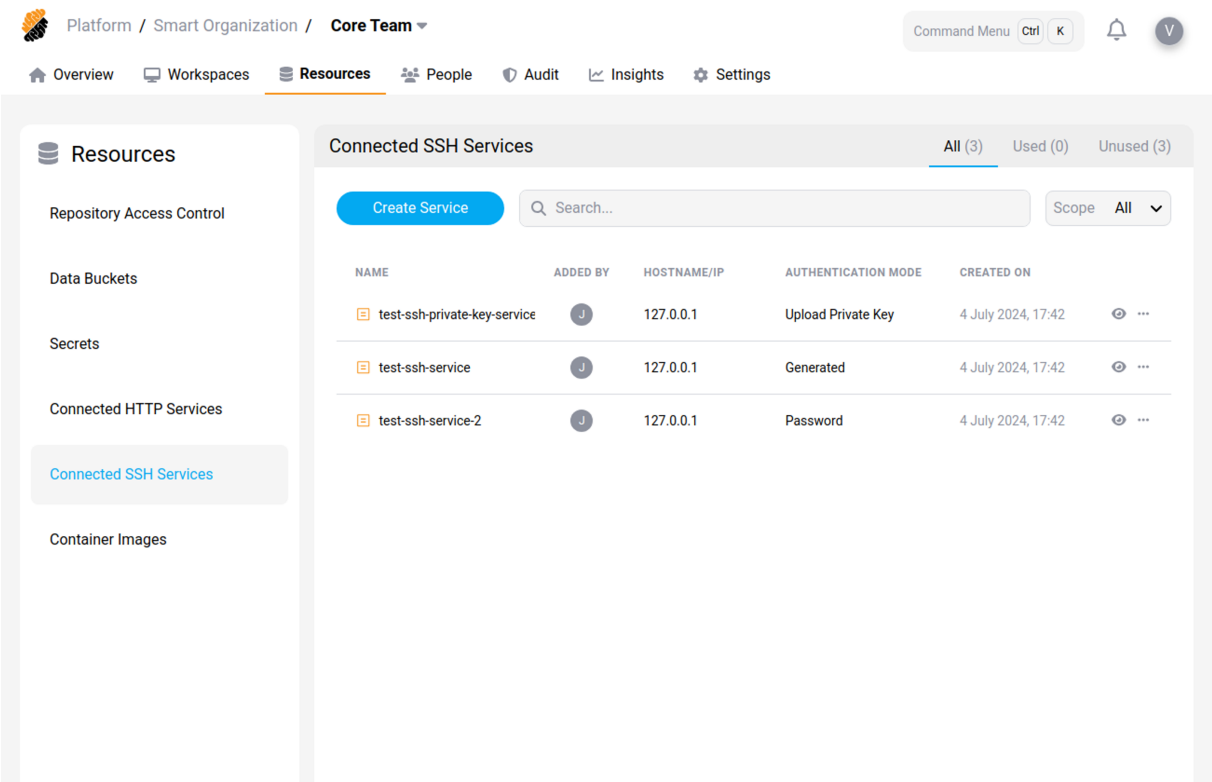
protocol. To enable users to connect to SSH services, you must either set up the SSH service as a project resource or users must configure a personal SSH identity in their profile settings.

> **Tip**
>
> The nature and protocol of services that can be attached to workspaces depend on your platform's implementation.

As with other resource types, SSH services are attached to workspaces when the workspace is created or when
its settings are updated. For the attach step and the alternative option, see
Enable outbound SSH.

- View Connected SSH Services
- Add an SSH Service

View Connected SSH Services



Connected SSH services are defined by the following characteristics:

- **Basic information**: name and scope of use (platform, organization, or project).
- **Class Level**: defines the visibility of the service, based on the user's permissions.
- **Attach Asset Information**: allows you to provide a description of the service.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

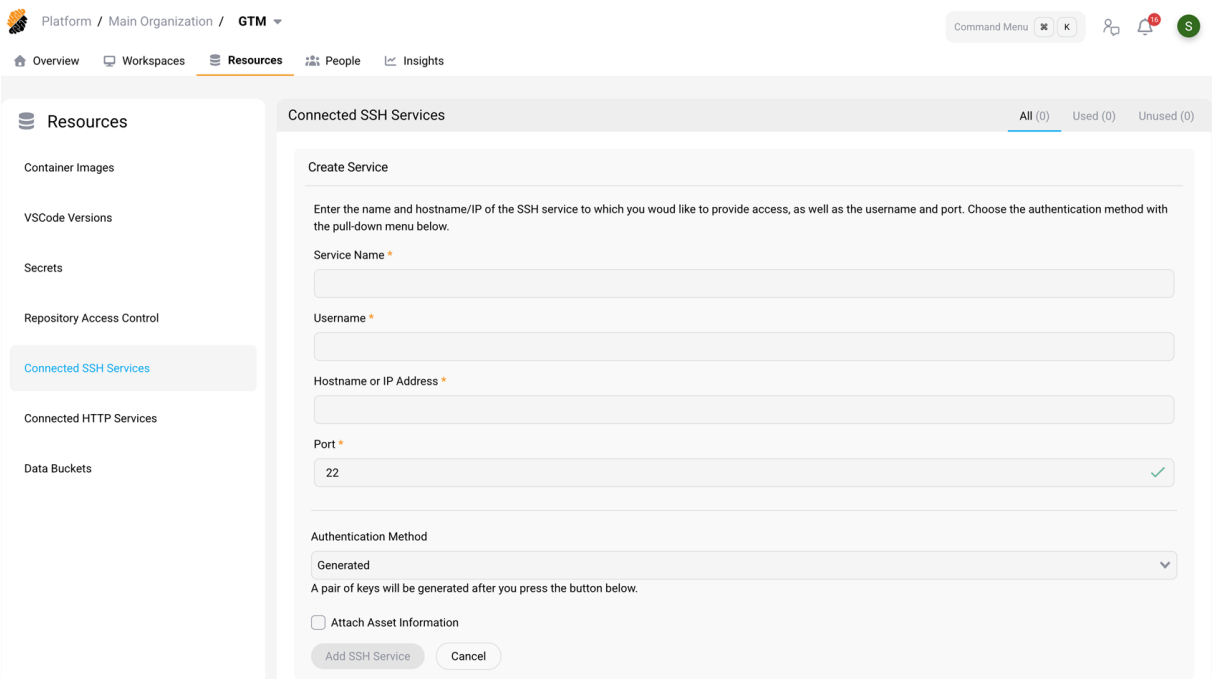
- **Hostname or IP Address**: the address of the SSH host.
- **Authentication Method**: the mechanism used to authenticate with the service.

Add an SSH Service

Requires the **Resources** permission set to **Manage**.

Select **Create Service**, then enter:

1. **Service Name**, a name to identify the host. It must not contain spaces or capital letters.
2. **Username**, the username used to access the host.
3. **Hostname or IP Address**, which points to the host location.
4. **Port**, the port the SSH service listens on.
5. **Authentication method**, an authentication method to access the service, and choose one of the methods:
    - "Generated": A pair of keys will be generating when adding the SSH service
    - "Upload Private Key": Upload the private key that will be used to authenticate you to the ssh service
    - "Password": Insert the password associated to your ssh username previously entered
6. **Asset Information**, a description of the service.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Enable outbound SSH
- Connect to external servers and services via SSH

Choose between data buckets and mount points

September 29, 2026

Citrix SecurSpaces™ provides two mechanisms for making external data available inside workspaces: **Data Buckets** and **Mount Points**. Both allow teams to share datasets, files, and artifacts across workspaces within a project, but they differ in how data is stored, synchronized, and accessed.

This guide explains how each option works and when to use it.

At a glance

	Data Buckets	Mount Points
How data reaches the workspace	Snapshot copied to the cluster	Live connection to external storage
Real-time collaboration	No —changes require a manual publish cycle	Yes —changes are immediately visible to all connected workspaces
Data duplication	Yes —each snapshot is replicated onto the cluster	No —data stays in the external storage
Versioning	Built-in —a new version is created each time data is published from a workspace	Not built-in —use external storage versioning if needed
Supported storage providers	Amazon S3, Azure Blob Storage, Google Cloud Storage	Amazon EFS, Amazon S3 Files, Azure File Storage
Setup required	Import a bucket (project owner)	Platform administrator enables the feature; project owner creates Mount Points
Best for	Reproducible experiments, versioned datasets, point-in-time references	Large shared datasets, real-time collaboration, terabyte-scale data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Data Buckets

A Data Bucket connects an external S3-compatible storage location to the platform. When a Data Bucket is attached to a workspace, the platform creates a **point-in-time snapshot** of the bucket contents and replicates it to the cluster. The snapshot is mounted as a folder in the workspace file system.

How it works

1. A project owner imports a Data Bucket by providing a name, bucket URL, and storage provider.
2. A developer attaches the Data Bucket to a workspace during workspace creation.
3. The platform replicates the bucket contents to the cluster and mounts the snapshot as a folder.
4. If the developer modifies data in a read-write bucket and publishes the changes, the platform creates a new version of the snapshot.

When to use Data Buckets

- **Reproducibility matters.** Each snapshot is versioned so that you can return to a known state of the data at any time.
- **Datasets are small to moderately sized.** Snapshots duplicate data onto the cluster, so replication time and disk usage scale with dataset size.
- **Isolation is preferred.** Each workspace gets its own copy of the data. Changes in one workspace do not affect others until explicitly published.

Limitations

- Replication can take a long time for large datasets.
- Snapshots consume cluster disk space proportional to the data size.
- Changes are not shared in real time —other users see updates only after a new version is published.

Mount Points

A Mount Point connects external file-based storage directly to the workspace file system without copying data to the cluster. The workspace reads from and writes to the external storage in real time.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How it works

1. A platform administrator enables Mount Point storage types under **System Configuration > Integrations**.
2. A project owner creates a Mount Point by specifying the storage provider, connection details, default mount path, and access permissions.
3. A developer attaches the Mount Point during workspace creation. The mount path can be customized, and a source sub-path can be specified to mount a specific subdirectory.
4. The external storage is mounted at the specified path (for example, `/mnt/shared-data`). All workspaces connected to the same Mount Point see the same files in real time.

When to use Mount Points

- **Real-time collaboration.** Multiple developers need to read or write shared files simultaneously, and changes must be visible immediately.
- **Large datasets.** Data is too large to replicate efficiently —terabytes of ML training data, model artifacts, or shared experiment results.
- **Avoiding data duplication.** You want to reduce cluster disk usage by keeping data in the external storage rather than copying it onto the cluster.

Limitations

- Requires a platform administrator to enable the feature before project owners can create Mount Points.
- No built-in versioning —if you need point-in-time snapshots, manage them in the external storage provider.
- One cloud provider per cluster. If both AWS and Azure Mount Point storage classes are present, the provider
  cannot be resolved and the feature is withdrawn. See
  Prepare AWS storage for Mount Points.
- Not supported in multi-region configurations.

Choosing between the two

Use the following questions to guide your decision:

1. **Do multiple developers need to see each other's changes in real time?**
   Yes → Mount Points. Data Buckets require a manual publish cycle to share updates.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. **Do you need built-in versioning to reproduce past states of the data?**
   Yes → Data Buckets. Each publish creates a versioned snapshot. Mount Points do not automatically version data.
3. **Is the dataset very large (hundreds of GB or more)?**
   Yes → Mount Points. They avoid the time and disk cost of replicating data to the cluster.
4. **Do you need each developer to work on an isolated copy of the data?**
   Yes → Data Buckets. Each workspace has its own snapshot, independent of other workspaces.
5. **Which storage provider are you using?**
   Data Buckets support Amazon S3, Azure Blob Storage, and Google Cloud Storage. Mount Points support Amazon EFS, Amazon S3 Files, and Azure File storage. For the AWS path, see Prepare AWS storage for Mount Points.

Both options can be used within the same project. For example, a team might use a Mount Point for a shared training-data directory and a Data Bucket for versioned experiment results.

Related topics

- Data Buckets
- Mount Points
- Enable Mount Point Storage
- Create a Workspace

Related information

- Code repositories
- Connected HTTP services
- Connected SSH services
- Data buckets
- Mount points
- Project resources

Enable outbound SSH

August 18, 2026

By default the SecurSpaces proxy service blocks outbound SSH connections from workspaces. Until you enable one

© 1997–2026 Citrix Systems, Inc. All rights reserved.

of the two options below, a developer who tries to reach an external host over SSH has the connection closed.

This page is for project owners and security officers. For the developer side —generating a key, registering
it, and connecting —see
Connect to external servers and services via SSH.

Choose an option

Consideration	Personal SSH Identity	Connected SSH Services
Best for	Individual developers who each need their own key, for example to authenticate with Azure DevOps or GitHub	Teams where several people connect to the same SSH service
Who sets it up	You enable the toggle; each developer generates and registers their own key	You create the service resource and attach it to workspaces; developers connect with no further setup
Key ownership	Each user owns their key	The platform or an administrator holds the key
Registration effort	Each user registers their public key on every target service	Registered once, and everyone using the workspace connects
Scope	User level, across every workspace where the toggle is on	Resource level, attached per workspace or template
Best fit	Access to personal accounts and services	Shared infrastructure such as build servers, staging VMs, and cloud instances

Option 1: Enable Personal SSH Identity

Requires the **Security** permission set to **Manage**, which project owners and security officers hold.

With Personal SSH Identity the platform generates and holds a key pair for each user. The private key is
applied when the developer connects and is never exposed inside the workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

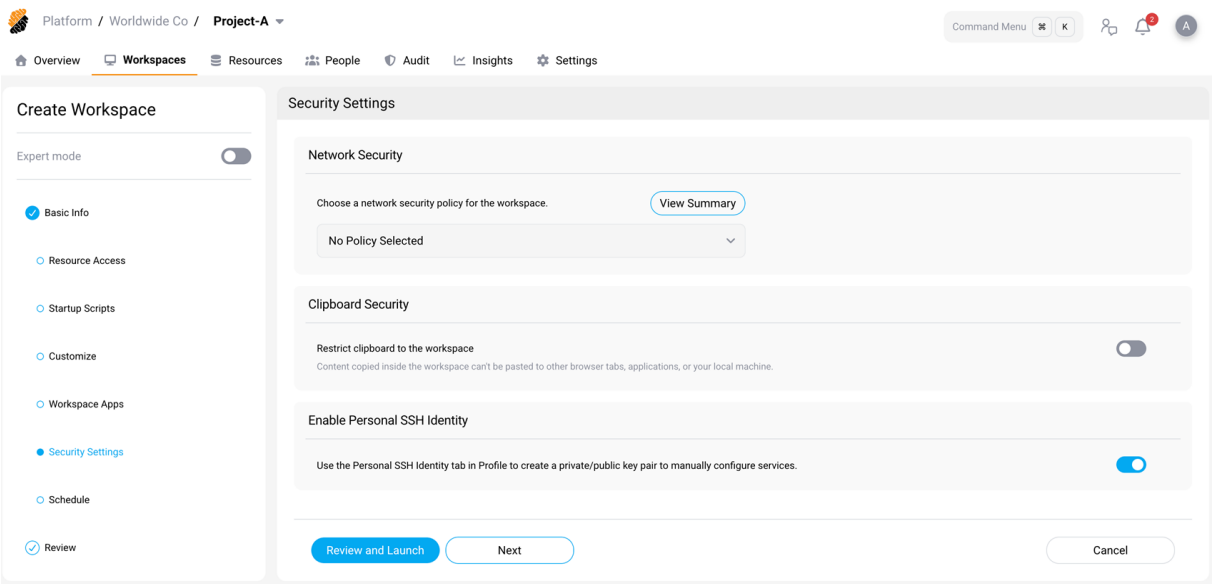
You can enable the setting on a single workspace, or on a template so that every workspace created from it
inherits the setting.

On a workspace

1. Go to the project's **Workspaces** tab.
2. Create or edit a workspace.
3. In the left panel, select **Security Settings**.
4. Switch **Enable Personal SSH Identity** on.
5. Select **Review and Launch**, or **Next**, to save.

On a workspace template

1. Go to the project's **Workspaces** tab and select the **Templates** view.
2. Create or edit a template.
3. In **Security Settings**, switch **Enable Personal SSH Identity** on.
4. Save the template.



Outbound SSH is now allowed from that workspace. Each developer then generates their own key in their profile
and registers the public key on the target service. See
Personal SSH Identity.

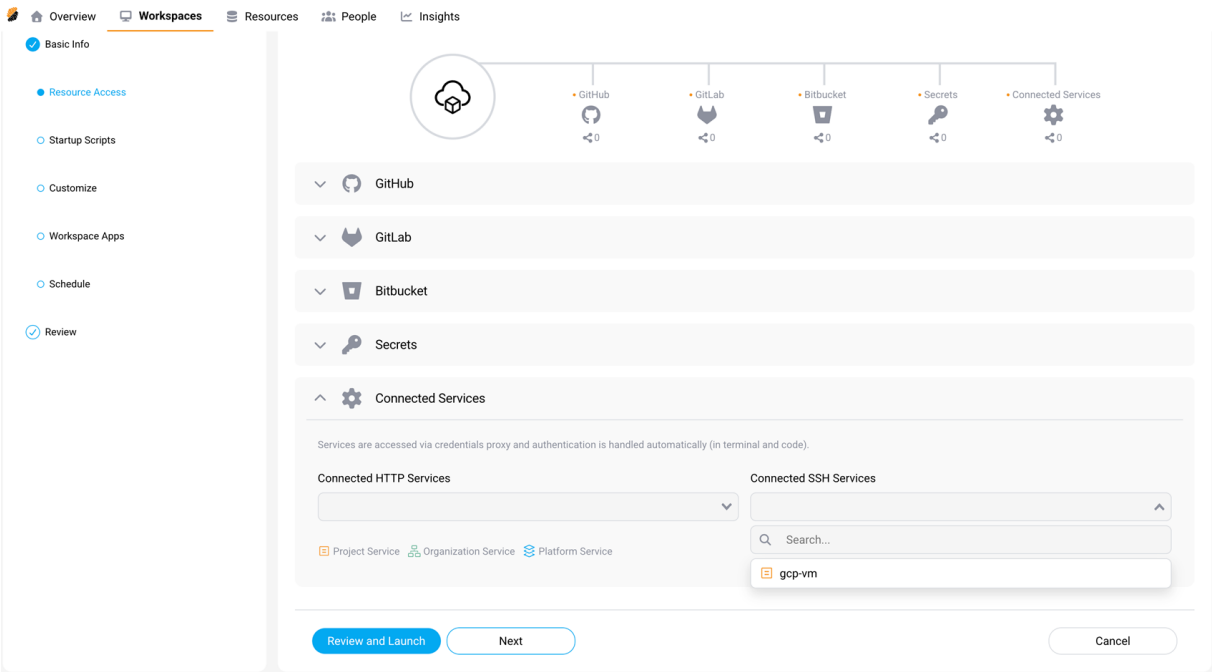
© 1997–2026 Citrix Systems, Inc. All rights reserved.

Option 2: Attach a Connected SSH Service

Creating the service requires the **Resources** permission set to **Manage**.

A Connected SSH Service defines a target host once, with its credentials, and shares it across workspaces.
Developers connect with `ssh <service-name>` and never handle a key.

1. Create the service as a project resource. For the form and its fields, see Connected SSH Services.
2. Attach it to a workspace:
    a) Create or edit a workspace.
    b) In the left panel, select **Resource Access**.
    c) Under **Connected Services**, open the **Connected SSH Services** list.
    d) Select your service.
    e) Select **Review and Launch**, or **Next**, to save.



Related information

- Connect to external servers and services via SSH
- Connected SSH Services
- Network policies

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SecurSpaces Flex workspace templates

September 29, 2026

This article describes workspace templates in Citrix SecurSpaces™ Flex: what they are, how they work, and how to create and manage them. For workspace size options and credit consumption, contact your Citrix sales team.

What is a workspace template?

A workspace template is the primary deployment artifact in Citrix SecurSpaces Flex. It defines the environment that a developer receives when they start a workspace.

A workspace template includes:

- **Container image** —The base Linux container image with preinstalled tools, runtimes, and configuration.
- **Preinstalled tools** —Languages, SDKs, CLIs, and development tools baked into the image.
- **Resource limits** —Maximum vCPU and RAM the workspace is allowed to consume (separate from the workspace size profile; used to set hard limits within a compute tier).
- **Access controls** —Which users or user groups can launch workspaces from this template.
- **Startup scripts** —Scripts that run at workspace initialization: cloning repositories, configuring dot files, mounting secrets, or running setup commands.

What the customer owns

The customer owns and hosts the container image. Citrix does not provide or manage the container image content. The customer is responsible for:

- Building and maintaining the container image.
- Hosting the image in a container registry accessible from the Citrix SecurSpaces execution plane (for example, Azure Container Registry, Docker Hub, or a private registry).
- Patching and updating the image as needed.

Citrix scans the container image for known vulnerabilities when a workspace is launched. Scan results are surfaced to the administrator in the Citrix SecurSpaces console.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Decoupled size and template

Workspace size and workspace template are configured independently. A single template can be paired with any workspace size (Light, Medium, or Heavy) on demand. This means:

- You do not need a separate template per compute tier.
- Developers can switch workspace sizes without requiring a new template.
- Platform engineering teams can standardize on a small number of templates and let workspace size vary by workload.

Create a workspace template

To create a workspace template, you must have the **Administrator** role in the Citrix SecurSpaces console.

1. Sign in to your SecurSpaces environment at your organization's access URL (for example, `https://yourcompany.flex.sds.citrix.com`).
2. Go to **Administration** > **Workspace Templates**.
3. Select **Create template**.
4. Complete the template configuration:

Field	Description
Name	A descriptive name for the template (for example, `backend-java-21` or `data-platform`).
Description	Optional. A short description shown to administrators.
Container image	The full image reference including registry, image name, and tag (for example, `myregistry.azurecr.io/dev-image:latest`).
Resource limits	Optional. Maximum vCPU and RAM limits for workspaces created from this template.
Startup script	Optional. A shell script that runs at workspace initialization.
Access	The users or groups permitted to launch workspaces from this template.

1. Select **Save** to create the template.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The template is now available for workspace creation and is listed on the **Workspace Templates** page.

Managed workspace sizes

In Flex deployments, workspace sizes are managed separately from workspace templates. Citrix publishes the
available size tiers, and each tier can have a current version, older supported versions, and retired versions.
This keeps CPU, memory, storage, and storage-class values aligned with the Flex service catalog.

Administrators can still control where a size tier is available. They can assign or remove a tier for an organization or project, and can update existing workspaces to the current version of the same tier. They
cannot create, edit, or delete the Citrix-managed tier definitions.

When a newer version of a tier is available, affected workspaces show an update prompt. The update keeps the
workspace on the same tier and applies the current tier version. Retired tier versions are not available for
new workspaces, but existing workspaces remain visible so they can be reviewed and updated.

When an administrator enables or disables a managed tier, or applies a bulk update to move workspaces to the
current version of a tier, SecurSpaces records the successful action in the audit log. The audit entry includes
who performed the action, when it happened, and the result, such as the tier state or the number of workspaces
updated.

Update a workspace template

Updating a template changes the configuration for new workspaces created from it. Existing running workspaces are not affected until they are restarted.

1. Go to **Administration** > **Workspace Templates**.
2. Select the template you want to update.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

3. Edit the relevant fields.
4. Select **Save**.

> **Note:**
>
> If you update the container image tag (for example, from `v1.0` to `v1.1`), new workspaces will use the updated image. Developers with running workspaces must stop and restart their workspace to pick up the new image.

Template versioning

Workspace templates support versioning. Each saved version of a template is recorded. Administrators can:

- View the version history of a template.
- Roll back to a previous version if a new image or configuration causes issues.

Delete a workspace template

Deleting a template removes it from the available template list. Existing running workspaces that were created from the deleted template continue to run until they are stopped.

1. Go to **Administration** > **Workspace Templates**.
2. Select the template you want to delete.
3. Select **Delete**.
4. Confirm the deletion in the dialog.

More information

- SecurSpaces Flex overview —the starting point for all Flex documentation
- Platform Flex credit consumption
- Network connections
- Technical security overview
- Workspace templates in Citrix SecurSpaces (self-hosted) —for workspace template concepts that apply to both deployment models

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Deploy

August 19, 2026

This section is for the team standing SecurSpaces up on their own infrastructure. SecurSpaces is customer-hosted: it runs on a Kubernetes cluster you provide and control.

Check System requirements and
Network destinations before you begin.
The second matters most for air-gapped or firewalled environments.

Install

Task	Page
Deploy a single-VM evaluation environment	Deploy an evaluation environment
Install with the CLI	Install with the Strong Installer CLI
Deploy on Amazon EKS	Amazon EKS
Deploy on Azure AKS	Azure AKS
Deploy on Red Hat OpenShift	Red Hat OpenShift

Configure the environment

Task	Page
Set up the ingress controller	Ingress
Allow SSH connections to workspaces	SSH TCP forwarding
Use an external database	External database
Route through an external proxy	External proxy
Run workspaces in more than one region	Multiple regions
Provide GPUs to workspaces	GPU-enabled workspaces on AKS

Keep it current

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Task	Page
Upgrade an existing deployment	Upgrade

Related information

- Administer —configuration after installation
- Trust boundaries —what you own

Deploy an evaluation environment

September 24, 2026

The 1-Click VM runs the whole Citrix SecurSpaces™ platform on a single virtual machine (VM). Use it for a proof of
concept (POC) or a demonstration. Preparing the VM is the part that differs between platforms; the installation,
validation and management steps are the same.

> **Note:**
>
> The 1-Click VM is not for production use and cannot be upgraded or migrated to a production installation. For
> production, see Deploy to a cluster.

The deployment sequence

1. **Prepare the VM.** Use the page for your platform in the table below. In the cloud, the installer creates the VM
   for you.
2. **Install SecurSpaces.** See Deploy in the cloud or
   Install on a local VM.
3. **Validate the deployment.** See Validate the deployment.
4. **Manage the environment.** See
   Manage the evaluation environment.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Choose your platform

Platform	Page	Notes
AWS, Azure or Google Cloud	Deploy in the cloud	The installer creates the VM, DNS and TLS
Citrix XenServer®	Prepare a XenServer VM	Ubuntu guest created in XenCenter
Microsoft Hyper-V	Prepare a Hyper-V VM	Ubuntu guest on a Windows PC
Other hypervisors or bare metal	Install on a local VM	Any host that meets the requirements

Before you start

- Check the host requirements in System requirements.
- Choose a machine capacity. See Sizing.
- Have an administrator email address ready.

For how the 1-Click VM works, see The 1-Click VM.

Next step

Choose your platform:

- Deploy in the cloud
- Prepare a XenServer VM
- Prepare a Hyper-V VM
- Install on a local VM, for other hypervisors and bare metal

Related information

- Deploy to a cluster
- Licensing

Deploy in the cloud

September 24, 2026

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This page deploys the Citrix SecurSpaces™ 1-Click VM to AWS, Azure or Google Cloud. The installer creates the VM,
installs the platform, and configures DNS and TLS certificates.

Prerequisites

- Docker on your local machine.
- Credentials for AWS, Azure or Google Cloud, with the permissions in Cloud permissions.
- On AWS or Google Cloud, inbound access to the ports in Open the required ports.
- An administrator email address.
- A machine capacity. See Sizing.

Open the required ports

The VM needs three inbound ports:

Port	Used for
443	The SecurSpaces console and the Cloud IDE, over HTTPS
2222	SSH connections from local IDEs to workspaces
22	SSH access to the VM itself, for administration

What you have to do depends on the cloud provider.

Azure. No action needed. The installer creates a network security group that allows all three ports from any address.

AWS. Open the ports yourself. The installer does not create a security group, so the VM uses the default security group of your default VPC, which does not allow inbound traffic from the internet. Before you deploy,
add inbound rules for TCP ports 443, 2222, and 22 to that security group.

GCP. Open the ports yourself. The installer does not create firewall rules. It places the VM on the **`default`** network with the network tag `https-server`. Before you deploy, create firewall rules on the
`default` network that allow TCP ports 443, 2222, and 22 to instances with the `https-server` tag.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning:**
>
> Allow these ports only from the addresses your evaluators connect from, rather than from any address. On
> Azure, the rules the installer creates allow any address; restrict their source after deployment if your
> security policy requires it.

Cloud permissions

The installer deploys with the cloud identity you sign in with when it prompts for cloud credentials. That
identity needs permission to create the VM and the resources around it. What it needs depends on the cloud
provider.

Opening the ports in Open the required ports on AWS or GCP is a separate, manual
step. Whoever does it needs permission to change the security group or create firewall rules, which the
permissions below do not include.

Azure

Sign in with a user account or a service principal. It needs permission to create the VM, its network, and a
load balancer.

The installer asks for a resource group, and suggests `strong-network-demo-rg`. If the resource group
already exists, the installer uses it. If it does not, the installer creates it. Which scope you grant
depends on that:

Resource group	Grant the role at
Already exists	The resource group
Created by the installer	The subscription

To avoid granting a role on the whole subscription, create the resource group yourself first and grant the
role on it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Choose one of the following.

Built-in role. Assign **Contributor** at the scope from the table. This is the simplest option.

Narrower built-in roles. Assign both **Virtual Machine Contributor** and **Network Contributor** at that
scope. The installer creates no storage account, so no storage role is needed.

Custom role. For the fewest permissions, create a custom role with these actions and assign it at that scope. Replace `{ subscription-id }` with your subscription ID:

```
{

    "Name": "SecurSpaces 1-Click VM deployer",
    "IsCustom": true,
    "Description": "Deploy a SecurSpaces 1-Click VM",
    "Actions": [
        "Microsoft.Compute/virtualMachines/*",
        "Microsoft.Compute/disks/*",
        "Microsoft.Compute/virtualMachines/extensions/*",
        "Microsoft.Network/virtualNetworks/*",
        "Microsoft.Network/publicIPAddresses/*",
        "Microsoft.Network/networkInterfaces/*",
        "Microsoft.Network/networkSecurityGroups/*",
        "Microsoft.Network/loadBalancers/*",
        "Microsoft.Resources/resourceGroups/*",
        "Microsoft.Resources/subscriptions/resourceGroups/read"
    ],
    "NotActions": [],
    "DataActions": [],
    "NotDataActions": [],
    "AssignableScopes": ["/subscriptions/{
 subscription-id }
 "]
 }
```

For example, to assign the narrower built-in roles to a service principal on an existing resource group:

```
az role assignment create \
  --assignee <service-principal-id> \
  --role "Virtual Machine Contributor" \
  --scope "/subscriptions/<subscription-id>/resourceGroups/<resource-
     group-name>"

az role assignment create \
  --assignee <service-principal-id> \
  --role "Network Contributor" \
  --scope "/subscriptions/<subscription-id>/resourceGroups/<resource-
     group-name>"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AWS

The identity needs permission to create an EC2 instance, a key pair, and an Elastic IP address. The installer
creates no security group and attaches no IAM role to the VM, so neither security group nor IAM permissions
are needed.

Choose one of the following.

Managed policy. Attach **AmazonEC2FullAccess**. This is the simplest option.

Custom policy. For the fewest permissions, attach a policy with these actions:

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeImages",
                "ec2:DescribeAvailabilityZones",
                "ec2:CreateKeyPair",
                "ec2:DeleteKeyPair",
                "ec2:DescribeKeyPairs",
                "ec2:AllocateAddress",
                "ec2:ReleaseAddress",
                "ec2:AssociateAddress",
                "ec2:DisassociateAddress",
                "ec2:DescribeAddresses",
                "ec2:CreateTags",
                "ec2:DescribeTags",
                "ec2:DescribeVolumes",
                "ec2:CreateVolume",
                "ec2:DeleteVolume",
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "*"
         }

    ]
 }
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

GCP

The identity needs permission to create a Compute Engine instance and a static external IP address in the
project you select. The installer attaches no service account to the VM, so no service account permissions
are needed.

The installer also lists your projects so you can pick one, and checks that the Compute Engine API is enabled
in it. If the API is not enabled, the installer tries to enable it.

Choose one of the following.

Predefined roles. Grant **Compute Admin** (`roles/compute.admin`). This is the simplest option. For
narrower access, grant **Compute Instance Admin (v1)** (`roles/compute.instanceAdmin.v1`) and **Compute
Network User** (`roles/compute.networkUser`) instead.

Custom role. For the fewest permissions, create a role with these permissions:

```
{

    "title": "SecurSpaces 1-Click VM deployer",
    "description": "Deploy a SecurSpaces 1-Click VM on GCP",
    "stage": "GA",
    "includedPermissions": [
        "compute.instances.create",
        "compute.instances.delete",
        "compute.instances.get",
        "compute.instances.list",
        "compute.instances.setMetadata",
        "compute.instances.setTags",
        "compute.instances.start",
        "compute.instances.stop",
        "compute.addresses.create",
        "compute.addresses.delete",
        "compute.addresses.get",
        "compute.addresses.list",
        "compute.addresses.use",
        "compute.disks.create",
        "compute.disks.delete",
        "compute.disks.get",
        "compute.disks.list",
        "compute.disks.use",
        "compute.images.get",
        "compute.images.list",
        "compute.images.useReadOnly",
        "compute.zones.get",
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        "compute.zones.list",
        "compute.regions.get",
        "compute.regions.list",
        "compute.machineTypes.get",
        "compute.machineTypes.list",
        "compute.networks.get",
        "compute.subnetworks.use",
        "compute.subnetworks.useExternalIp",
        "resourcemanager.projects.get",
        "serviceusage.services.list"
    ]
}
```

Whichever option you choose, the identity also needs `serviceusage.services.list`, so the installer can
confirm the Compute Engine API is enabled. Without it, the installer cannot tell that the API is enabled and
keeps asking you to enable it. If the API is not yet enabled, enable it yourself before you deploy, or grant
`serviceusage.services.enable` so the installer can do it.

Run the installer

1. Create a directory for the deployment files and run the installer in it:

   ```
   mkdir -p ~/sds-cloud-deployment
   cd ~/sds-cloud-deployment
   docker run -it --rm -v ${
    PWD }
    :/strong-network/shared strongnetwork/strong_installer:<version>
   ```

   Replace `<version>` with the release to deploy, for example `2026.9.4`.

2. At the `sds-cli` prompt, enter:

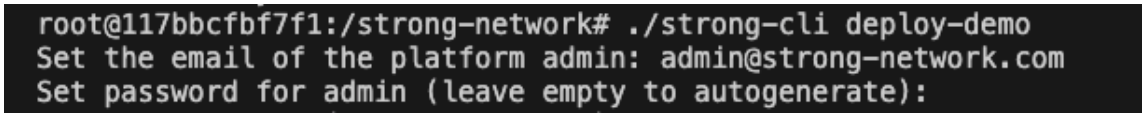
   ```
   deploy-demo
   ```

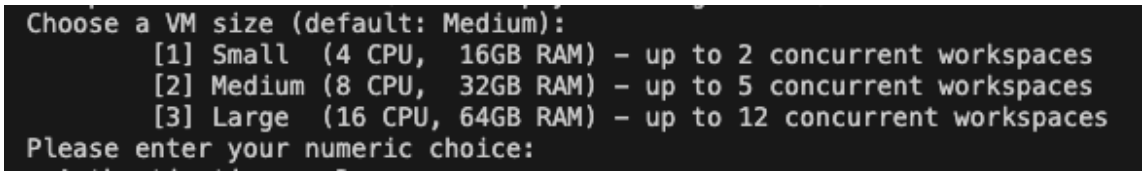
3. Complete the prompts:

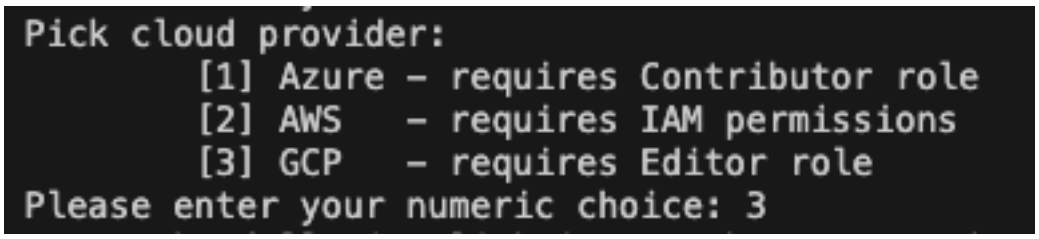
Prompt	Enter
Administrator credentials	Email address and password
VM size	Your machine capacity
Cloud provider	AWS, Azure or GCP
Cloud credentials	Your cloud identity. For GCP, select a project from the list.

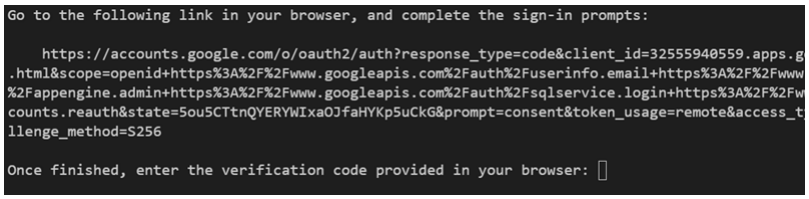
© 1997–2026 Citrix Systems, Inc. All rights reserved.

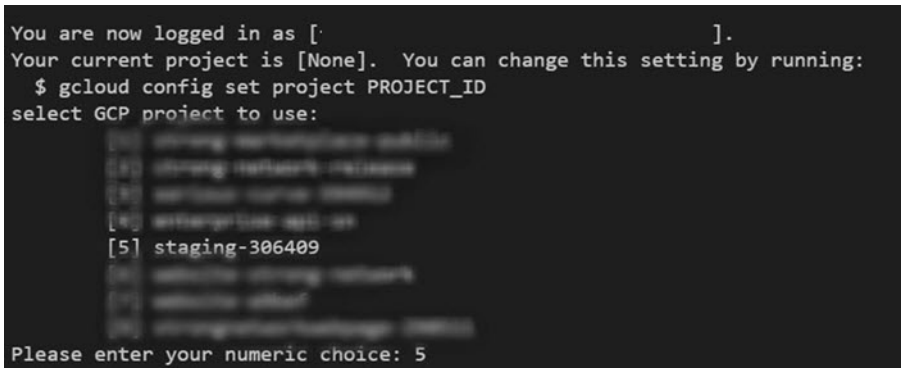
Prompt	Enter
Region	US, EU or Asia, or a specific datacenter

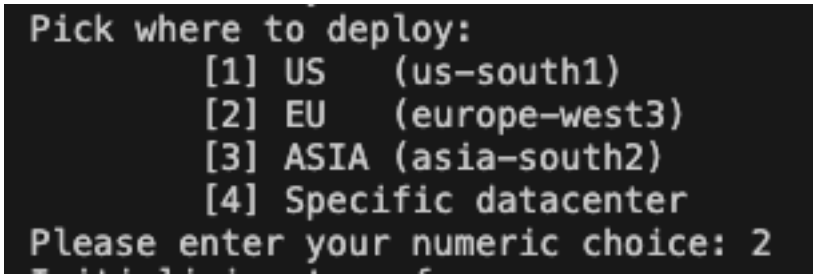












4. Wait for the installer to finish. Note the platform URL and the administrator credentials it displays.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Caution:**
>
> The installer output and the deployment directory contain credentials and cloud state. Store them privately, and
> keep the directory: you need it to remove the cloud resources.

The VM is running and the platform URL is available.

Next step

Continue with Validate the deployment.

Prepare a XenServer VM

September 24, 2026

This page creates the Ubuntu VM on Citrix XenServer® that Citrix SecurSpaces™ is installed on. When the VM is ready,
continue with Install on a local VM.

Prerequisites

- A XenServer host managed with XenCenter, with free resources for your machine capacity. See Sizing.
- An Ubuntu 22.04 or 24.04 Desktop ISO in a storage repository.
- A virtual network with outbound internet access that client devices can reach.

Create the VM

1. In XenCenter, select **New VM**, and select the Ubuntu Desktop ISO as the installation source.
2. Allocate vCPUs, memory and disk for your machine capacity.
3. Connect the network interface to the virtual network.
4. Install Ubuntu Desktop. Create a user account with `sudo` permissions.

Prepare the guest

In a terminal on the VM:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Install and enable SSH:

```
sudo apt update
sudo apt install -y openssh-server
sudo systemctl enable --now ssh
```

2. Note the VM's IP address:

```
hostname -I
```

3. To keep a clean restore point, take a snapshot of the VM in XenCenter.

The VM is running, clean and reachable.

Troubleshooting

Issue	Resolution
The AVX check fails on the guest	The XenServer host CPU does not support AVX. Use a compatible host.
Client devices cannot reach the VM	Connect the VM to a network that the client devices can route to.

Next step

Continue with Install on a local VM.

Prepare a Hyper-V VM

September 24, 2026

This page creates the Ubuntu VM on Microsoft Hyper-V that Citrix SecurSpaces™ is installed on. The Windows host can
also run the installer. When the VM is ready, continue with
Install on a local VM.

Prerequisites

- A Windows edition with Hyper-V enabled, for example Windows 11 Pro.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Enough free memory on the host for a fixed allocation that matches your machine capacity. See Sizing.
- Free disk space for the virtual disk and checkpoints.
- An Ubuntu 22.04 or 24.04 Desktop ISO from Ubuntu releases.
- To run the installer on the Windows host: Docker Desktop running Linux containers.

Create the VM

1. In **Hyper-V Manager**, select your local host.
2. Select **New** > **Virtual Machine**, and enter a name, for example `SecurSpaces-Lab`.
3. Select **Generation 2**.
4. Set **Startup memory** to the memory for your machine capacity, for example `16384` MB for Small. Clear **Use**
   Dynamic Memory for this virtual machine.
5. Connect the network adapter to **Default Switch**. To reach the console from other devices, connect it to an
   external virtual switch instead.
6. Create a virtual hard disk for your machine capacity, for example 200 GB for Small.
7. Select the Ubuntu Desktop ISO as the boot image, and finish the wizard.
8. Open the VM's **Settings**:
   - In **Processor**, set the number of virtual processors for your machine capacity, for example 4 for Small.
   - In **Security**, keep **Enable Secure Boot** selected, and select the **Microsoft UEFI Certificate Authority** template.
9. Start the VM, select **Connect**, and install Ubuntu Desktop. Create a user account with `sudo` permissions.
10. Shut down the VM, remove the ISO from the DVD drive, and start the VM.

> **Tip:**
>
> Deleting a VM in Hyper-V does not delete its VHDX. When you repeat the installation, create a new virtual disk.

Prepare the guest

In a terminal on the VM:

1. Install and enable SSH and curl:

```
sudo apt update
sudo apt install -y openssh-server curl
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
sudo systemctl enable --now ssh
```

2. Install the Hyper-V tools for the running kernel, and start the KVP daemon, so that Hyper-V reports the VM's IP
   address:

```
sudo apt install -y "linux-cloud-tools-$(uname -r)"
sudo udevadm control --reload-rules
sudo udevadm trigger --action=add --subsystem-match=misc
sudo systemctl enable --now hv-kvp-daemon
```

3. Stop the VM from suspending when idle:

```
sudo systemctl mask sleep.target suspend.target hibernate.target
    hybrid-sleep.target
```

4. Note the VM's IP address:

```
hostname -I
```

5. From PowerShell on the Windows host, check that you can connect:

```
ssh <user>@<vm-ip>
```

6. To keep a clean restore point, shut down the VM and create a checkpoint.

The VM is running, clean and reachable.

Run later steps from Windows

When you run the installer on the Windows host, use these PowerShell commands in the later steps.

1. Run the installer:

```
New-Item -ItemType Directory -Path "$env:USERPROFILE\
    SecurSpacesInstaller"
Set-Location "$env:USERPROFILE\SecurSpacesInstaller"
docker run --rm -it `
  --mount "type=bind,source=$($PWD.Path),target=/strong-network/
      shared" `
  strongnetwork/strong_installer:<version>
```

2. Copy the deployment script to the VM:

```
scp .\deploy.sh <user>@<vm-ip>:~/deploy.sh
```

3. Check DNS and HTTPS:

```
Resolve-DnsName <name>.try.sds.citrix.com
curl.exe --head https://<name>.try.sds.citrix.com/
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

4. After a restart, check the VM's state and IP address, in PowerShell as administrator:

```
Get-VM -Name SecurSpaces-Lab
Get-VMNetworkAdapter -VMName SecurSpaces-Lab | Select-Object -
    ExpandProperty IPAddresses
```

Troubleshooting

Issue	Resolution
The VM does not start, with error `0x800705AA`	Hyper-V cannot allocate the memory. Free memory on the host by closing applications or other VMs.
Hyper-V shows no IP address for the VM	Install `linux-cloud-tools-$(uname -r)` and start `hv-kvp-daemon`. The generic `linux-cloud-tools-virtual` package can match a different kernel.
The AVX check fails on the guest	The host CPU does not support AVX. Use a compatible host.
The installer container does not start	Make sure Docker Desktop is running Linux containers and can reach Docker Hub.
The platform URL resolves to the wrong address	The VM's address on the Default Switch changed. Compare `hostname -I` with the DNS record.
Other devices cannot reach the console	The Default Switch is reachable from the Windows host only. Use an external virtual switch.

Next step

Continue with Install on a local VM.

Install on a local VM

September 24, 2026

This page installs Citrix SecurSpaces™ on an Ubuntu VM or host that you have prepared. You run the installer on a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

separate machine, the installer host. It generates a deployment script that you run on the target host.

To prepare a VM first, see Prepare a XenServer VM or Prepare a Hyper-V VM.

Prerequisites

- A target host that meets the 1-Click VM requirements, with no Docker, Kubernetes or K3s installed.
- An installer host with Docker, outbound HTTPS access to Docker Hub, and 20 GB of free disk space.
- An administrator email address.
- A machine capacity. See Sizing.

Check the target host

1. On the target host, run:

```
cat /etc/os-release
grep -o 'avx[^ ]*' /proc/cpuinfo | head -1
free -h
df -h /
hostname -I
```

2. Check that:
   - The operating system is Ubuntu 22.04 or 24.04.
   - The AVX command returns output. If it does not, use a different host.
   - The memory and disk match your machine capacity.
3. Note the IP address that client devices will use.

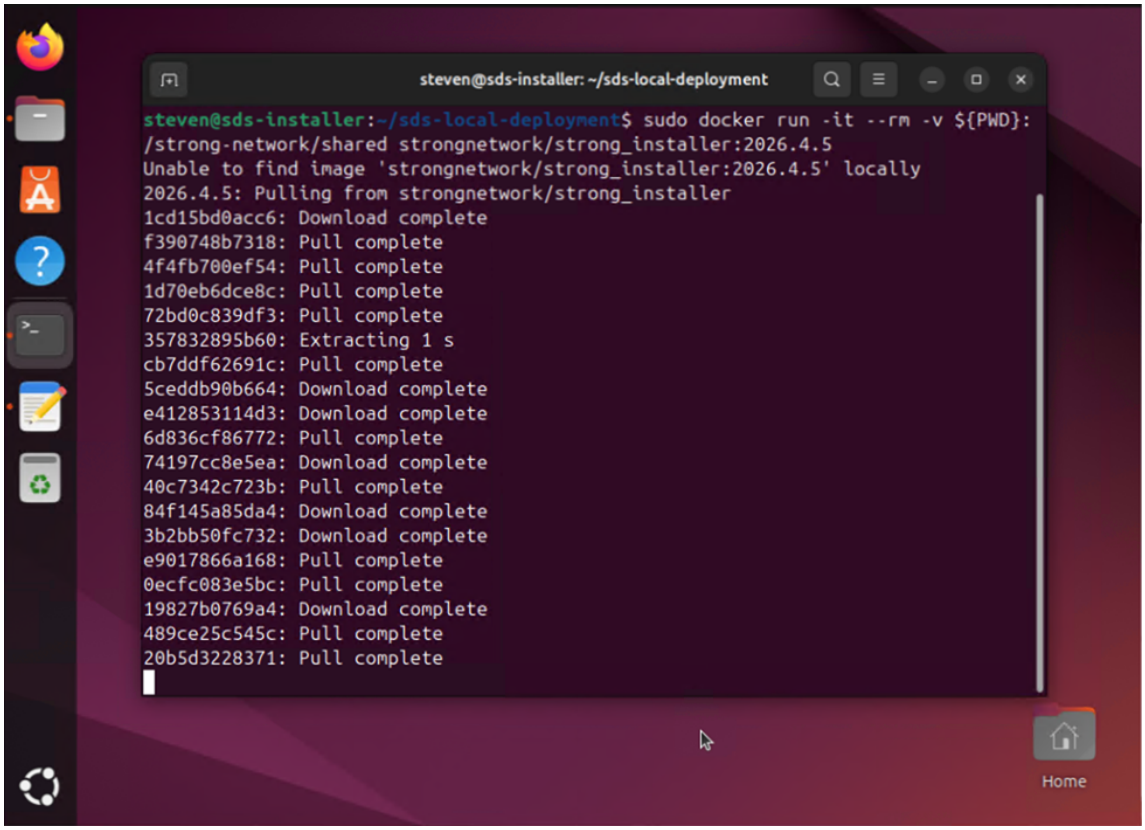
Run the installer

1. On the installer host, create a private directory and run the installer in it:

```
mkdir -p ~/sds-local-deployment
cd ~/sds-local-deployment
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared strongnetwork/strong_installer:<version>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

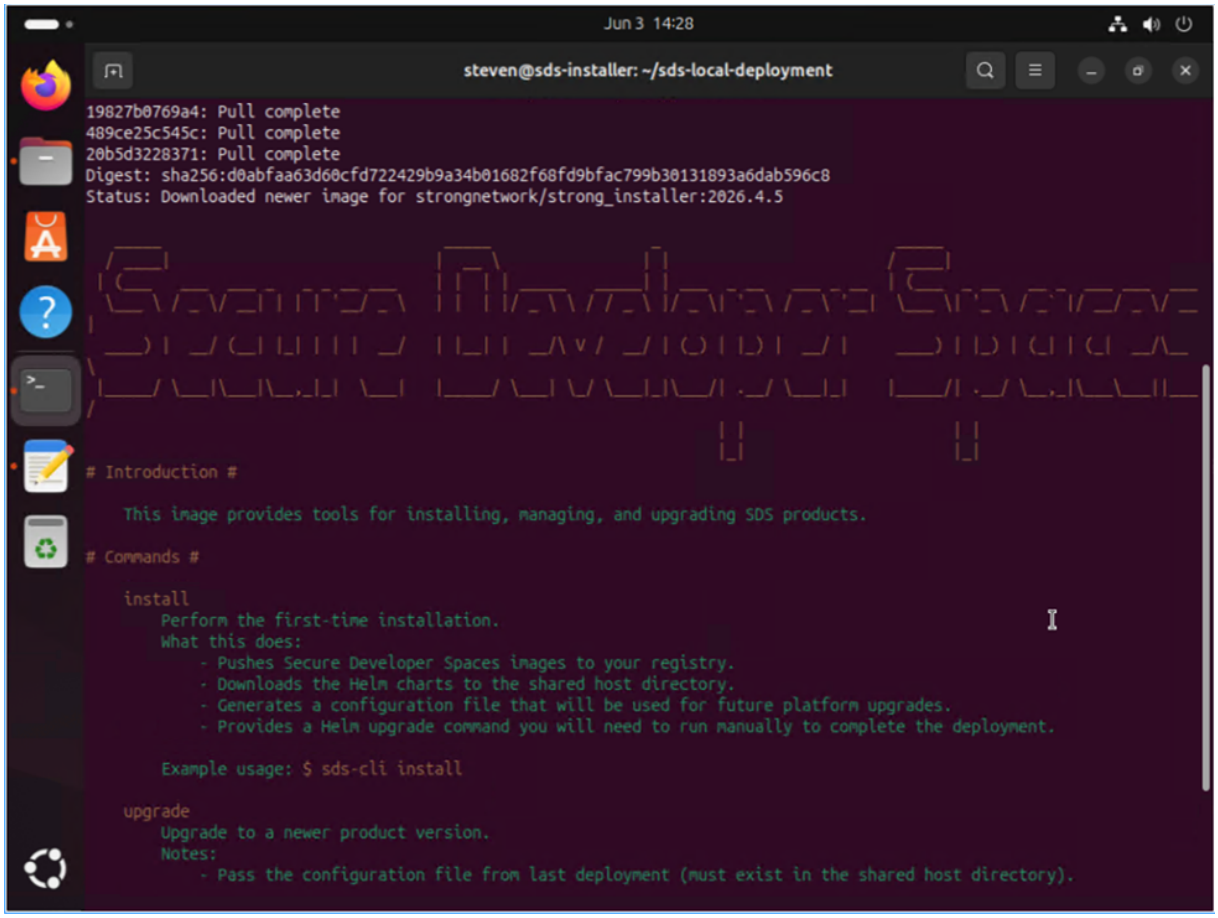
Replace <version> with the release to deploy, for example 2026.9.4. On Windows, see Run later steps from Windows.



2. If the installer offers a newer version, select **No** to stay on the version you chose.

The sds-cli prompt opens. It accepts SecurSpaces CLI subcommands only. See SDS CLI.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

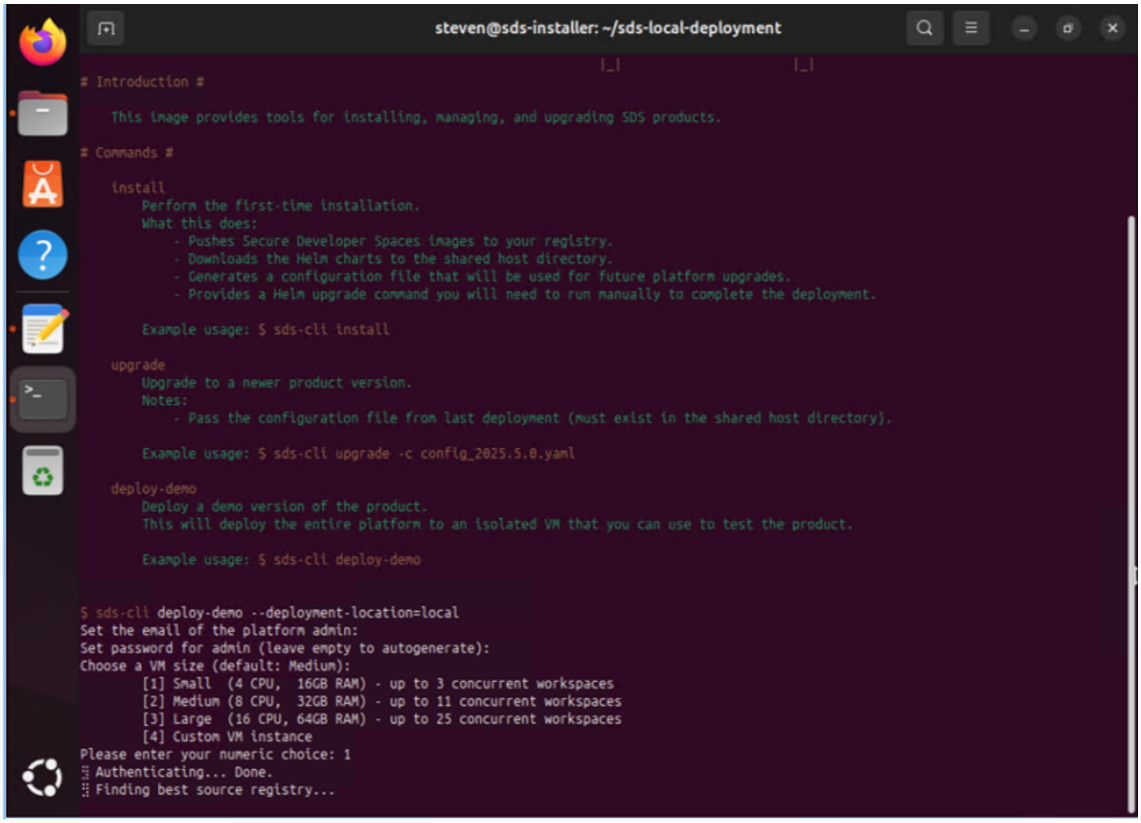


Generate the deployment script

1. At the `sds-cli` prompt, enter this command on one line:

```
deploy-demo --deployment-location=local
```

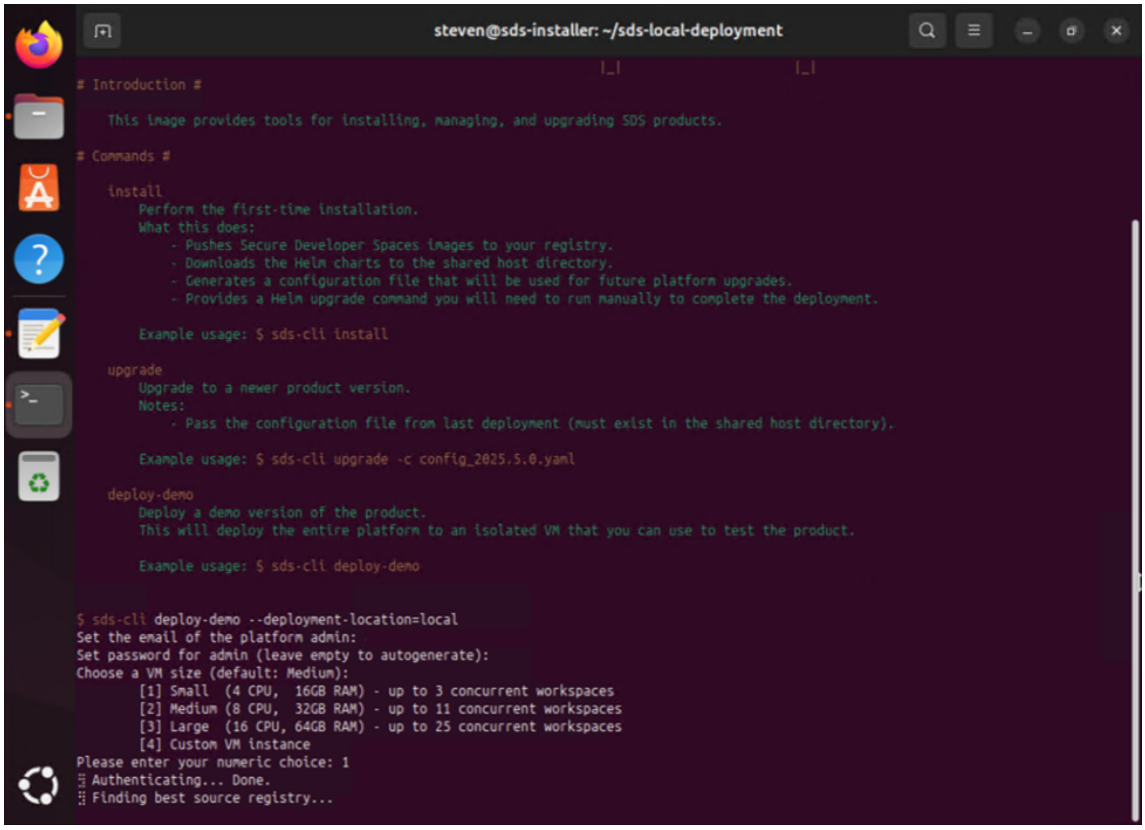
© 1997–2026 Citrix Systems, Inc. All rights reserved.



2. Complete the prompts:

Prompt	Enter
Administrator email	The address you sign in with
Administrator password	A strong password, or leave empty to generate one
Machine capacity	The capacity that matches the target host. The default is Medium.
Region	**1** US, **2** EU or **3** Asia

© 1997–2026 Citrix Systems, Inc. All rights reserved.



3. Under **Generated Local Deployment Script**, copy the script from `#!/bin/bash` up to, but not including, the
   separator line.
4. Save it on the installer host as `deploy.sh`, in UTF-8 without BOM and with LF line endings.
5. Note the assigned URL, for example `<name>.try.sds.citrix.com`, and the administrator credentials.

> **Caution:**
>
> The script and the installer output contain secrets. Store them privately.

Leave the installer open for the next step.

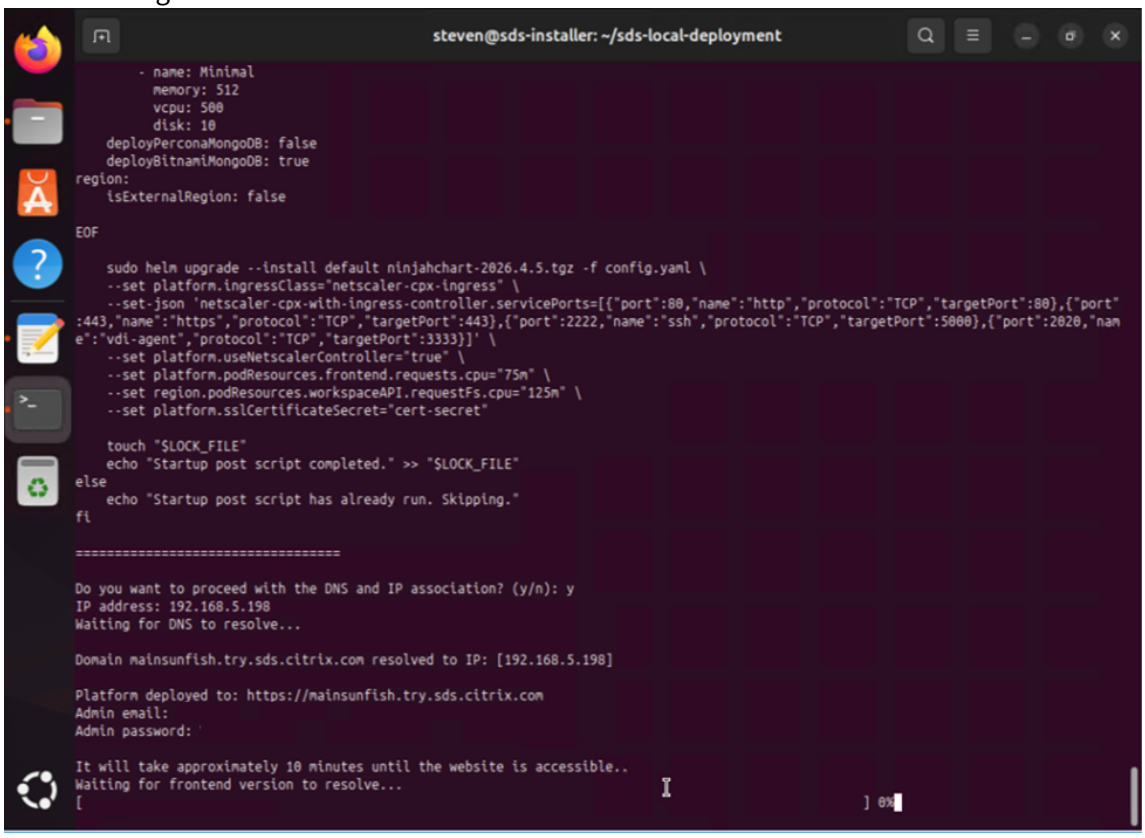
Associate the IP address with DNS

> **Important:**
>
> Complete this step within 1 hour of generating the script. You can run the script at the same time.

1. When the installer asks whether to proceed with the DNS and IP association, enter `y`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Enter the target host's IP address.



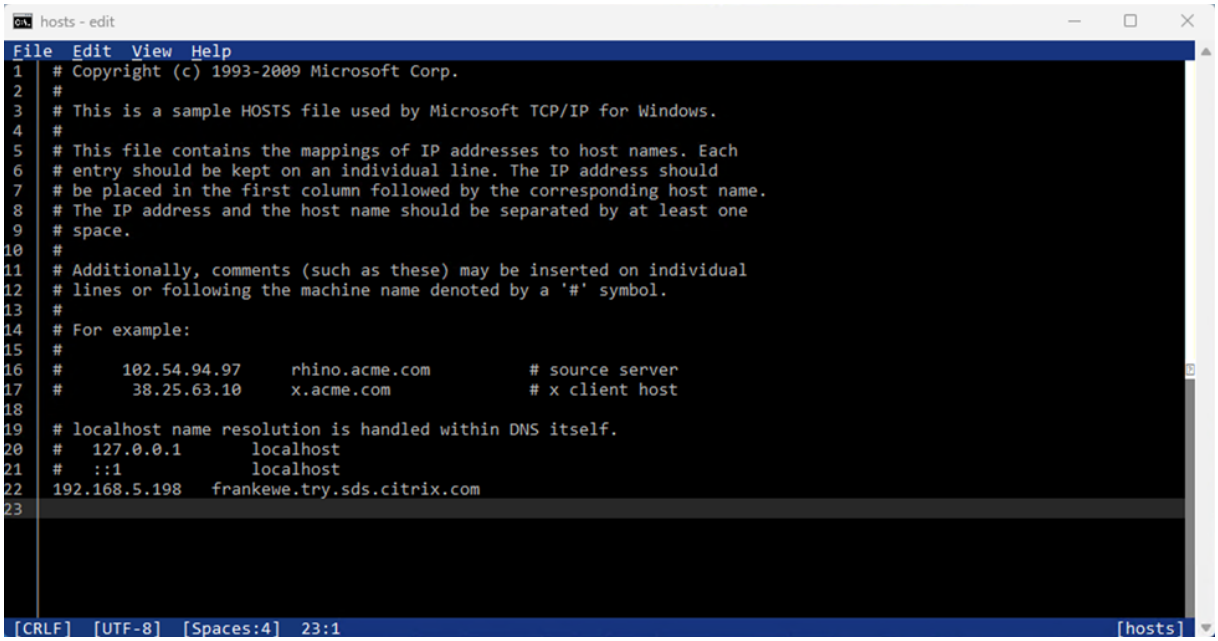
3. From a client device, check that the assigned hostname resolves to that address:

```
nslookup <name>.try.sds.citrix.com
```

If client devices cannot resolve the hostname, add a record to their DNS server, or add this line to the hosts file on
each client device:

```
<host-ip>  <name>.try.sds.citrix.com
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Run the deployment script

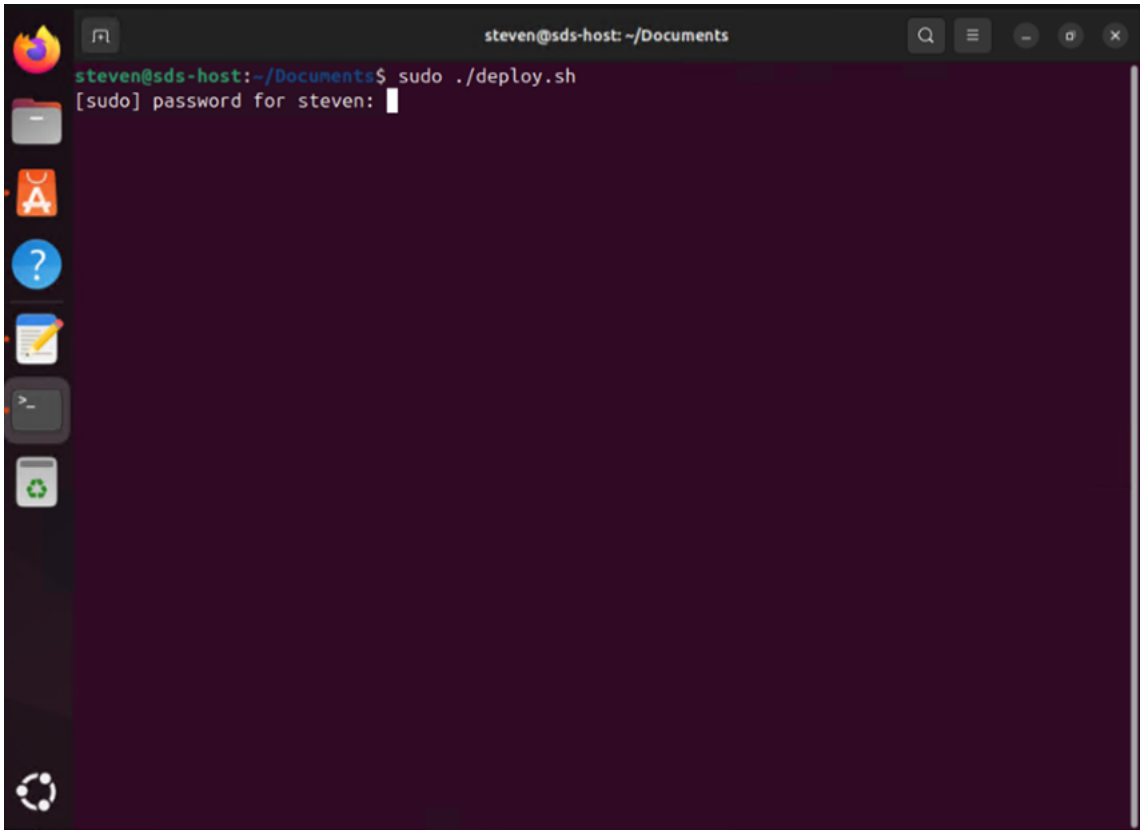
1. Copy `deploy.sh` to the target host, and connect to it:

```
scp deploy.sh <user>@<host-ip>:~/deploy.sh
ssh <user>@<host-ip>
```

2. On the target host, restrict the file's permissions and run it:

```
chmod 600 ~/deploy.sh
sudo sh -c 'umask 077; exec bash ~/deploy.sh'
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.



3. When the script finishes, check the cluster:

```
sudo kubectl get nodes
sudo kubectl get pods -A
```

Wait until the node is `Ready` and every pod shows all its containers ready in the **READY** column.

Wait for initialization

A background job applies the evaluation license and creates sample data after the script finishes.

1. Find the job and follow its log:

```
sudo kubectl get jobs
sudo kubectl logs job/<job-name>
```

2. Wait for `One Click VM Demo Data Successfully Initialized.` in the log.

> **Caution:**
>
> Until initialization finishes, the setup wizard can show a **License** step with **Community** selected. Do not
> select Community. Wait, then reload the page.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Issue	Resolution
The installer container does not start	Make sure Docker is running and can reach Docker Hub.
`unknown command` at the `sds-cli` prompt	Enter the subcommand on one line, without `sds-cli`, a path or line-continuation characters.
The script fails immediately	Make sure `deploy.sh` contains only the script, starts with `#!/bin/bash`, and uses LF line endings without BOM.
The script fails later	The target host is not clean. Restore a clean snapshot or reinstall Ubuntu, then run the script again.
The platform URL does not resolve	Check the DNS record or hosts file. If more than 1 hour has passed, run the installer again to generate a new script.

Next step

Continue with Validate the deployment.

Validate the deployment

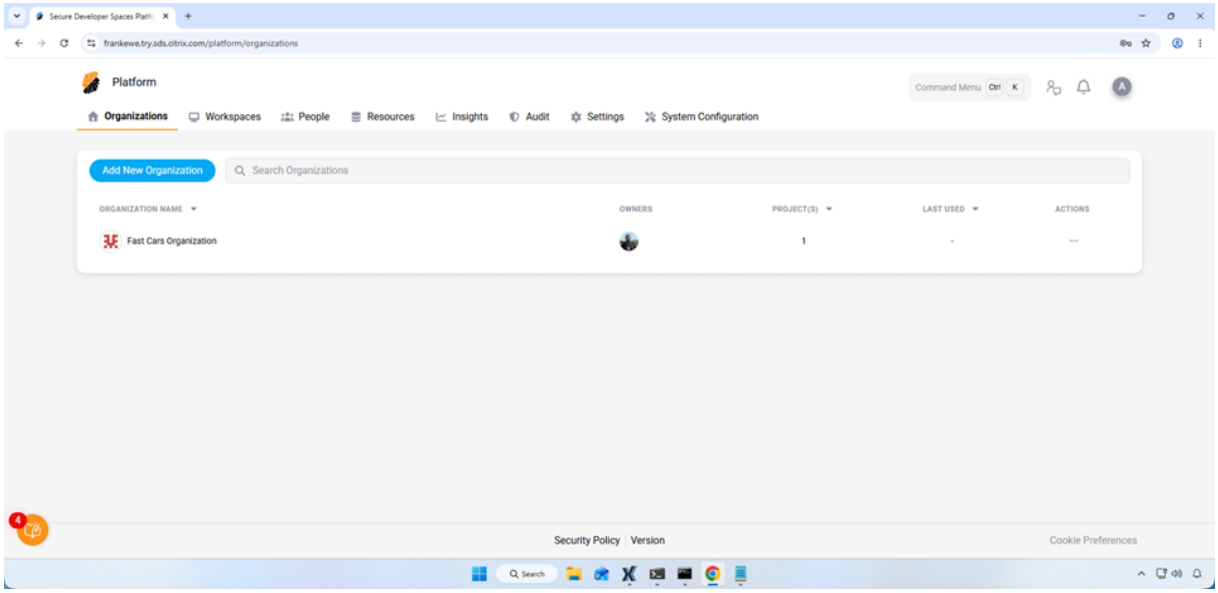
September 24, 2026

This page checks that a new Citrix SecurSpaces™ evaluation environment works, for cloud and local deployments.

Sign in

1. In a browser, go to `https://<name>.try.sds.citrix.com`. Use the hostname, not the IP address.
2. Enter the administrator email, select **Proceed**, and enter the administrator password from the installer.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



If sign-in fails with “invalid username or password”, wait 5 minutes and try again.

Complete the setup wizard

1. On **Register Sign-In Domain**, select **Skip**. To use your identity provider instead, see Identity providers.
2. On **Code Repository Applications**, select **Skip**. To connect a provider later, see Code repositories.
3. On **Organization**, enter an organization name and a project name, for example `Test Lab` and `Hello SecurSpaces`, and select **Finish**.

Check the license

Go to **Platform** > **System Configuration** > **License**, and check the license type, limits and expiry date. The evaluation license is valid for 6 months, and the TLS certificate for 90 days. See Licensing.

Create a workspace

1. In your project, select **Create your first Workspace**.
2. Select **Create Custom Workspace**.
3. On **Basic Info**, enter:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	Value
Name	`hello-securspaces`
Owner	Your user
Workspace image	Default Generic Image
Region	Default Region
Resources	Minimal

4. Select **Review and Launch**, and then **Launch**.
5. Wait for the status to change to **Running**.

Run code

1. In the **Access** column, select the VS Code icon. If the browser blocks the window, allow pop-ups for the console
   URL.
2. If the IDE asks whether to trust `/home/developer`, select **Manage** > **Trust**.
3. Open a terminal, and run:

```
mkdir -p ~/hello-securspaces
cd ~/hello-securspaces
printf 'print("Hello from Citrix SecurSpaces!")\n' > hello.py
python3 hello.py
sha256sum hello.py
```

4. Check that the output is `Hello from Citrix SecurSpaces!`, and note the checksum.

Check that files persist

1. On the **Workspaces** page, open the workspace's action menu, select **Pause**, and wait for **Paused**.
2. Select **Run**, and wait for **Running**.
3. Reopen the IDE, and run:

```
python3 ~/hello-securspaces/hello.py
sha256sum ~/hello-securspaces/hello.py
```

4. Check that the output and the checksum are the same as before.

The deployment works.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Next step

Continue with Manage the evaluation environment.

To learn SecurSpaces as a developer, follow Your first workspace.

Manage the evaluation environment

September 24, 2026

This page applies to cloud and local Citrix SecurSpaces™ evaluation environments. For platform-specific problems, see
the troubleshooting section of the page for your platform.

Restart the VM

1. Shut down or restart the VM from the operating system, for example with `sudo reboot`.
2. For a local VM, check that its IP address has not changed. If it has, update the DNS record or hosts file.
3. Connect to the VM, and wait until the node and every pod are ready:

   ```
   sudo kubectl get nodes
   sudo kubectl get pods -A
   ```

   This can take several minutes. Do not reinstall while services start.
4. If a workspace shows **Error**, select **Pause**, wait for **Paused**, and select **Run**.

Troubleshooting

Issue	Resolution
The setup wizard shows **License** with **Community** selected	Initialization has not finished. Wait, then reload the page. Do not select Community.
Sign-in fails right after deployment	Services are still starting. Wait 5 minutes and try again.
The platform URL resolves to the wrong address	The VM's IP address changed. Update the DNS record or hosts file.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Issue	Resolution
A workspace shows **Error** after a restart	Wait until every pod is ready, then pause and run the workspace.
The browser reports a certificate error after 90 days	The TLS certificate has expired. It is valid for 90 days.
The console reports an expired license after 6 months	The evaluation license has expired. Check **Platform** > **System Configuration** > **License**.

Remove the environment

1. Save any notes, results or screenshots you need.
2. Remove the platform:
   - **Cloud:** remove the resources the installer created.
   - **Local:** on the target host, run `sudo /usr/local/bin/k3s-uninstall.sh`, or delete the VM. On Hyper-V, also delete the VHDX file and checkpoints.
3. Remove any DNS records or hosts file entries you added.
4. On the installer host, delete the deployment script, the installer output and the deployment directory.

Related information

- Deploy an evaluation environment
- Deploy to a cluster

Deploy to a cluster

August 19, 2026

Citrix SecurSpaces™ runs on any Kubernetes-compatible cluster, including OpenShift. Preparing that cluster
is the part that differs between platforms; everything after it is the same.

> **Note**
>
> This page is for a production deployment. **If you only need to evaluate SecurSpaces, you do**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **not need a cluster at all** —the 1-Click VM runs the whole platform on a single virtual machine, and is the quickest route to a working environment for a proof of concept or a demonstration. See Deploy an evaluation environment. It is not intended for production use.

The deployment sequence

1. **Prepare your infrastructure** —a Kubernetes cluster, a container registry, and a MongoDB database. This is platform-specific; use the page for your platform below.
2. **Install with the Strong Installer CLI** —pull the Helm chart, push images to your registry, and create the configuration file.
3. **Deploy to the cluster** — install the ingress controller, upload the TLS secret, and apply the Helm chart.
4. **Configure the platform** —identity, repositories, and access. See the table at the end of the install page.

> **Important:**
>
> Every deployment needs an ingress controller, and the chart installs one for you only if you use NetScaler CPX. Decide which controller you are using before you start, because it affects the cluster you build and how workspace SSH is exposed. See Ingress.

Choose your platform

Platform	Page	Notes
Amazon EKS	Amazon EKS	EKS, ECR, and MongoDB Atlas
Azure AKS	Azure AKS	AKS, ACR, and MongoDB Atlas or Cosmos DB
Google GKE	Google GKE	GKE Standard, Artifact Registry, and MongoDB Atlas. Autopilot is not supported

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Platform	Page	Notes
Red Hat OpenShift	Red Hat OpenShift	Self-contained guide, including the OpenShift-specific wildcard route policy
Nutanix Kubernetes Platform	Nutanix Kubernetes Platform	NKP 2.17 or later, with NetScaler CPX ingress

Self-hosted clusters and other managed distributions follow the same sequence: create the cluster, provide a registry and a database, then install. See System requirements for the supported Kubernetes versions and sizing.

Before you start

Requirement	Detail
Domains	A host domain, and a wildcard subdomain for proxy support such as `*.proxy.example.com`
TLS	Certificates for both domains
Database	MongoDB, hosted or in-cluster. See The SecurSpaces database
Registry	A container registry you control, which the installer pushes images to
Sizing	Node pools for the service and workspace tiers. See Sizing
Licence	A SecurSpaces licence. See Licensing

For an evaluation rather than a production deployment, the 1-Click VM is considerably faster. See Deploy an evaluation environment.

Related information

- Architecture
- System requirements

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- NGINX
- Multiple regions
- Upgrade

Amazon EKS

August 19, 2026

This page prepares the AWS infrastructure Citrix SecurSpaces™ needs: a Kubernetes cluster to run on, a
container registry to pull images from, and a MongoDB database to store platform state.

When you have finished, continue with
Install with the Strong Installer CLI, which is the
same for every platform.

Prerequisites

- An AWS account with permissions to create EKS, ECR, and networking resources.
- A MongoDB Atlas subscription.
- Two domain names: a regular one such as `example.com`, and a wildcard subdomain for proxy support such as
  `*.proxy.example.com`.
- TLS certificates for those domains.
- An identity provider, such as Okta, if you intend to use SAML.
- An ingress controller, or a decision about which one to use. SecurSpaces cannot serve traffic without
  one, and only NetScaler CPX is deployed by the chart. See
  Ingress.

For sizing and supported versions, see
System requirements.

Install the following tools:

- AWS Command Line Interface
- eksctl
- kubectl

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Create an EKS cluster

1. Configure the AWS CLI with your credentials:

   ```
   aws configure
   ```

   Enter your AWS access key ID, secret access key, default region, and output format when prompted.

The reference sizing for around 500 developers is two `m7i.2xlarge` service nodes (8 vCPU, 32 GiB) and an
autoscaling workspace pool of `m7i.4xlarge` (16 vCPU, 64 GiB). See
Sizing.

1. Create the cluster:

   ```
   eksctl create cluster \
     --name $MY_EKS_CLUSTER_NAME \
     --region $AWS_REGION \
     --nodegroup-name standard-nodes \
     --node-type m7i.2xlarge \
     --nodes 2 \
     --nodes-min 2 \
     --nodes-max 4 \
     --managed
   ```

2. Add a node group for workspaces, with autoscaling. Workspaces are the elastic part of the deployment, so
   give this group room to grow:

   ```
   eksctl create nodegroup \
     --cluster $MY_EKS_CLUSTER_NAME \
     --region $AWS_REGION \
     --name workspace-nodes \
     --node-type m7i.4xlarge \
     --nodes 2 \
     --nodes-min 2 \
     --nodes-max 10 \
     --managed
   ```

3. Verify the connection:

   ```
   kubectl get nodes
   ```

Create an Amazon ECR registry

SecurSpaces pulls its service images and your workspace images from a registry you control. The installer

© 1997–2026 Citrix Systems, Inc. All rights reserved.

pushes them there in the next stage.

1. Create a repository for the platform images:

```
aws ecr create-repository \
  --repository-name strong-network \
  --image-scanning-configuration scanOnPush=true
```

2. Create a repository for workspace images:

```
aws ecr create-repository \
  --repository-name ws-images \
  --image-scanning-configuration scanOnPush=true
```

3. Authenticate Docker to the registry:

```
aws ecr get-login-password --region $AWS_REGION | docker login --
    username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.
    $AWS_REGION.amazonaws.com
```

4. For a multi-region deployment, set up cross-region replication:

```
aws ecr put-replication-configuration \
  --replication-configuration '{
 "rules":[{
 "destinations":[{
 "region":"us-west-2","registryId":"$AWS_ACCOUNT_ID" }
 ] }
 ] }
 '
```

Create a MongoDB database

SecurSpaces stores its platform state in MongoDB. For what it holds and how large to expect it to grow, see
The SecurSpaces database.

1. In MongoDB Atlas, open the database page and select **Create**.

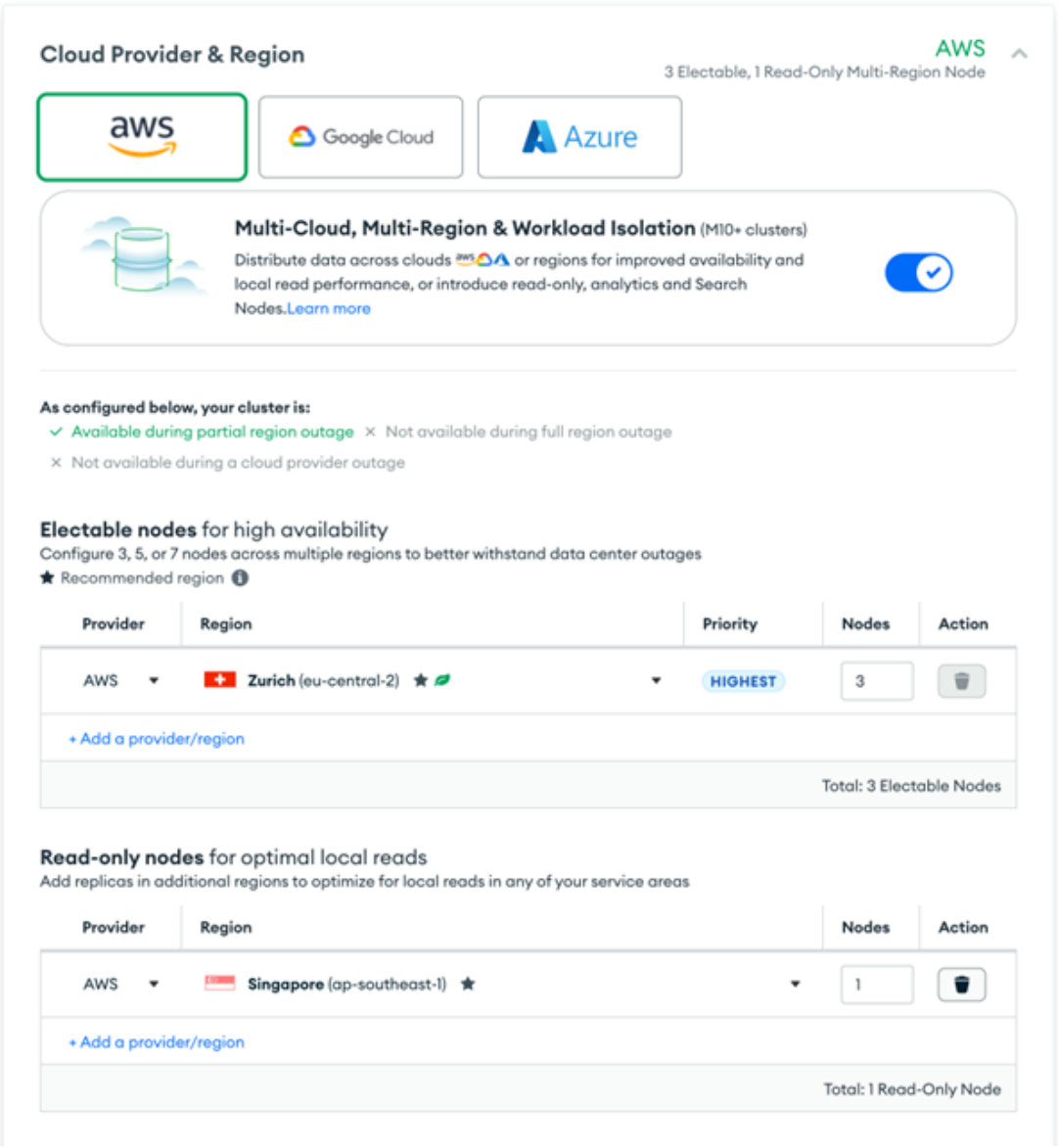


2. Configure the deployment and select **Create Deployment**:

   - Select **AWS** as the cloud provider.

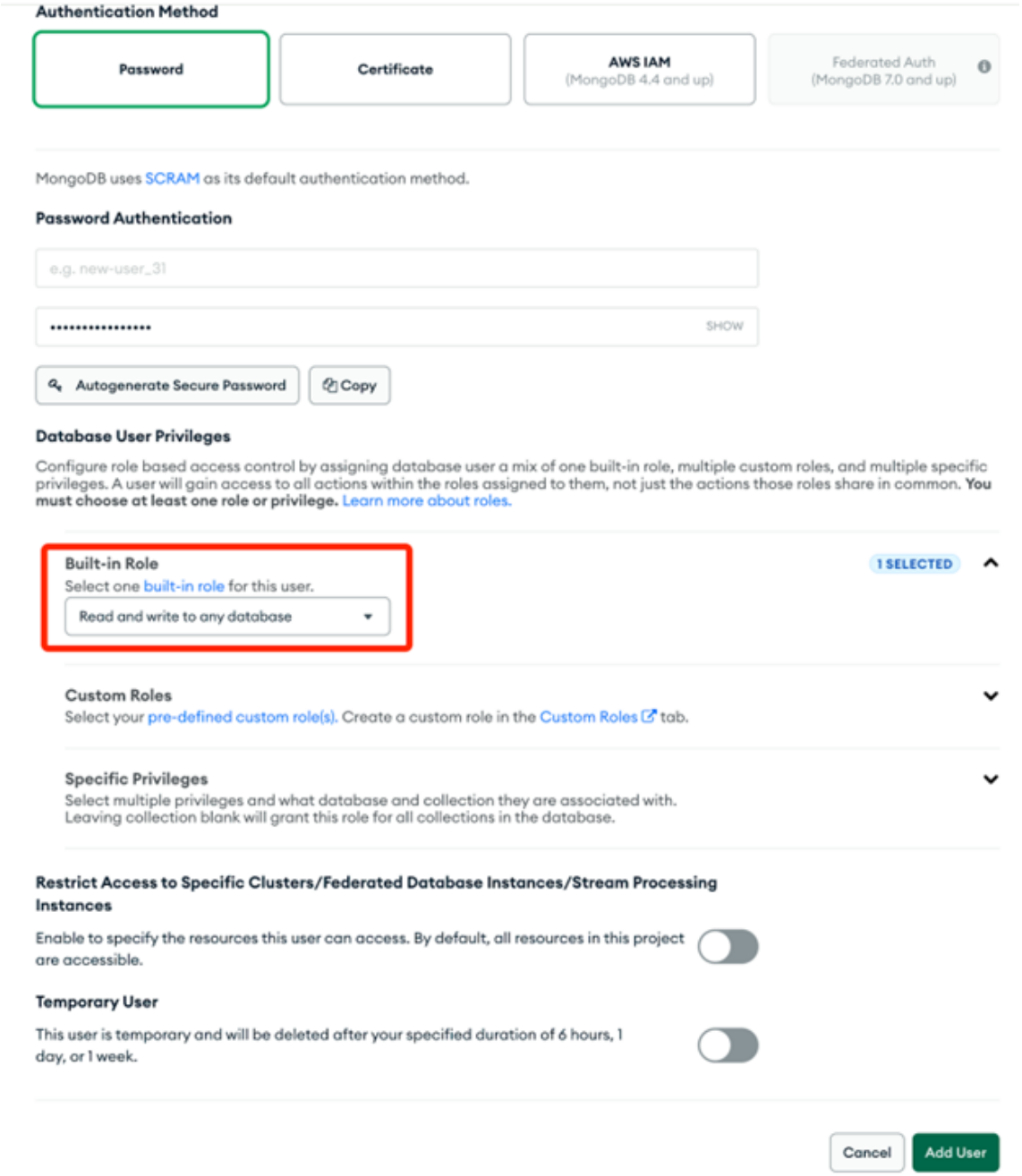
© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Choose the same region as the EKS cluster.
- Select the M10 tier with autoscaling.



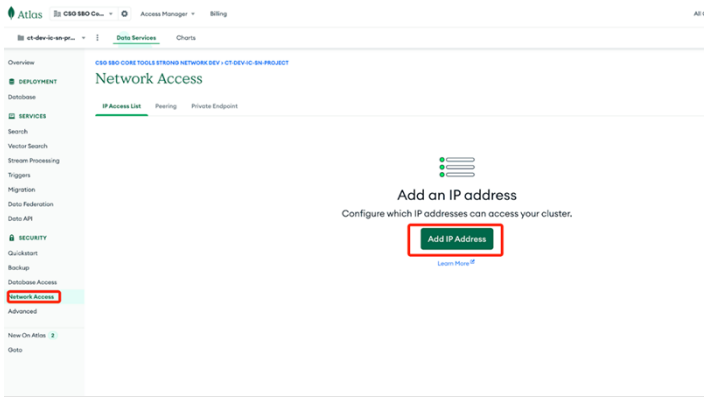
3. On the **Database Access** page, add a user.

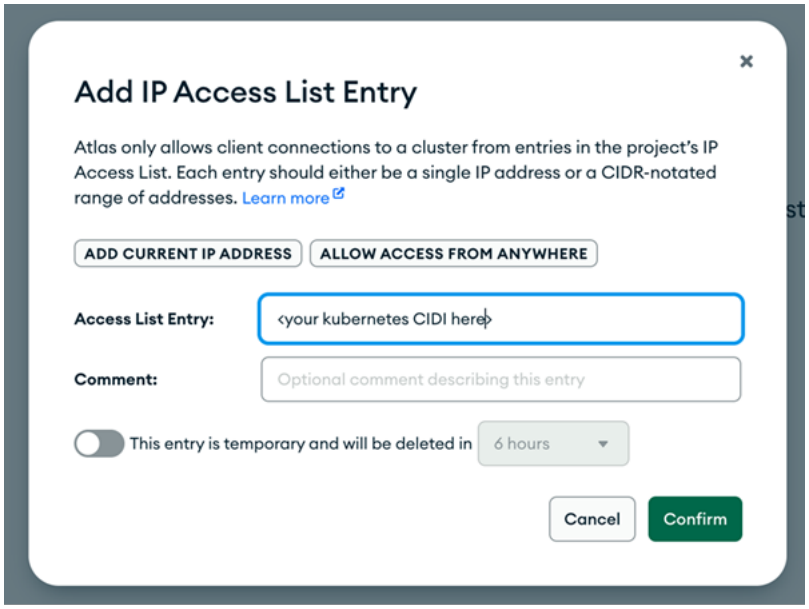
© 1997–2026 Citrix Systems, Inc. All rights reserved.



4. On the **Network Access** page, allow your EKS cluster to reach the database. Add the CIDR blocks for the
   cluster VPC, or peer the VPC with Atlas, which is the more secure option.

© 1997–2026 Citrix Systems, Inc. All rights reserved.





You can also run MongoDB inside the cluster instead of using Atlas. See Configure an external database.

Next step

Your AWS infrastructure is ready. Continue with Install with the Strong Installer CLI.

During that installation you also set up the ingress controller. If you have not chosen one, do that first —see Ingress.

Related information

- Kubernetes deployment
- System requirements

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Ingress
- GPU-enabled workspaces

Azure AKS

August 19, 2026

This page prepares the Azure infrastructure Citrix SecurSpaces™ needs: a Kubernetes cluster to run on, a
container registry to pull images from, and a MongoDB database to store platform state.

When you have finished, continue with
Install with the Strong Installer CLI, which is the
same for every platform.

Prerequisites

- An Azure subscription with permissions to create AKS, ACR, and networking resources.
- A MongoDB Atlas subscription, or Azure Cosmos DB for MongoDB.
- Two domain names: a regular one such as `example.com`, and a wildcard subdomain for proxy support such as
  `*.proxy.example.com`.
- TLS certificates for those domains.
- An identity provider, such as Okta, if you intend to use SAML.
- An ingress controller, or a decision about which one to use. SecurSpaces cannot serve traffic without
  one, and only NetScaler CPX is deployed by the chart. See
  Ingress.

For sizing and supported versions, see
System requirements.

Install the following tools:

- Azure CLI
- kubectl

Create an AKS cluster

1. Create a resource group. This is the logical container for the deployment, and its location determines

© 1997–2026 Citrix Systems, Inc. All rights reserved.

where resources run when you do not specify one:

```
az group create --name $MY_RESOURCE_GROUP_NAME --location $REGION
```

2. Create the cluster. The following example creates a cluster with a system-assigned managed identity:

```
az aks create \
  --resource-group $MY_RESOURCE_GROUP_NAME \
  --name $MY_AKS_CLUSTER_NAME \
  --node-count 1 \
  --generate-ssh-keys
```

The example creates a single node. Size the cluster for your expected load: the reference sizing for
around 500 developers is two `Standard_D8as_v6` service nodes (8 vCPU, 32 GiB) and an autoscaling
workspace pool of `Standard_D16as_v6` (16 vCPU, 64 GiB). See Sizing.

3. Download the credentials and point `kubectl` at the cluster:

```
az aks get-credentials --resource-group $MY_RESOURCE_GROUP_NAME --
    name $MY_AKS_CLUSTER_NAME
```

4. Verify the connection:

```
kubectl get nodes
```

Create an Azure Container Registry

SecurSpaces pulls its service images and your workspace images from a registry you control. The installer
pushes them there in the next stage.

1. Create the registry. The name must be unique within Azure and contain 5 to 50 lowercase alphanumeric
characters:

```
az acr create --resource-group myResourceGroup --name
    mycontainerregistry --sku Basic
```

2. Create a service principal with the `AcrOwner` role. The command returns the `appId`, `displayName`,
`password`, and `tenant`:

```
acr_resource_id=$(az acr show --name <acr-name> --query id --
    output tsv)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
az ad sp create-for-rbac --name <service-principal-name> --role
    AcrOwner --scopes ${
 acr_resource_id }
  --sdk-auth
```

3. Verify the service principal by signing in to the registry:

```
docker login <acr-name>.azurecr.io --username <service-principal-
    id>
```

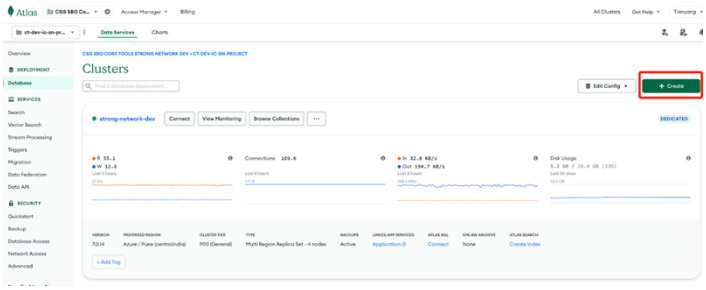
4. Attach the registry to the cluster, so AKS can pull from ACR without further credentials:

```
az aks update --name myAKSCluster --resource-group myResourceGroup
     --attach-acr <acr-name>
```

Create a MongoDB database

SecurSpaces stores its platform state in MongoDB. For what it holds and how large to expect it to grow, see
The SecurSpaces database.

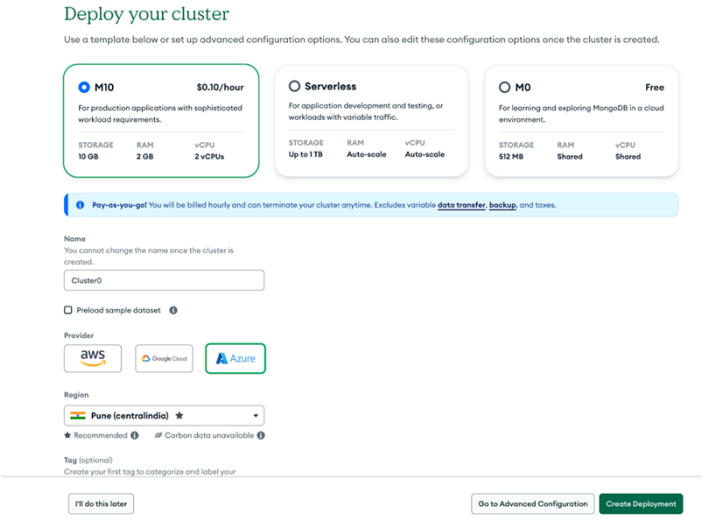
1. In MongoDB Atlas, open the database page and select **Create**.



2. Configure the deployment and select **Create Deployment**:

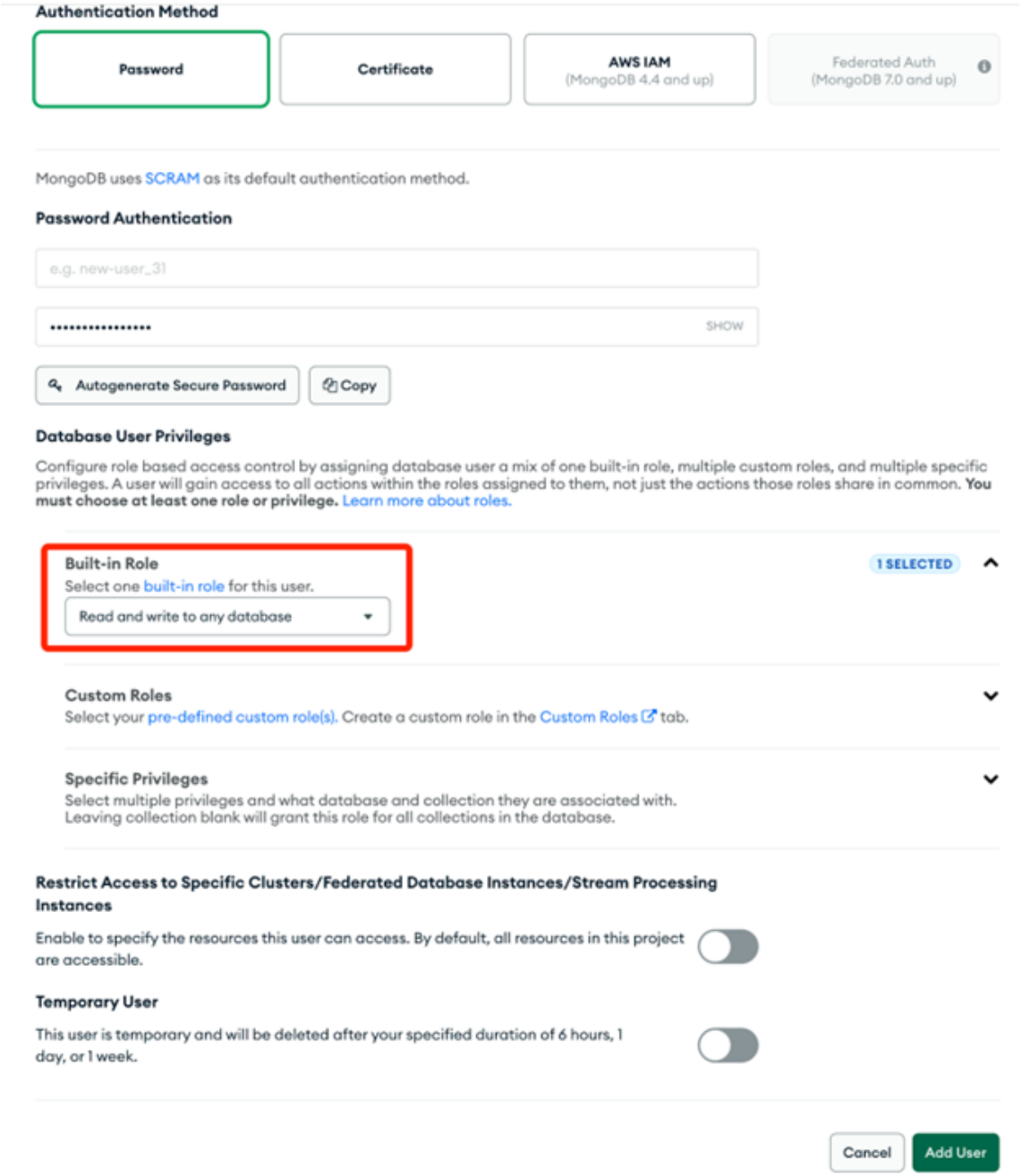
   - Select **Azure** as the cloud provider.
   - Choose the same region as the AKS cluster.
   - Select the M10 tier with autoscaling.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



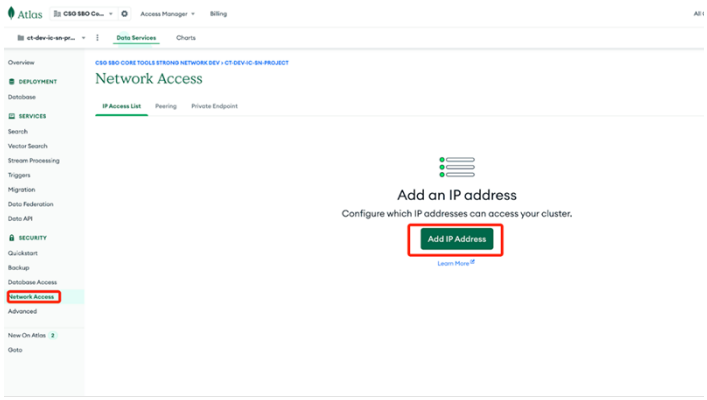
3. On the **Database Access** page, add a user.

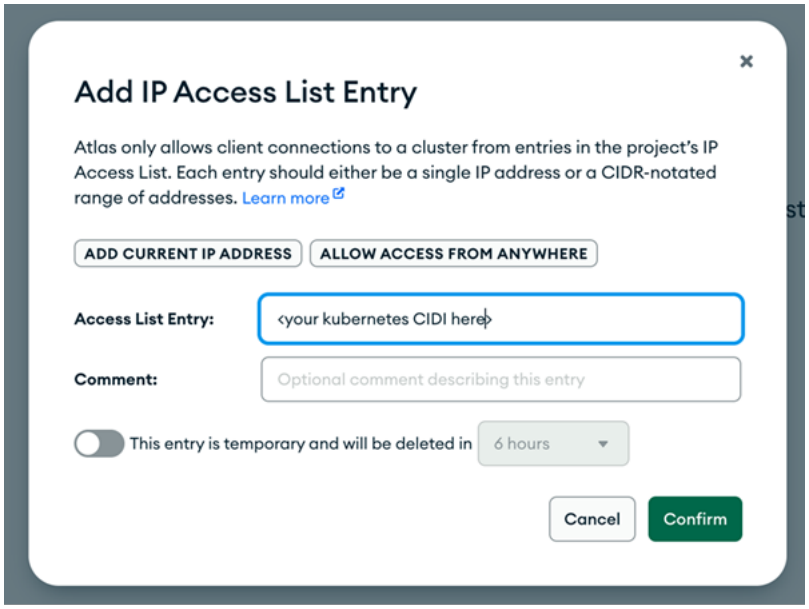
© 1997–2026 Citrix Systems, Inc. All rights reserved.



4. On the **Network Access** page, allow your AKS cluster to reach the database. Add the cluster's outbound
   IP ranges, or use private endpoints, which is the more secure option.

© 1997–2026 Citrix Systems, Inc. All rights reserved.





You can also run MongoDB inside the cluster instead of using a hosted service. See Configure an external database.

Next step

Your Azure infrastructure is ready. Continue with Install with the Strong Installer CLI.

During that installation you also set up the ingress controller. If you have not chosen one, do that first —see Ingress.

Related information

- Kubernetes deployment
- System requirements

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Ingress
- GPU-enabled workspaces on AKS

Google GKE

October 01, 2026

This page prepares the Google Cloud infrastructure Citrix SecurSpaces™ needs: a Kubernetes cluster to run on,
a container registry to pull images from, and a MongoDB database to store platform state.

When you have finished, continue with
Install with the Strong Installer CLI, which is the
same for every platform.

> **Important:**
>
> **Use GKE Standard, not GKE Autopilot.** SecurSpaces does not install on Autopilot clusters — the
> installation fails. Autopilot is the default mode when you create a cluster in the Google Cloud console, so
> select Standard explicitly.
>
> **NetScaler CPX does not currently work on GKE.** This is a known issue, a fix is in progress, and there is
> no date yet. Contact Citrix for the current status before planning a deployment that depends on it, and use
> one of these instead:
>
> - NGINX —the closest substitute, and the
>   option this documentation covers in full
> - Kubernetes Gateway API —supported by the chart, with the
>   GKE-specific setting described below

Prerequisites

- A Google Cloud project with permissions to create GKE, Artifact Registry, and networking resources.
- A MongoDB Atlas subscription if you use Atlas, or a plan to deploy internal Percona MongoDB.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Two domain names: a regular one such as `example.com`, and a wildcard subdomain for proxy support such as `*.proxy.example.com`.
- TLS certificates for those domains.
- An identity provider, such as Okta, if you intend to use SAML.
- An ingress controller, or a decision about which one to use. SecurSpaces cannot serve traffic without one, and only NetScaler CPX is deployed by the chart. See Ingress.

For sizing and supported versions, see System requirements and Sizing.

Install the following tools:

- Google Cloud CLI
- kubectl

Create a GKE cluster

1. Sign in and select your project:

```
gcloud auth login
gcloud config set project $PROJECT_ID
```

The reference sizing for around 500 developers is two `n2-standard-8` service nodes (8 vCPU, 32 GiB) and an autoscaling workspace pool of `n2-standard-16` (16 vCPU, 64 GiB). See Sizing.

1. Create a Standard cluster with a node pool for the platform services:

```
gcloud container clusters create $CLUSTER_NAME \
  --project $PROJECT_ID \
  --region $REGION \
  --num-nodes 2 \
  --machine-type n2-standard-8 \
  --enable-ip-alias
```

`gcloud container clusters create` produces a Standard cluster. Autopilot requires the separate `create-auto` command, which must not be used.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Add a node pool for workspaces, with autoscaling. Workspaces are the elastic part of the deployment, so
   give this pool room to grow:

```
gcloud container node-pools create workspace-nodes \
  --cluster $CLUSTER_NAME \
  --project $PROJECT_ID \
  --region $REGION \
  --machine-type n2-standard-16 \
  --num-nodes 2 \
  --enable-autoscaling \
  --min-nodes 2 \
  --max-nodes 10
```

3. Fetch cluster credentials and verify the connection:

```
gcloud container clusters get-credentials $CLUSTER_NAME --project
   $PROJECT_ID --region $REGION
kubectl get nodes
```

   The installer runs this command for you during an upgrade, using the cluster name, project, and region in
   your configuration file.

Create an Artifact Registry repository

SecurSpaces pulls its service images and your workspace images from a registry you control. The installer
pushes them there in the next stage.

1. Enable the API and create a repository for the platform images:

```
gcloud services enable artifactregistry.googleapis.com --project
   $PROJECT_ID

gcloud artifacts repositories create strong-network \
  --repository-format=docker \
  --location=$REGION \
  --project=$PROJECT_ID
```

2. Create a repository for workspace images:

```
gcloud artifacts repositories create ws-images \
  --repository-format=docker \
  --location=$REGION \
  --project=$PROJECT_ID
```

3. Authenticate Docker to the registry:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
gcloud auth configure-docker $REGION-docker.pkg.dev
```

The installer recognizes an Artifact Registry or Container Registry URL —anything containing `docker.pkg.dev` or `gcr.io` —and authenticates to it with the Google Cloud CLI. You can sign in as a user
account, or supply a service account key with `--service-account-file-path`, which is the usual choice for a
pipeline. See
SDS CLI reference.

Create a MongoDB database

SecurSpaces stores its platform state in MongoDB. For what it holds and how large to expect it to grow, see
The SecurSpaces database.

MongoDB Atlas

1. In MongoDB Atlas, open the database page and select **Create**.
2. Configure the deployment and select **Create Deployment**:
   - Select **Google Cloud** as the cloud provider.
   - Choose the same region as the GKE cluster.
   - Select the M10 tier with autoscaling.
3. Create a database user, and note the user name and password. The installer asks for them.
4. Allow access from your cluster's egress addresses in **Network Access**.
5. Copy the connection string from **Connect > Drivers**.

Internal Percona MongoDB with keyless GCS backups

To run MongoDB inside GKE, select Percona during installation (`platform.deployPerconaMongoDB` unset or
`true`). The updated chart deploys Percona Operator 1.23.0, MongoDB 7.0.37-20, and PBM 2.15.0.

For keyless backups to Google Cloud Storage:

- Enable Workload Identity Federation for GKE and the GKE metadata server on the node pools running
  the database and operator pods.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Create a backup bucket and grant access directly to both the database and operator Kubernetes service
  accounts. The operator also needs access for retention cleanup.
- Configure `perconaServerMongoDB.serviceAccount` and `perconaServerMongoDB.backup` in Helm values.
  Use native `gcs` storage without `credentialsSecret`; no service account JSON key or HMAC key is needed.

Backups are disabled by default. For setup commands, schedules, PITR, retention, and verification, see
GCS Workload Identity.

For an existing SDS 2026.9 deployment, first run the installer-supplied Percona upgrade script to reach 1.22.0 and apply the 1.23 CRDs, then run Helm to complete the move to 1.23.0. See
Upgrade.

If you use the Kubernetes Gateway API

GKE needs one extra setting when SecurSpaces uses Gateway API resources rather than Ingress resources. Set
both of these in your configuration file:

```yaml
platform:
  useGatewayAPI: true
  gatewayAPI:
    gatewayClassName: <your-gateway-class>
    enableHealthCheckPolicy: true
```

`enableHealthCheckPolicy` creates the `networking.gke.io/v1` HealthCheckPolicy resources that GKE requires to
health-check the backends. It applies to GKE only.

Most deployments use an ingress controller instead. See
Ingress.

Next step

Your Google Cloud infrastructure is ready. Continue with
Install with the Strong Installer CLI.

During that installation you also set up the ingress controller. If you have not chosen one, do that first —see Ingress.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Kubernetes platforms
- System requirements
- Sizing
- Ingress

Deploy SecurSpaces on Red Hat OpenShift

September 29, 2026

The Citrix SecurSpaces™ platform provides a secure and productive cloud development environment that can be deployed on public, private clouds, and self-hosted servers. It also works in full air-gapped modes, such as high-security settings.
The primary purpose of SecurSpaces platform is to provide secure, cloud development environments (CDEs) that boost developer productivity while ensuring enterprise-level security. It allows organizations to streamline the provisioning and management of coding environments, improving efficiency and collaboration among internal and external teams. By centralizing development resources and integrating automated security features, the platform reduces the risk of data leaks and intellectual property theft, enabling safe remote work and supporting DevSecOps practices.
Specifically, typical usage scenarios encompass IT efficiency, developer productivity, and governance of the development process.
This deployment guide provides the steps to deploy the SecurSpaces to an existing Red Hat OpenShift cluster. The CDEs can be accessed through cloud or local integrated development environments (IDEs), offering a secure and isolated environment for developers.

This guide covers preparing an OpenShift cluster and installing the SecurSpaces platform on it. Unlike the
EKS and AKS pages, it is self-contained, because OpenShift's route and security-context model changes several
of the installation steps.

Once the platform is running, see
Kubernetes for what to configure next.

Prerequisites

The SecurSpaces CDE platform deployment has the following prerequisites:

- Container registry to which you have push permissions

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- An up-and-running OpenShift cluster
- Basic tools for working with that OpenShift cluster: **oc** and **kubectl**
  You should already be authenticated in both **oc** and **kubectl** with your OpenShift cluster.

Deployment Steps

Obtain a SecurSpaces platform license

You need a valid license to use the SecurSpaces platform. Please contact your ATS or Solution Architect to request that one be issued to you.
To issue a license, we only need to know which URL will be used to access the SecurSpaces platform. In case of OpenShift, we need to find out the cluster domain first, which we can easily do in two ways:

1. From the web console URL: The cluster domain is part of the OpenShift web console address. If you access the console at `https://console-OpenShift-console.apps.cluster.example.com`, then your cluster domain is `apps.cluster.example.com`
2. Using the OpenShift CLI (oc):

```
oc get ingresses.config/cluster -o jsonpath='{
  .spec.domain }
  '
```

   - You will get an output like: `apps.sm9kda46je6e9a9c7b.swedencentral.aroapp.io`
   - Once you have the cluster domain, the final SecurSpaces platform domain will be in the form of: `sn.<cluster-domain>`
   - In our case, it will look like the following: `sn.apps.<CLUSTER_ID>.<REGION>.aroapp.io`

Now that we have the domain, we will register it with the SecurSpaces License server.

Check the Wildcard policy of your OpenShift cluster

To be able to create and use workspaces on the SecurSpaces platform, which have dynamic routes, you need to make sure that the wildcard policy of your OpenShift cluster routing is set to **WildcardsAllowed**.

To do that, you can follow these steps:

- Find out the name of your ingress controller by running:

```
oc get ingresscontroller -n openshift-ingress-operator -o jsonpath='
   {
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  .items[*].metadata.name }
'
```

- If you made a default installation of an OpenShift cluster and haven't changed any of the settings, the name will most likely be default.
- Check if your ingress controller has a WildcardsAllowed policy already set, using the name of the controller obtained from the previous step. In our case, it is default, so the command looks like:

```
  oc get ingresscontroller default -n openshift-ingress-operator -o
      jsonpath='{
.spec.routeAdmission.wildcardPolicy }
'
```

- If the output is WildcardsAllowed, then everything is set, and you can go to the next step of the guide.
- If the output is empty, it means we need to set the policy manually. This can be done by editing the configuration of that particular ingress controller (named default) by executing:

```
  oc patch ingresscontroller default -n openshift-ingress-operator --
      type=merge -p '{
"spec":{
"routeAdmission":{
"wildcardPolicy":"WildcardsAllowed" }
 }
 }
'
```

- You can now check that the previous command was successful by executing and getting **WildcardsAllowed** as a response:

```
  oc get ingresscontroller default -n openshift-ingress-operator -o
      jsonpath='{
.spec.routeAdmission.wildcardPolicy }
'
```

- If you want to see the complete ingress controller configuration, you can execute:

```
  oc get ingresscontroller default -n openshift-ingress-operator -o
      yaml
```

More info on the subject of Ingress controllers and wildcard policies in particular can be found at: https://docs.redhat.com/en/documentation/OpenShift_container_platform/4.18/html/operator_apis/ingresscontroller-operator-OpenShift-io-v1#spec-routeadmission

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Create a Kubernetes namespace for your deployment

Select the name of the namespace that you want to use for SecurSpaces deployment. It is advised not to reuse existing namespaces for deploying our platform. For this guide, we will use the name SecurSpaces.

- First, you need to check if that namespace already exists by running:

```
kubectl get namespaces | grep strong-network
```

- If it does not exist, you can create it by running:

```
kubectl create namespace strong-network
```

Run the SecurSpaces installer

You can start the installation process by running the following command from the console of your device:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared strongnetwork/strong_installer:2026.9.4
```

After this is executed and the installer image is pulled and run, you can run the installer. The installer will create a configuration file that should be used to upgrade the product in the future. It will also download the Helm chart to the shared host directory and push SecurSpaces images to your container registry.

Since we provide a large number of pre-built workspace images by default, which occupy a significant amount of space, you can reduce the number of images that we pull or push by editing the list of workspace images that we will pull or push, or by not downloading them at all. This could also be useful if you (the client) plan to use your workspace images, so you do not need SecurSpaces' prebuilt ones.

In case you need only specific workspace images but do not want to download the rest, you can edit the **workspace_images.txt** file and remove the ones that you do not need.

Then you can run:

```
sds-cli install
```

You will have to go through a wizard that helps you create the YAML configuration file.

The example below was captured on an earlier release. The current installer asks fewer questions — it no

© 1997–2026 Citrix Systems, Inc. All rights reserved.

longer prompts for the workspace domain, the database encryption key, the JWT secret, node affinities, or
two-factor authentication —and names the chart and configuration file after the version you install.

```
Enter the main domain of the platform (e.g, strong-network.example.com)
    : `sn.apps.<CLUSTER_ID>.<REGION>.aroapp.io`
The default domain used to access workspaces is: proxy.sn.apps.<
    CLUSTER_ID>.<REGION>.aroapp.io`
Do you want to change it? (y/n): n
Set the email of the platform admin: admin:admin@example.com
Set password for admin (leave empty to autogenerate):
Define the base64 database encryption key (leave empty to autogenerate)
    :
Define the JWT secret (leave empty to autogenerate):
Do you want to enable node affinities for workspaces and strong network
     services? (y/n): n
Use two-factor authentication for users? (y/n): n
Installing version 2025.5.0...
```

In case you get an error in the terminal like the following:

```
Failed to fetch access token with error: permission denied, try using
 a service account...
```

This means that your domain is not registered correctly on the SecurSpaces License server, and therefore, we cannot automatically generate the access token for our container registry.

If you did not encounter any problems during the wizard, your whole wizard will look something like the output below.

```
root@0f7ca7f89e78:/strong-network# sds-cli install
Enter the main domain of the platform (e.g, strong-network.example.com)
    : sn.apps.<CLUSTER_ID>.<REGION>.aroapp.io
The default domain used to access workspaces is: proxy.sn.apps.<
    CLUSTER_ID>.<REGION>.aroapp.io
Do you want to change it? (y/n): n
Set the email of the platform admin: admin:admin@example.com
Set password for admin (leave empty to autogenerate):
Define the base64 database encryption key (leave empty to autogenerate)
    :
Define the JWT secret (leave empty to autogenerate):
Do you want to enable node affinities for workspaces and Citrix Secure
    Developer Spaces services? (y/n): n
Use two-factor authentication for users? (y/n): n
Installing version 2026.4.5...
 Authenticating... Done.
 Finding best source registry... Selected EU region.
Pulling helm chart version 2026.4.5..
File ninjahchart-2026.4.5.tgz downloaded to /strong-network/shared/
    ninjahchart-2026.4.5.tgz
Destination Registry URL:<YOUR_REGISTRY>.azurecr.io
Detected Azure Registry Type..
Select Type For Authentication:
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        [1] User Account
        [2] Service Principal
Please Enter Your Numeric Choice: 2
Enter service principal ID: <YOUR_SERVICE_PRINCIPAL_ID>
Enter service principal password: Authentication successful.
src: europe-docker.pkg.dev/strong-network-release/images/frontend
   :2026.4.5
dst: http://<YOUR_REGISTRY>.azurecr.io/frontend:2026.4.5
Getting image source signatures
Copying blob fb6ab0edbcbd done   |
Copying blob ab9385d5595c done   |
Copying blob 0c1c1add5fb1 done   |
Copying blob 4f4fb700ef54 skipped: already exists
Copying config 3564c26c52 done   |
Writing manifest to image destination
 Finishing up... Done.
Run your deployment like this:
    helm upgrade --install \\
            --namespace default \\
            --create-namespace \\
            -f "./config_2026.4.4.yaml" \\
            release ./ninjahchart-2026.4.5.tgz
root@0f7ca7f89e78:/strong-network#`
```

Add Docker registry secrets to Kubernetes

Docker images for the SecurSpaces platform itself and later for workspace images need to be hosted in a suitable location. This is why you need to provide your Container registry (or Artifactory), because the SecurSpaces installer is going to pull the images from SecurSpaces private GCP Artifactory and push them to your registry. SecurSpaces container registry is likely not going to be public, and this is why you need to authenticate with it. The credentials for this authentication must be present during the SecurSpaces platform installation so that images can be pulled for each of the SecurSpaces service pods.
Once you have the credentials ready, you can add them by running:

```
kubectl create secret docker-registry acr-secret   --namespace=strong-
   network   --docker-server=<YOUR_REGISTRY>.azurecr.io    --docker-
   username=<YOUR_SERVICE_PRINCIPAL_ID>-token   --docker-password=<
   YOUR_REGISTRY_PASSWORD>
```

This creates a new Docker registry secret named '**acr-secret**' in the '**strong-network**' namespace.

You can verify that the secret is successfully added by executing:

```
`kubectl get secrets | grep acr-secret`
```

Now you need to manually edit the YAML configuration that you got as a result of running the Strong installer and add the following line in the platform section: `imagePullSecrets: acr-secret`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set proper Security Context Constraints

Since the OpenShift security model differs from the standard Kubernetes security model, you must apply additional policies; otherwise, SecurSpaces services or workspaces will not start properly. You need to set policies for both service deployments and workspaces that will be made on the platform.

Set policies for the service accounts of platform services Execute the instructions below:

```
oc adm policy -n strong-network add-scc-to-user privileged -z release-central-service
oc adm policy -n strong-network add-scc-to-user privileged -z release-coordinator
oc adm policy -n strong-network add-scc-to-user privileged -z release-proxy
oc adm policy -n strong-network add-scc-to-user privileged -z release-workspace-api
oc adm policy -n strong-network add-scc-to-user privileged -z release-mongodb
oc adm policy -n strong-network add-scc-to-user privileged -z release-frontend
oc adm policy -n strong-network add-scc-to-user privileged -z release-cron-service-account
```

This will add the elevated permissions to all platform services'service accounts that SecurSpaces uses.

The `-z` flag indicates that what follows is a service account name, for example, release-proxy.

Create a new service account, its policy, and assign it

Create a file named **workspaces-policy.yaml**:

```
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: anyuid provides all features of the
       restricted SCC
      but allows users to run with any UID and any GID.
  creationTimestamp: "2024-08-07T12:25:28Z"
  generation: 1
  name: strongnetwork-scc
  resourceVersion: "998061"
  uid:
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- NET_ADMIN
- NET_RAW
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups:
- system:cluster-admins
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:
- MKNOD
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users: []
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret
```

Apply it by running:

```
kubectl apply -f workspaces-policy.yaml
```

Now you need to check if a service account named **statefulset-sa** already exists:

```
oc get serviceaccounts | grep statefulset-sa
```

Service account already exists

If the service account already exists, you can check if it is already assigned the **strongnetwork-scc SecurityContextConstraints**. You can do so by running:

```
`oc get rolebindings,clusterrolebindings --all-namespaces -o wide |
   grep statefulset-sa`
```

You can expect to get an output like the one below, which would mean that it is using a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

strongnetwork-scc:
If this is the case, you do not need to do any extra steps regarding the service account and its policies.

Service account does not exist

If the service account does not exist, you need to create one and assign the **strongnetwork-scc** policy to it:

```
oc create serviceaccount statefulset-sa -n strong-network
oc adm policy add-scc-to-user strongnetwork-scc -z statefulset-sa
```

Editing the YAML configuration

Finally, you need to manually edit the YAML configuration that you got as a result of running the Strong installer.
You need to add the following lines in the platform section:

```
useOpenShift: true
```

You need to add the following lines in the region section:

```
clusterConfig:
        privilegedWorkspaces: true
        customServiceAccount: statefulset-sa
```

Final YAML

After all the edits made in previous steps, the YAML configuration should look like this:

```
mongodb:
    auth:
        replicaSetKey: <MONGODB_REPLICA_SET_KEY>
        password: <MONGODB_PASSWORD>
        rootPassword: <MONGODB_ROOT_PASSWORD>
platform:
    imageTag: 2026.9.4
    dockerRegistry: <YOUR_REGISTRY>.azurecr.io/
    workspaceRegistry: <YOUR_REGISTRY>.azurecr.io/ws-images/
    publicRegistryLatestTag: 2.3.9
    sidecarTag: 2026.9.4
    hostName: sn.apps.<CLUSTER_ID>.<REGION>.aroapp.io
    centralProxyHostname: proxy.sn.apps.<CLUSTER_ID>.<REGION>.aroapp.io
    userAdminEmail: admin@example.com
    userAdminPassword: <ADMIN_PASSWORD>
    secretKeyReposB64: <BASE64_ENCRYPTION_KEY>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    jwtSecret: <JWT_SECRET>
    sslCertificateSecret: ""
    sslCertificateSecretProxy: ""
    twoFaDisabled: true
    useOpenShift: true
    imagePullSecrets: acr-secret
region:
    clusterConfig:
        privilegedWorkspaces: true
        customServiceAccount: statefulset-sa
```

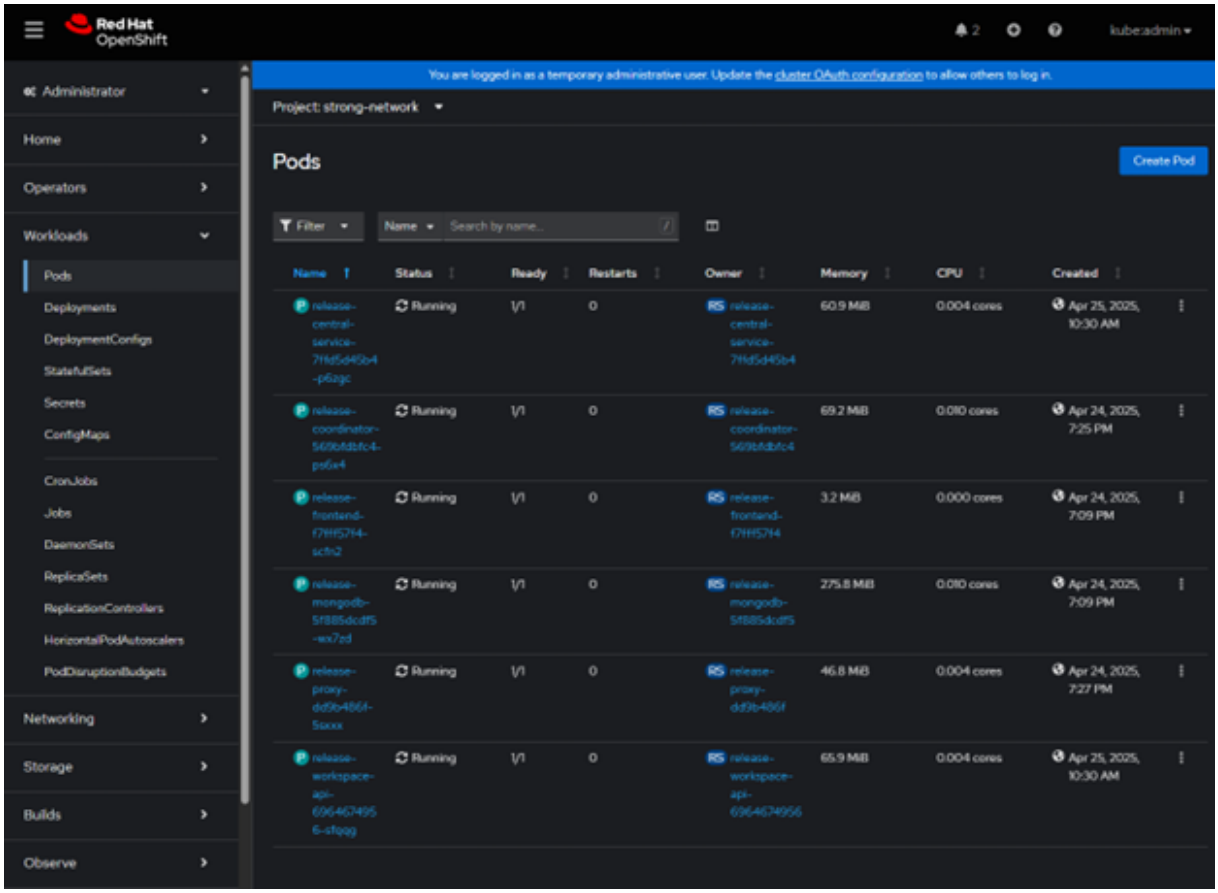
Deployment

Now what's left is to run the helm upgrade and deploy the application:

```
helm upgrade --install \\
             --namespace strong-network \\
             -f "./config_2026.9.4.yaml" \\
             release ./ninjahchart-2026.9.4.tgz
```

The terminal will freeze for a few seconds while deployment is in progress.
Now you can go to the OpenShift console and **Workloads -> Pods** and verify that all pods are running correctly:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



If there is a problem with a particular pod, click on that pod and have a look at its Logs/Events. When everything is ok with all the pods, you can proceed to log in to the platform on the URL that is defined by the **hostName** value in the YAML, using the **userAdminEmail** and **userAdminPassword** from the YAML.

After creating the initial organization and project, you can try creating a workspace and see if it deploys correctly. If the workspace does not deploy correctly, examine the logs of the relevant Pod/StatefulSet, as explained above.

Nutanix Kubernetes Platform

August 20, 2026

Citrix SecurSpaces™ runs on the Nutanix Kubernetes Platform (NKP). This page covers what is specific to NKP:
the cluster prerequisites, the node labels, the certificate, and the ingress settings the chart needs.

Everything after that is the standard installation. When you have finished here, continue with Install with the Strong Installer CLI.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Nutanix publishes a full end-to-end walkthrough as a solutions tech note, which is useful for the NKP side of
the deployment:
Citrix Secure Developer Spaces on Nutanix Kubernetes Platform.

> **Important:**
>
> The Nutanix tech note was written against SecurSpaces `2025.10.13` and has not been revised since. Its
> installer image tag, chart file name, and CLI command are all out of date —it uses `strong-cli`, which is
> now `sds-cli`. Use it for the NKP-specific steps, and follow the SecurSpaces documentation for the
> installation itself.

Prerequisites

- NKP 2.17 or later, deployed as either a self-managed or a workload cluster.
- A free IP address in the Kubernetes service load balancer range, for the SecurSpaces load balancer.
- A kubeconfig with privileged access to the cluster.
- Three domain names:

Purpose	Example
Primary	`sds.example.com`
Proxy	`proxy.sds.example.com`
Wildcard subdomain	`*.proxy.sds.example.com`

- A TLS certificate whose Subject Alternative Names cover all three domains.
- A SecurSpaces platform license.

For supported Kubernetes versions and capacity planning, see
System requirements and
Sizing.

Create the namespace

```
1  kubectl create namespace strong-network
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Label the nodes

SecurSpaces separates service nodes from workspace nodes. Label them so the chart can schedule each pod type
on the right pool:

```
kubectl label nodes <service-node-name> strong.network/type=service
kubectl label nodes <workspace-node-name> strong.network/type=workspace
```

For how many nodes of each type to provision, see Sizing.

Install the certificate

Store the certificate and its private key as a TLS secret in the namespace:

```
kubectl create secret tls sds-example-com-cert \
  --key=/path/to/private_key.pem \
  --cert=/path/to/cert.pem \
  -n strong-network
```

If you use cert-manager, request the certificate instead. This example issues a self-signed certificate, which is suitable for testing but not for production:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: sds-example-com-cert
  namespace: strong-network
spec:
  secretName: sds-example-com-cert
  issuerRef:
    name: selfsigned-issuer
    kind: ClusterIssuer
  dnsNames:
    - sds.example.com
    - proxy.sds.example.com
    - "*.proxy.sds.example.com"
```

Configure the chart for NKP

Run the installer as described in
Install with the Strong Installer CLI. Select **Percona MongoDB** when the installer asks for the database type, unless you are using an external database.

Then add the NKP-specific values to the generated configuration file. NKP ships NetScaler CPX as its ingress

© 1997–2026 Citrix Systems, Inc. All rights reserved.

controller, so the chart needs telling to emit NetScaler ingress annotations:

```
platform:
  useNetscalerController: true
  ingressClass: netscaler-cpx-ingress
  sslCertificateSecret: sds-example-com-cert
```

`ingressClass` must match the class your NetScaler CPX controller actually watches. The chart default is `netscaler-cpx-ingress`; confirm yours with `kubectl get ingressclass` and set both sides to the same value.

> **Note:**
>
> If your chart version requires node affinity to be switched on explicitly, also set `platform.enableNodeAffinity: true`. The node labels above have no effect unless affinity is enabled. The Nutanix tech note marks this as required for `2025.10.13`.

Deploy

```
helm upgrade --install \
    --namespace strong-network \
    --create-namespace \
    -f ./config_<version>.yaml \
    release ./ninjachart-<version>.tgz
```

Check that the platform came up:

```
kubectl get pods -n strong-network
kubectl get ingress -n strong-network
```

Then open `https://sds.example.com/login` and sign in with the administrator account you set during installation.

Related information

- Kubernetes platforms
- Install with the Strong Installer CLI
- System requirements
- Sizing

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Install with the Strong Installer CLI

October 01, 2026

The Strong Installer CLI (`sds-cli`) helps you install, upgrade, and manage Citrix SecurSpaces™ deployments. You run the CLI from the Strong Installer container.

Overview

Use this article to:

- Install SecurSpaces for the first time.
- Upgrade an existing deployment.
- Push container images to your registry.

Prerequisites

Before you begin, make sure you have:

- Docker installed and running on your local machine.
- Credentials for your container registry.
- Network access to Docker Hub and your destination registry.
- Enough disk space to download and push container images.

Run Strong Installer

Strong Installer is available as a Docker container on Docker Hub. Run this command to start it:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:2026.9.4
```

> **Note:**
>
> `${ PWD }` mounts your current directory to `/strong-network/shared` inside the container. Strong Installer saves configuration files and Helm charts to this location.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Install SecurSpaces

Before you begin

Make sure you have:

- The destination container registry URL.
- Authentication credentials for your registry.
- The service and workspace versions you want to install.

Installation steps

1. Start the Strong Installer container:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:2026.9.4
```

2. Run the installation command:

```
sds-cli install
```

   Optional: Specify the service and workspace versions with flags:

```
sds-cli install -v 2026.9.4 -w 2.3.9
```

3. Enter the initial configuration values when prompted.
4. Follow the prompts to authenticate to your cloud provider or registry.
5. Enter the full destination registry URL.
6. Wait while the CLI creates the configuration file and pushes images to your registry.

Registry authentication

Strong Installer detects the registry type from the URL and prompts for the appropriate credentials. For the
credential type each registry expects, and for the flags that supply them non-interactively, see SDS CLI reference.

> **Important:**
>
> After you authenticate, the Helm chart file is downloaded to your shared directory. Secure this file immediately, as it contains your deployment configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Deploy to the cluster

Installing pushes the images and writes the configuration file. Deploying applies that configuration to your
Kubernetes cluster. Prepare the cluster first —see
Kubernetes for your platform.

Before you begin

Make sure you have:

- A prepared Kubernetes cluster, with `kubectl` pointing at it.
- An ingress controller. The chart's default `ingressClass` is `nginx`, but NetScaler CPX is recommended
  because the chart deploys and configures it for you. See
  Ingress.
- A TLS certificate and key for your host domain.
- The Helm chart and configuration file produced by the install step.

Install an ingress controller

SecurSpaces will not start serving traffic without one. Unless you are using NetScaler CPX, which the
chart deploys for you, install and configure the controller before you apply the chart.

- NGINX —install with Helm, then apply the recommended ConfigMap values
- NetScaler CPX —deployed by the chart, no
  separate installation
- Ingress —if you have not chosen a controller
  yet

Whichever you use, `platform.ingressClass` in your configuration file must match the class the controller
actually watches. Confirm it with `kubectl get ingressclass`.

Upload the TLS secret

The SecurSpaces ingress configuration references a TLS secret. Create it before deploying:

```
1 kubectl create secret tls my-ssl-secret --cert=path/to/tls.crt --key=
    path/to/tls.key
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Deploy the chart

```
helm upgrade --install my_release ninjahchart-${
 SN_VERSION }
 .tgz -f config.yml
```

Verify the deployment

List the running pods:

```
kubectl get pods
```

The output resembles the following:

```
pod/release-central-service-7794bf97db      1/1 Running 0 3h24m
pod/release-central-service-55f74bb5d       1/1 Running 0 3h24m
pod/release-mongodb-869ddfb7c-7q58h         1/1 Running 0 3h24m
pod/release-coordinator-7f9c4c95bd5fd       1/1 Running 0 3h24m
pod/release-proxy-569859ddb9-748jj          1/1 Running 0 3h24m
pod/release-frontend-7b4b8b97d4-67d2k       1/1 Running 0 3h24m
pod/release-workspace-api-74cb48675c-9x4v8 1/1 Running 0 3h24m
```

Then browse to your deployment URL and sign in with the administrator credentials set in your Helm values.

What to configure next

Task	Page
Connect an identity provider	Identity and access
Connect code repositories	Code repositories
Allow SSH access to workspaces	SSH TCP forwarding
Add a region	Multiple regions
Create your first project and team	Set up a project for your team

Upgrade an existing deployment

Before you begin

Make sure you have:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- The configuration file from your previous deployment, for example, `config_2026.9.0.yaml`.
- Access to the same container registry used in the original deployment.
- A recent database backup. For internal Percona, review the stepwise upgrade procedure.

Procedure

1. Start the Strong Installer container with the target release version:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:<target-version>
```

2. Run the upgrade command with your existing configuration file:

```
sds-cli upgrade -c config_2026.9.0.yaml
```

3. Wait for the CLI to download the chart, update the configuration, and push images. For Percona deployments (`platform.deployPerconaMongoDB` unset or **`true`**), it also copies the upgrade kit to `/strong-network/shared/percona-upgrade/` and prints the script instructions. No new Percona prompts are added to the CLI; rerunning it replaces the kit with the installer version's scripts.

4. When upgrading from SDS 2026.9 to the chart with Percona Operator 1.23.0, run the kit from the shared directory before Helm. Replace the namespace, release, and chart version with your values:

```
./percona-upgrade/upgrade-percona.sh default release ./ninjahchart
   -<version>.tgz
```

   The script asks for `yes`, takes a backup, upgrades through 1.21.2 and 1.22.0, and applies the 1.23 CRDs.
   Helm 3 does not upgrade existing CRDs. See Upgrade for prerequisites, write pauses, and recovery commands.
   Skip this step for external MongoDB.

5. Run the Helm command printed by the installer, then validate the platform. For Percona, this completes the move to operator 1.23.0, MongoDB 7.0.37-20, and PBM 2.15.0. Helm blocks an existing Percona deployment that has not reached 1.22 or lacks the required 1.23 CRDs.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Upgrade without a configuration file

If you do not have the original configuration file, run the upgrade with the `--ignore-config` flag:

```
sds-cli upgrade --ignore-config
```

The CLI then prompts for each value it would otherwise read from the configuration file, including the host
domain and the destination registry URL, and does not write a new configuration file. See Upgrading without a configuration file.

> **Warning:**
>
> If you upgrade without a configuration file, you must enter all deployment parameters manually. Use this option only if the configuration file is unavailable.

Automated upgrade

For CI/CD pipelines or other automated deployments, you can run the installer upgrade stage with one
command. Apply the printed Helm command afterward. For the move to Percona 1.23, first complete the
Percona script step above; the script requires confirmation.

Example: Google Artifact Registry

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:2026.9.4 \
    upgrade \
    -c config_2026.9.0.yaml \
    --source-registry-region eu \
    --service-account-file-path shared/your-service-account.json
```

The destination registry URL is read from the configuration file passed with `-c`. There is no flag to set it.

Flags

For the full flag list, including the registry-specific authentication flags for Google Artifact Registry, Amazon ECR, Azure Container Registry and generic Docker login, see
SDS CLI reference.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Tip:**
>
> Use `--use-docker-login` with `--registry-username` and `--registry-password` to skip cloud provider authentication prompts. This is useful for CI/CD pipelines that do not support interactive prompts.

Push images only

To push container images without running a full installation or upgrade, use the `push-images` command.

Procedure

1. Start the Strong Installer container:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:2026.9.4
```

2. Run the `push-images` command:

```
sds-cli push-images \
    --source-registry-region eu \
    -c config.yaml \
    --verbose
```

Push images with Docker login (non-interactive)

For CI/CD pipelines, use the `--use-docker-login` flag to skip cloud authentication prompts:

```
sds-cli push-images \
    --source-registry-region eu \
    --use-docker-login \
    --registry-username "your-username" \
    --registry-password "your-password" \
    -c config.yaml
```

Flags

For the full `push-images` flag list, including the skip options, see SDS CLI reference.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Authentication prompts in CI/CD

Issue	Resolution
The CLI prompts you to choose an authentication method even when credentials are provided.	Use `--use-docker-login` with `--registry-username` and `--registry-password` to skip cloud provider authentication.

```
sds-cli push-images \
    --use-docker-login \
    --registry-username "$USERNAME" \
    --registry-password "$PASSWORD" \
    ...
```

Docker permission denied

Issue	Resolution
Permission denied errors when you run Docker commands	Make sure your user is added to the `docker` group, or prefix the command with `sudo`.

Image push failures

Issue	Resolution
Images fail to push to the destination registry.	Verify that your registry credentials are correct. Make sure the destination registry exists and is accessible. Check network connectivity to both the source and destination registries. Use `--verbose` for detailed error messages.

Related information

- SDS CLI reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Upgrade
- Requirements

Ingress

September 22, 2026

Every Citrix SecurSpaces™ deployment needs an ingress controller in front of it. The platform terminates TLS
at the ingress, and workspace traffic —including WebSocket connections for the Cloud IDE and, optionally,
SSH —passes through it.

Supported controllers

Controller	Deployed by the SecurSpaces chart	Configuration needed
NetScaler CPX (recommended)	Yes	None. See NetScaler CPX
NGINX	No	Tuning values, and manual setup for SSH
HAProxy	No	Set the ingress class
Kubernetes Gateway API	No	Set a gateway class instead of using Ingress resources
Istio	No	Match your gateway, and manual setup for SSH. See Istio

**NetScaler CPX is the recommended choice, and the only one the SecurSpaces chart can deploy and configure
for you.** If you have no existing preference, use it.

> **Important:**
>
> **NetScaler CPX does not currently work on Google GKE.** This is a known issue with a fix in progress and
> no date yet. On GKE, use NGINX or the Kubernetes Gateway API, and contact Citrix for the current status.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> See Google GKE.

What the choice affects

Two things differ meaningfully between controllers.

WebSocket and HTTP/2 handling. The Cloud IDE holds a WebSocket connection for as long as a workspace is
open. A controller that closes idle connections early, or buffers responses aggressively, produces disconnects that look like platform faults. With NetScaler CPX the chart applies the right profiles itself. With NGINX you set them by hand —see
NGINX.

SSH access to workspaces. SSH needs a TCP path through the ingress, on a port other than 443.

- With **NetScaler CPX**, the chart creates the SSH ingress and the service port for you. Nothing to do.
- With **NGINX**, you create a ConfigMap, patch the controller deployment, and expose the port yourself. See
  SSH TCP forwarding with NGINX.
- With **Istio**, the chart creates no SSH ingress. You expose TCP to the workspace API yourself, as with
  NGINX. See Istio.

That difference is the main practical reason to prefer NetScaler CPX.

Selecting a controller

The controller is chosen in the configuration file the installer generates, before you run Helm. See Install with the Strong Installer CLI.

Setting	Purpose
`platform.ingressClass`	The ingress class your controller watches
`platform.useNetscalerController`	Deploy and configure NetScaler CPX
`platform.useNginxController`	Apply NGINX-specific ingress annotations
`platform.useHAProxyController`	Apply HAProxy-specific ingress annotations
`platform.useGatewayAPI`	Use Gateway API resources instead of Ingress resources

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	Purpose
`platform.useIstio`	Use Istio Gateway and VirtualService resources instead of Ingress resources
`platform.istio.gatewayNamespace`	Namespace where the Istio ingress gateway workload runs
`platform.istio.gatewaySelector`	Labels that select the Istio ingress gateway workload

Set the annotation flag that matches your controller. `ingressClass` must match the class the controller
actually watches —confirm it with `kubectl get ingressclass` rather than assuming the default.

Related information

- NetScaler CPX
- NGINX
- Istio
- SSH TCP forwarding with NGINX
- System requirements

NetScaler CPX

August 20, 2026

NetScaler CPX is the recommended ingress controller for Citrix SecurSpaces™, and the only one the SecurSpaces
Helm chart deploys for you.

It works out of the box. The chart installs NetScaler CPX, wires it to the platform, and applies the settings SecurSpaces needs. You do not have to configure the controller separately.

> **Important:**
>
> **NetScaler CPX does not currently work on Google GKE.** This is a known issue with a fix in progress and
> no date yet. On GKE, use NGINX or the
> Kubernetes Gateway API, and contact Citrix for the current status. Other platforms are unaf-

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> fected.

What the chart sets up

Understanding what is already handled saves you repeating it by hand.

Area	What you get
The controller	NetScaler CPX is installed as part of the SecurSpaces release, with its licence accepted and exposed through a LoadBalancer service
Ports	80 and 443 for the platform, plus a port for SSH to workspaces and one for the VDI agent
Connection handling	HTTP and TCP profiles with WebSocket and HTTP/2 enabled, which is what the Cloud IDE needs to hold a session open
Backend security	Client authentication on the front end, and TLS to the frontend and central services
SSH to workspaces	A dedicated SSH ingress, created automatically, routed to the workspace service over TCP

The SSH ingress is the difference worth knowing about. With other controllers you build that path yourself.

Selecting it

NetScaler CPX is chosen in the configuration file the installer generates:

```
platform:
  useNetscalerController: true
  ingressClass: netscaler-cpx-ingress
  sslCertificateSecret: <your-tls-secret>
```

`ingressClass` must match the class the controller watches. `netscaler-cpx-ingress` is the value the chart
uses; if you change one, change both.

`sslCertificateSecret` names a TLS secret in the release namespace, covering your platform domain, your proxy
domain, and the wildcard proxy subdomain. See
Install with the Strong Installer CLI.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Verify

After the release is deployed:

```
kubectl get pods -n strong-network
kubectl get ingress -n strong-network
kubectl get svc -n strong-network
```

You should see the CPX pod running, an external address on the LoadBalancer service, and an ingress for the
platform alongside one for SSH.

If the LoadBalancer address stays pending, your cluster has no load balancer provider or no free address in
its range. That is a cluster-level problem rather than a SecurSpaces one.

Using an existing NetScaler

If you already run NetScaler in front of your cluster, point SecurSpaces at it rather than letting the chart
deploy CPX: leave `useNetscalerController` off, and set `ingressClass` to the class your existing controller
watches. The chart still applies the NetScaler-specific annotations when the class name begins with `netscaler`.

Related information

- NetScaler CPX documentation —the product documentation for CPX itself, including scaling, licensing, and advanced configuration
- Ingress
- Install with the Strong Installer CLI
- System requirements

NGINX

August 24, 2026

These values apply to the **ingress-nginx** controller. If you use NetScaler CPX, the SecurSpaces chart applies the equivalent profiles itself and there is nothing to tune. See
NetScaler CPX.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Citrix SecurSpaces™ carries long-lived connections —the Cloud IDE, SSH sessions, and workspace apps all hold
a socket open for the length of a working session. The default ingress-nginx settings are tuned for short
request/response traffic, so the following values are recommended for a SecurSpaces deployment.

Install the controller

SecurSpaces does not install NGINX for you. The cluster needs an ingress-nginx controller before you deploy
the Helm chart.

Many managed Kubernetes services have a recommended way to do this —check the
ingress-nginx deployment guide
for your cloud provider first. If your provider is not listed, install it with Helm:

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
helm install --namespace nginx --create-namespace ingress-nginx ingress-nginx/ingress-nginx \
  --set controller.service.type=LoadBalancer
```

On AWS, add the annotation that provisions a Network Load Balancer:

```
  --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-type"="nlb"
```

Confirm the ingress class the controller watches, and use that value for `platform.ingressClass` in your
SecurSpaces configuration:

```
kubectl get ingressclass
```

Tune the controller

Apply the following values before you put the platform into use. They are not optional in practice: the defaults produce dropped Cloud IDE sessions under normal working conditions.

A default ConfigMap exists in the `ingress-nginx` namespace, usually named `ingress-nginx-controller`. The
name varies with how the ingress controller was installed.

```
kubectl edit configmap ingress-nginx-controller
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Recommended values

```
apiVersion: v1
data:
  allow-snippet-annotations: "true"
  enable-brotli: "true"
  keep-alive: 120s
  keep-alive-requests: "10000"
  use-gzip: "true"
  use-http2: "true"
kind: ConfigMap
```

Value	Effect
`allow-snippet-annotations`	Controls whether the controller honors snippet annotations. See the note below —this one has consequences beyond performance.
`enable-brotli`	Compresses responses with Brotli where the client supports it.
`keep-alive`	Holds an idle upstream connection open for 120 seconds rather than the default 75.
`keep-alive-requests`	Allows 10,000 requests per keep-alive connection rather than the default 1,000, which matters for the long sessions the Cloud IDE creates.
`use-gzip`	Compresses responses for clients that do not support Brotli.
`use-http2`	Serves HTTP/2 to clients that support it.

About `allow-snippet-annotations`

The SecurSpaces Helm chart reads this value from the ConfigMap when it renders, and behaves differently
depending on what it finds:

- **`"true"`** —the chart adds its own `server-snippet` annotation to the ingress, which installs SecurSpaces error pages for HTTP 400, 401, 403, 404, 405, 408, and 413.
- **`"false"`** —the chart omits that annotation. The platform still works, but those responses fall back to
  the ingress controller's default error pages.

If you set `platform.nginxConfig` or `platform.nginxConfigSubdomain` in your Helm val-

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ues, the chart emits a
`configuration-snippet` annotation, which the controller only honors when `allow-snippet-annotations` is
`"true"`.

> **Note:**
>
> Snippet annotations let any user who can create an Ingress object inject NGINX configuration. Recent
> ingress-nginx releases default this to `"false"` for that reason. Setting it to `"true"` is a trade-off
> between the SecurSpaces error pages and that exposure. If your cluster policy forbids snippet annotations,
> leave it `"false"` —the platform is fully functional either way.

Related information

- SSH TCP forwarding
- Install with the Strong Installer CLI
- Ports and protocols

Istio

September 22, 2026

Use Istio when a cluster standard requires platform traffic to enter through an Istio ingress gateway. When
you enable it, the Citrix SecurSpaces™ chart renders Istio Gateway, VirtualService, and DestinationRule
resources instead of Kubernetes Ingress resources.

SecurSpaces does not install Istio. Install the Istio ingress gateway and confirm it is running before you
deploy the chart.

> **Important:**
>
> The chart creates no SSH ingress when Istio is enabled. If your developers connect to workspaces over SSH,
> you build that path yourself. See SSH to workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Before you start

Make sure you have:

- **Istio 1.22 or later.** The chart uses the stable `networking.istio.io/v1` API.
- **An Istio ingress gateway already running** in the cluster.
- **Permission to create resources in the gateway namespace.** The chart creates a Secret holding the
  internal services certificate authority there, along with the DestinationRules that select the gateway
  workload. Istio resolves both in the namespace of the gateway rather than the release namespace.
- **The labels that identify your gateway workload.** Confirm them rather than assuming the default:

```
kubectl get pods -n <gateway-namespace> --show-labels
```

Select Istio

Istio is chosen in the configuration file the installer generates, before you run Helm:

```
platform:
  useIstio: true
  istio:
    gatewayNamespace: istio-system
    gatewaySelector:
      istio: ingressgateway
  sslCertificateSecret: <your-tls-secret>
```

Value	What it sets
`platform.useIstio`	Renders Istio resources instead of Ingress resources
`platform.istio.gatewayNamespace`	The namespace where the Istio ingress gateway workload runs
`platform.istio.gatewaySelector`	The labels that select the Istio ingress gateway workload

The defaults shown above match a gateway installed with `istioctl install`. `platform.ingressClass` and the
controller-specific flags are ignored while `useIstio` is on.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Managed Istio distributions

Managed distributions label their gateway differently and run it in their own namespace. The Azure Kubernetes Service Istio add-on, for example, needs:

```
platform:
  useIstio: true
  istio:
    gatewayNamespace: aks-istio-ingress
    gatewaySelector:
      istio: aks-istio-ingressgateway-external
```

What the chart creates

Resource	What it does
Gateway for the platform host	Serves HTTP on port 80 with a redirect to HTTPS, and HTTPS on port 443
Gateway for workspace subdomains	Serves the proxy hostname and its wildcard subdomain
VirtualService for the platform host	Routes each platform path to the service that serves it
VirtualService for workspace subdomains	Routes subdomains to the workspace API, and coordinator to the coordinator
DestinationRules, in the gateway namespace	Re-originate TLS to each service and verify the certificate
Services certificate authority Secret	Gives the gateway the authority it verifies those services against

On the platform host, the chart routes gRPC-Web, SCIM, OAuth, SAML, and REST paths to the central service,
workspace and browser paths to the workspace API, and everything else to the frontend.

Public TLS terminates at the Istio ingress gateway. The gateway then opens a second TLS connection to the
SecurSpaces services, so traffic stays encrypted on the way through the cluster.

In an external region, the chart exposes only the workspace API, on that region's own API host.

SecurSpaces stays outside the mesh

The SecurSpaces service pods opt out of Istio sidecar injection. The services already encrypt traffic between themselves with the internal services certificate authority, and keeping them out of the data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

path
means cluster-wide mesh policy does not govern them.

This matters if your mesh applies either of the following, because both would otherwise break the platform:

- An `outboundTrafficPolicy` of `REGISTRY_ONLY`, which would block outbound calls to Git providers,
  identity providers, and container registries.
- A default-deny `AuthorizationPolicy`, which would block the calls between SecurSpaces services.

The ingress gateway still routes to the services through the DestinationRules the chart creates.

SSH to workspaces

The Istio resources the chart creates carry HTTP and HTTPS only. Unlike NetScaler CPX™, which creates an SSH
ingress and the service port for you, Istio leaves that path to you.

To offer SSH, expose TCP traffic to the workspace API service on port 2222 through your gateway, then enable
SSH at the platform, organization, project, and workspace levels. The platform-side steps are the same
whichever controller you use —see
SSH TCP forwarding, and start at
Enable SSH Access in the Platform.

Verify

After the release is deployed, confirm the resources exist and the gateway has an external address:

```
kubectl get gateway,virtualservice -n strong-network
kubectl get destinationrule -n <gateway-namespace>
kubectl get svc -n <gateway-namespace>
```

Expected result: two Gateways and two VirtualServices in the release namespace, a DestinationRule for each
SecurSpaces service in the gateway namespace, and an external address on the gateway service.

If the platform host returns a 503, the gateway is matching the Gateway resource but cannot reach a service.
Check that the DestinationRules landed in the gateway namespace and that their workload selector

© 1997–2026 Citrix Systems, Inc. All rights reserved.

matches the
gateway labels.

Related information

- Ingress
- SSH TCP forwarding
- Install with the Strong Installer CLI
- Ports and protocols

SSH TCP forwarding with NGINX

August 20, 2026

This guide describes how to configure the SecurSpaces platform to enable SSH access to Workspaces. SSH access allows developers to securely connect to the remote filesystem of a workspace and use remote IDE features in tools such as Visual Studio Code, JetBrains Gateway, Cursor, or Windsurf.

> **Note:**
>
> The ingress configuration on this page is for **NGINX**. If you use NetScaler CPX, the SecurSpaces chart
> creates the SSH ingress and the service port for you, so you can skip to
> Enable SSH Access in the Platform. See
> NetScaler CPX.
>
> When using Kubernetes distributions such as MicroK8s, replace the deployment application with a `DaemonSet`.

Overview

The SSH access feature is optional and must be enabled at multiple levels:

- Platform
- Organization
- Project
- Individual workspace

This guide walks through:

1. Configuring the nginx load balancer to forward TCP requests for SSH access.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Enabling SSH access in the platform, organization, and project settings.
3. Using SSH to connect to Workspaces.

Configure NGINX for SSH Access

The nginx load balancer must be configured to handle SSH requests. This is a relatively quick process. You will need to:

1. Create a ConfigMap named ssh-mapping in the nginx namespace which maps the SSH port to the SSH port of the SN workspace service (designated 12345)
2. Edit the DeploymentApp of the nginx ingress controller so that it applies the new ConfigMap in the –tcp-services-configmap flag.
3. Expose port 12345 in the Service of the nginx ingress controller.

Create a ConfigMap

To create the ConfigMap, you first switch to the namespace of the nginx controller - by default it should be called nginx. Then, simply run the command to create the ConfigMap:

```
kubectl create configmap ssh-mapping
```

Edit the ConfigMap's data field to include a mapping from the SSH port to your release's workspace API (it's listening on port 2222, which is hardcoded, please do not change this value). To do this, edit the config map:

```
kubectl edit configmap ssh-mapping
```

Update the `data` field:

```
apiVersion: v1
data:
  "12345": default/release-workspace-api:2222
kind: ConfigMap
```

> **Important:**
>
> Port `2222` is hardcoded in the Workspace API. Do not change this value.

Update the NGINX Ingress Controller Deployment

Edit the DeploymentApp if the nginx ingress controller deployment to include the `--tcp-services-configmap` argument:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
kubectl edit deployment ingress-nginx-controller
```

Add the following to the Arguments of the controller (under the Args header):

```
spec:
--tcp-services-configmap=$(POD_NAMESPACE)/ssh-mapping
```

Expose Port in the Service

Expose the port in the service of the nginx controller. Add the following entry under the ports field of the service:

```
kubectl edit svc nginx-ingress-controller
```

```
Ports:
appProtocol: http
      name: http
      nodePort: 30875
      port: 80
      protocol: TCP
      targetPort: http
appProtocol: https
      name: https
      nodePort: 31800
      port: 443
      protocol: TCP
      targetPort: https
name: ssh
      port: 12345
      protocol: TCP
      targetPort: 12345
```

Once complete, TCP requests to port 12345 will be forwarded to the workspace service.

Enable SSH Access in the Platform

SSH access must be enabled at the platform, organization, and project levels individually.

Platform Level

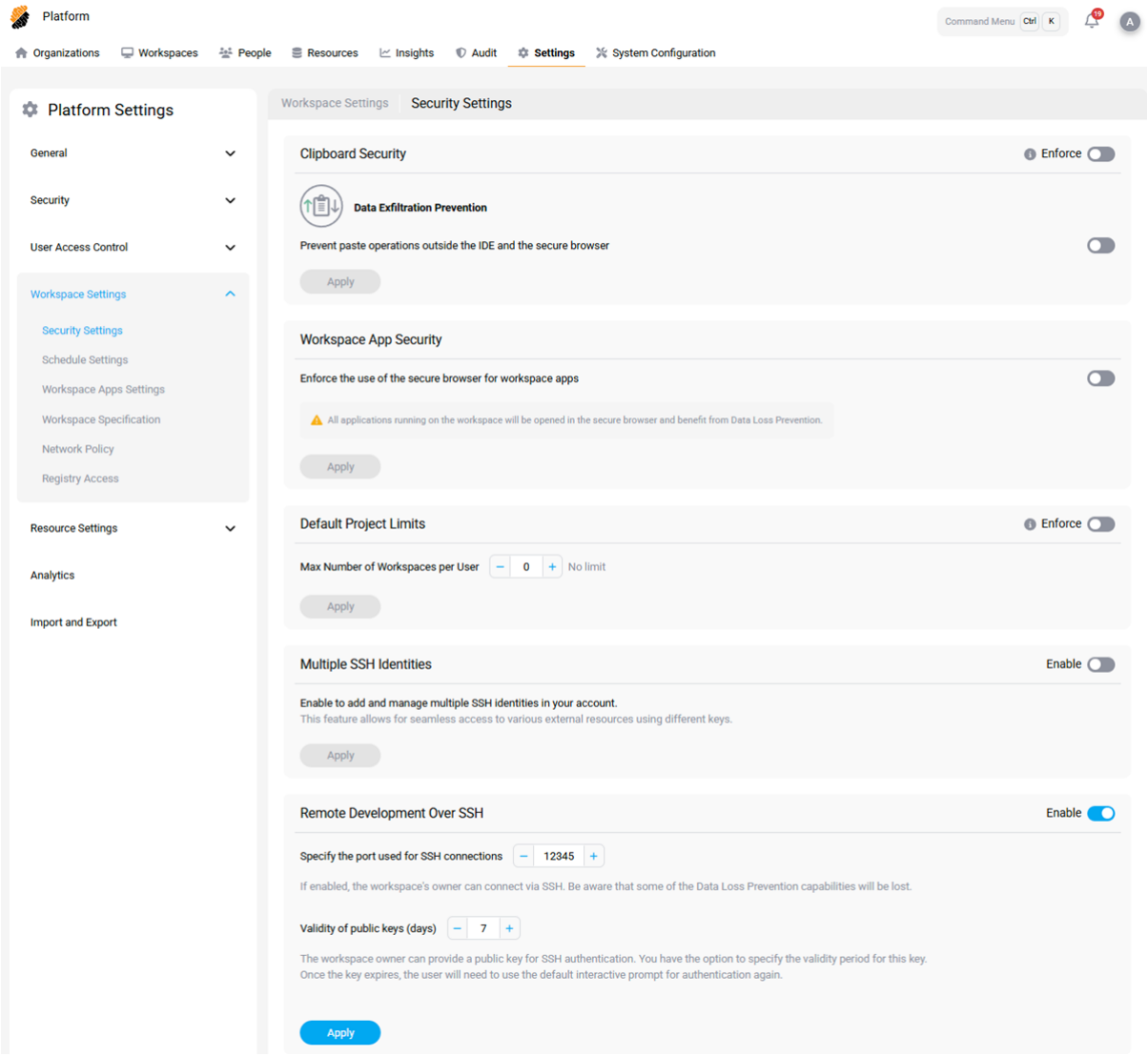
> **Required role**: Administrator or security officer

To enable the feature, navigate to the Workspace Platform Settings page:
Platform Overview → Settings → Workspace Settings → Security Settings

- Locate the **Remote Development Over SSH** section.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Toggle the feature **on**.
- Ensure the SSH port matches the exposed port in the nginx load balancer (12345).



The administrator can also configure the **validity period** for public keys used in authentication with the Workspace.

When connecting to a workspace with SSH, the user will be given the following prompt:

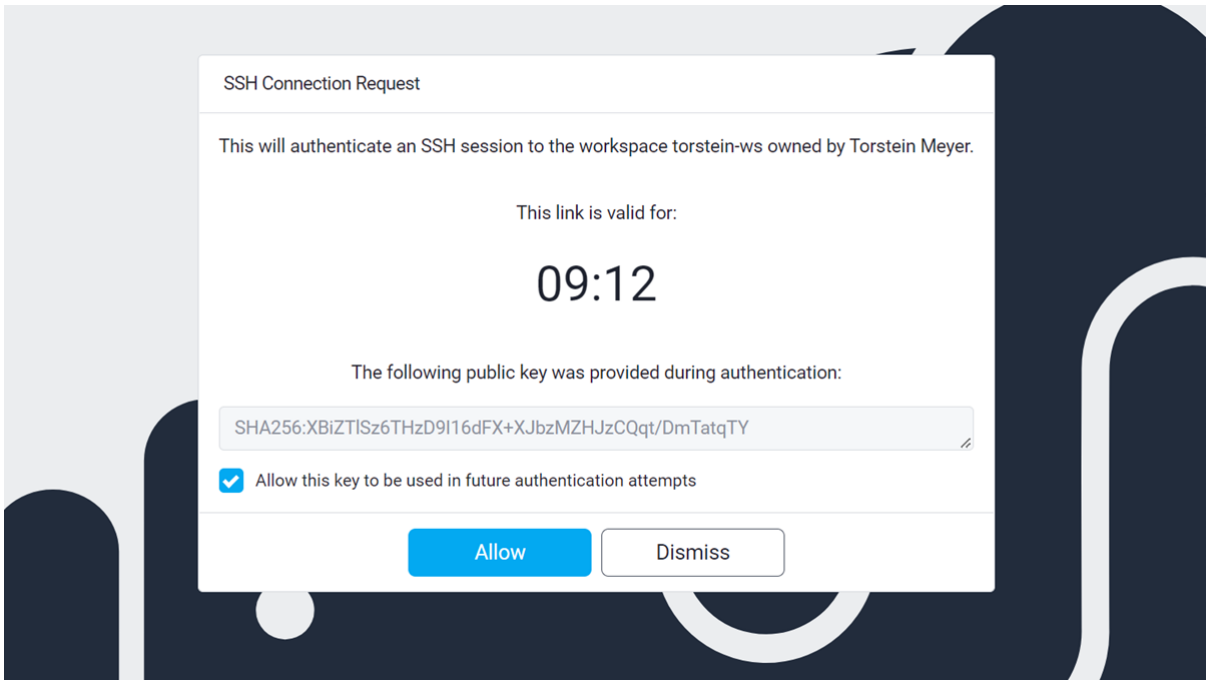
```
C:\Users\torstein>ssh ws-53748696236688@ssh.proxy.cloudcoder.network
you need to approve the SSH connection request on the platform. Access the following URL to proceed:
https://cloudcoder.network/ssh_to_workspace_approval/ebf75f3273c1de91b41097cd7b1f56c12b3bc43e336ec072008d7419b5fc1edf
```

When opening the provided, link they will reach the following page:

From here, the user can either allow or dismiss the request. Additionally, the user can choose to allow the public key provided during the authentication process to skip seeing the prompt in future authentication attempts. This key will only be valid for a set amount of time, which is configurable by the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

administrator through the validity of public keys setting.



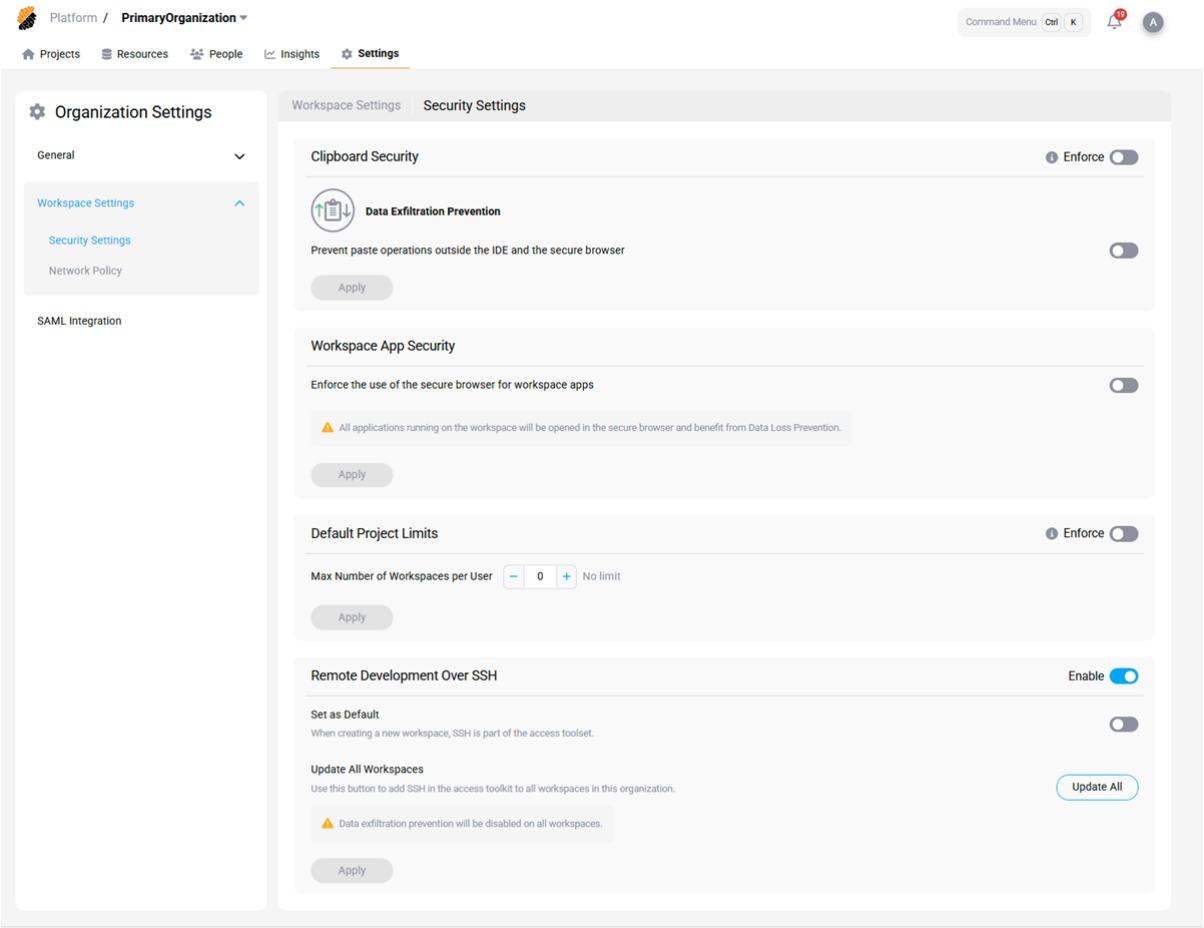
Organization Level

Required role: Organization owner or Administrator

To enable the feature, navigate to the Workspace Security Settings page:
Organization Overview → Settings → Workspace Settings → Security Settings

- Toggle **Remote Development Over SSH** to enable.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



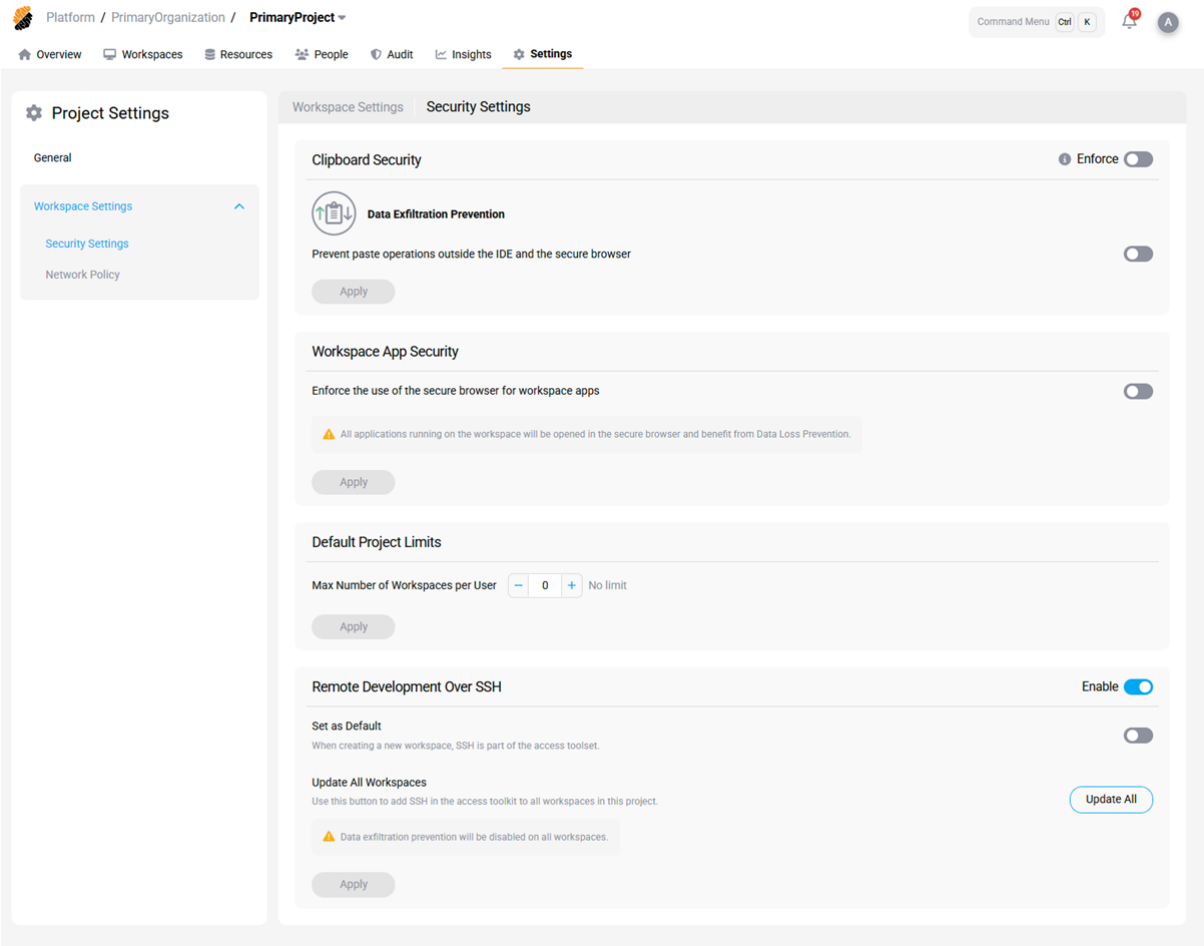
Project Level

> **Required role**: Project owner or Administrator

To enable the feature, navigate to the Workspace Security Settings page:
Project Overview → Settings → Workspace Settings → Security Settings

- Toggle **Remote Development Over SSH** to enable.
- Optionally:
  - Enable SSH as part of the default access item for new workspaces.
  - Update all existing workspaces to include SSH access.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Use SSH to connect to Workspaces

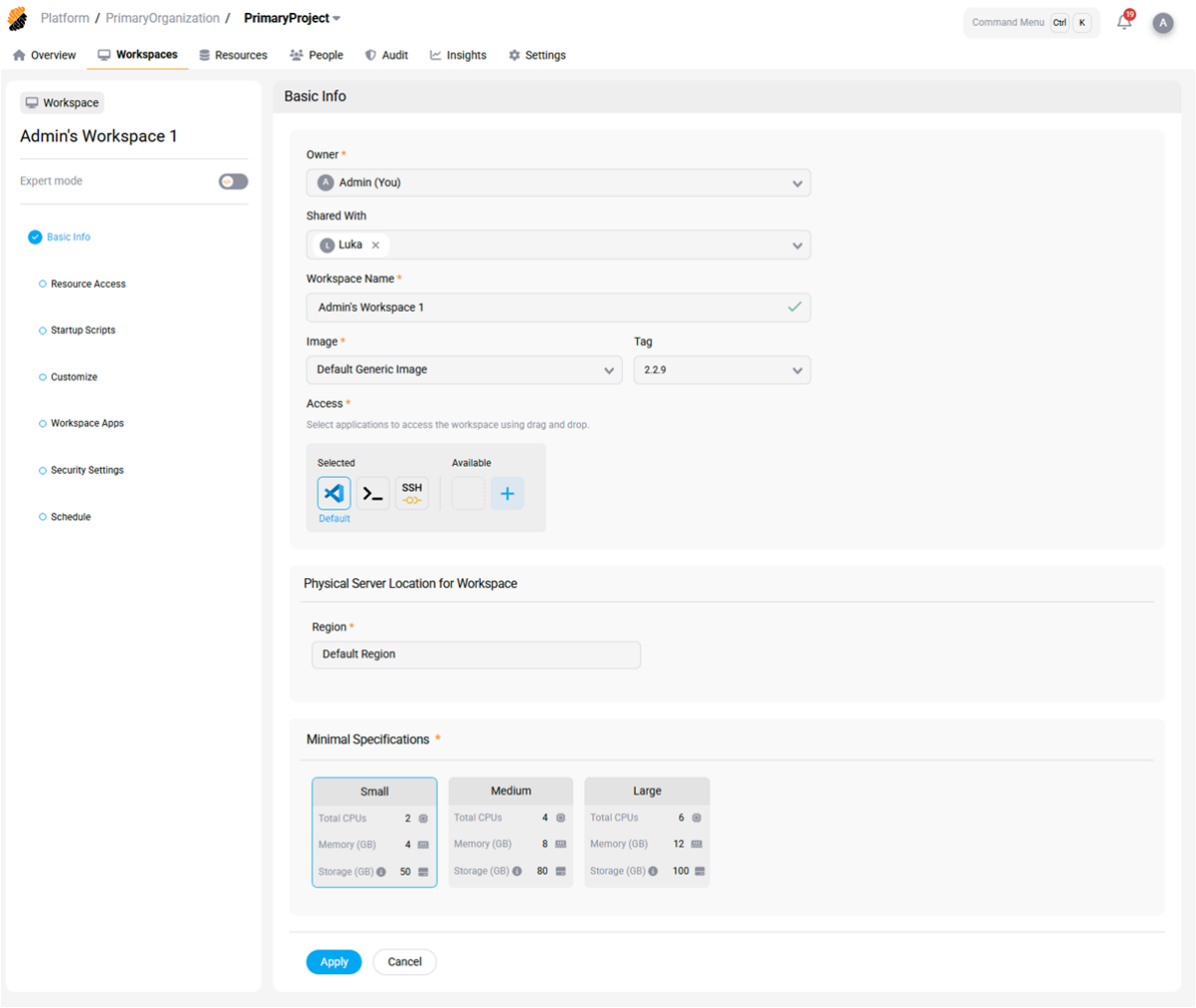
Once enabled, developers can connect to running Workspaces via SSH.

Enable SSH on Individual Workspaces

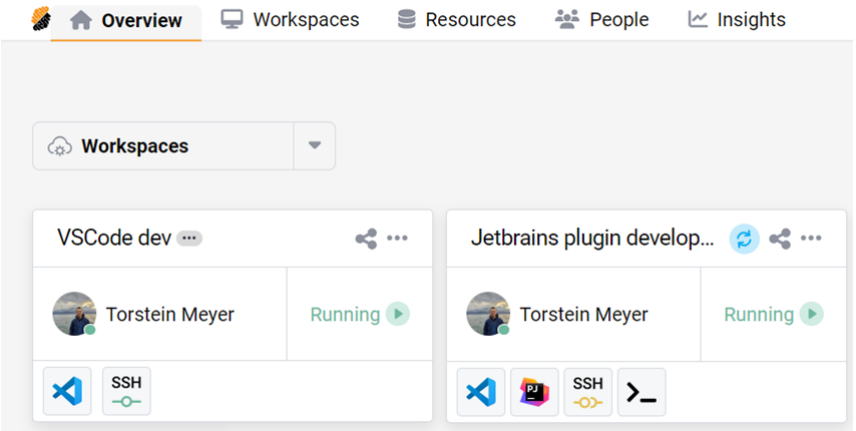
With the SSH feature enabled, developers on the platform can make use of the feature. As an additional safety measure, the feature can also be enabled or disabled on each specific Workspace. By default, SSH is disabled on individual Workspaces. To enable:

- Edit the Workspace.
- On the **Basic Info** page, under **Access**, drag the **SSH** icon from *Available* to *Selected*.
- Click **Apply**.

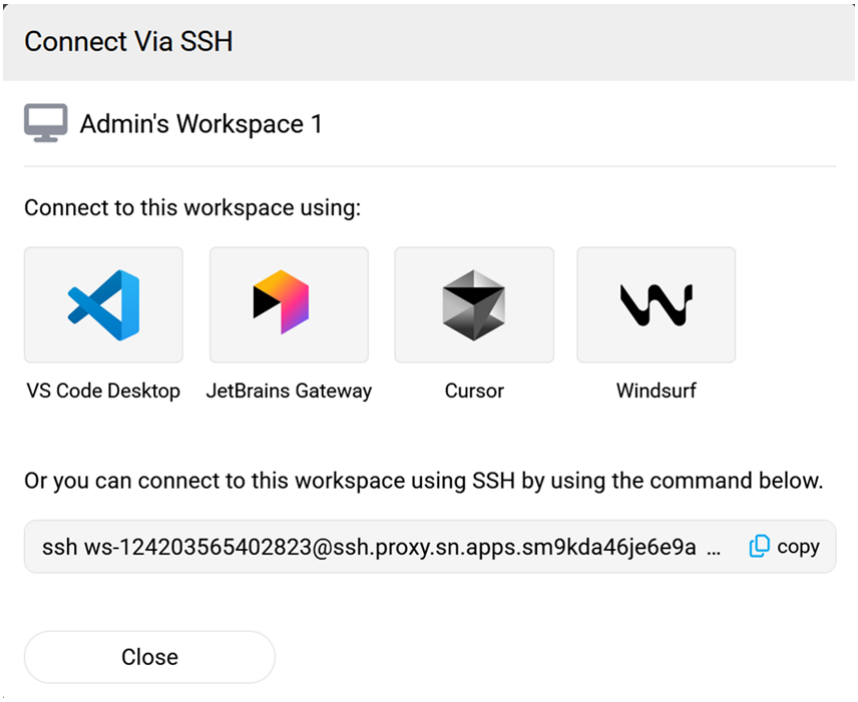
© 1997–2026 Citrix Systems, Inc. All rights reserved.



On workspaces with SSH enabled, the owner of the workspace will be able to access the workspace using SSH when the workspace is in a running state. To do so, first open the Connect Via SSH modal by clicking the SSH icon on the workspace card:



© 1997–2026 Citrix Systems, Inc. All rights reserved.



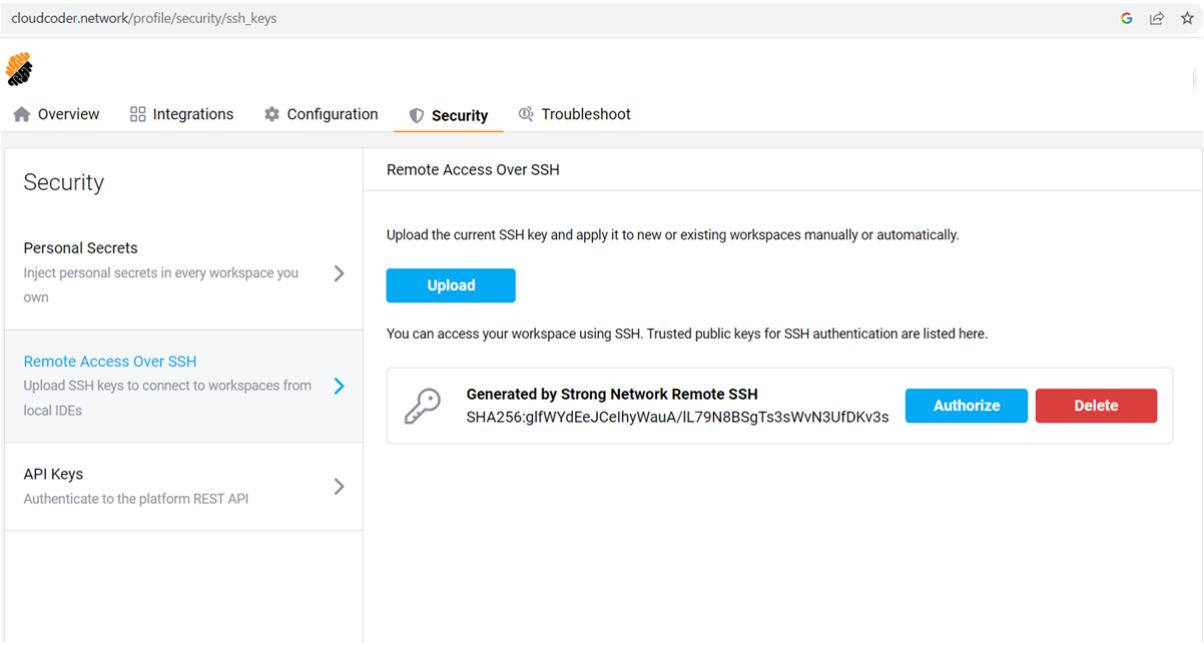
This will open the **Connect Via SSH** modal. Here the user can either connect directly to their local VS Code Desktop and/or JetBrains Gateway editors, or copy the SSH command in the format:

```
ssh ws-{
 id }
 .ssh.proxy.{
 domain }
  -p {
 port }
```

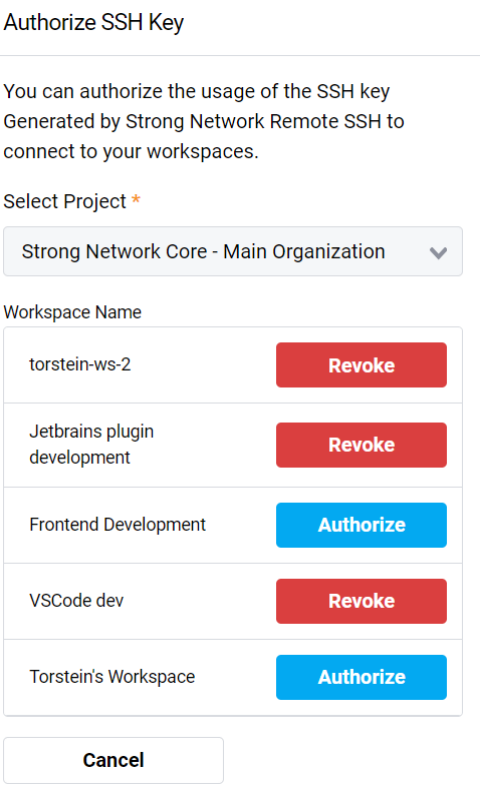
You can then use this command to access the workspace as you would any ordinary SSH server.

The user can authenticate using a public key. To do this, the public key must be uploaded to the platform and authorized for use in the workspace. Uploading the key can be done on the profile page:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Clicking **Authorize** will allow the user to specify the key's access to specific workspaces:



© 1997–2026 Citrix Systems, Inc. All rights reserved.

SSH to Workspace with Local IDEs

This feature can be used with VS Code Remote Development and/or Jetbrains Gateway to use an IDE on your local machine but the filesystem on the remote machine.

External database

June 09, 2026

You can configure SecurSpaces to authenticate to an external MongoDB deployment with X.509 certificates instead of username and password
credentials. Use this option for deployments that require certificate-based authentication, such as MongoDB Atlas environments managed
by enterprise security teams.

Prerequisites

- An external MongoDB deployment that supports `MONGODB-X509`
- A MongoDB X.509 database user whose distinguished name matches the client certificate subject
- A client certificate and private key in PEM format
- The certificate authority bundle if your MongoDB deployment uses a private certificate authority
- Access to update the SecurSpaces Helm values and run `helm upgrade`

Configure X.509 authentication with inline certificate values

Base64 encode the certificate files and add them to `platform.externalMongodb.auth`:

```
platform:
  externalMongodb:
    hostName: "cluster.example.mongodb.net"
    protocol: "mongodb+srv"
    additionalParameters: "retryWrites=true&w=majority"
    authMechanism: "MONGODB-X509"
    auth:
      database: "strong-network"
      caCertPemB64: "<base64-ca-certificate>"
      certPemB64: "<base64-client-certificate>"
      certKeyPemB64: "<base64-client-private-key>"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If your MongoDB deployment uses a publicly trusted certificate authority, you can omit `caCertPemB64`. SecurSpaces then uses the system certificate pool.

Configure X.509 authentication with an existing Kubernetes Secret

If you already manage the certificate material in Kubernetes, reference the Secret and the key names:

```
platform:
  externalMongodb:
    hostName: "cluster.example.mongodb.net"
    protocol: "mongodb+srv"
    additionalParameters: "retryWrites=true&w=majority"
    authMechanism: "MONGODB-X509"
    auth:
      database: "strong-network"
      existingSecret: "sds-mongodb-x509"
      existingSecretKeys:
        caCert: "ca.pem"
        cert: "tls.crt"
        certKey: "tls.key"
```

The Secret must exist in the SecurSpaces deployment namespace before you apply the Helm upgrade.

Apply the configuration

1. Update your SecurSpaces Helm values file.
2. Confirm that the external MongoDB host, protocol, and additional parameters match your MongoDB deployment.
3. Run your standard `helm upgrade` command.
4. Confirm that SecurSpaces services start successfully and can connect to the external MongoDB deployment.

After the upgrade, SecurSpaces connects to the external MongoDB deployment by using the configured X.509 client certificate.

Related information

- The SecurSpaces database
- Database sizing
- Back up the database

© 1997–2026 Citrix Systems, Inc. All rights reserved.

External proxy configuration

Many enterprise environments require all outbound HTTP/HTTPS traffic to flow through a corporate proxy —for egress filtering, TLS inspection, or compliance logging. SecurSpaces supports this natively through a single Helm values block: `platform.externalProxy`.

When configured, **all outbound traffic from both Citrix SecurSpaces™ platform services and developer workspaces** routes through the specified proxy. SecurSpaces handles proxy configuration automatically inside workspaces —it injects the standard `HTTP_PROXY`, `HTTPS_PROXY`, and `NO_PROXY` environment variables, and most CLI tools (`curl`, `git`, `apt`, `pip`, `npm`, language runtimes) honor them by default.

> **Note:**
>
> This page is about routing SecurSpaces traffic through **your organization's** outbound proxy. It is not
> about the SecurSpaces proxy service, which is an internal platform component that every workspace
> connection passes through and which needs no configuration. See
> How the SecurSpaces proxy service works.

Configuration block

The configuration lives under `platform.externalProxy` in your Helm values file:

```yaml
platform:
  externalProxy:
    httpsProxyUrl: ""
    httpProxyUrl: ""
    certificate: ""
    noProxyList: ""
```

If you omit the `externalProxy` block entirely, traffic flows directly from the cluster without a proxy.

Fields

`httpsProxyUrl`

The proxy URL used for HTTPS traffic. Format: `http://host:port` or `https://host:port` (the *protocol of the proxy itself*, not the traffic being proxied).

If `httpsProxyUrl` is set but `httpProxyUrl` isn't, HTTPS traffic uses this value, and HTTP traffic falls back to it as well.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

`httpProxyUrl`

The proxy URL used for HTTP traffic. Same format as above.

If `httpProxyUrl` is set but `httpsProxyUrl` isn't, HTTP traffic uses this value, and HTTPS traffic falls back to it.

In most corporate setups, a single proxy handles both protocols, so it's common to configure just one of the two and rely on the fallback. Set both explicitly only if your environment uses different proxies for HTTP and HTTPS.

`certificate`

If your proxy performs TLS inspection (sometimes called TLS interception, man-in-the-middle proxying, or "break-and-inspect"), it presents its own certificate to clients instead of the origin server's. To trust it, provide the proxy CA certificate here.

The certificate must be **PEM-encoded**, then **base64-encoded**, before being placed in the values file. Generate the value with:

```
base64 -w 0 < proxy-ca.pem
```

(or `base64 -i proxy-ca.pem` on macOS). Paste the resulting single-line string as the value.

If you need to trust **multiple certificates** (for example, a root CA plus an intermediate, or several proxies in different regions), concatenate all the PEM blocks into a single file *before* base64-encoding. Each certificate must keep its full `-----BEGIN CERTIFICATE-----` / `-----END CERTIFICATE-----` markers:

```
cat root-ca.pem intermediate-ca.pem proxy-ca.pem > combined.pem
base64 -w 0 < combined.pem
```

The order doesn't matter for trust evaluation —all certificates in the bundle are added to the trust store.

When set, the certificate is added to the trust store of every SecurSpaces service container and every workspace container, so HTTPS calls through the proxy validate correctly without per-application configuration.

Leave empty if your proxy doesn't perform TLS inspection or uses a publicly trusted certificate.

`noProxyList`

A comma-separated list of hosts, domains, and IP addresses that bypass the proxy. Use this for internal services that the proxy can't reach or that don't need inspection —your internal Git server, a private container registry, internal package mirrors, the Kubernetes API, and so on.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Format rules:

- Comma-separated, **no spaces** between entries
- A leading dot (`.example.com`) matches the domain and all its subdomains
- A bare hostname (`example.com`) matches that host only
- IP addresses (`192.168.1.10`) and CIDR ranges (`10.0.0.0/8`) are supported
- Ports can be appended with a colon (`example.com:8443`)

In-cluster traffic (anything that resolves to a `cluster.local` address or a pod/service IP) is already excluded and doesn't need to be listed here.

Worked example

A typical corporate deployment with a TLS-inspecting proxy at `proxy.corp.example.com:8080`, an internal Git server, and a private registry:

```
platform:
  externalProxy:
    httpsProxyUrl: "http://proxy.corp.example.com:8080"
    httpProxyUrl: "http://proxy.corp.example.com:8080"
    certificate: "
       LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURxRENDQXBDZ0F3SUJBZ0lVRkV4
       ..."
    noProxyList: ".corp.example.com,git.internal,registry.internal
       ,10.0.0.0/8"
```

What this does:

- All outbound HTTP and HTTPS traffic from SecurSpaces services and workspaces routes through `proxy.corp.example.com:8080`.
- The proxy CA certificate is trusted everywhere, so TLS inspection works without certificate errors.
- Anything under `*.corp.example.com`, plus the two internal hostnames and the `10.0.0.0/8` private range, bypasses the proxy and connects directly.

Applying changes

SecurSpaces reads `platform.externalProxy` at pod startup. After you update the values and run `helm upgrade`, services pick up the new configuration when their pods restart. Existing workspace pods continue to use the proxy settings they started with —restart the workspace to pick up changes.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Certificate errors (`x509: certificate signed by unknown authority`, `SSL_ERROR_BAD_CERT_`): The proxy performs TLS inspection, but `certificate` is unset or contains the wrong CA. Verify with `openssl s_client -connect <any-https-host>:443 -proxy proxy.corp.example.com:8080` from inside a workspace and check which certificate is presented.

Internal hosts unreachable: A `noProxyList` entry is missing. Subdomain matching requires the leading dot.

Base64 decoding errors at deploy time: The `certificate` value must be a single line with no embedded newlines. Re-run `base64 -w 0` to flatten it.

Deploy SecurSpaces in multiple regions

This guide describes how to deploy Citrix SecurSpaces™ across multiple Kubernetes clusters or regions. A multi-region setup improves developer experience by reducing latency and improving performance by routing users to the closest regional cluster.

Core concept

A multi-region deployment consists of:

- **Primary (central) deployment** –Hosts the main database and services.
- **Regional deployments** –Stateless deployments that connect to the primary deployment's database and services.

To ensure seamless operation, critical configuration values (especially secrets for authentication and encryption) must be synchronized from the primary deployment to each regional deployment. This synchronization is done by copying specific values from the primary Helm deployment to the regional Helm chart.

How to deploy to additional regions

Like the primary deployment, regions are managed through Helm charts.

While the regional Helm charts are similar to those of the primary deployment, the main difference is that several values that are usually **auto-generated during the first deployment** must be **manually copied** from the primary deployment to the regional Helm charts.

These values include:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Database authentication credentials and connection parameters
- Secrets for signing cookies and tokens
- Secrets for encrypting stored values

Populate the Helm charts

1. Copy the values from the primary deployment's `platform` section.
2. Add a `region` section and set `isExternalRegion` to **`true`**.

Example `values.yaml`:

```yaml
platform:
  imageTag: ""                # Image tag for services
  hostName: ""                # Main domain used to access the platform, e.g. strong-network.example.com
  centralProxyHostname: ""    # Wildcard domain for workspaces, e.g. proxy.strong-network.example.com
  jwtSecret: ""               # Use the same jwtSecret as in the main deployment
  secretKeyReposB64: ""       # Example: openssl rand -base64 16
  # Include all other values from the primary deployment's platform section
  # ...
region:
  isExternalRegion: true      # For regional deployments, set this to true
```

> **Note:**
>
> When `isExternalRegion` is set to **`true`**, set `platform.internalMongodb` to **`false`**.

Required fields and their mappings

The following fields must match between the primary and regional deployments:

Field name	Description
hostName	Domain name of the deployment (used by users and API).
centralProxyHostname	Workspace sub-domain of the main deployment (usually `proxy.<hostName>`)
jwtSecret	Secret for signing tokens and cookies.
secretKeyReposB64	Secret for encrypting values.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Retrieve secrets from the primary deployment

Run the following commands in the namespace of the **primary deployment cluster** to extract the required values.

Get the `hostName` value

```
kubectl get secrets strong-network-secret -o yaml
```

Copy the `hostName` value from the output.

Get the `secretKeyReposB64` value

```
kubectl get secrets strong-network-secret -o yaml
```

Copy the `secretKeyReposB64` value, then **base64 decode it** before pasting it into the regional Helm charts.

For example:

```
echo "<base64-encoded-value>" | base64 --decode
```

Next steps

- Verify that all required secrets and configuration values are synchronized between the primary and regional deployments.
- Deploy the Helm chart for the regional cluster.
- Confirm that developers can connect to the nearest regional deployment with minimal latency.

GPU-enabled workspaces on AKS

This guide explains how to enable and validate GPU-enabled Citrix SecurSpaces™ workspaces on an Azure Kubernetes Service (AKS) cluster using NVIDIA GPUs.

Prerequisites

Before you begin, ensure you have the following:

- An AKS cluster with permissions to manage node pools and deploy Kubernetes add-ons
- A GPU-capable node pool (example VM size: **Standard_NC24ads_A100_v4**)
- Kubernetes CLI access (**kubectl**)
- A workspace template/spec option that can request GPUs (for example **GPUs: 1**)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add a GPU node pool

Create or scale a GPU node pool in AKS using the Azure portal, Azure CLI, or Infrastructure as Code (IaC).
After the nodes are ready, confirm they have joined the cluster:

```
kubectl get nodes -o wide
```

Install the NVIDIA device plugin

You must deploy the NVIDIA device plugin as a DaemonSet to allow the cluster to communicate with the GPU hardware.

1. Deploy the plugin:

```
kubectl create -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.17.1/deployments/static/nvidia-device-plugin.yml
```

1. Verify the plugin is running:

```
kubectl -n kube-system get ds nvidia-device-plugin-daemonset
kubectl -n kube-system get pods -l name=nvidia-device-plugin-ds -o wide
```

1. Confirm the GPU resource is advertised by the node. Replace with your specific node name:

```
kubectl describe node <GPU_NODE_NAME> | grep -E "nvidia.com/gpu|Capacity|Allocatable" -n
```

Expected result: The node displays `nvidia.com/gpu` under **Capacity/Allocatable**.

Reserve GPU nodes for GPU workloads (Recommended)

To ensure GPU nodes are reserved for specific tasks, apply labels and taints.

Label GPU nodes for workspace scheduling

Apply a label to the node for workspace scheduling:

```
kubectl label node <GPU_NODE_NAME> strong.network/type=workspace
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Taint GPU nodes to prevent non-GPU workloads

Apply a taint to prevent non-GPU workloads from scheduling on these nodes:

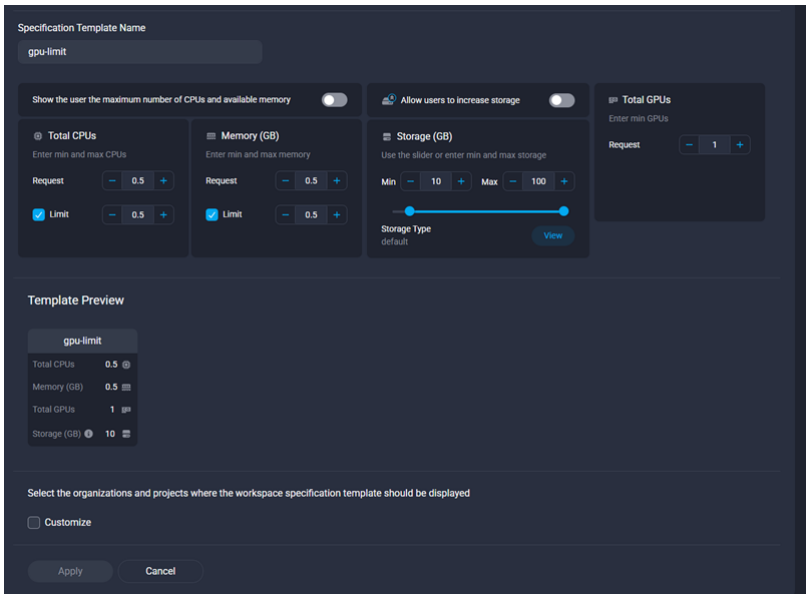
```
kubectl taint node <GPU_NODE_NAME> nvidia.com/gpu=present:NoSchedule
```

> **NOTE:**
>
> GPU-enabled workspace pods must include a matching toleration so they can schedule onto these nodes.

Create a GPU-enabled SecurSpaces workspace template

In your admin portal, create or update a workspace template that requests GPU resources. Ensure you configure:

- GPUs: 1 (or the value required for your workload).
- Resource limits: Size CPU, memory, and disk appropriately for GPU tasks.



Create the SecurSpaces workspace

Select your GPU-enabled template to create a new workspace.

Validate GPU allocation (Kubernetes)

After the workspace is deployed, verify that the GPU is correctly allocated and accessible.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Confirm the workspace pod is running on a GPU node

Verify the workspace pod is running on a designated GPU node:

```
kubectl get pod <workspace-pod> -n <namespace> -o wide
```

Expected result:

- **Status**: `Running`
- **Node**: GPU-capable node.

Confirm GPU is requested by the workload container

```
kubectl describe pod <workspace-pod> -n <namespace>
```

Expected result:

- The main workspace container includes nvidia.com/gpu: 1 in requests/limits
- The pod has the necessary toleration (if GPU nodes are tainted)

Validate GPU access inside the SecurSpaces workspace

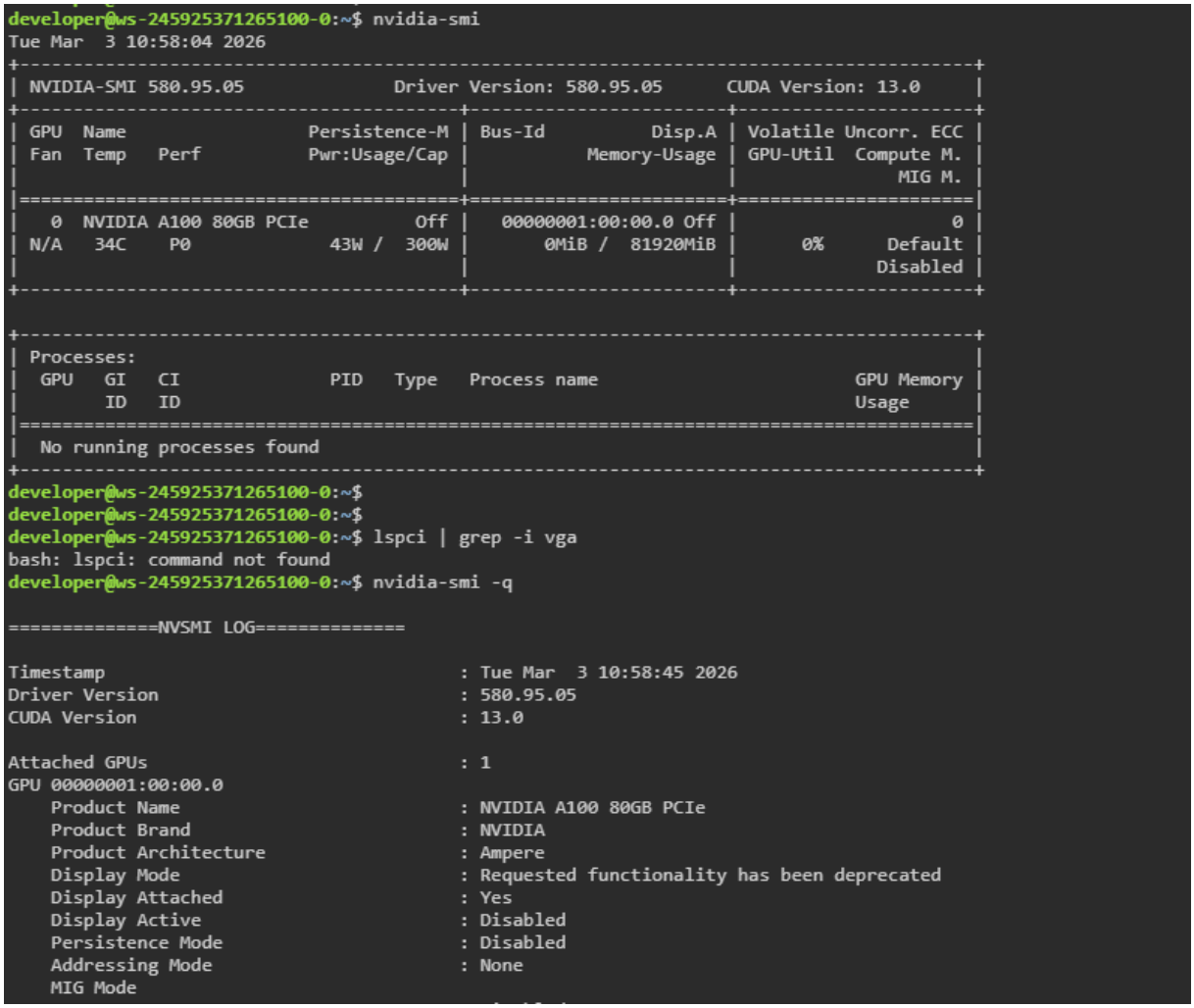
From inside the SecurSpaces workspace container terminal, run:

```
nvidia-smi
```

Expected result:

- NVIDIA GPU details are displayed (model, driver version, utilization).

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Troubleshooting

GPUs are not detected by Kubernetes

If the device plugin is running but GPU capacity is not visible on nodes (common after node pool changes), restart the node runtime and redeploy the plugin:

```
kubectl debug <GPU_NODE_NAME> -it --image=ubuntu -- chroot /host
    systemctl restart containerd
kubectl delete daemonset -n kube-system nvidia-device-plugin-daemonset
kubectl create -f https://raw.githubusercontent.com/NVIDIA/k8s-device-
    plugin/v0.17.1/deployments/static/nvidia-device-plugin.yml
```

Then re-check node capacity:

```
kubectl describe node <GPU_NODE_NAME> | grep -E "nvidia.com/gpu|
    Capacity|Allocatable" -n
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Common issues

Issue	Potential Cause
Pod Pending: “Insufficient nvidia.com/gpu”	Not enough free GPUs on the node pool, or the pod is requesting more GPUs than expected.
Pod schedules, but nvidia-smi fails	Driver/device plugin not functioning correctly, or the container image lacks required NVIDIA tooling.
Pod won’t schedule onto GPU nodes	GPU nodes are tainted but the pod lacks the required toleration, or node selectors/affinity prevent scheduling.
GPU resources not visible inside the container	NVIDIA device plugin not running correctly, or the container image is missing necessary NVIDIA libraries.

Upgrade

October 01, 2026

Upgrade SecurSpaces by running the Docker-based installer, preparing the internal database if you use
Percona, and applying the Helm command printed by the installer.

For an upgrade from SDS 2026.9 to the chart with Percona Operator 1.23.0, run the supplied Percona upgrade
script **before** Helm. Percona supports operator upgrades one minor version at a time. The script takes the database through 1.21.2 and 1.22.0; Helm completes the upgrade to 1.23.0.

Deployments using external MongoDB follow the usual installer and Helm flow without the Percona step.

Prerequisites

Before starting the upgrade, ensure the following:

- A recent backup of the SecurSpaces configuration database. See Back up the database.
- Access to the terminal with Docker installed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Current working directory `${ PWD }` contains the correct configuration file for your existing deployment.
- Necessary permissions to run Docker and apply Helm upgrades to your cluster.
- Kubernetes context is correctly configured.
- For internal Percona, `kubectl` and permission to apply cluster-scoped custom resource definitions (CRDs),
  update the operator Deployment and database custom resource, and access database pods for the backup.
- Space in the working directory for the `mongodump` archive written by the upgrade script.
- Only the intended Percona operator watches this database. Check for other operators watching all namespaces.

Run the Installer

Launch the installer for the target release. Replace `<target-version>` with the installer version that
supplies the chart with Percona Operator 1.23.0:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
  strongnetwork/strong_installer:<target-version>
```

> **Note:**
>
> `${ PWD }` refers to your current working directory. This directory must contain the configuration file used in your current deployment.

Execute the Upgrade Command

Once inside the Docker container, run the upgrade command using your existing configuration file:

```
sds-cli upgrade -c config_<your-current-version>.yaml
```

Example:

```
sds-cli upgrade -c config_2026.9.0.yaml
```

The installer downloads the new chart, updates the configuration, prints the usual `helm upgrade --install ... release` command, and pushes the images. There are no new Percona questions
in `sds-cli upgrade`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If `platform.deployPerconaMongoDB` is unset or **`true`**, it also copies the Percona upgrade kit to
`/strong-network/shared/percona-upgrade/` and prints the upgrade, restore, and backup-delete instructions.
The kit contains three scripts and the CRDs for 1.21.2 and 1.22.0. A rerun replaces this copy so that the scripts match the installer version. The directory is available as `./percona-upgrade/` on the host through
the shared-directory mount.

For external MongoDB (`platform.deployPerconaMongoDB: false`), the Percona kit and instructions are skipped.

Prepare internal Percona MongoDB for 1.23

Run this step after `sds-cli upgrade` finishes and before the printed Helm command. Use a terminal with
access to the cluster, and work from the shared directory containing the chart and `percona-upgrade/`.

The script takes the namespace, Helm release name, and chart archive path, in that order:

```
./percona-upgrade/upgrade-percona.sh <namespace> <release> ./
   ninjahchart-<version>.tgz
```

For namespace **`default`** and release `release`:

```
./percona-upgrade/upgrade-percona.sh default release ./ninjahchart-<
   version>.tgz
```

The script:

1. Exits with “nothing to do”if the database is already on 1.23.
2. Checks that the supplied chart contains the 1.23 CRDs.
3. Asks you to enter `yes`, waits for the database to be `ready`, and writes a `mongodump` backup to the
   current directory. Keep this archive until you have validated the platform.
4. Upgrades through the intermediate versions below. At each step it applies that version’s CRDs, sets
   the operator image, patches the database custom resource (`crVersion`, MongoDB image, and PBM image),
   and waits for `ready` with every database pod on the new image.
5. Applies the 1.23 CRDs from the chart and scales the old operator to `0`, ready for Helm to start 1.23.0.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Stage	Operator and database `crVersion`	MongoDB image version
Starting deployment (SDS 2026.9)	1.20.1	7.0.26-14
First script step	1.21.2	7.0.28-15
Second script step	1.22.0	7.0.30-16
Final Helm upgrade	1.23.0	7.0.37-20

CRDs must be applied outside Helm. Helm 3 does not upgrade existing CRDs. Running the script performs
these explicit CRD updates, including the 1.23 CRDs needed before the final Helm upgrade.

Each version step is a rolling restart. Expect a write pause of up to about 15 seconds per step; plan the upgrade window accordingly.

Apply the Helm Upgrade

After the Percona script completes successfully, run the Helm command printed by the installer. For external MongoDB, run it directly after the installer finishes.

Helm starts Percona Operator 1.23.0, which moves the database to `crVersion: 1.23.0`, MongoDB
`7.0.37-20`, and PBM `2.15.0`.

The chart blocks an existing Percona deployment that has not reached 1.22 or lacks the required 1.23 CRDs.
An already-upgraded 1.23 deployment can undergo subsequent Helm upgrades.

Post-Upgrade Verification

Once the Helm upgrade is applied:

- Verify that all pods are running and healthy:

```
kubectl get pods -n <your-namespace>
```

- Check service availability and logs:

```
kubectl logs <pod-name> -n <your-namespace>
```

- Confirm that the platform version has been updated successfully.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- For Percona, confirm the database is `ready`, `crVersion` is `1.23.0`, and database pods use MongoDB
  `7.0.37-20` and PBM `2.15.0`:

```
kubectl get psmdb <release>-psmdb-db -n <namespace>
kubectl get psmdb <release>-psmdb-db -n <namespace> -o yaml
kubectl get pods -n <namespace> -o wide
```

- Sign in as the platform administrator and confirm organizations, projects, and services are available.
- If backups are enabled, confirm that they continue to complete and the storage, schedule, and PITR
  settings are retained. For keyless GCS backups, see
  Back up the database.

Clean up or restore the upgrade backup

After platform validation, delete the upgrade backup from the directory where you ran the script:

```
./percona-upgrade/delete-percona-backup.sh
```

If you need to recover the data, use the restore script with the archive created during the upgrade:

```
./percona-upgrade/restore-percona-backup.sh default release <backup>.
   archive.gz
```

Replace **`default`** and `release` with your namespace and Helm release. This restores **data only**; it does
not downgrade the operator, CRDs, MongoDB, or PBM. Follow the quiesce and validation guidance in Restore the database.

Troubleshooting

Helm blocks the Percona upgrade

Running Helm before preparing Percona can produce the following error (wrapped here for readability):

```
Error: UPGRADE FAILED: execution error at (ninjahchart/templates/
   percona-instance.yaml:14:4):
The Percona MongoDB database protein-psmdb-db is on operator version
   1.20.1 with CRDs unknown.
This chart needs it on 1.22 or later with CRDs 1.23:
run percona-upgrade/upgrade-percona.sh from the SDS installer first.
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Run `upgrade-percona.sh` with the correct namespace, release, and new chart, then rerun the printed Helm
command. Do not skip the intermediate operator versions or the explicit CRD updates.

Conflicting Percona operators

If the database does not settle at the expected version or become ready, check whether another Percona
operator watches the same namespace, including operators configured to watch all namespaces. A leftover
operator can reconcile the database back to a competing configuration. Identify and scale the unintended
operator to `0` before continuing.

Other upgrade issues

If you encounter issues during the upgrade:

- Review the installer output for error messages.
- Ensure your configuration file matches the expected format.
- Check Docker and Kubernetes logs for additional context.

Administer

August 19, 2026

This section is for **platform administrators** running Citrix SecurSpaces™ for the whole organization. It
covers everything configured once and applied everywhere: who can sign in, what workspaces are allowed to do,
which external systems are connected, and how the platform is licensed and monitored.

Most tasks here need platform administrator rights. Some security-sensitive areas need the
security officer role.

If you are setting the platform up for the first time, install it first —see
Deploy.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Start here

Task	Page
Find the platform-level settings and what each area covers	System Configuration
Upload a license and turn platform features on	Licensing
Manage platform configuration as code	Terraform provider
Allow AI coding agents safely	Run AI coding agents in workspaces

Getting people in

Task	Page
Connect an identity provider and sync users	Identity and access
Grant platform-wide roles such as security officer	Platform roles
Control who can reach the platform	User access control

Setting the rules

Task	Page
Configure platform-wide options and defaults	Platform settings
Set what workspaces are allowed to do	Workspace policy
Control when workspaces pause and are deleted	Workspace lifecycle and idle detection
Restrict data movement and enforce security options	Security settings

Connecting other systems

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Task	Page
Connect repositories, registries, and third-party tools	Integrations
Provide registry access and IDE versions	Platform resources
Prepare storage for mount points	Storage

Running the platform

Task	Page
Review licensing, analytics, and usage	Operations
Size, back up, and restore the database	The SecurSpaces database
Review audit events and forward them to a SIEM	Audit
See how resources are allocated and consumed	Insights
Serve workspaces from more than one region	Regions

System configuration

System Configuration is where you manage Citrix SecurSpaces™ at the platform level. From one place, you handle your license and its features, assign platform roles, connect identity providers, turn on integrations, set up regions, and adjust a few other platform-wide options.

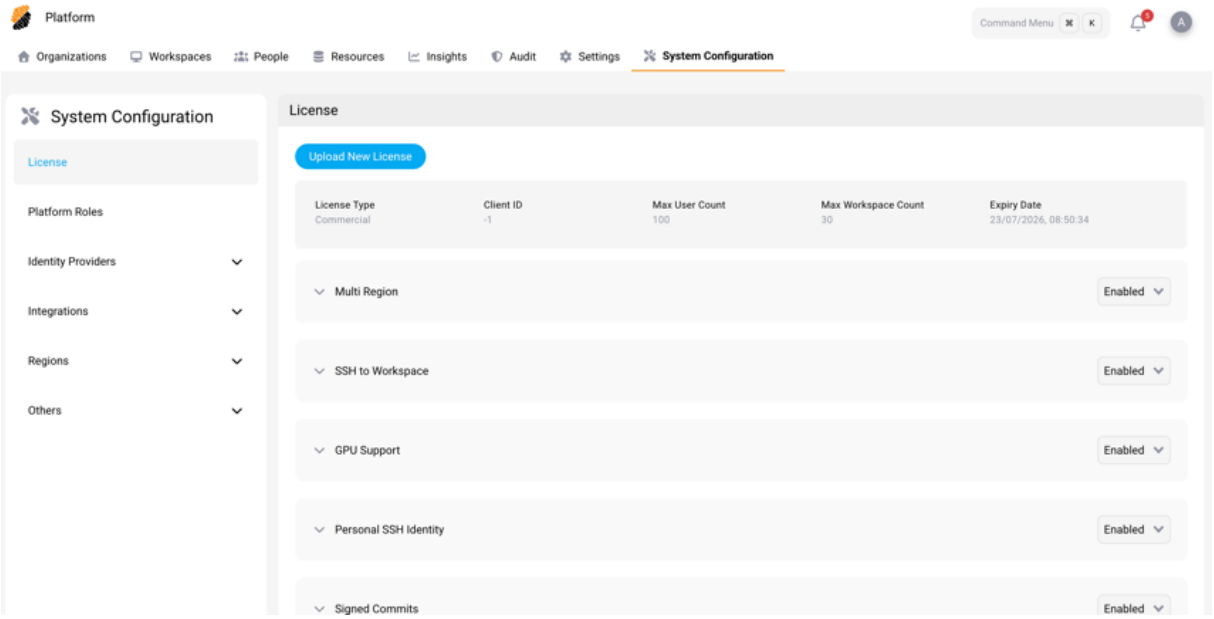
> **Note:**
>
> Only platform administrators can open System Configuration. It's separate from **Platform Settings**, which covers other platform-wide options such as general settings, security settings, and analytics.

Open System Configuration

Sign in as a platform administrator and select **System Configuration** in the top navigation bar. It opens on the **License** page.

To go straight there, browse to `https://<your-sds-hostname>/platform/system_configuration`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Configuration areas

The left navigation groups everything into these areas:

Area	What you do here
License	Upload your license, review what it covers, and turn platform features on or off.
Platform Roles	Give users and groups platform-wide roles, such as security officer.
Identity Providers	Connect OAuth, SAML, and SCIM providers so users sign in and sync automatically.
Integrations	Connect code repositories, third-party applications, storage, and Citrix DaaS™.
Regions	Add regions and warm nodes for multi-region deployments.
Others	Set up the email gateway, data breach contact, SSH server key, terms of service, and project labels.

Related information

- Platform roles

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Platform Settings
- Roles and permissions

Terraform provider

August 21, 2026

Citrix SecurSpaces™ publishes a Terraform provider so that platform configuration can be managed as code
rather than through the console. It is useful when you create projects and templates repeatedly, when configuration has to be reviewed before it is applied, or when an audit needs a record of who changed what.

The provider manages platform objects. It does not deploy the platform itself —for that, see Install with the Strong Installer CLI.

What it manages

Resource	Manages
`strong_user`	Platform users
`strong_organization`	Organizations and their owner
`strong_project`	Projects, their owner, and their members
`strong_user_group`	User groups and their membership
`strong_workspace_template`	Workspace templates, including specifications, images, scripts, secrets, apps, and security settings

Get the provider

The provider is distributed through your SecurSpaces deployment rather than the public Terraform Registry, so
you download it with the same installer container you use for upgrades. Access is tied to your platform entitlement.

1. Run the installer container as described in
   Install with the Strong Installer CLI.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Fetch the provider:

```
sds-cli get-terraform -d <your-sds-hostname>
```

The newest version is downloaded into the shared directory. Add `-v <version>` to select a specific one,
or `--verbose` to list what is available first.

3. Install it where Terraform looks for local providers, and mark it executable:

```
mkdir -p ~/.terraform.d/plugins/strong.network/strong-network/
    strong/<version>/linux_amd64/
cp terraform-provider-strong-<version> \
  ~/.terraform.d/plugins/strong.network/strong-network/strong/<
      version>/linux_amd64/terraform-provider-strong
chmod +x ~/.terraform.d/plugins/strong.network/strong-network/
    strong/<version>/linux_amd64/terraform-provider-strong
```

> **Note:**
>
> Because the provider is installed from the filesystem rather than a registry, `terraform init` does not
> verify a checksum for it. Fetch it through `sds-cli` rather than copying it between machines, and keep the
> version you deployed recorded with the rest of your infrastructure code.

Configure the provider

Declare the provider and point it at your deployment:

```
terraform {

  required_providers {

    strong = {

      source  = "strong.network/strong-network/strong"
      version = "<version>"
     }

   }

 }


provider "strong" {

  api_token      = var.api_token
  deployment_url = var.deployment_url
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

Argument	Required	Description
`api_token`	Yes	A platform API key. Create one under **Profile > Security > API Keys**
`deployment_url`	Yes	The URL of your SecurSpaces deployment

The token is marked sensitive by the provider, so Terraform does not print it. Supply it through a variable
or the environment rather than writing it into a file, and give it only the permissions the configuration needs —see Roles and permissions.

Resources

`strong_user`

Argument	Type	Required	Description
`email`	string	Yes	The user's email address
`user_type`	integer	Yes	`1` for an administrator, `3` for a developer
`identity_provider`	integer	Yes	The identity provider, as a numeric value. `1` Google, `2` Microsoft, `3` SAML, `7` generic OIDC, `0` none
`full_name`	string	No	The user's full name

`strong_organization`

Argument	Type	Required	Description
`name`	string	Yes	The organization name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Argument	Type	Required	Description
owner_id	string	Yes	The user ID or email of the owner

strong_project

Argument	Type	Required	Description
name	string	Yes	The project name
owner_id	string	Yes	The user ID or email of the owner
organization_id	string	Yes	The organization the project belongs to
member	block list	No	Project members. Each block takes id and role

strong_user_group

Argument	Type	Required	Description
name	string	Yes	The group name
description	string	No	A description of the group
members	set of strings	No	User IDs in the group

strong_workspace_template

The largest resource. It covers what a template defines in the console, including:

- **Identity and scope** —name, project_id, version, description, region_id
- **Image** —workspace_image with id and tag
- **Resources** —workspace_specs with cpu, memory, and disk
- **Startup** —before_startup_script, after_startup_script, default_folder

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Secrets** —`injected_secrets_as_env`, `injected_secrets_as_file`, and local equivalents taking `secret_name` and `content`
- **Applications** —`workspace_apps` with `port`, `name`, and `use_https`
- **Security** —`policy_ids` for network policies, `apps_security`, `personal_ssh_identity`, `workspace_access_items`
- **Clipboard** —`clipboard_settings`, covering `monitor`, `clipboard_restricted`, paste restrictions, character limits, and `enable_supervised_copy`
- **Schedule** —`workspace_schedule`, `timeout_outside_schedule`, `idle_timeout`

For what each setting does, see Workspace templates.

Example

Creating an organization, a project inside it, and a group:

```
resource "strong_user" "lead" {

  email             = "team-lead@example.com"
  user_type         = 3
  full_name         = "Team Lead"
  identity_provider = 3
 }


resource "strong_organization" "engineering" {

  name     = "engineering"
  owner_id = strong_user.lead.id
 }


resource "strong_project" "platform" {

  name            = "platform-services"
  owner_id        = strong_user.lead.id
  organization_id = strong_organization.engineering.id
 }


resource "strong_user_group" "reviewers" {

  name        = "reviewers"
  description = "Engineers who review platform changes"
  members     = [strong_user.lead.id]
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

Apply it in the usual way:

```
terraform init
terraform plan
terraform apply
```

Every resource supports import, so you can bring objects that already exist under Terraform management with
`terraform import`.

Related information

- SDS CLI reference
- API
- Workspace templates
- Roles and permissions

Run AI coding agents in workspaces

August 21, 2026

Developers increasingly run AI coding agents that edit files, execute commands, install packages, and call
external services on their behalf. An agent doing that on a laptop is largely ungoverned. Running it inside a
Citrix SecurSpaces™ workspace puts the same controls around it that already apply to the developer.

Agents are used two ways, and the difference decides how much of this page applies to you:

Mode	What it looks like
Attended	A developer opens their workspace in the Cloud IDE or a local IDE and runs an agent from inside it, watching what it does
Unattended	A pipeline or an orchestrator creates a workspace, an agent runs on its own, and the result is collected afterwards

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Attended use is where most teams start, and it needs no special configuration: the workspace a developer
already has applies its controls to the agent automatically. Unattended runs need more decisions, and most of
this page is about those.

This page is for platform administrators. It covers what an agent can reach, how to keep concurrent runs from
colliding, what it costs, and what you can see afterwards.

Why a workspace is the right boundary

An agent inherits the environment it runs in. On an endpoint that means the developer's credentials, their
whole filesystem, and unrestricted network access.

In a workspace, the boundary is already drawn:

- **Credentials are brokered, not held.** Repository and service credentials are injected at runtime and
  stay with the platform. An agent uses them without being able to read them out and without them ever
  landing on the endpoint.
- **Egress is policy-controlled.** An agent reaches only the destinations the workspace's network policy
  allows, so an agent that decides to fetch a package from an unexpected host does not get to.
- **The filesystem is scoped.** The agent sees the workspace, not the developer's machine.
- **Everything is audited.** Agent activity produces the same audit events as developer activity.

Attended agents

A developer runs an agent inside the workspace the same way they would on a laptop: open the workspace,
start Copilot, Claude Code, Cursor, or whatever the team uses, and work alongside it. Both connection
methods support this —the Cloud IDE in a browser tab, or a local IDE connected over SSH. See
Choose how you connect.

Nothing extra is required to govern it. The agent runs as the workspace user, so it inherits the network policy, the brokered credentials, the role, and the clipboard rules that already apply to that developer, and its AI traffic is attributed like any other. An administrator decides those controls once, in the template; the developer does not opt in and cannot opt out.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Two things are worth setting up so that developers are not fighting the environment:

- **Agent tooling belongs in the image or a startup script**, not in each developer's manual setup. CLI
  agents installed at a system path, and VS Code extensions supplied as `.vsix` files, are available the
  moment a workspace starts. See
  Manage workspace images.
- **The model endpoint has to be reachable.** An agent that cannot reach its provider fails in ways that
  look like a broken workspace. Include the endpoint in the network policy, or route it through the
  gateway.

> **Note:**
>
> Data protection controls are weaker when a developer connects a local IDE over SSH than when they use the
> Cloud IDE, because content leaves the browser session. Where that matters, govern it by policy. See
> Layered controls.

Unattended agents

The rest of this section concerns runs that nobody is watching. If your developers only use agents interactively, the controls above are already doing their job.

Isolate one run from another

Agents running in parallel that share a working directory interfere with each other: competing edits, clashing
ports, and dependency changes made for one task breaking another.

Give each concurrent run its own workspace. A workspace is a container with its own filesystem, its own
process space, and its own network identity, so two runs cannot collide.

Create each one from a workspace template, so
that the image, resources, credentials, scripts, and policy are defined once and every run starts identical.

Templates can be managed as code with the
Terraform provider. The provider manages

© 1997–2026 Citrix Systems, Inc. All rights reserved.

templates, projects, users, groups, and organizations —it does not create individual workspaces, so per-run
creation uses the API or the UI.

The REST API creates workspaces when a pipeline or an
orchestrator does the triggering:

Endpoint	Purpose
`POST /v1/projects/{ project_id } /workspaces/template`	Create a workspace from a template
`POST /v1/projects/{ project_id } /workspaces`	Create a workspace from an explicit specification
`POST /v1/projects/{ project_id } /workspaces/{ ws_id } /state`	Start or pause an existing workspace
`DELETE` on the workspace	Discard the run

Set the state on create so the workspace comes up running rather than paused, and delete it when the run
finishes.

> **Note:**
>
> Isolation is per workspace, not per session inside one. Two agents in the same workspace share a filesystem
> and will interfere. If concurrency matters, create a workspace for each run.

Start the agent inside the workspace

Creating a workspace does not run anything. SecurSpaces has no API for executing a command in a running
workspace, so the agent has to be started from inside. There are two ways to arrange that for a run nobody
is attending.

A startup script. A workspace runs a pre-startup script before the IDE comes up and a post-startup script
once the workspace is ready. Put the agent invocation in the post-startup script and the workspace begins the
run on its own, with no external trigger. This is the option to use for unattended runs. Reserve the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

pre-startup script for preparation the agent depends on, not for the agent itself, because a failure there can stop the workspace coming up at all.

Startup scripts can be set at three levels, and for agent workloads two of them matter:

Level	Effect
Template	Every workspace created from the template runs the same script. Use when the runs are alike
Workspace	The script is set on the individual workspace at creation. Use when each run needs a different command

The workspace level is what lets an orchestrator vary the work per run. Creating a workspace from a template
inherits the template's script, whereas creating one from an explicit specification accepts a startup script
in the request, so the caller can supply the command for that run without maintaining a template for every
variation.

For how to configure them, see
Where to configure startup scripts.

SSH. Retrieve the workspace's connection details from the API, connect, and run the command. Use this
when the orchestrator needs to decide what to run after the workspace exists, or to collect results when the
run ends. See Connect over SSH.

A pipeline-driven run therefore looks like this: create the workspace with a post-startup script that launches the agent, let the run proceed, collect the result, delete the workspace.

> **Important:**
>
> Startup scripts run on every workspace start, not only on the first one. If idle detection pauses a workspace and it is later resumed, the post-startup script runs again. Make the script idempotent, or have
> it check whether the run has already completed, so that resuming a workspace does not silently start the
> work a second time.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Decide what an agent can reach

An agent is only as constrained as the workspace it runs in, so build the constraint into the template rather
than relying on the agent to behave.

Network egress. Attach a network policy that
allows only the destinations the work needs —your repositories, your artifact registry, your model endpoint.
An agent cannot exfiltrate to a host the policy does not permit, whatever it is asked to do.

Credentials. Use project resources rather
than personal secrets for agent workloads, so the credential belongs to the project and can be rotated or
revoked without touching a person's account. Give the workspace only the resources that task requires.

Permissions. The workspace runs as a user with a role. Give agent workspaces a role with the narrowest
permissions that work —see Roles and permissions.

Data protection. Clipboard and workspace app controls apply to an agent session as they do to a developer
session. See Layered controls.

Control what it costs

Agent runs consume compute differently from people. They are bursty, they can run unattended, and a loop that
does not terminate will keep a workspace busy indefinitely. An attended agent is bounded by the developer's
own session, so this section matters most for unattended runs —with the exception of token spend, which
applies to both.

- **Idle detection** pauses a workspace that stops being used, which is the main protection against a
  forgotten run billing overnight. See
  Workspace lifecycle and idle detection.
- **Workspace specifications** cap CPU, memory, and disk per run. Size agent templates deliberately; an
  agent does not need a developer's full workstation.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Concurrency** is what drives node count, not the number of workspaces that exist. See Sizing.

> **Important:**
>
> An agent that runs unattended will not pause itself. Idle detection is what stops an abandoned run
> consuming a node until someone notices, so confirm it is enabled before you allow unattended runs.

See what happened

Two sources, answering different questions.

Audit records what the agent did on the platform: workspace start and stop, resource access, clipboard and
transfer events, and monitored network activity. See
Audit.

AI Gateway records what the agent spent. When AI traffic is routed through a gateway, SecurSpaces tags
every request with the user, project, and workspace it came from, so token consumption can be attributed
rather than arriving as a single unattributed bill. See
Configure AI Gateway routing.

That attribution is what makes agent usage manageable: without it you know the total and nothing else.

A workable starting point

If developers are running agents interactively, you are most of the way there already. Confirm three things
on the templates they use:

1. The network policy allows the model endpoint, as well as your repositories and registry.
2. The agent tooling is in the image or a startup script, rather than each developer installing it.
3. AI traffic is routed through the gateway, so consumption is attributed rather than anonymous.

For unattended runs, add:

1. A template for agent runs, with a specification that fits the workload rather than a developer's desktop.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Project-scoped credentials, and no more than the task requires.
3. Idle detection confirmed enabled, so an abandoned run does not hold a node.
4. The agent invocation in a post-startup script, written to be safe if the workspace is resumed.
5. One workspace per concurrent run, deleted when the run finishes.

Start with a small population and one team. The template is the thing to get right —once it is, every run
inherits it.

Related information

- Give your AI assistant access to the documentation
- Configure AI Gateway routing
- Network policies
- Templates
- Sizing
- Connect over SSH
- What persists in a workspace
- REST API

Configure platform login

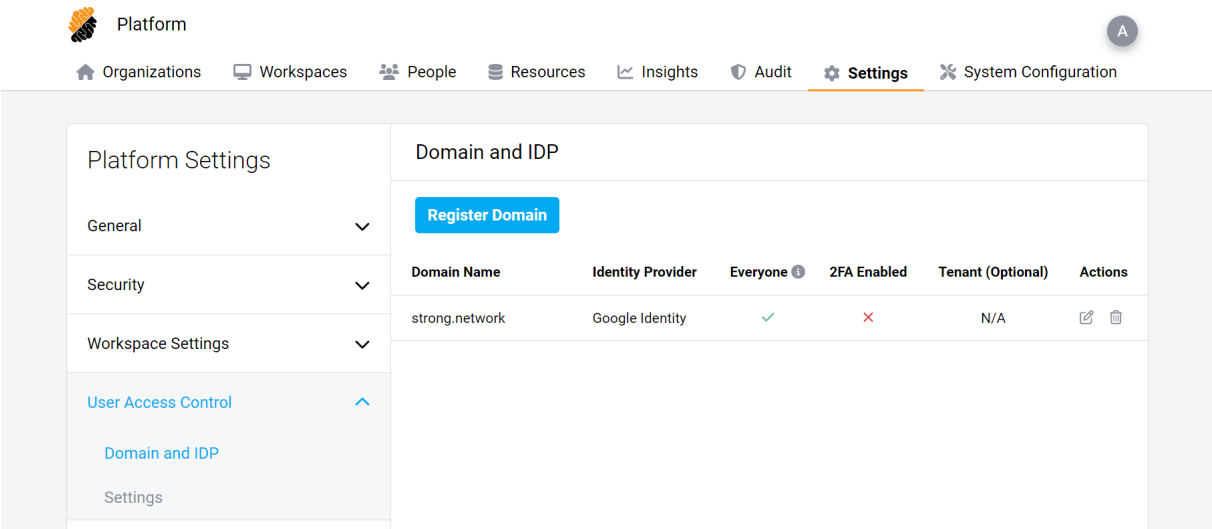
Configure Login for Users

There are five ways users can log in to the platform:

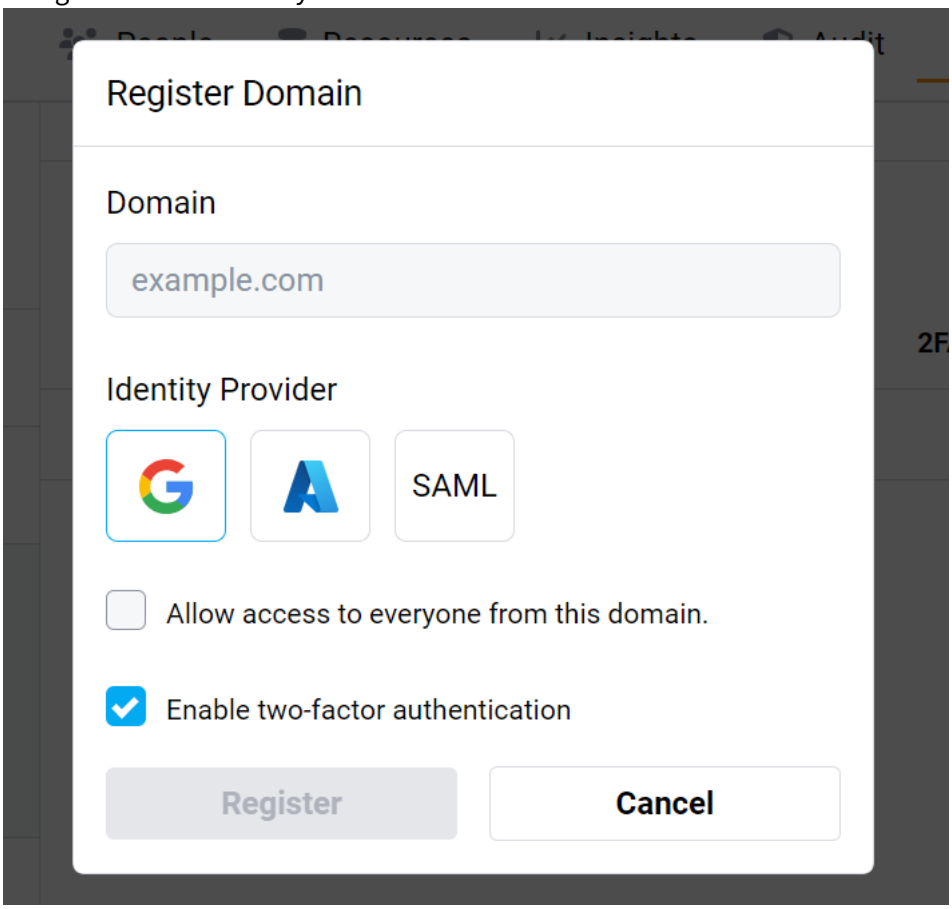
- Google OAuth provider
- Microsoft OAuth provider (Azure)
- SAML
- OpenID Connect
- Username and password

After configuring the Identity Provider of choice (any of the first 4 options), it can be used to authenticate users of specified domains. These can be configured under User Access Control, in the submenu "Domain and IDP".

© 1997–2026 Citrix Systems, Inc. All rights reserved.



If a domain is added, it means that when adding a user to the platform, that user will authenticate using the chosen Identity Provider.



In this menu, you may choose to check “Allow access to everyone from this domain”which will create user accounts on the fly, without the need to create the account beforehand. This is called Just-in-Time provisioning. This new user will not have any organization or project assigned to them.

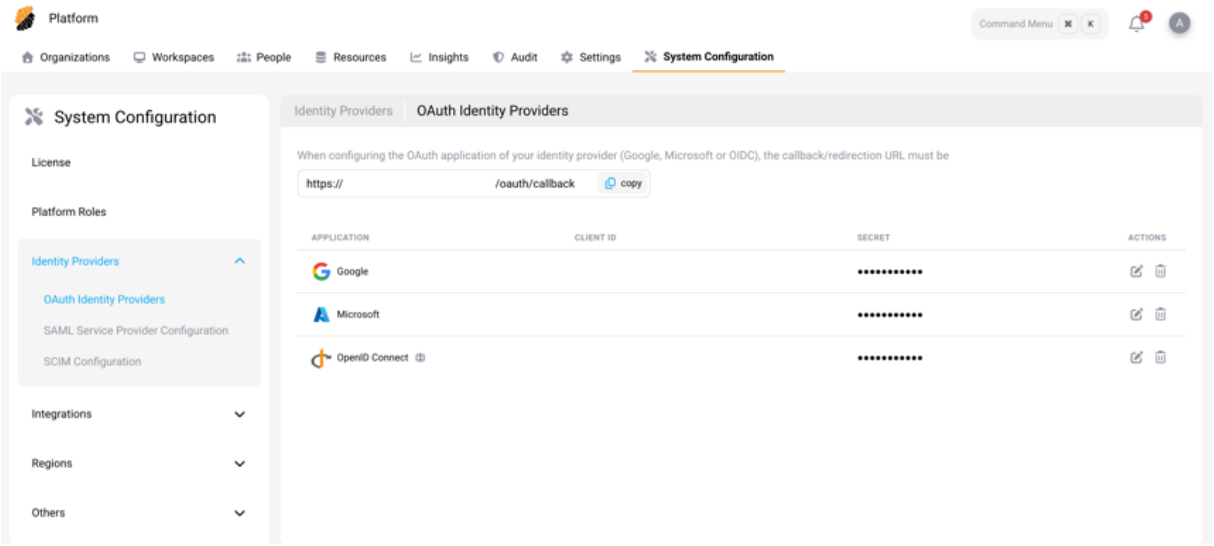
© 1997–2026 Citrix Systems, Inc. All rights reserved.

You may also enable two-factor authentication which will use OTP on any user from the specified domain. If two-factor authentication is desired, we recommend setting it up either in your Identity Provider or in the platform to avoid asking the user to do the process twice.

Identity providers

Let your users sign in with accounts they already have, and keep your user list in sync automatically. In **Identity Providers**, you connect Citrix SecurSpaces™ to your organization's identity systems.

Find these settings at **System Configuration > Identity Providers**.



You can set up three types of provider:

- OAuth Identity Providers
- SAML Service Provider Configuration
- SCIM Configuration

OAuth Identity Providers

Choose the OAuth providers your users sign in with, such as Google, Microsoft (Azure), or any OpenID Connect provider.

To set one up, follow the matching guide:

- Google OAuth
- Microsoft OAuth
- OpenID Connect

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SAML Service Provider Configuration

Connect SecurSpaces to your SAML identity provider so users authenticate through it. Upload your provider's metadata from a URL or an XML file, and the platform handles the rest.

For step-by-step instructions, see SAML.

SCIM Configuration

Keep users and groups current without manual work. With SCIM, your identity provider adds and removes users for you. Generate an authorization token here, then give it to your SCIM provider to link the two.

For step-by-step instructions, see SCIM.

Related information

- Configure Platform Login
- System Configuration

Platform roles

August 19, 2026

Most access in Citrix SecurSpaces™ is granted through project roles, which apply inside a single project.
Platform roles are different: they apply across the whole platform, whichever organization or project the
person belongs to.

Platform roles are assigned in
System Configuration, under
Platform Roles. Only a platform administrator can open that page.

Available roles

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Role	What it grants
Security officer	Access to security-sensitive areas such as system analytics and audit data. Only a platform administrator can assign or remove this role.
Data privacy officer	Names the contact for data privacy responsibilities on the platform.

The security officer role also carries privileges that project roles cannot grant. For example, a security officer can turn on **Bypass Schedule** for a workspace, so that it keeps running instead of pausing when idle. See
Workspace idle detection.

Assign a platform role

1. Sign in as a platform administrator and open **System Configuration**.
2. Select **Platform Roles**.
3. Select the role you want to assign.
4. Search for a user or group, and add them.

Remove a platform role

1. Open **System Configuration > Platform Roles**.
2. Select the role, then select the user or group.
3. Confirm the removal.

Related information

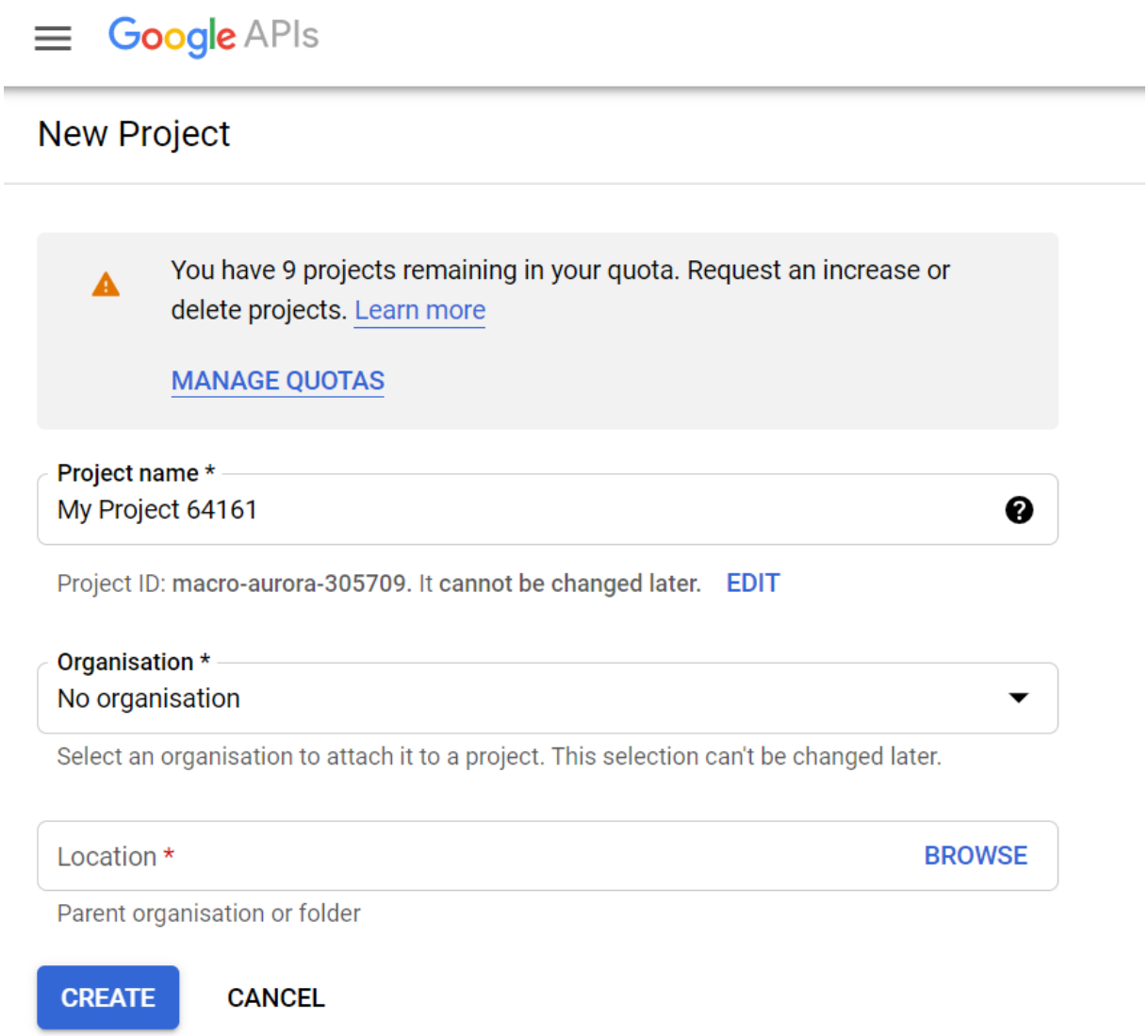
- System Configuration
- Roles and permissions
- User access control

Google OAuth

To create an OAuth Client to use Google as an Identity Provider, follow these steps to obtain the OAuth Client ID and Secret required in the platform configuration:

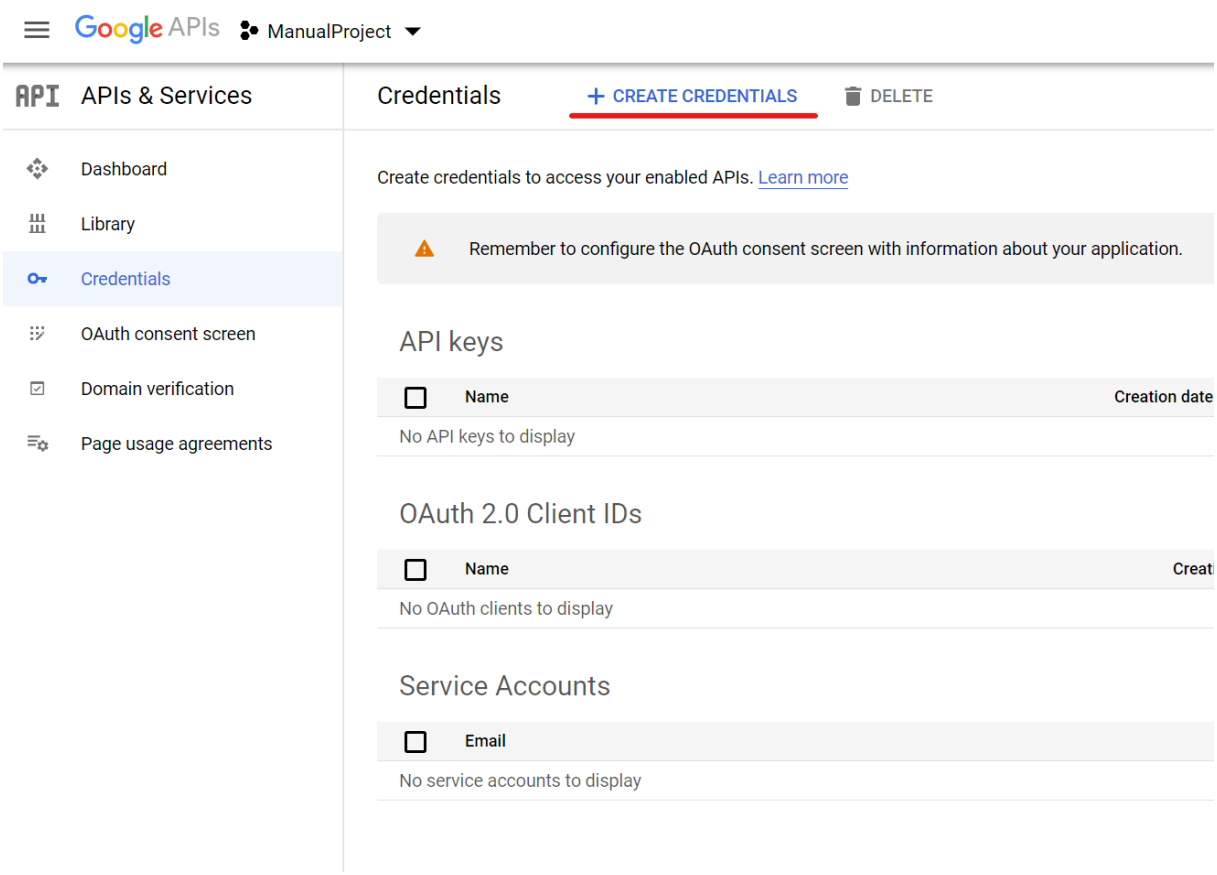
© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Go to the Google API Console and create a new project (or use an existing one). The project name, organization, and location are left at your discretion.



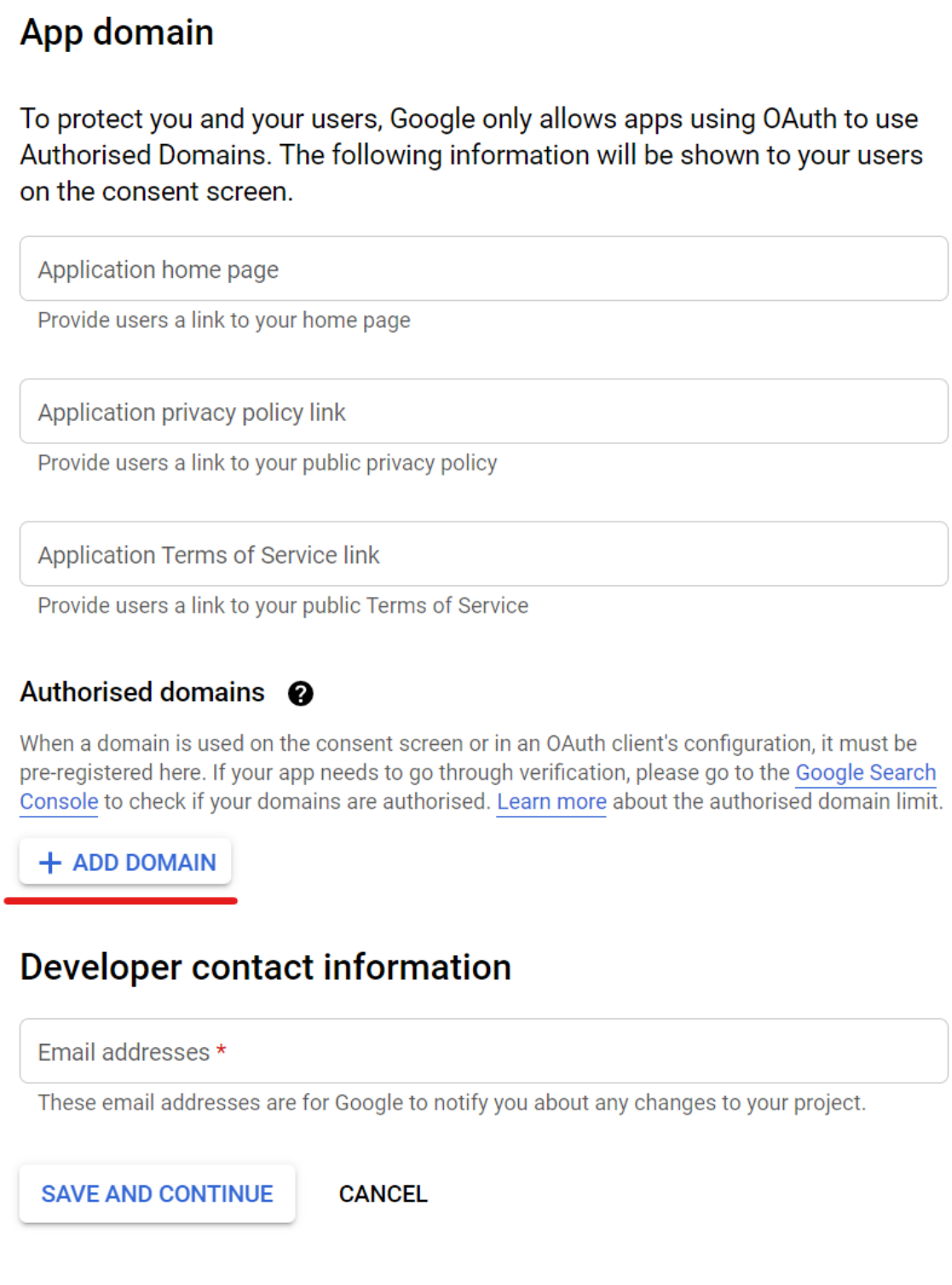
- Inside the project, click on "+ Create Credentials"and select "OAuth client ID"from the submenu.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



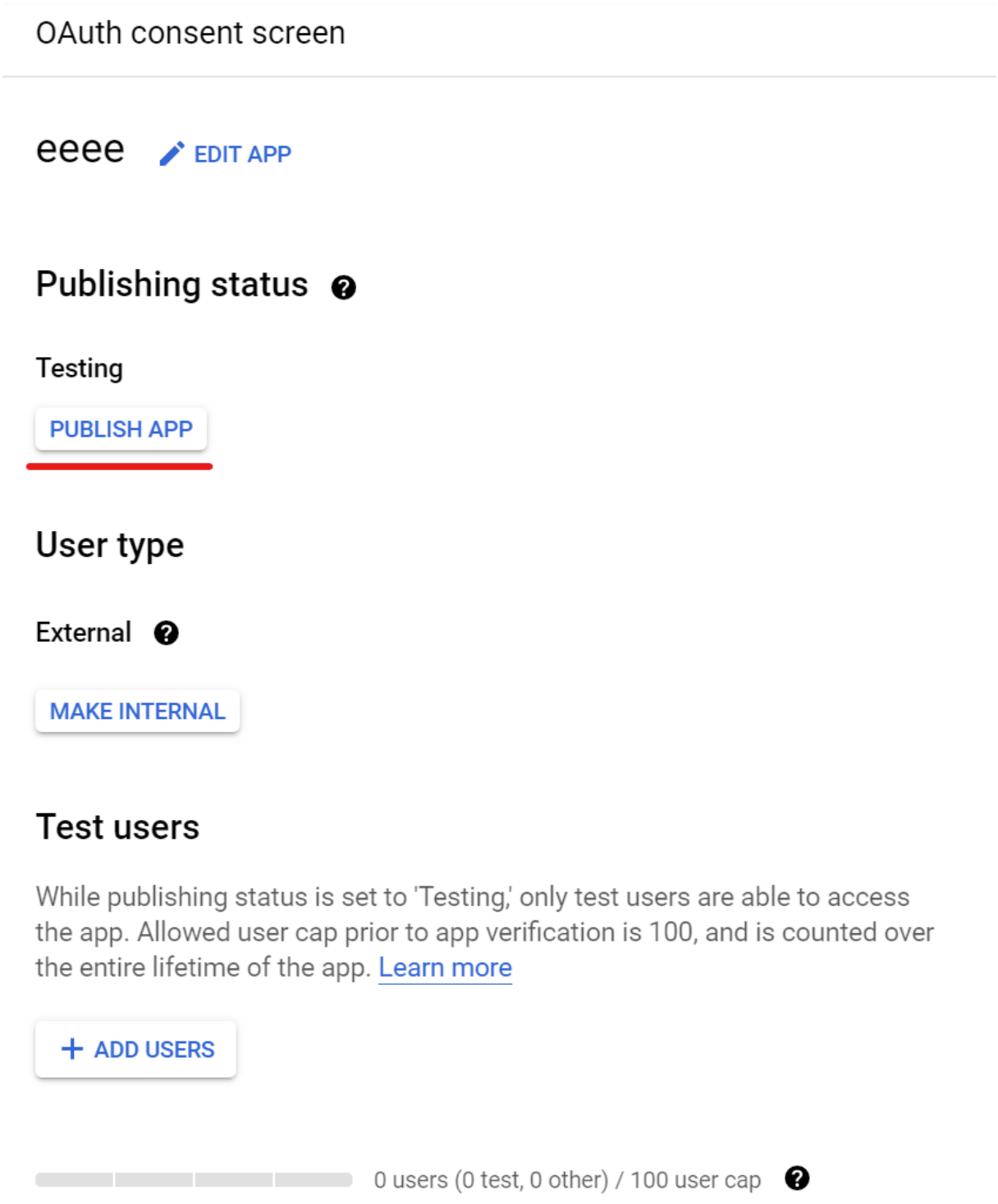
- You will be presented with a warning to first configure an OAuth consent screen. Click on it. Select an external consent screen and click create. Fill in the fields at your discretion. The app name will be seen by users trying to log in to the platform.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- In the authorized domain, specify the domain in which the platform is deployed.
- Click on “Save and Continue”in the following menus without adding anything until you reach the summary page, then click on “Back to Dashboard”.
- Click on “Publish App”.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

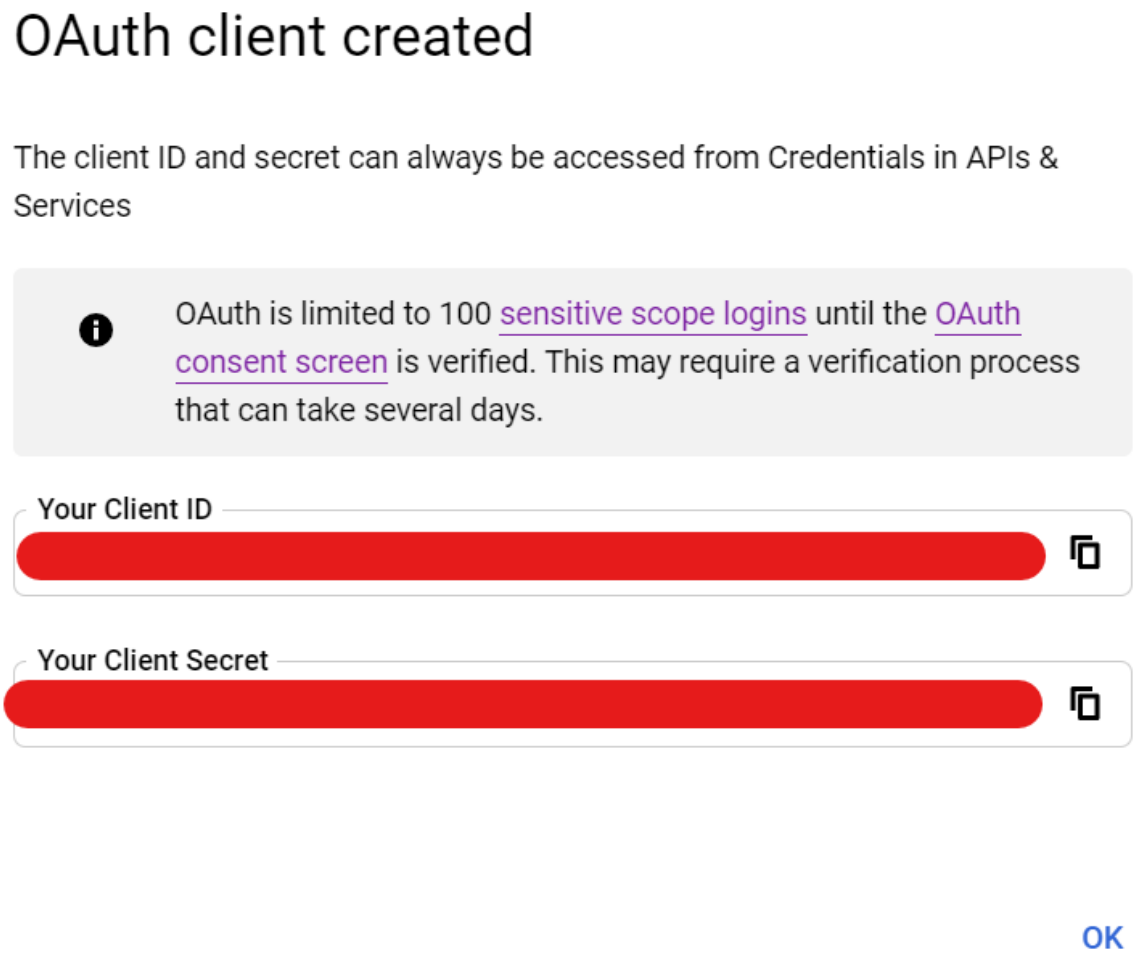


- Return to the Credentials page and create the credentials for an OAuth client ID.
- On this page, set the application type to “Web application”. The name is left at your discretion.
- In “Authorised JavaScript origins”, specify the domain name in which the platform is deployed. In “Authorised redirect URIs”, enter the redirect URL, similar to:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

    - `https://example.com/oauth/callback`
    - Where “example.com”should be set to the proper domain name.

- Click on “Create”and note the Client ID and Secret for the platform configuration.



Related information

- Microsoft Entra
- OpenID Connect
- Platform roles
- Identity providers
- SAML
- Identity and access

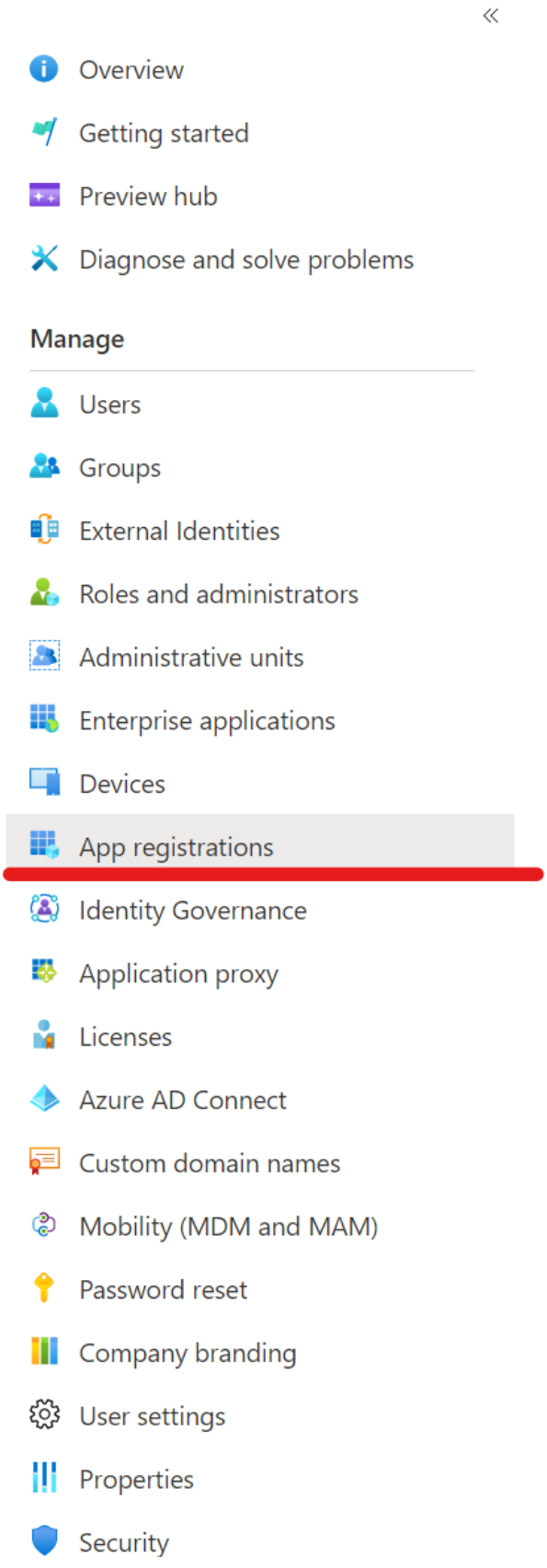
© 1997–2026 Citrix Systems, Inc. All rights reserved.

Microsoft Entra

The platform supports integration with Azure Active Directory for logging in with your Microsoft Azure account. To configure it:

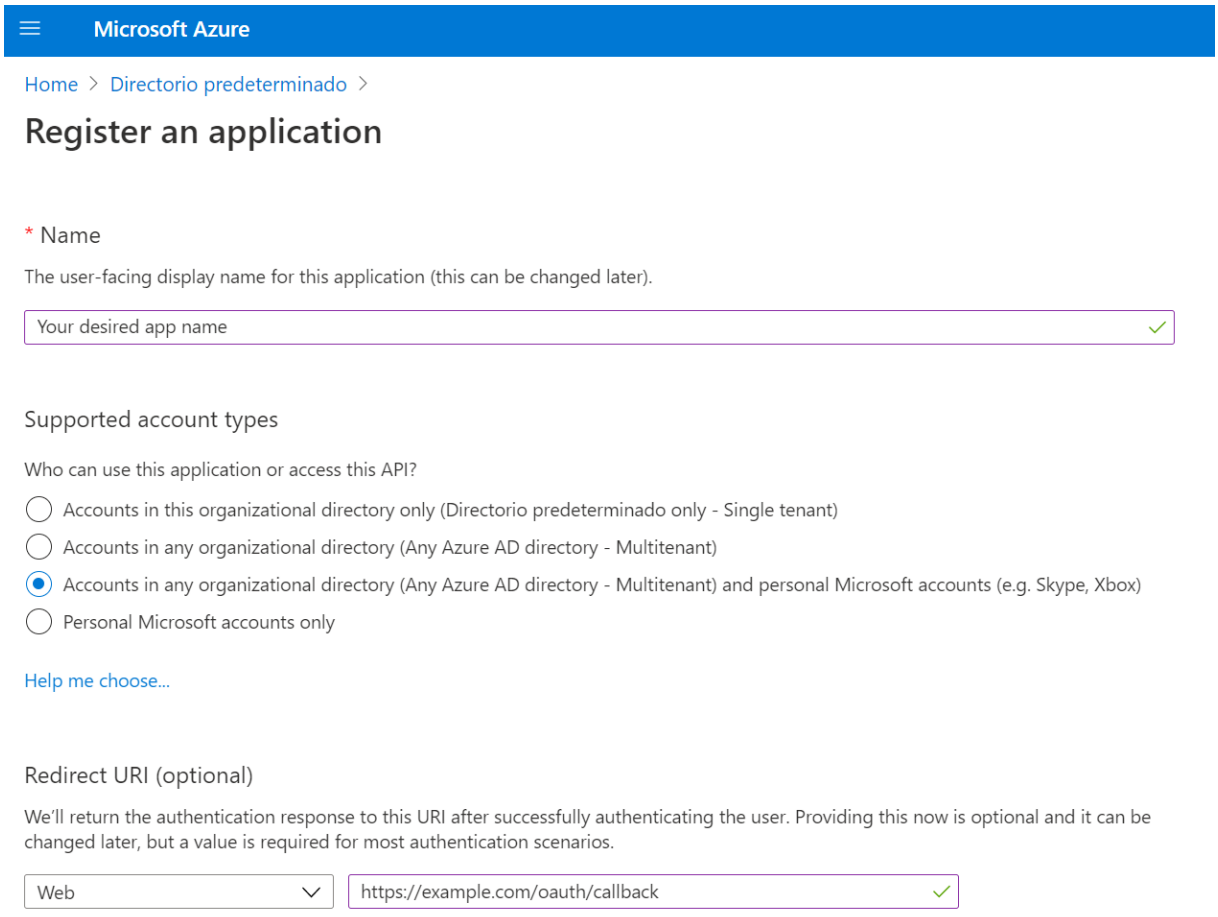
- Go to the Microsoft Azure portal.
- Navigate to the Azure Active Directory.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



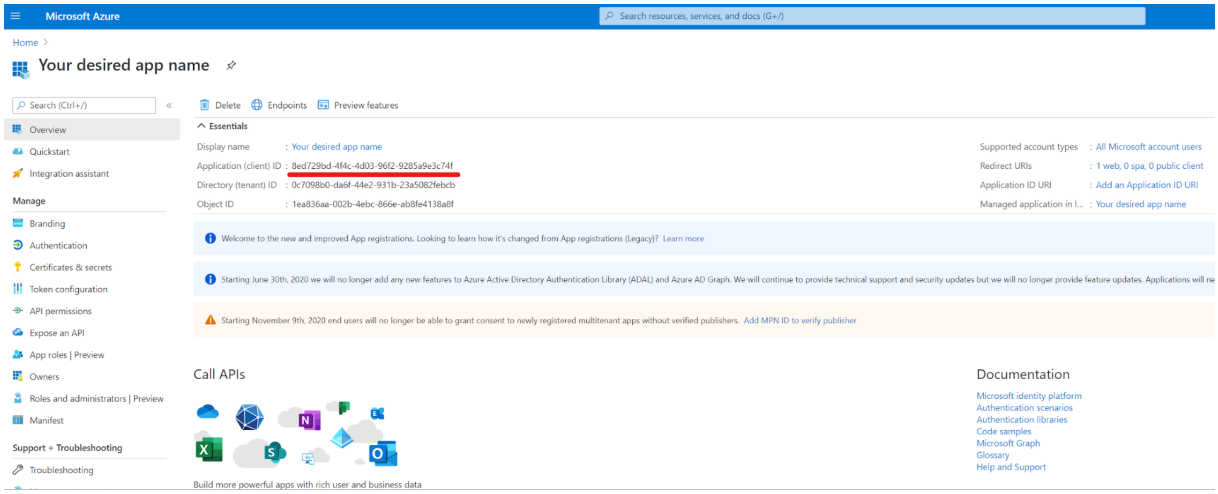
© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Click on “App registrations”and then “New registration”. Set the following:
- **App name:** Choose a name that will be publicly visible to users logging into the platform.
- **Supported account types:** We recommend selecting “Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g., Skype, Xbox)”to allow registered users to log in with their public domain accounts.
- **Redirect URI:** Set the selector to “Web”and enter a URI similar to `https://example.com/oauth/callback`.

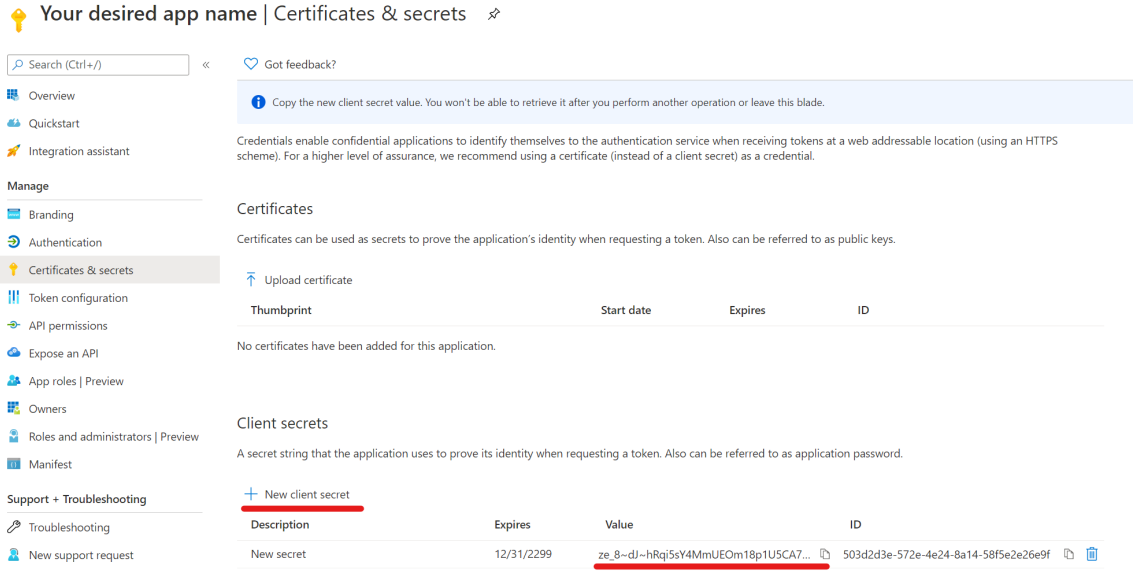


- Click on “Register”at the bottom.
- On the next page, note the OAuth Client ID for the platform configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- To obtain the secret, go to “Certificates & secrets”of the newly created app and click on “New client secret”. Enter this secret in the platform configuration to complete the setup.



Single Logout (SLO) for Microsoft Azure

To enable Single Logout for the OIDC flow with Azure, configure the following:

- To log out users from Microsoft when they log out of the Citrix SecurSpaces™ platform, add another URL in the Redirect URI section with just the domain name used by the SecurSpaces platform. This URL is used to redirect users back after they log out of their Microsoft accounts.
- Add the optional claim called “login_hint”to the ID token:
- Go to “Token configuration”and click on “Add optional claim”.
- Select ID as token type and then select “login_hint”.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- To log the user out of the SecurSpaces platform when they log out of their Microsoft account, add the optional claim called "sid"to the ID token type.
- Add a Logout URL under the "Authentication"menu with the structure `https://[domain_name]/auth/logout`, where `domain_name` is the domain under which you have the SecurSpaces platform. This endpoint will be called by Microsoft when a user logs out to also log out the user from the SecurSpaces platform.

Related information

- Google OAuth
- OpenID Connect
- Platform roles
- Identity providers
- SAML
- Identity and access

Connect an OpenID Connect provider

This platform supports integration with OpenID Connect for logging in.

Registering the Application

- Go to the OpenID Provider's Developer Portal.
- Navigate to the Applications or Clients section.
- Click on "Create New Application"or equivalent. Set the following:
- **App Name**: Choose a name that will be displayed to users logging in.
- **Application Type**: Select "Web Application".
- **Redirect URIs**: Add the following URI to handle login redirects: `https://example.com/oauth/callback`
- **Logout Redirect URI**: Add the following URI to handle logout redirects: `https://example.com/auth/logout`
- Save the application.

Note the Client ID and Client Secret generated during this process. These will be required for platform configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Configuring Scopes and Claims

Under the Scopes or Permissions section of your application, ensure the following scopes are included:

- `openid`
- `email`
- `profile`
- Any additional scopes your platform requires.

Configure claims if necessary. Common claims include:

- `sub`: Unique identifier for the user.
- `email`: User's email address.
- `name`: Full name of the user.
- `preferred_username`: Username or handle.

Enabling Single Logout (SLO)

To enable Single Logout (SLO) for OpenID Connect:

Navigate to the Advanced Settings or Logout Configuration section.

Enable Single Logout if supported by the provider.

Add the Logout Redirect URI configured earlier:

```
https://example.com/auth/logout
```

Optionally, add the following claims to the ID token:

- `sid`: Session identifier.
- `logout_hint`: Provides context for logging out.

Related information

- Google OAuth
- Microsoft Entra
- Platform roles
- Identity providers
- SAML
- Identity and access

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SAML

August 19, 2026

Citrix SecurSpaces™ acts as a SAML 2.0 service provider. Any conformant identity provider works — Okta,
Microsoft Entra ID, Ping, ADFS, and others —because the configuration is an exchange of metadata rather than
anything provider-specific.

Setting it up is a two-sided task: the identity provider needs to know about SecurSpaces, and SecurSpaces
needs to know about the identity provider.

Configure the identity provider

Create a SAML application in your identity provider with the following values, where <your-sds-hostname> is
the domain your platform is deployed on:

Setting	Value
Single Sign-On URL (Assertion Consumer Service URL)	https://<your-sds-hostname>/saml/acs
Audience URI (Service Provider Metadata URL)	https://<your-sds-hostname>/saml/metadata

If your identity provider prefers to import metadata rather than take values by hand, SecurSpaces publishes
its own service provider metadata. Use **Download metadata** on the SAML configuration page.

Attributes

The identity provider must send these attributes in the assertion:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Attribute	Type	What it does
`email`	**Required**	The email the user is registered with on the platform. Without it the connection does not work.
`displayName`	Recommended	The name shown for the user across the platform. When it is empty, `firstName` and `lastName` are used instead.
`firstName`	Optional	Used only when `displayName` is empty.
`lastName`	Optional	Used only when `displayName` is empty.
`uid`	Optional	A unique identifier, usually supplied so a user's actions can be traced across systems.

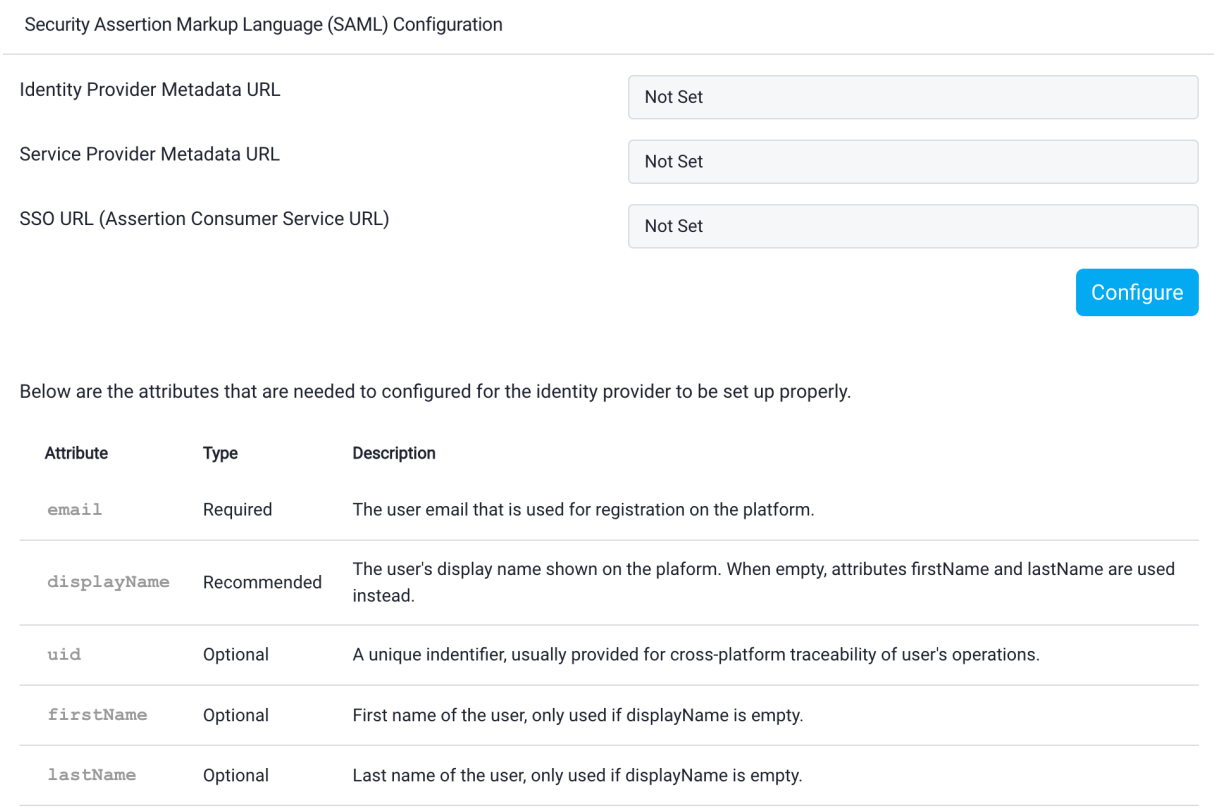
Send `displayName` if you can. Without it the platform assembles a name from `firstName` and `lastName`, and
if those are absent too, users are identified by email address throughout the console and the audit log.

Sending `uid` is worth doing wherever audit data is correlated with other systems, because email addresses
change and a stable identifier does not.

Configure SecurSpaces

1. Sign in as a platform administrator.
2. Go to **System Configuration > SAML Service Provider Configuration**, or browse to `https://<your-sds-hostname>/platform/system_configuration/saml_sp`.
3. Select **Configure**.
4. Provide the identity provider's metadata, either as an **Identity Provider Metadata URL** or by uploading its XML metadata file.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



A metadata URL is the better choice where your identity provider offers one, because certificate rotations
are picked up automatically rather than needing the file to be uploaded again.

Related information

- Identity providers
- SCIM —automatic user provisioning
- User access control
- Platform roles

SCIM configuration

September 21, 2026

The Citrix SecurSpaces™ platform adheres to the System for Cross-domain Identity Management (SCIM) 2.0
specification. Use SCIM for automatic provisioning, synchronization, and deprovisioning of users

© 1997–2026 Citrix Systems, Inc. All rights reserved.

and groups.
The SecurSpaces platform supports both the `/Users` and `/Groups` endpoints.

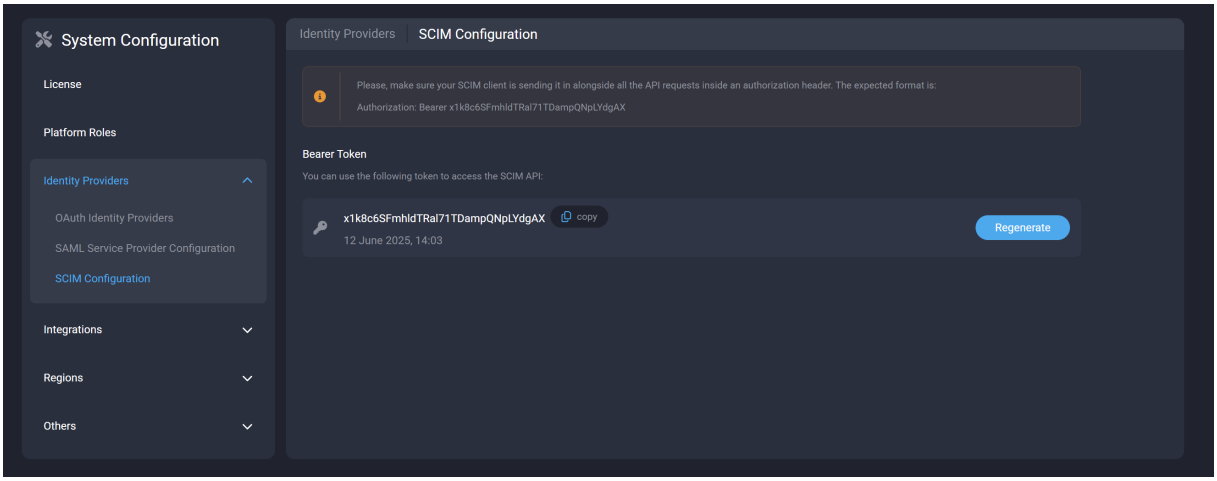
- Use the `/Users` endpoint to create, update, delete, and retrieve users. User responses include direct group
  memberships in the read-only `groups` attribute when the user belongs to one or more groups.
- Use the `/Groups` endpoint to create, update, delete, and retrieve groups. Group list responses include group
  members, so identity providers can validate memberships without requesting each group separately.

When your identity provider sends separate SCIM name fields, SecurSpaces maps `name.givenName` and
`name.familyName` to the user's first and last name. If the provider sends `name.formatted`, SecurSpaces uses
that formatted value as the user's display name.

Configure the SCIM provider

A token is required to authorize requests between your SCIM provider and the SecurSpaces platform. As an
administrator, you can obtain the token from **System Configuration > Identity Providers > SCIM Configuration**.

`https://<your-sds-hostname>/system_configuration/identity_providers/scim`



Ensure that your SCIM provider includes this token in all API requests, using the following authorization header
format:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

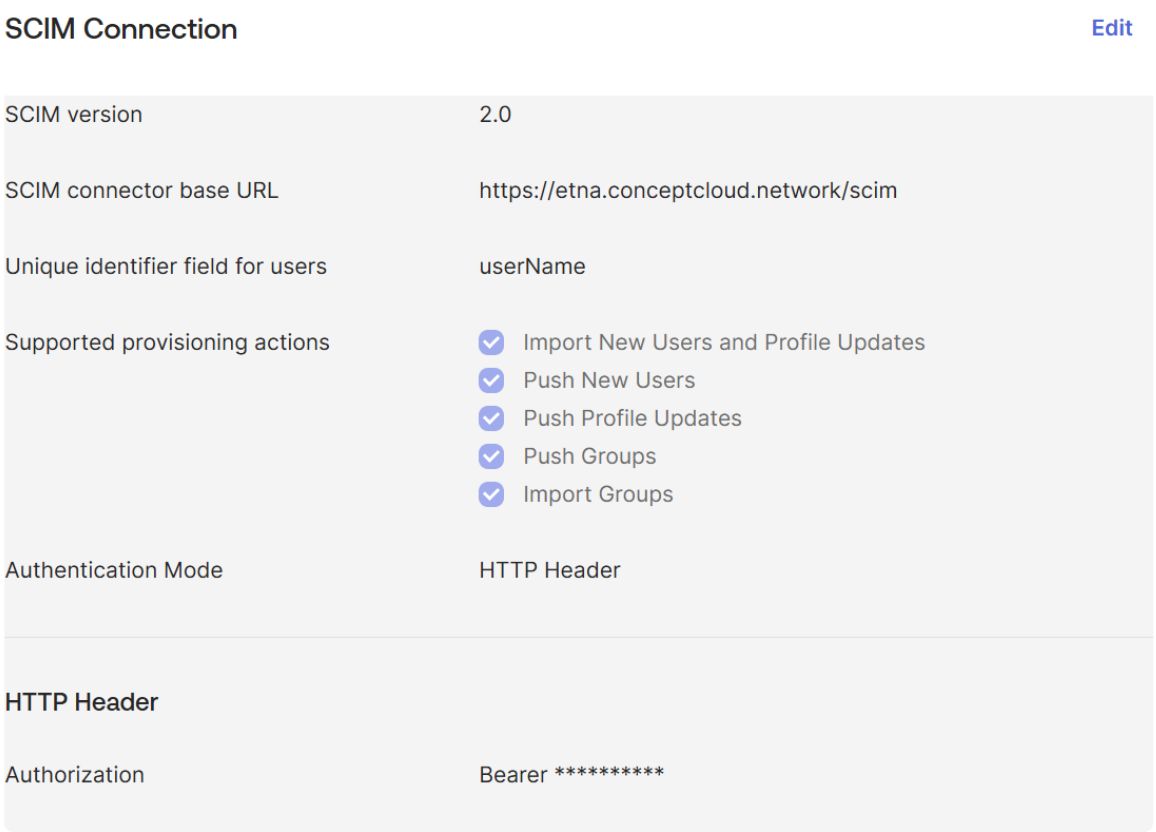
```
Authorization: Bearer <scim-token>
```

Okta

To use Okta, set these fields:

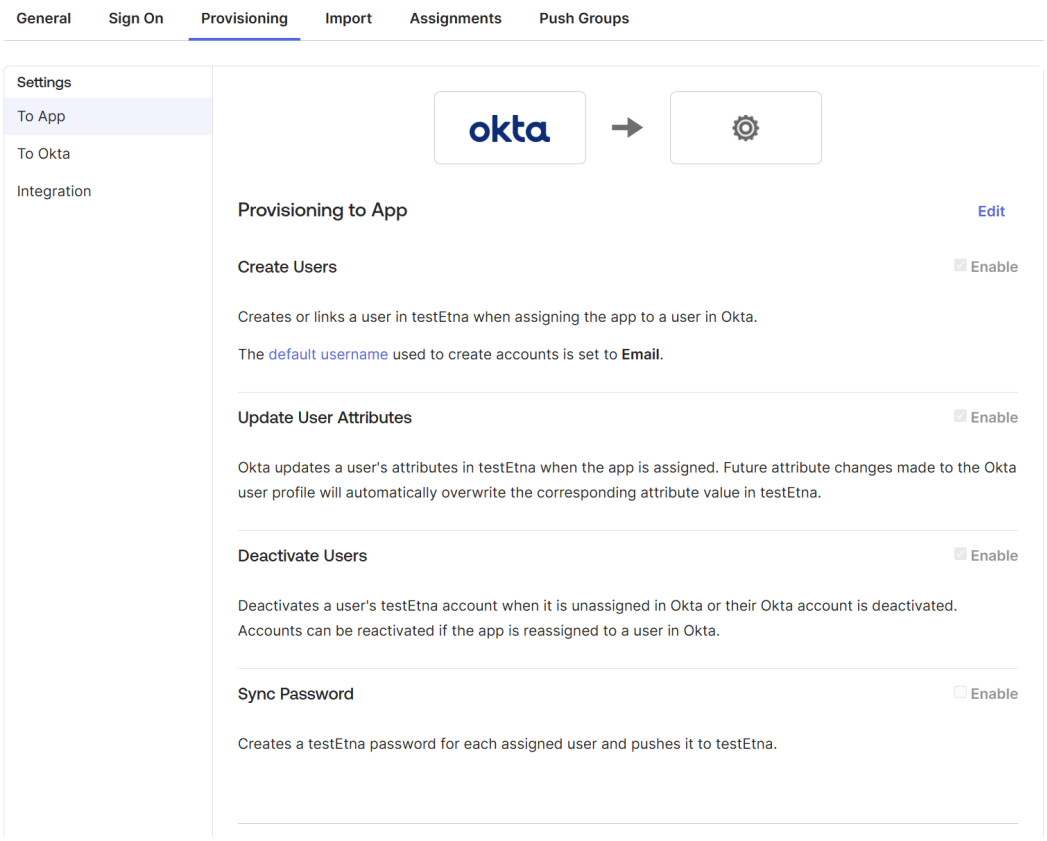
- **SCIM connector base URL:** `https://<your-sds-hostname>/scim`
- **Unique identifier field for users:** `userName`

The configuration looks similar to this example:



Under users, enable the options that match your provisioning workflow:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



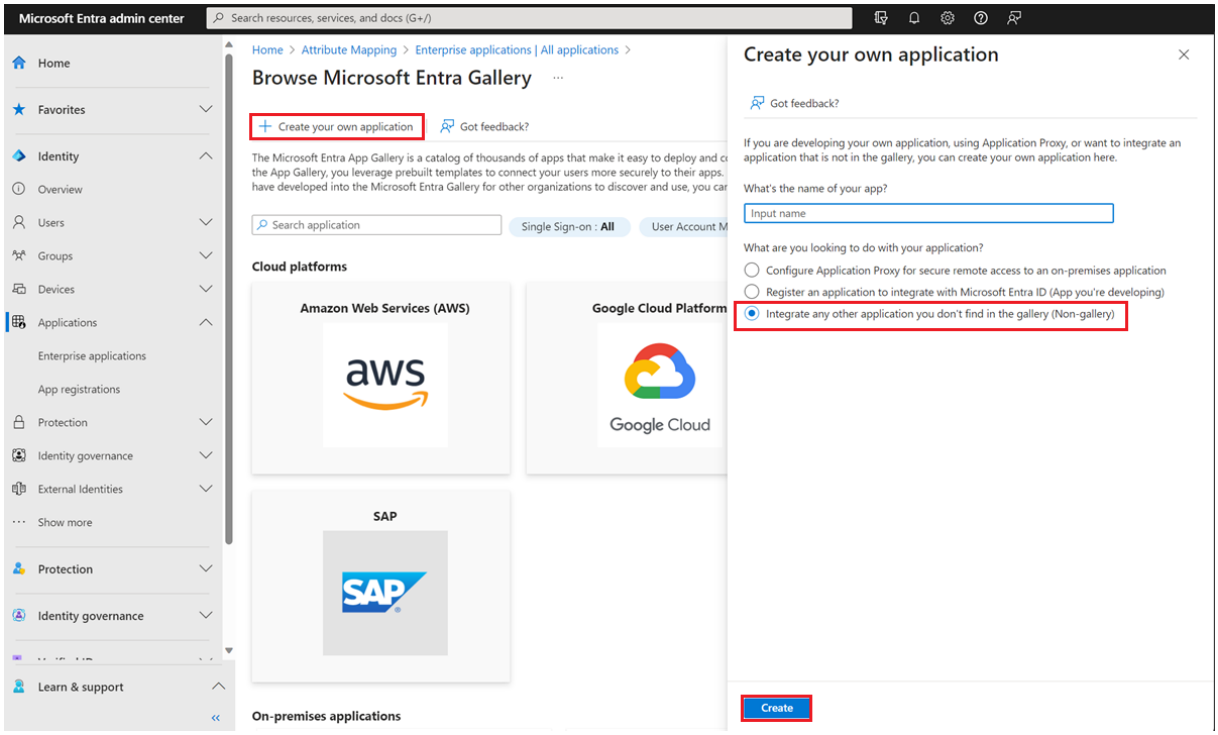
Microsoft Entra

To use Microsoft Entra, configure a custom Enterprise Application within your Microsoft Entra tenant.

Add an Enterprise Application

1. Sign in to the Microsoft Entra admin center with at least **Cloud Application Administrator** permissions.
2. Go to **Identity > Applications > Enterprise applications > All applications**.
3. Select **+ New application**.
4. Select **Create your own application**.
5. Enter a unique application name, and select **Integrate any other application you don't find in the gallery**.
6. Select **Create**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Configure provisioning After the application is created, follow these steps to set up the SCIM connection:

1. Go to **Identity > Applications > Enterprise applications**, and select your newly created application.
2. Under **Manage**, select **Provisioning**.
3. Select **+ New configuration**.
4. For **Select authentication method**, select the supported bearer token option.
5. For **Tenant URL**, enter your SCIM endpoint URL. For example, `https://<your-sds-hostname>/scim`.
6. For **Secret token**, enter the bearer token provided by SecurSpaces.
7. Select **Test Connection**.
8. Select **Save**.

> **Note:**
>
> A successful test connection might be required to continue.

PingOne

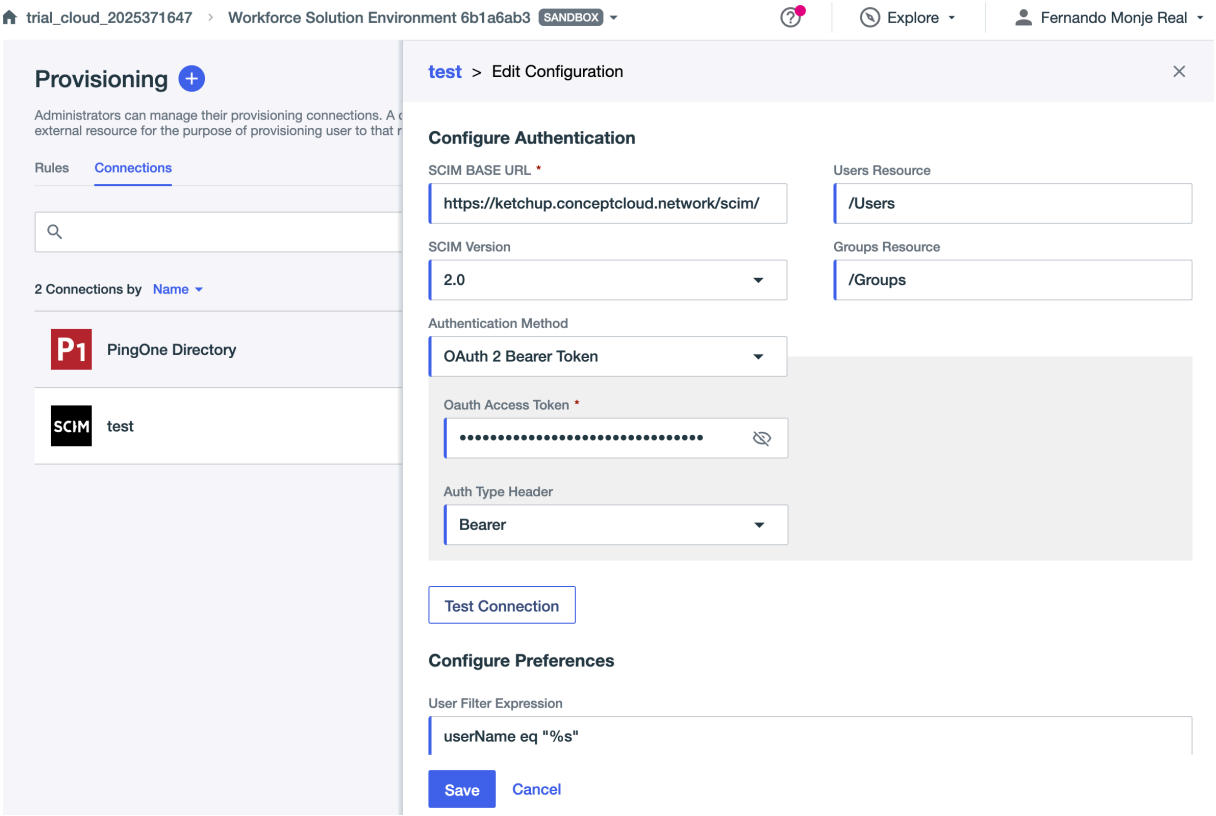
To use PingOne, set these fields:

- **SCIM base URL:** `https://<your-sds-hostname>/scim`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

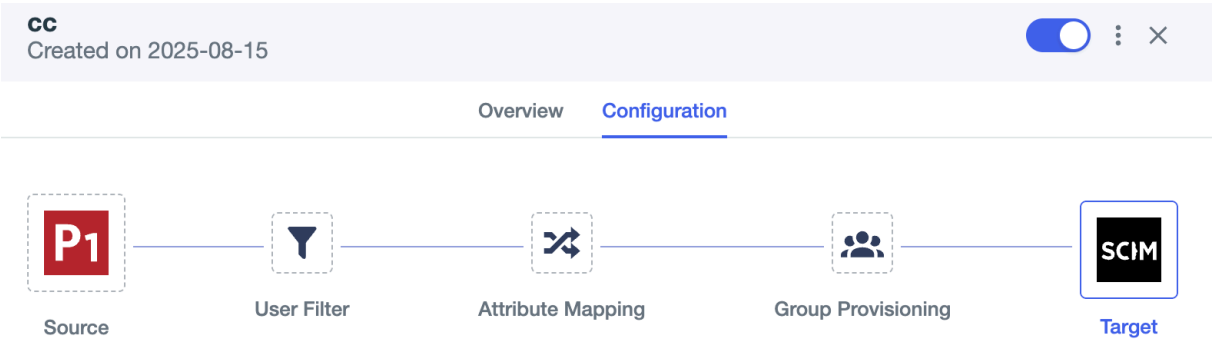
- **User Filter Expression:** Change `username` to `userName`, and change `Eq` to `eq`.

The configuration looks similar to this example:



After configuration, the result looks similar to these examples:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Selected Target



Name
test

Description

See Details

Authorization

SCIM BASE URL
https://ketchup.conceptcloud.network/scim/

Users Resource
/Users

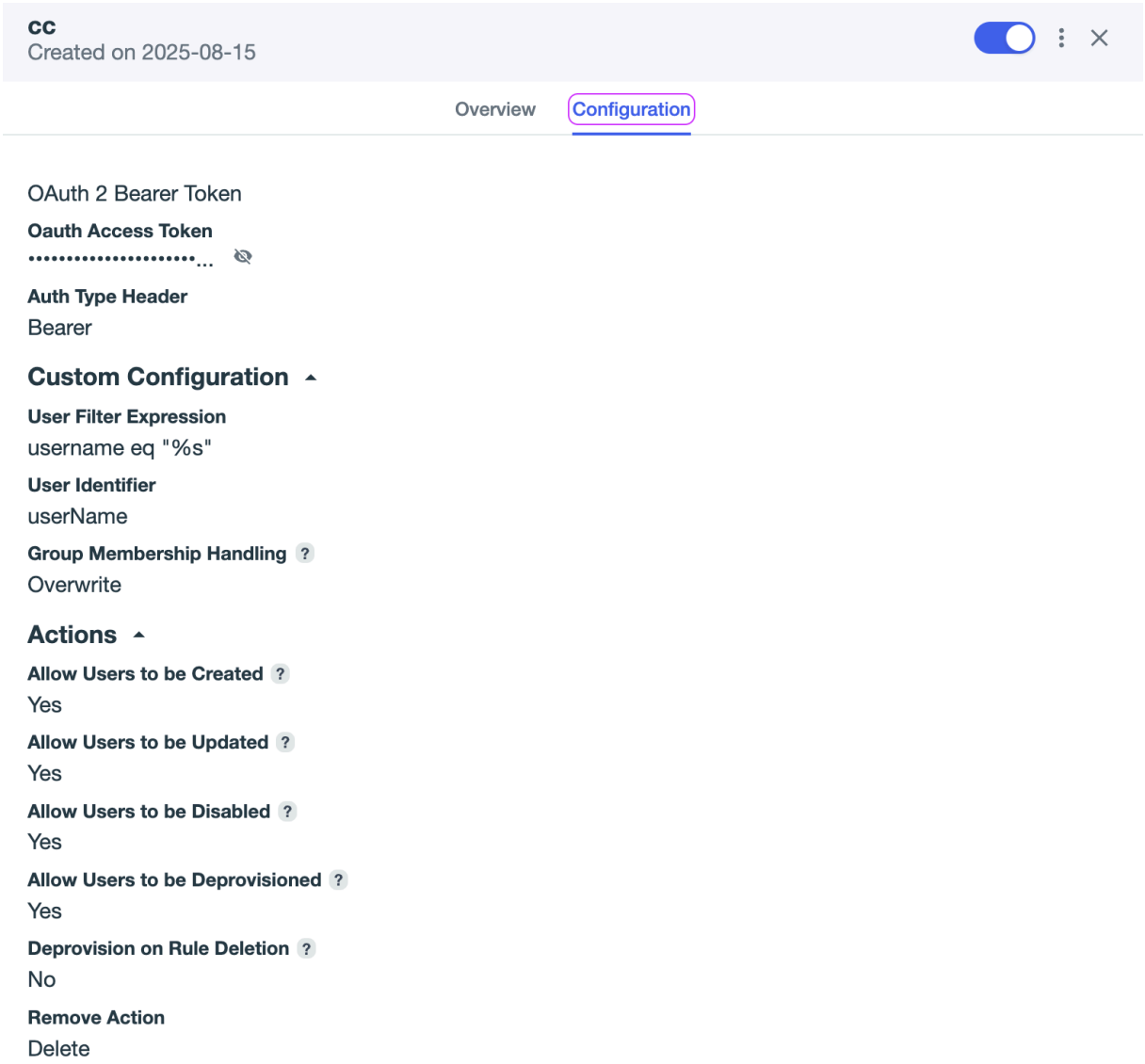
SCIM Version
2.0

Groups Resource
/Groups

Authentication Method
OAuth 2 Bearer Token

Oauth Access Token
••••••••••••••••••••••...

© 1997–2026 Citrix Systems, Inc. All rights reserved.



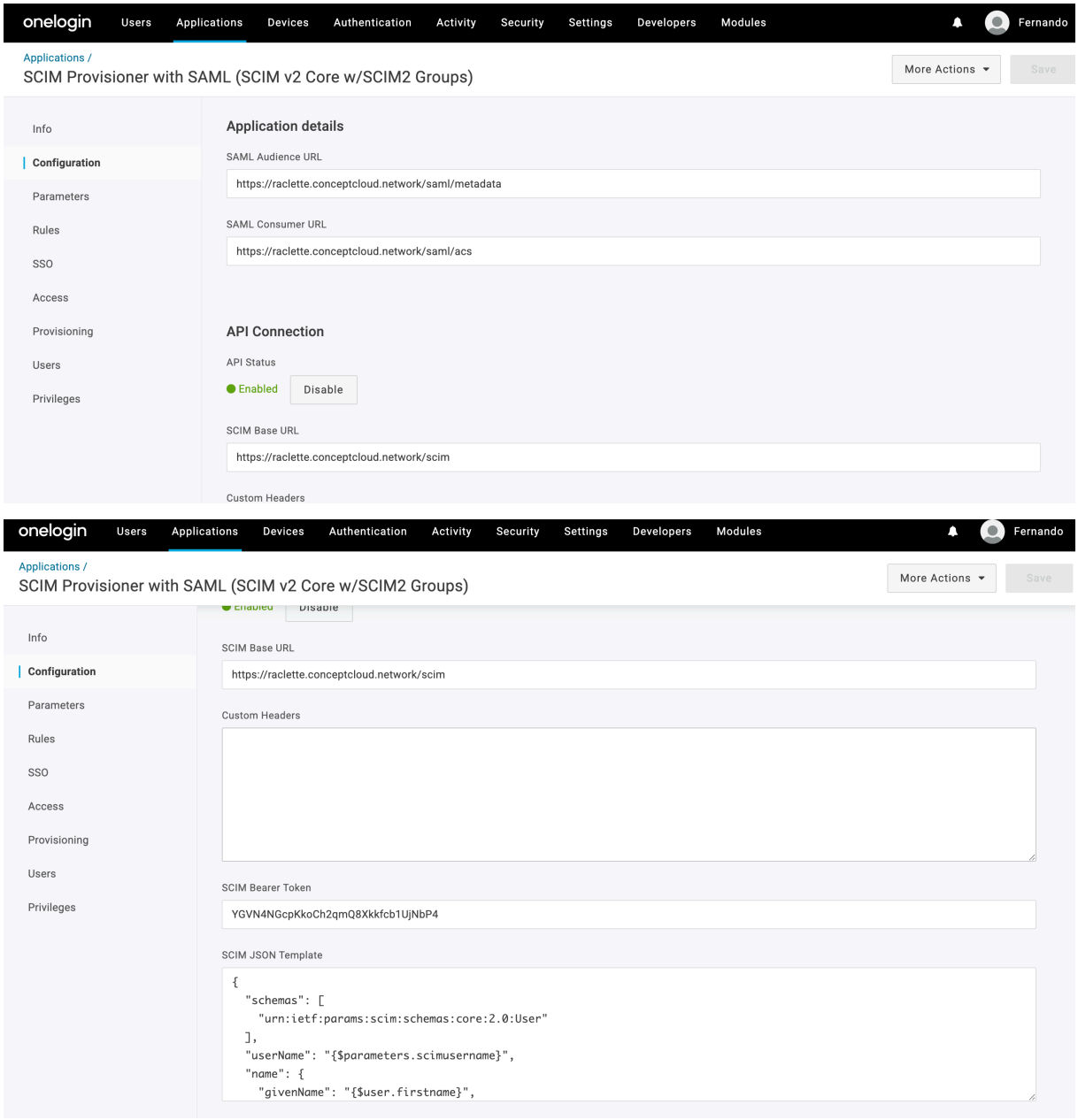
OneLogin

To use OneLogin, set these fields:

- **SCIM base URL:** `https://<your-sds-hostname>/scim`
- **scimusername:** Set the value to **Email**.

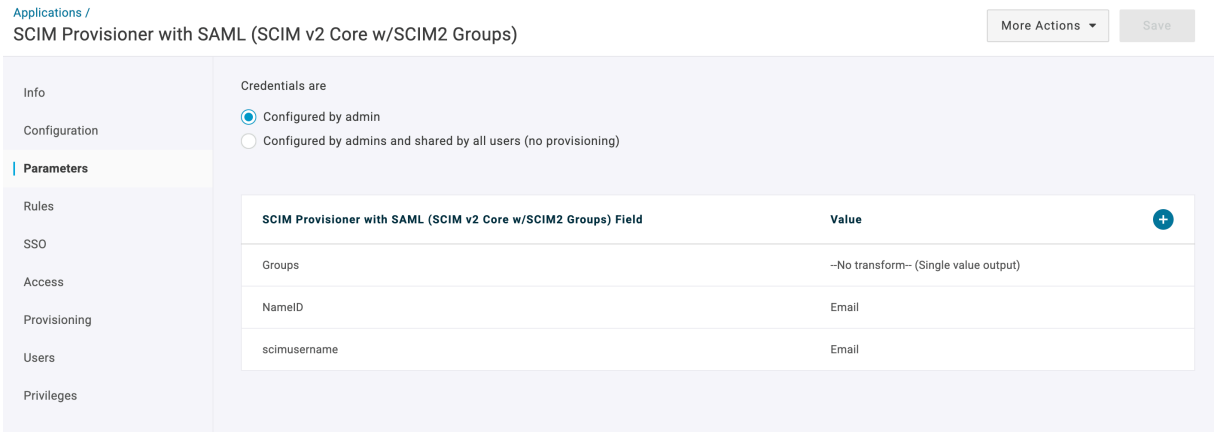
The configuration looks similar to these examples:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



The parameters section looks similar to this example:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

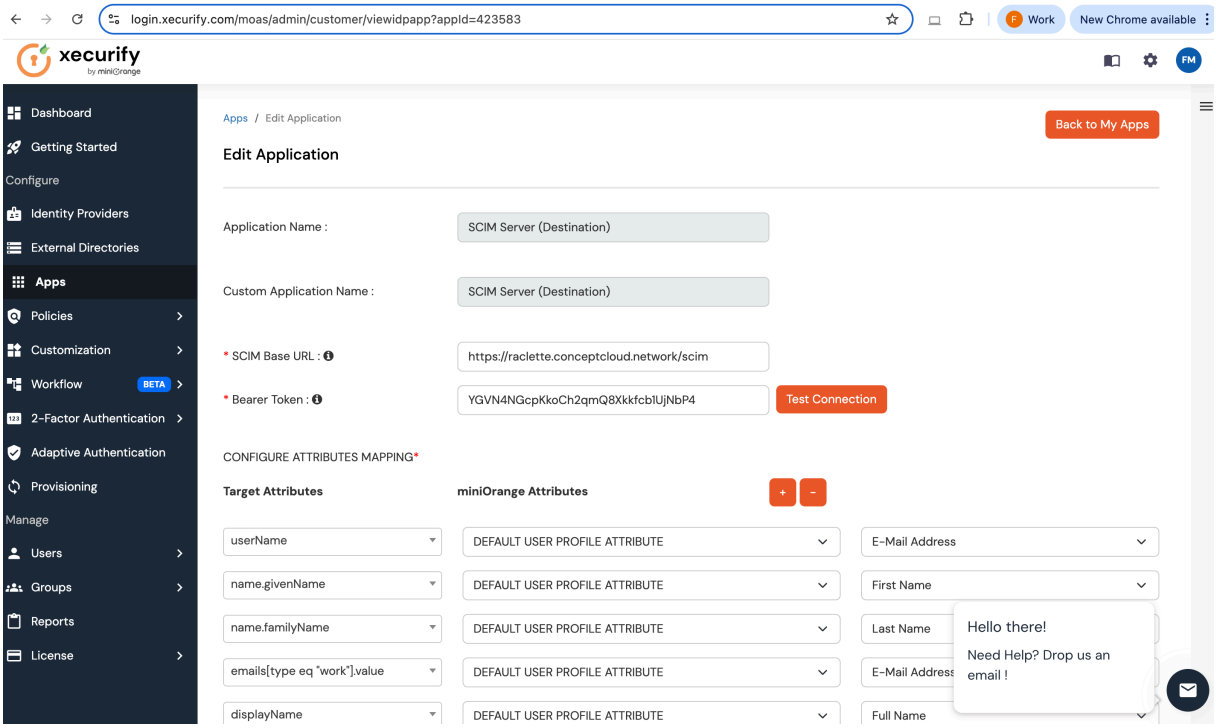


Xecurify

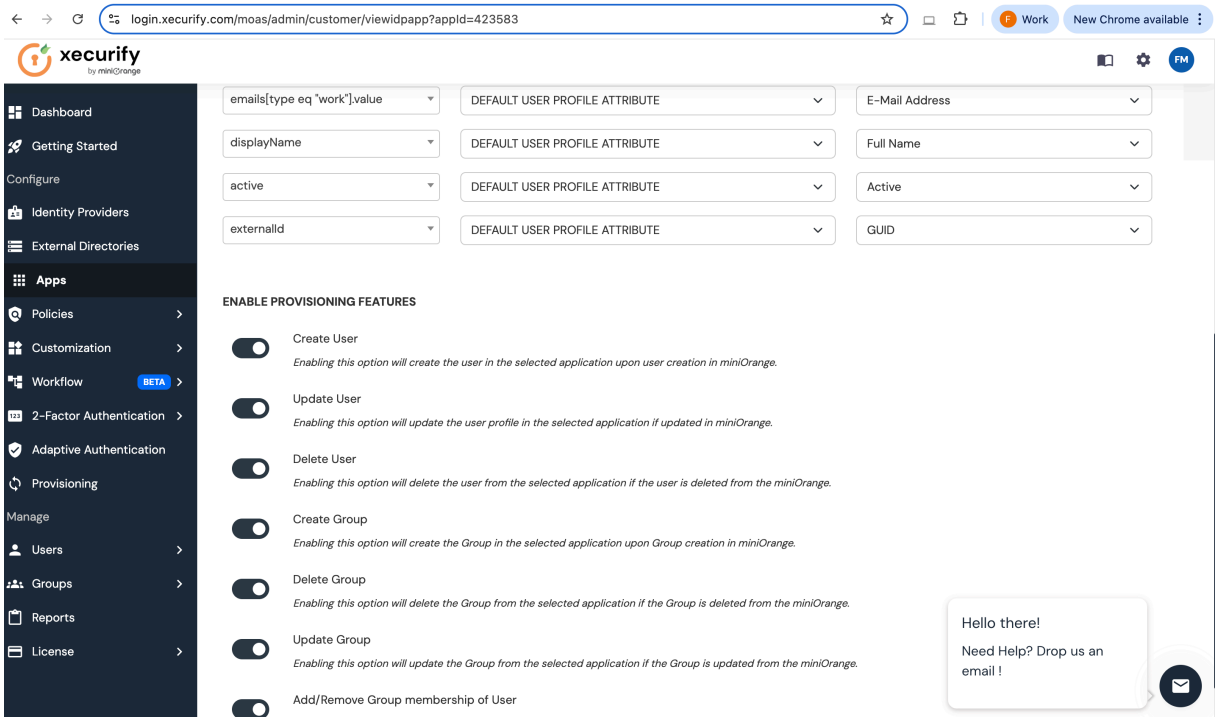
To use Xecurify, also known as miniOrange, set these fields:

- **SCIM Base URL:** `https://<your-sds-hostname>/scim`
- **userName:** Set the value to **E-Mail Address**.

The configuration looks similar to these examples:



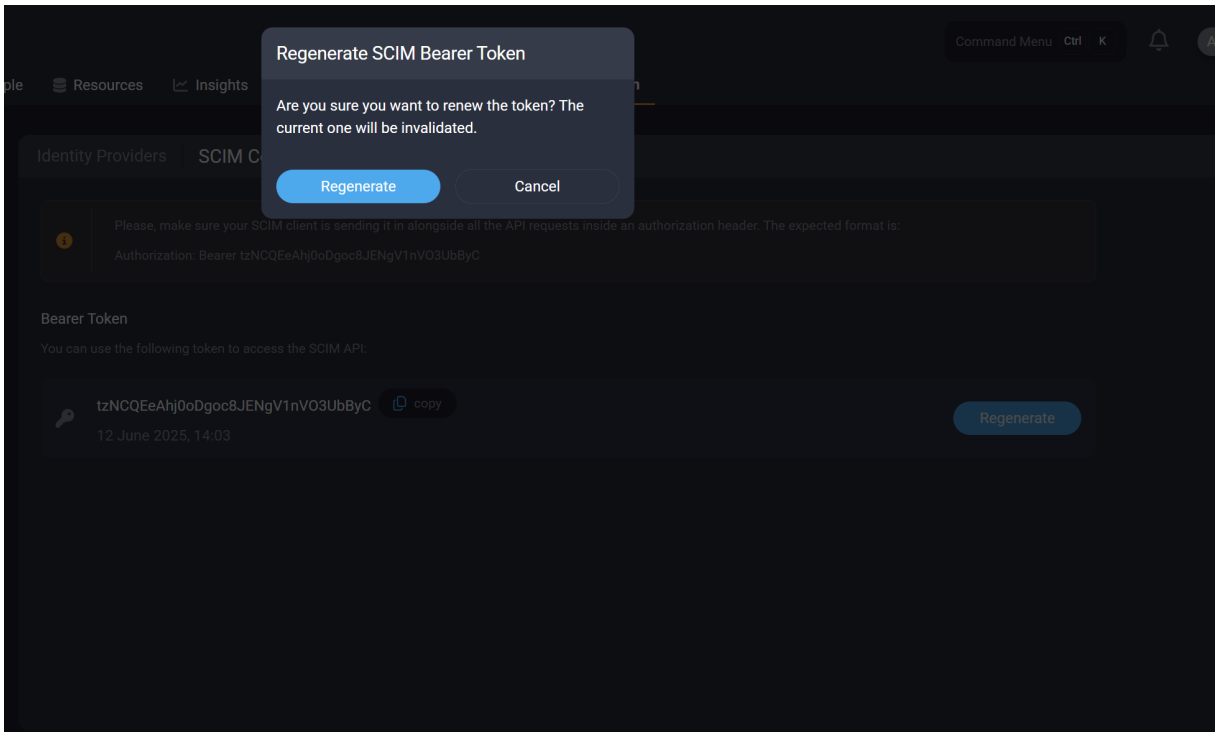
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Configure the SecurSpaces platform

No additional configuration is needed on the SecurSpaces platform. SCIM is enabled by default.

You can renew the token at any time. Renewing the token invalidates any previously issued tokens.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Google OAuth
- Microsoft Entra
- OpenID Connect
- Platform roles
- Identity providers
- Identity and access

User access control

August 19, 2026

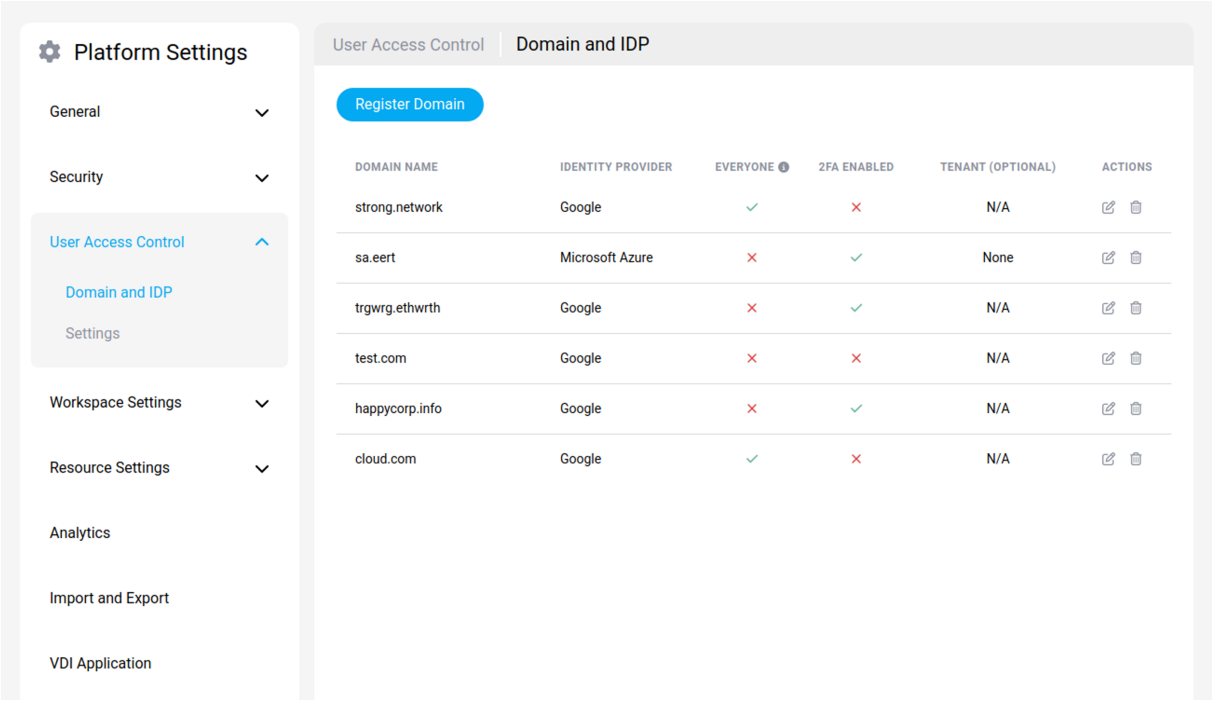
User Access Control decides two things at platform level: which identity provider authenticates a given
user, and what any user is permitted to do regardless of their role.

Find it under **System Configuration**. Changes here apply to everyone on the platform.

Registered domains and identity providers

Users are routed to an identity provider by the domain of their email address. Register a domain, associate
it with a configured provider, and anyone signing in from that domain is authenticated by it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



This is how a deployment supports more than one organization at once. Employees on your own domain can go to
your corporate identity provider, while contractors on another domain use a different one.

Two-factor authentication is enabled here, per registered domain.

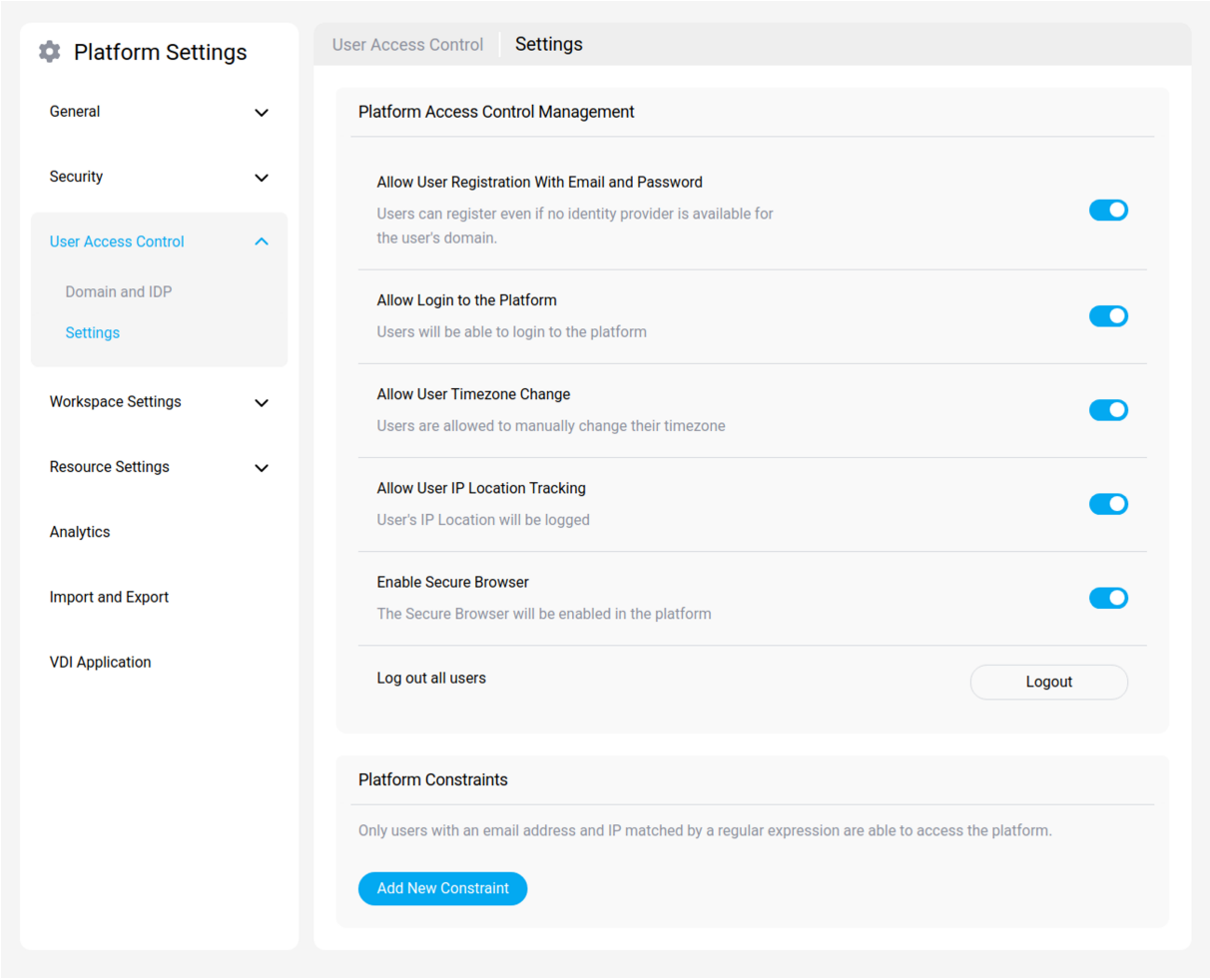
A domain with no identity provider is the case to watch. Whether those users can sign in at all depends on
the platform constraint below that governs registration with an email address and password.

See Identity providers for connecting a
provider, and SAML or
OpenID Connect for the protocols.

Platform access control and constraints

These settings apply to every user on the platform, whatever role they hold.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Setting	When enabled
Allow login to the platform	Users can sign in. Turning it off closes the platform to everyone, which is a maintenance and incident control rather than an everyday setting.
Allow user registration with email and password	Users can register even when no identity provider covers their domain. Leave it off where every user should come through an identity provider.
Allow user timezone change	Users may set their own timezone, which drives their working schedule and therefore when workspaces start and pause.
Allow user IP location tracking	The user's IP location is recorded in the logs.
Support accessible for all users	Every user can reach the platform Support menu.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The registration setting is the one with the widest effect. With it on, someone whose domain is not registered can create an account with a password, bypassing the identity provider routing above. Most deployments that have connected an identity provider should turn it off.

Image URL constraints are also part of this area. See
Registry access.

Related information

- Identity and access
- Platform roles
- Roles and permissions
- Control catalogue

Code repositories

August 19, 2026

Before a project owner can import a repository, the Git provider has to be connected at platform level. That
connection is what lets Citrix SecurSpaces™ authenticate on a developer's behalf, so nobody has to hold
repository credentials on their own device.

Connect a provider once. Every organization and project on the platform can then import repositories from it.

Choose a provider

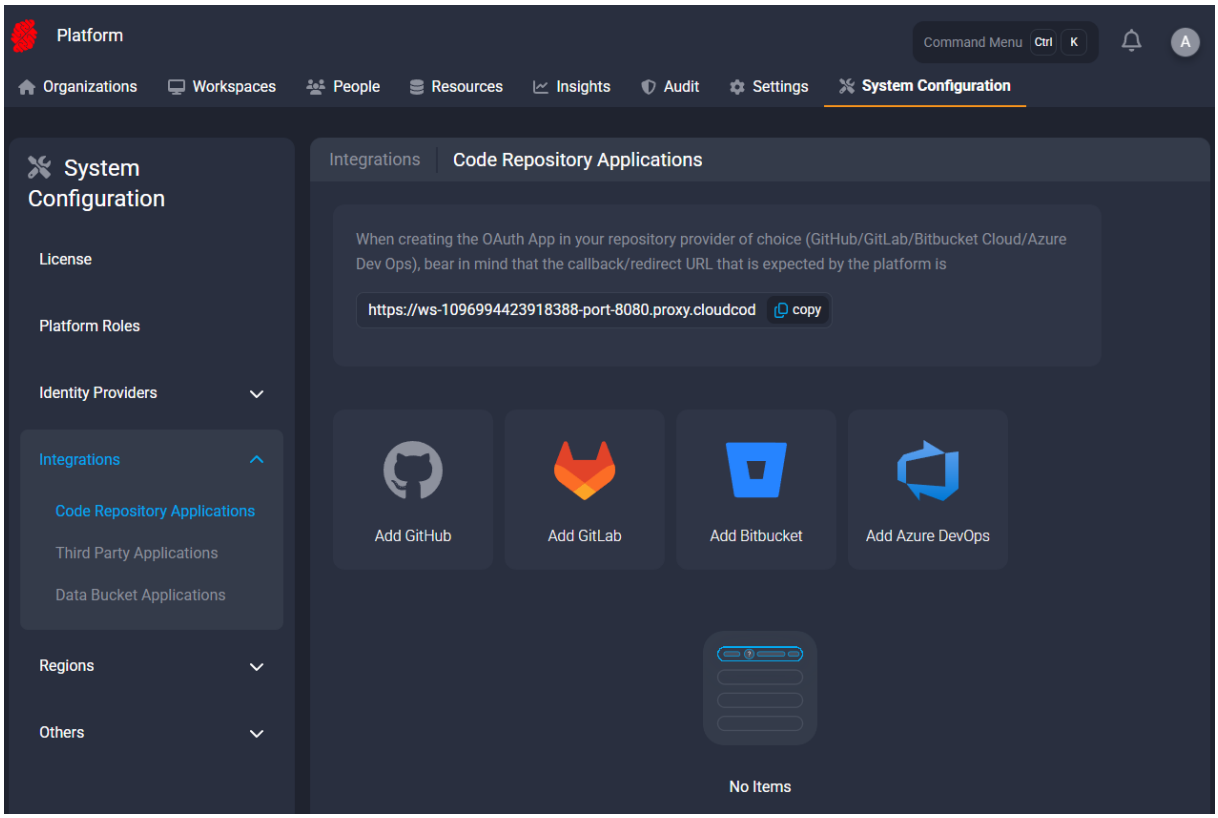
Provider	Hosting	What you need
GitHub	Managed and self-hosted (GitHub Enterprise Server)	An OAuth application in your GitHub organization
GitLab	Managed and self-hosted	An OAuth application in your GitLab instance
Bitbucket	Cloud and Server	Credentials that differ between Cloud and Server —see the page

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Provider	Hosting	What you need
Azure DevOps	Managed	An application registration in Microsoft Entra

You can connect more than one, including two instances of the same provider —for example a managed GitHub
organization alongside a self-hosted GitHub Enterprise Server.

A repository hosted somewhere else can still be used. A project owner adds it manually by URL, but without a
connected application the platform cannot enumerate repositories or inject credentials for it.



What happens after you connect

The provider appears as an option when a project owner imports a repository. See Code repositories for that side of
the task.

From then on, credentials are injected into the workspace at runtime and traffic to the provider passes through the platform proxy, so repository access is attributable and can be audited.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

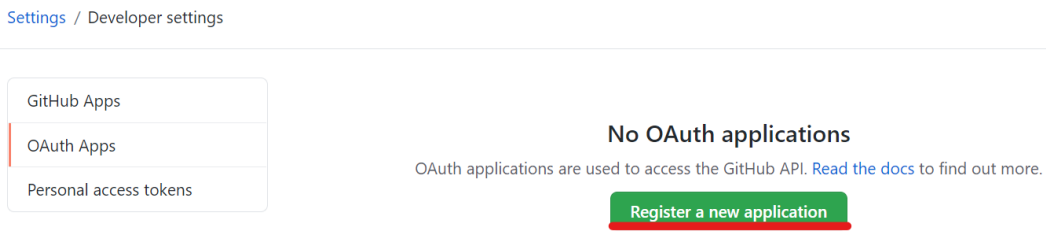
Related information

- Integrations
- Code repositories (project level)
- Trust boundaries

Connect GitHub

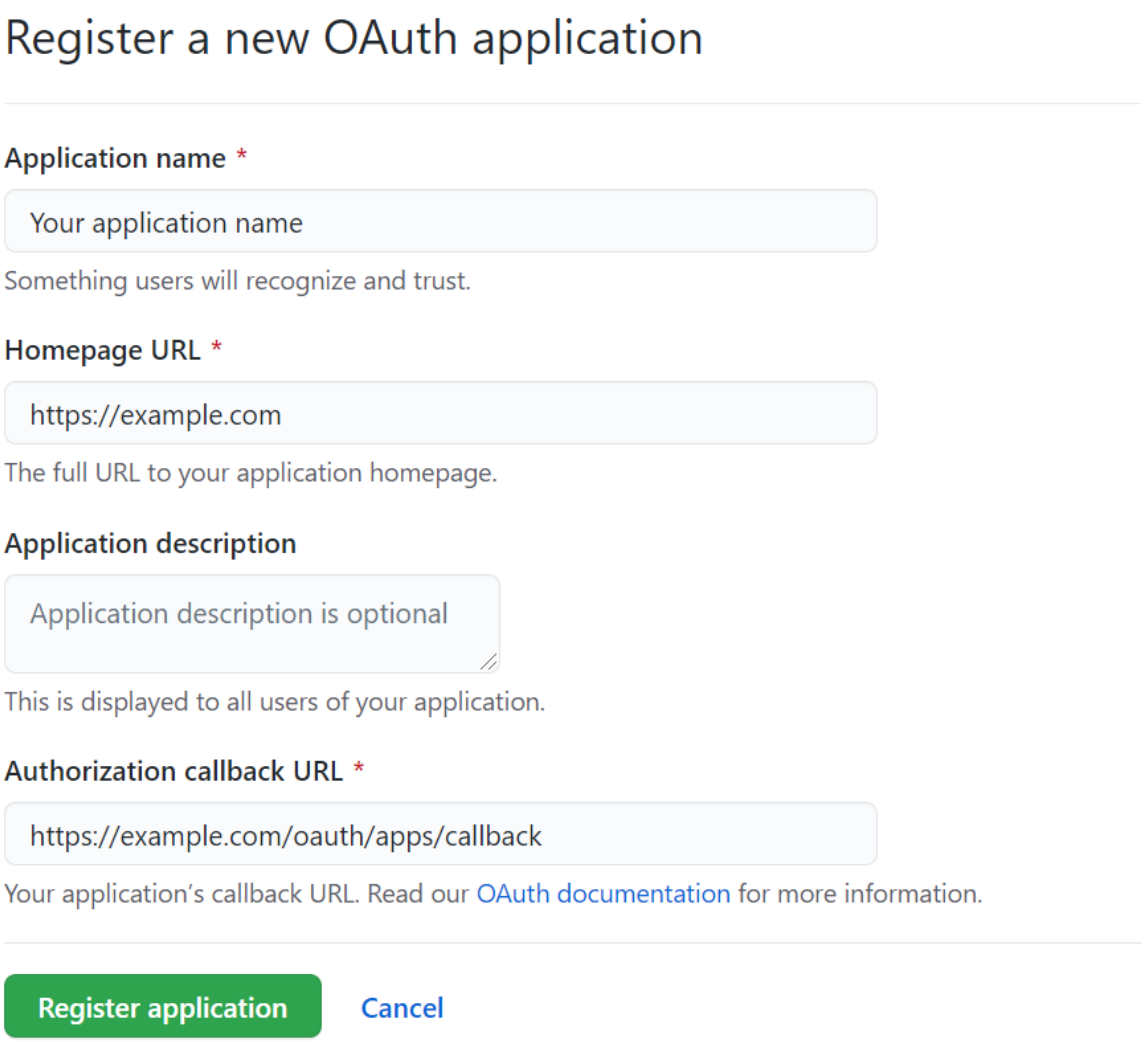
Follow these steps to create an OAuth App in GitHub to connect it to the platform:

- **Navigate to Developer Settings:**
- Using a GitHub account, go to its settings and then to “Developer settings”.
- Inside this menu, click on “OAuth Apps”.
- You can follow this https://github.com/settings/developers to reach this menu directly.



- **Register New Application:**
- Click on “Register new application”and you will be presented with a screen to set:
  - **Application Name:** At your discretion.
  - **Homepage URL:** The main route of the domain where the platform is running.
  - **Authorization Callback URL:** The URL should have a structure similar to `https://example.com/oauth/apps/callback`, where “example.com”should be replaced with the proper domain name (same as the Homepage URL).

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- **Complete the Registration:**
- When this process is done, click on the green button "Register application".
- You will be redirected to a new application page where you can see the Client ID and generate the Secret that needs to be set in the platform configuration.
- **Give Organization Access:**
- You will need to grant the organization access to this newly created OAuth app in the organization you want to connect to the platform.

Related information

- Azure DevOps

© 1997–2026 Citrix Systems, Inc. All rights reserved.

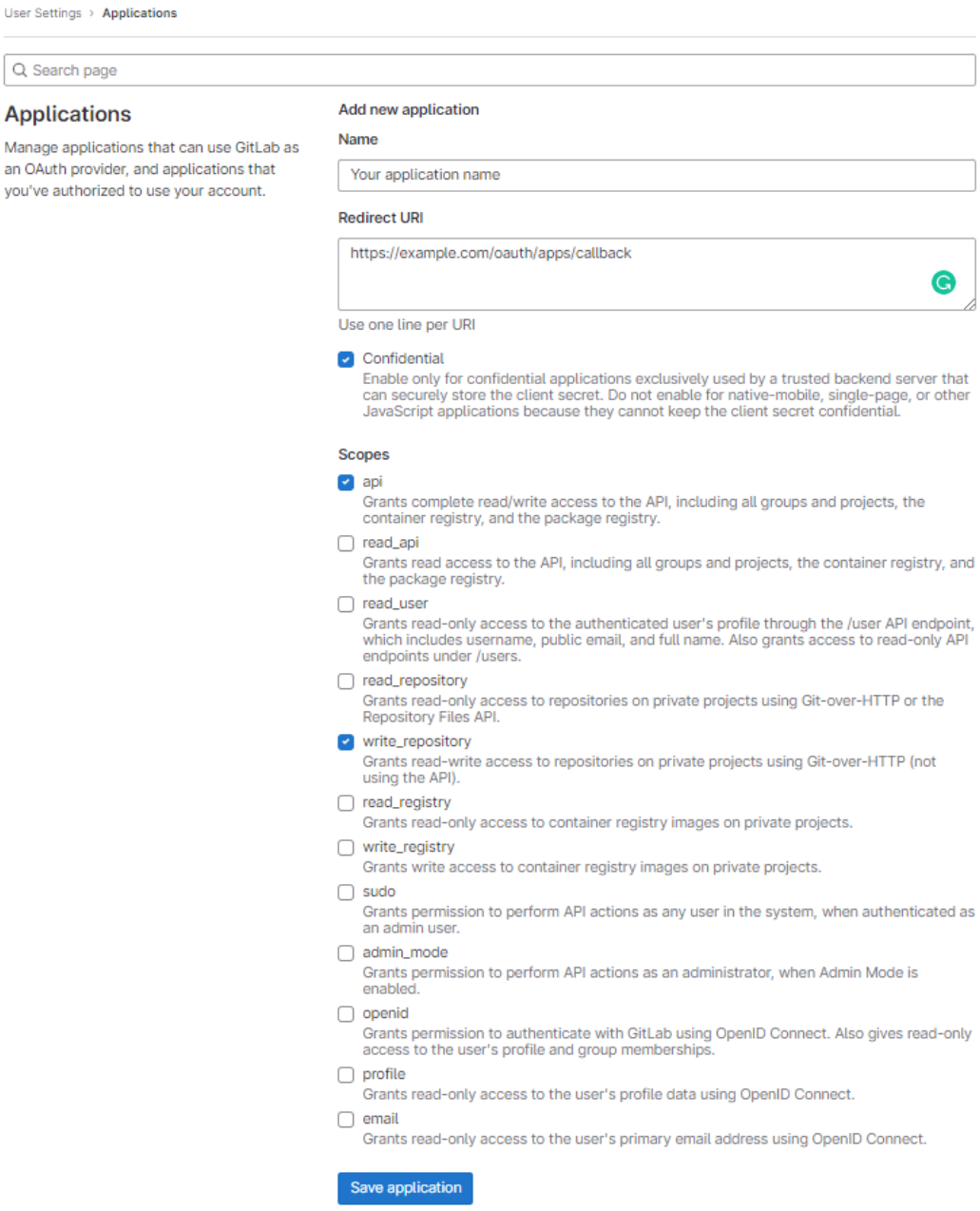
- Bitbucket
- GitLab
- Code repositories

Connect GitLab

Follow these steps to create an OAuth App in GitLab to connect it to the platform:

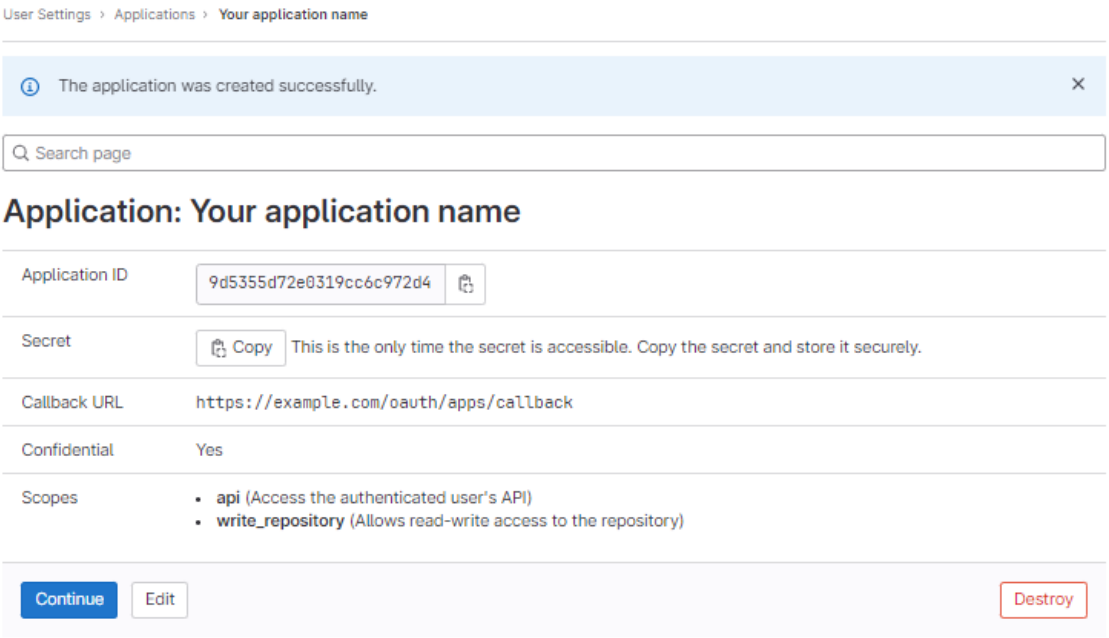
- **Navigate to Applications:**
- Using a GitLab account, go to user settings and then to "Applications".
- You can follow this https://gitlab.com/-/user_settings/applications to reach this menu directly.
- **Create a New OAuth App:**
- Click on "New application"and set the following fields:
  - **Name:** The name you want to give to the application. It will be public.
  - **Redirect URI:** The URL should have a structure similar to `https://example.com/oauth/apps/callback`, where "example.com"should be replaced with the proper domain name.
  - **Confidential:** This should already be selected by default; leave it as it is.
  - **Scopes:** Add the `api` and `write_repository` scopes. These are needed to automatically deploy deployment keys.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- **Complete the Registration:**
- After clicking the “Save application”button, you will be presented with the Client ID (called Application ID) and Secret, which you need to enter in the platform configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Related information

- Azure DevOps
- Bitbucket
- GitHub
- Code repositories

Connect Bitbucket Cloud

Follow these steps to create an OAuth App in Bitbucket Cloud to connect it to the platform:

- **Navigate to OAuth Consumers:**
- Using a Bitbucket account, go to the main organization settings and then to “OAuth consumers” .
- You can follow this `https://bitbucket.org/[YOUR_DOMAIN_NAME]/workspace/settings/api` to reach this menu directly.
- **Add a New Consumer:**
- Click on the “Add consumer”button and set the following fields:
    - **Name:** The name you want to give to the application. It will be public.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

    - **Callback URL:** The URL should have a structure similar to `https://example.com/oauth/apps/callback`, where "example.com"should be replaced with the proper domain name. This URL can be seen from the admin panel of the SecurSpaces platform.
    - **This is a private consumer:** This should already be selected by default; leave it as it is.
    - **Scopes:** Select "Read"under the Account section and "Write"under the Repositories section, and "Read"under the Pull requests section. This can also be checked in the Citrix SecurSpaces™ Platform when clicking the "Add Bitbucket"button.

- **Complete the Registration:**
- After clicking the "Save"button, you will be presented with the Client ID (called Key) and Secret, which you need to enter in the platform configuration.

Bitbucket Server or Data Center Integration as Code Repository Provider

In this section, we will see how to connect the SecurSpaces platform to a self-hosted Bitbucket instance:

- **Configure SecurSpaces Platform:**
- Go to the SecurSpaces platform settings and open the "Code Repository Applications"menu.
- Click on the "Add Bitbucket"button.
- Select the checkbox for "Bitbucket Server or Data Center (self-hosted)".
- **Set the Following Fields:**
- **Bitbucket App Name:** It can be anything. This is what users will see when using this Code Repository Provider.
- **Custom Domain:** Enter the URL where the Bitbucket instance is hosted. If no scheme is given, HTTPS will be chosen by default.
- **Enforce Users to Connect:** If selected, users will need to connect to Bitbucket before they can open their workspaces. This can prevent misconfiguration/permission issues on the user side.
- **Complete the Registration:**
- Click the "Create"button to complete the configuration on the SecurSpaces platform side.
- Save the "Bitbucket Server Public Key"for later use. This can also be found in the edit menu after clicking the "Create"button.
- **Configure Bitbucket Instance (Version 7.20 or Later):**
- Go to Administration > Applications > Application Links and click on "Create link":
    - **Application Type:** External application

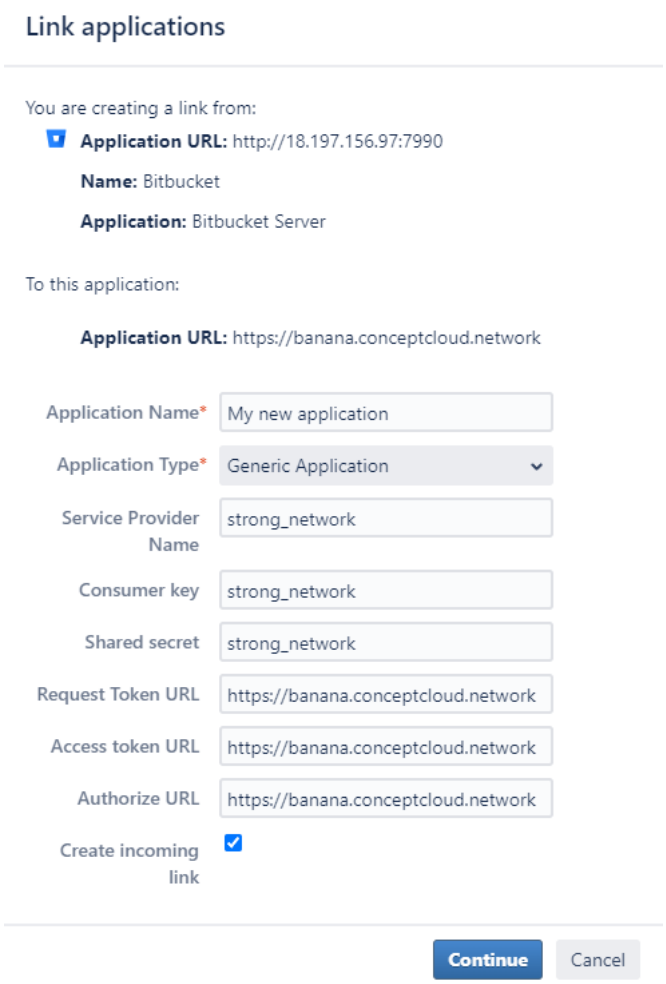
© 1997–2026 Citrix Systems, Inc. All rights reserved.

  - **Direction:** Incoming
  - Click on continue
  - Set a unique name
  - **Redirect URL:** Set to `https://example.com/oauth/apps/callback`, where "example.com"should be replaced with the proper domain name.
  - **Application Permissions:** Account: Write, Repositories: Admin
  - After clicking the "Save"button, enter "strong_network"for both Client ID and Client Secret.

- **Configure Bitbucket Instance (Version 7.20 or Earlier):**
- Go to Administration > Application Links.
- Enter the platform URL (e.g., `https://example.com`, where "example.com"should be replaced with the proper domain name).
- Click on "Create new link". If you see a "No response received"error, ignore it and click Continue.
- In the following menu, enter:

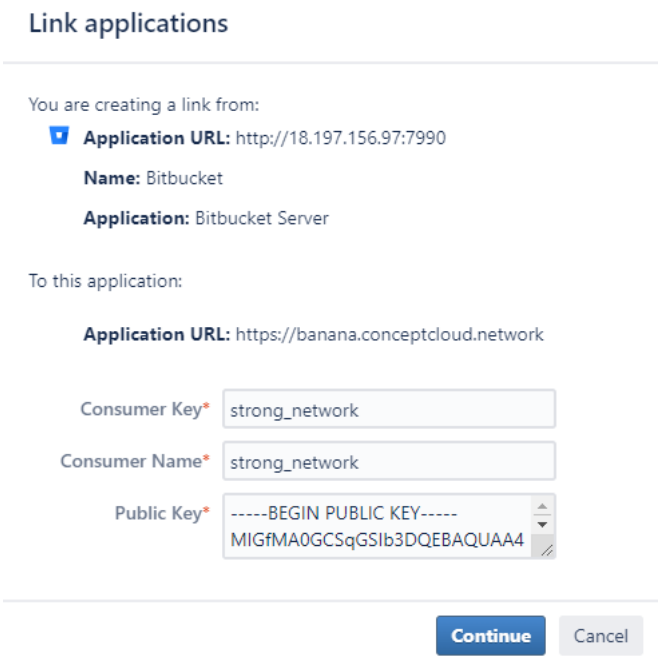
  - **Application Name:** It can be anything.
  - **Application Type:** Generic Application
  - **Service Provider Name:** It can be anything (recommended: "strong_network").
  - **Consumer Key:** Set to "strong_network".
  - **Shared Secret:** Set to "strong_network".
  - **Request Token URL:** Set to `http://example.com`, where "example.com"should be replaced with the proper domain name.
  - **Access Token URL:** Set to `http://example.com`, where "example.com"should be replaced with the proper domain name.
  - **Authorize URL:** Set to `http://example.com`, where "example.com"should be replaced with the proper domain name.
  - Check "Create incoming link"and click Continue.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- In the following menu, enter:
  - **Consumer Key:** Set to “strong_network”.
  - **Consumer Name:** Set to “strong_network”.
  - **Public Key:** Enter the value that can be seen in the platform.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- **Complete the Configuration:**
- Click on Continue. The configuration is complete.

Related information

- Azure DevOps
- GitHub
- GitLab
- Code repositories

Azure DevOps

Follow these steps to create an OAuth App in Azure DevOps to connect it to the platform.

- Using an Azure DevOps account, go to the following link:
  Register an application
- Click on the "Add consumer"button and set the following fields:
  - **Company Name:** Your company's name.
  - **Application Name:** The name you want to give to the application. It will be public.
  - **Application Website:** Set to `https://example.com/oauth/apps/callback` (replace `example.com` with the proper domain name).

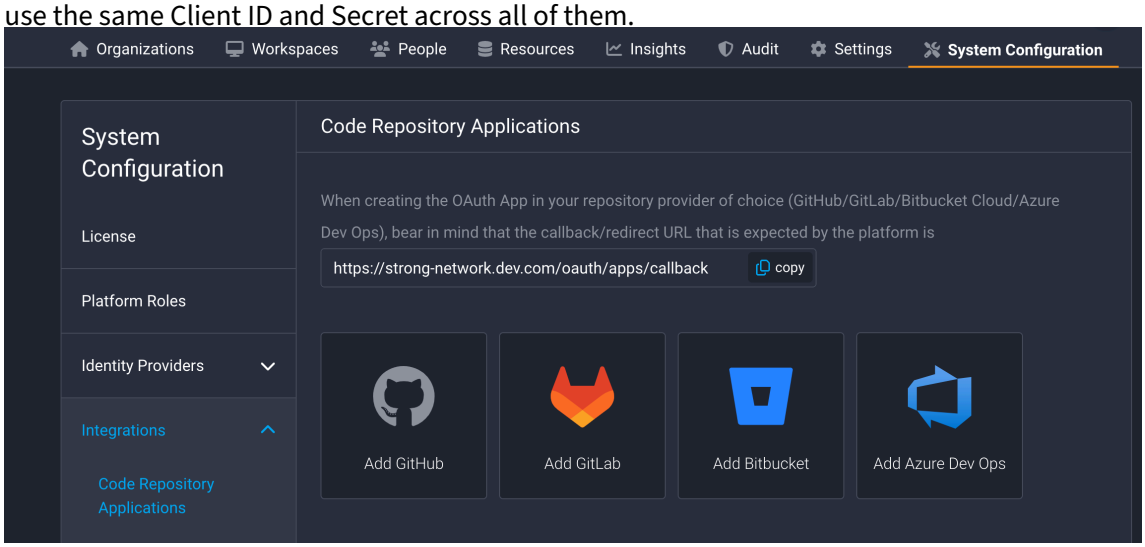
© 1997–2026 Citrix Systems, Inc. All rights reserved.

  - **Authorization Callback URL:** Set to `https://example.com/oauth/apps/callback` (replace `example.com` with the proper domain name). This URL can be found in the administrator panel of the Citrix SecurSpaces™ platform.
  - **Authorized Scopes:** `Code (read and write)` and `Project and team (read)`.

- Once done, click the "Create Application"button. You will be presented with the Client ID (called App ID) and the Secret (called Client Secret) after clicking the "Show"button. Enter these fields in the administrator configuration of the SecurSpaces platform.

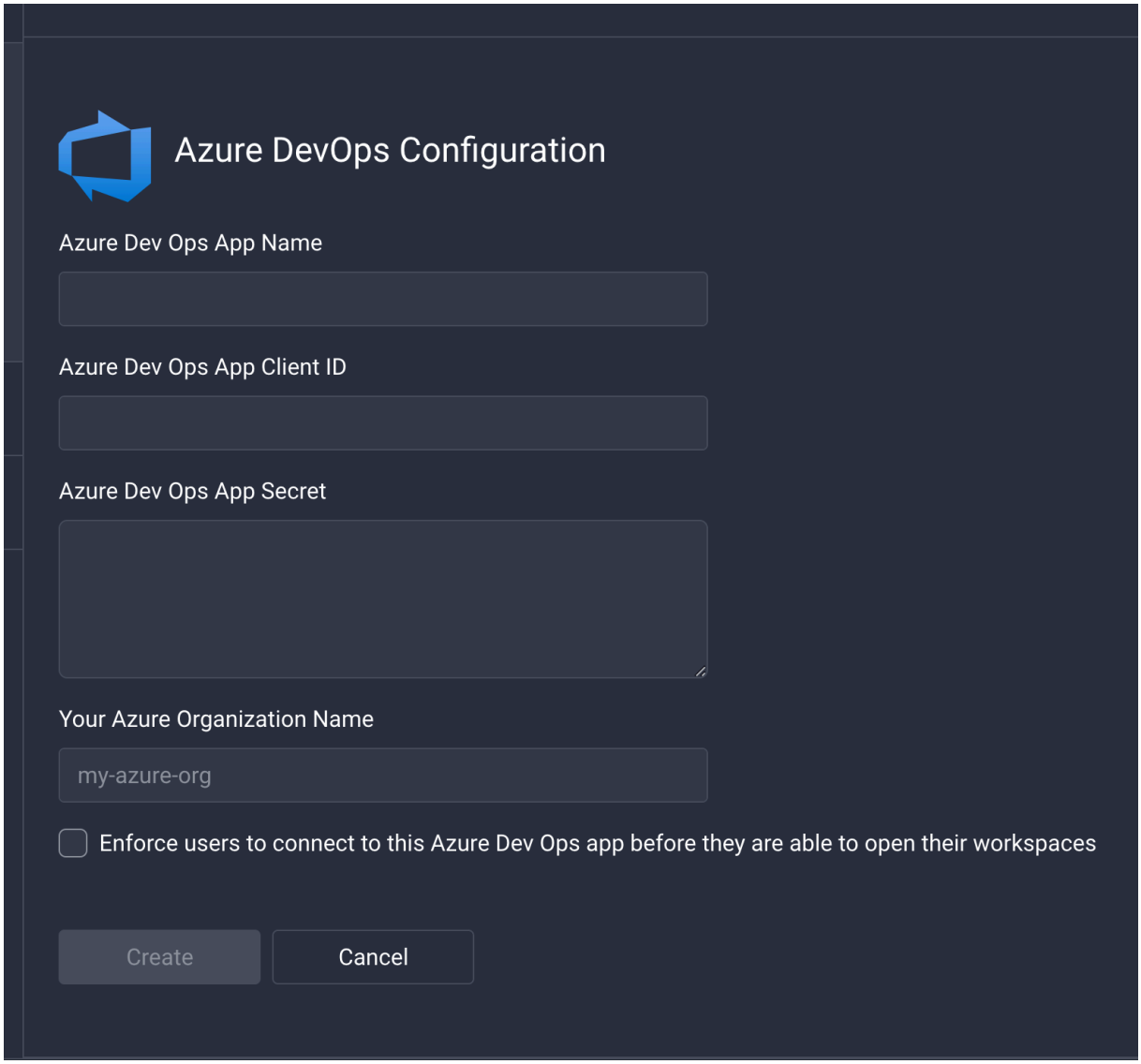
  - Register an application
  - `https://example.com/oauth/apps/callback`

- Specify the Azure Organization name. This application can only access repositories under this specific organization. To access repositories from different organizations, create multiple Azure DevOps Code Repository Applications, each with its corresponding organization name. You may use the same Client ID and Secret across all of them.



Paste Client ID, App Secret and Organization name from steps above:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Related information

- Bitbucket
- GitHub
- GitLab
- Code repositories

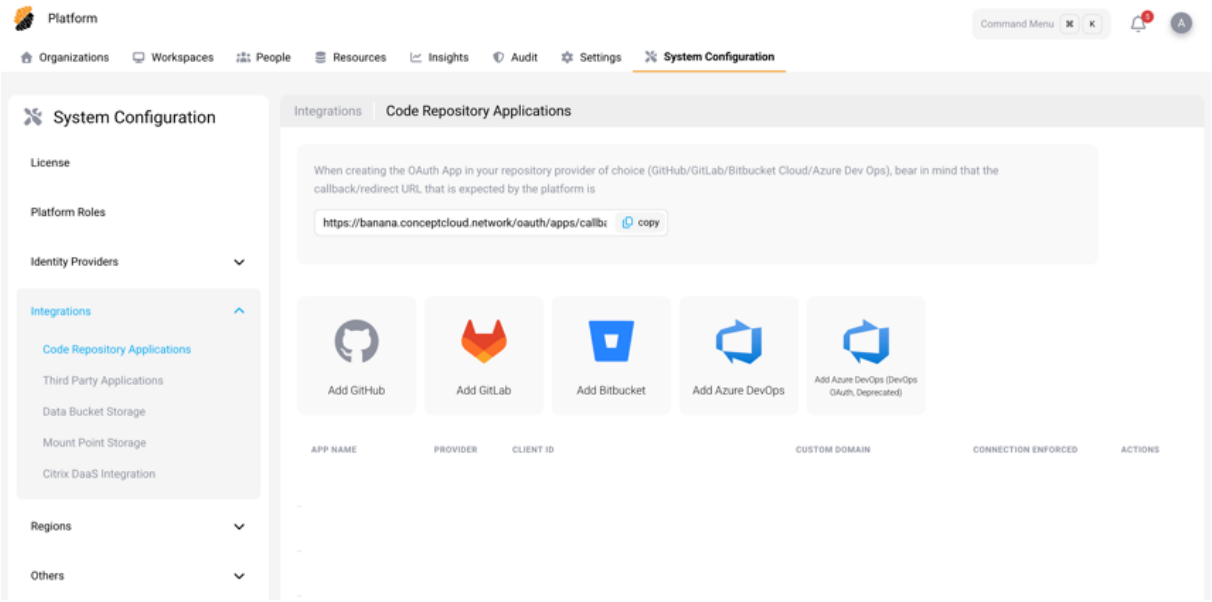
Integrations

September 29, 2026

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Connect Citrix SecurSpaces™ to the outside tools your teams rely on, from code repositories and third-party
apps to storage and Citrix DaaS™. Turn each one on and set it up in **Integrations**.

Find these settings at **System Configuration > Integrations**.



Here's what you can connect:

- Code Repository Applications
- Third Party Applications
- Data Bucket Storage
- Mount Point Storage
- Citrix DaaS Integration

Code Repository Applications

Add the repository providers your developers connect to their workspaces, such as GitHub, GitLab, Bitbucket, and Azure DevOps. You set up each one as an OAuth application.

For setup steps, see Setup Code Repositories Applications.

Third Party Applications

Add third-party apps, such as JFrog, that developers reach from their workspaces. You provide the credentials and callback URL each provider expects.

For an example, see JFrog.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Data Bucket Storage

Set up the storage behind **Data Buckets**, so teams can share versioned datasets, files, and artifacts across the workspaces in a project.

To learn more, see Data Buckets.

Mount Point Storage

Turn on **Mount Point Storage** to give teams direct access to external file storage. A Mount Point connects that storage straight to the workspace file system, so workspaces read and write in real time without copying data into the cluster.

Turn on a storage type here first. After that, project owners can create Mount Points. Supported storage
providers include Azure file storage, Amazon Elastic File System (EFS), and Amazon S3 Files.

Which switches appear depends on the cloud that backs Mount Points, and the platform detects that cloud from the
Mount Point storage classes present on the cluster rather than offering it as a choice. On an AWS cluster, a single
switch, **Enable AWS File Storage**, covers both Amazon EFS and Amazon S3 Files.

> **Important:**
>
> AWS Mount Points depend on AWS resources that SecurSpaces does not create, including the file system, the mount
> targets in each Availability Zone, the IAM roles, and the Kubernetes storage classes themselves. An administrator
> must create an eligible AWS storage class on the cluster before the AWS storage type appears here. Until then,
> there is nothing to turn on.

Mount Points aren't supported in multi-region configurations.

To learn more, see:

- Mount point storage, for what each switch does
  and what stays mounted after you turn one off
- Prepare AWS storage for Mount Points, for the
  AWS resources you create before the AWS option appears
- Sharing Data Between Users and
  Mount Points, for what project owners do next

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Citrix DaaS Integration

Link SecurSpaces to Citrix DaaS and Citrix Workspace so users reach both their workspaces and their Citrix
virtual apps and desktops with a single sign-in.

For setup steps, see Citrix DaaS Integration.

Related information

- System Configuration

Use HashiCorp Vault as a secret manager

You can use **HashiCorp Vault** to store all platform secrets instead of encrypting them in MongoDB.

Citrix SecurSpaces™ Platform connects to HashiCorp Vault using the **JWT authentication mechanism** provided by **Kubernetes**.

For more information, see Use Kubernetes for OIDC authentication

Prerequisites

The configuration depends on whether your Vault instance is deployed in the same Kubernetes cluster as the SecurSpaces Platform:

- **If Vault is deployed in the same cluster:**
  The OpenID Connect (OIDC) issuer endpoint is automatically reachable.
- **If Vault is deployed in a different cluster:**
  Ensure that the OIDC issuer endpoint of the SecurSpaces cluster is reachable by Vault.
  If it isn't, you must manually add the **signing public key(s)** of the SecurSpaces cluster.
  For details, see Use Kubernetes for OIDC authentication

Configuration

You can configure Vault in the SecurSpaces Platform using the following four Helm chart values:

```
# hashicorpVault:
#   If set, secrets are stored in Vault instead of the database.
#   vaultAddress: "https://example.com:8200"
#   vaultRoleName: "sds-role"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
#   customMountPath: "" # Default is "secret"
#   vaultCertB64: ""    # Base64-encoded PEM CA certificate (optional)
```

Parameter descriptions

Parameter	Description
vaultAddress	Specifies the Vault address. The Vault instance must be accessible from the SecurSpaces cluster. All platform services use this address to store and retrieve secrets.
vaultRoleName	Specifies the name of the Vault role configured for SecurSpaces. If different Kubernetes services use different service accounts, the `bound_subject` field may vary. You can omit this field when creating the role.
customMountPath	Specifies the Vault path where secrets are stored. Optional. Defaults to `secret`.
vaultCertB64	Specifies the Base64-encoded TLS certificate for Vault. Use this setting if Vault uses a self-signed certificate. Optional.

Related information

- AI Gateway
- Backstage plugin
- Citrix DaaS
- JFrog
- Publish JetBrains Gateway
- Integrations

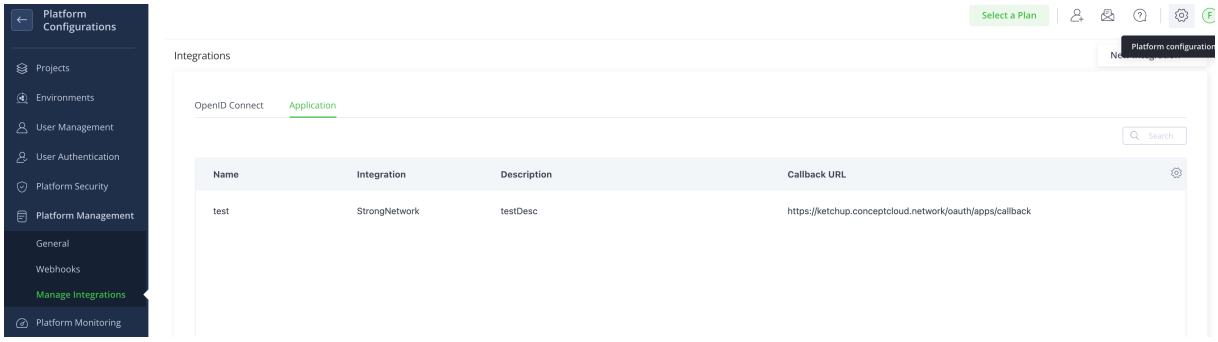
Register JFrog as a third-party app

You can follow these steps to connect your JFrog instance and the Citrix SecurSpaces™ platform.

At the moment this configuration can only be done in self-hosted JFrog instances or by asking the JFrog support team in the SaaS version.

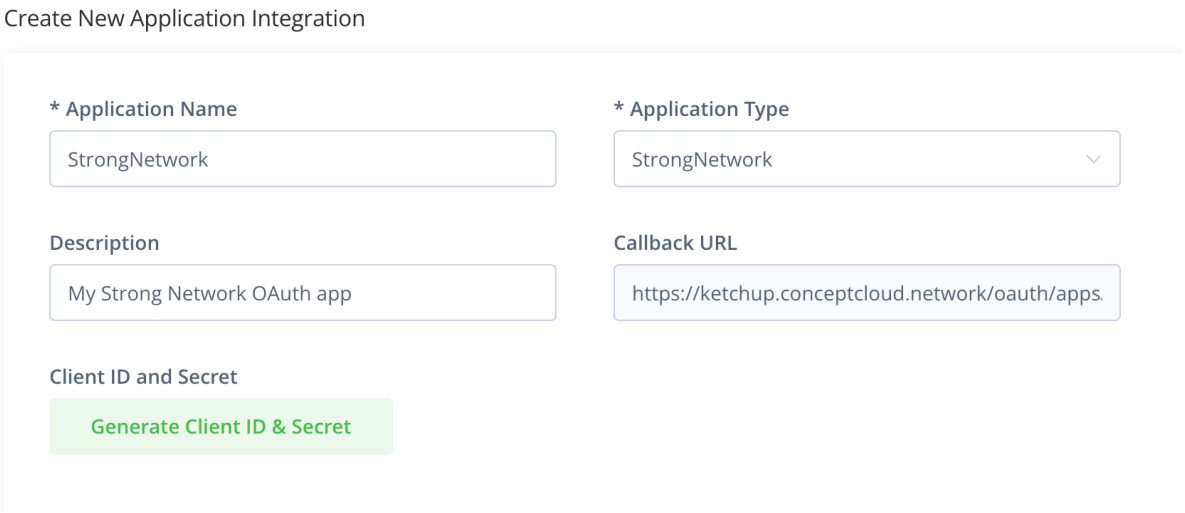
© 1997–2026 Citrix Systems, Inc. All rights reserved.

Log in to your JFrog deployment as the admin go to Platform Management, then Manage Integrations. Go to the tab called “Application”. You can also follow the link:
‘
https://[your_domain_name].jfrog.io/ui/admin/configuration/integrations
‘



Click on “New Integration”of type “Application”and fill in the following fields:

- **Application Name:** Up to you.
- **Application Type:** Select the template you added in the values.yaml file.
- **Description:** Up to you.
- **Callback URL:** You can find it in the Third Party Applications administrator menu in the SecurSpaces platform and has the format of `https://[your_strong_network_domain]/oauth/apps/callback`



Click on **`Generate Client ID & Secret`** and copy the values.

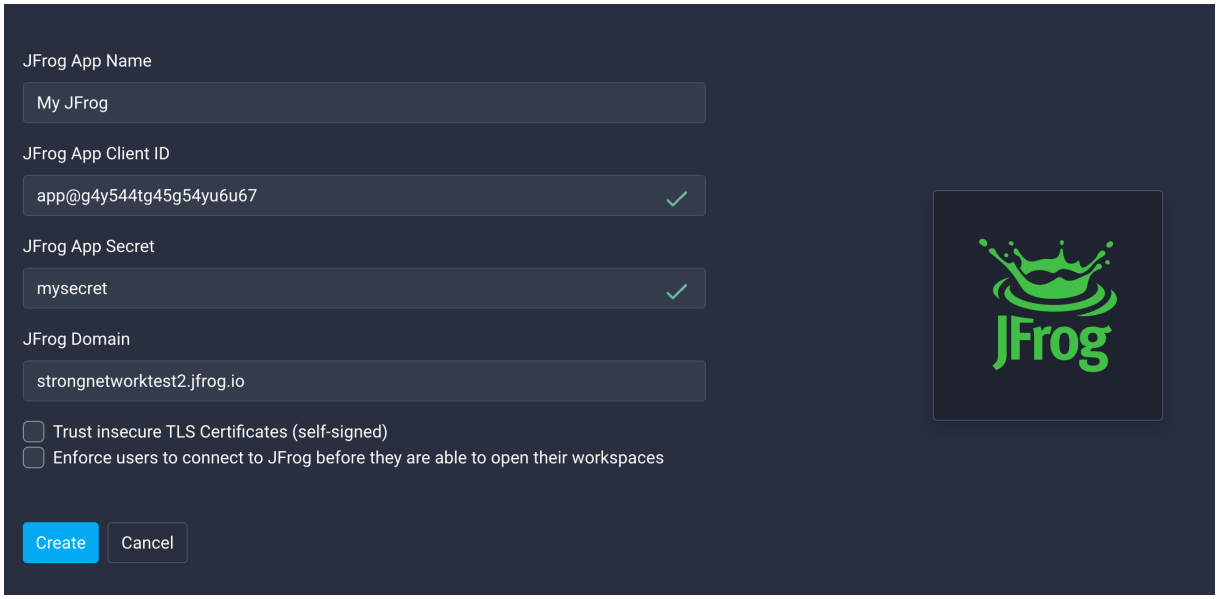
Lastly, sign in as an administrator to the SecurSpaces platform, go to System Configuration → Third Party Applications, and select JFrog. You will need to introduce:

- **Name:** Up to you, it will be displayed to the platform users

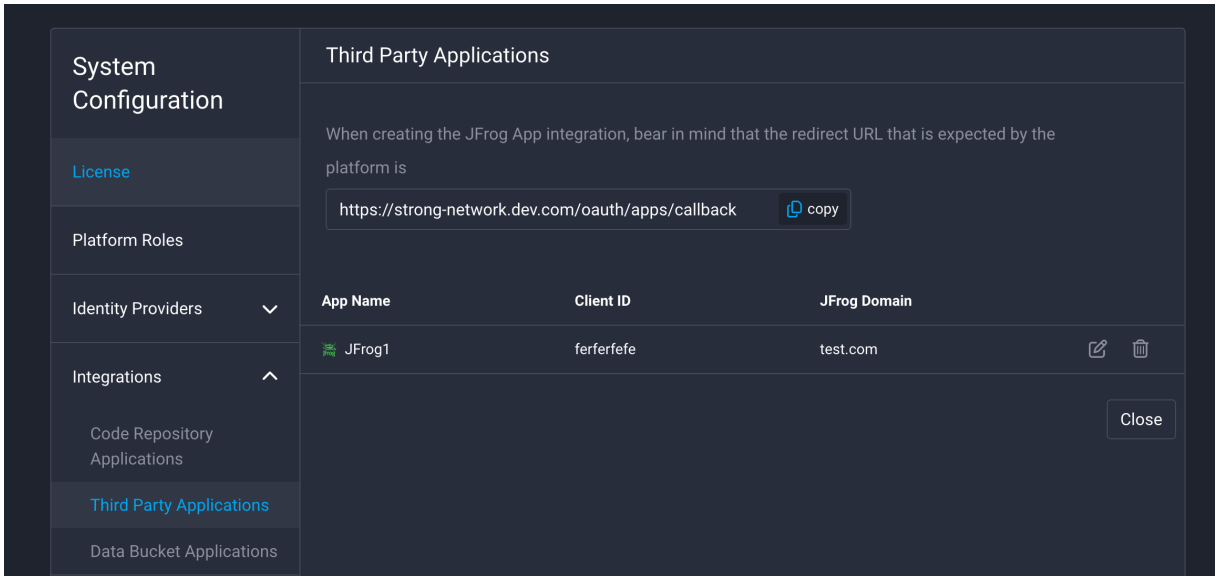
© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Client ID and Secret:** Values copied from JFrog
- **Domain:** Your JFrog domain

You can choose if you want the platform to trust insecure TLS certificates in case your JFrog deployment doesn't have a valid certificate. You may also want users to always connect to JFrog before they access their workspaces, in this case, they will get a popup where they have to connect before opening them. If you don't select this option they will get the popup but can dismiss it.



When the application is configured you can edit it by clicking on the edit icon on the right side.

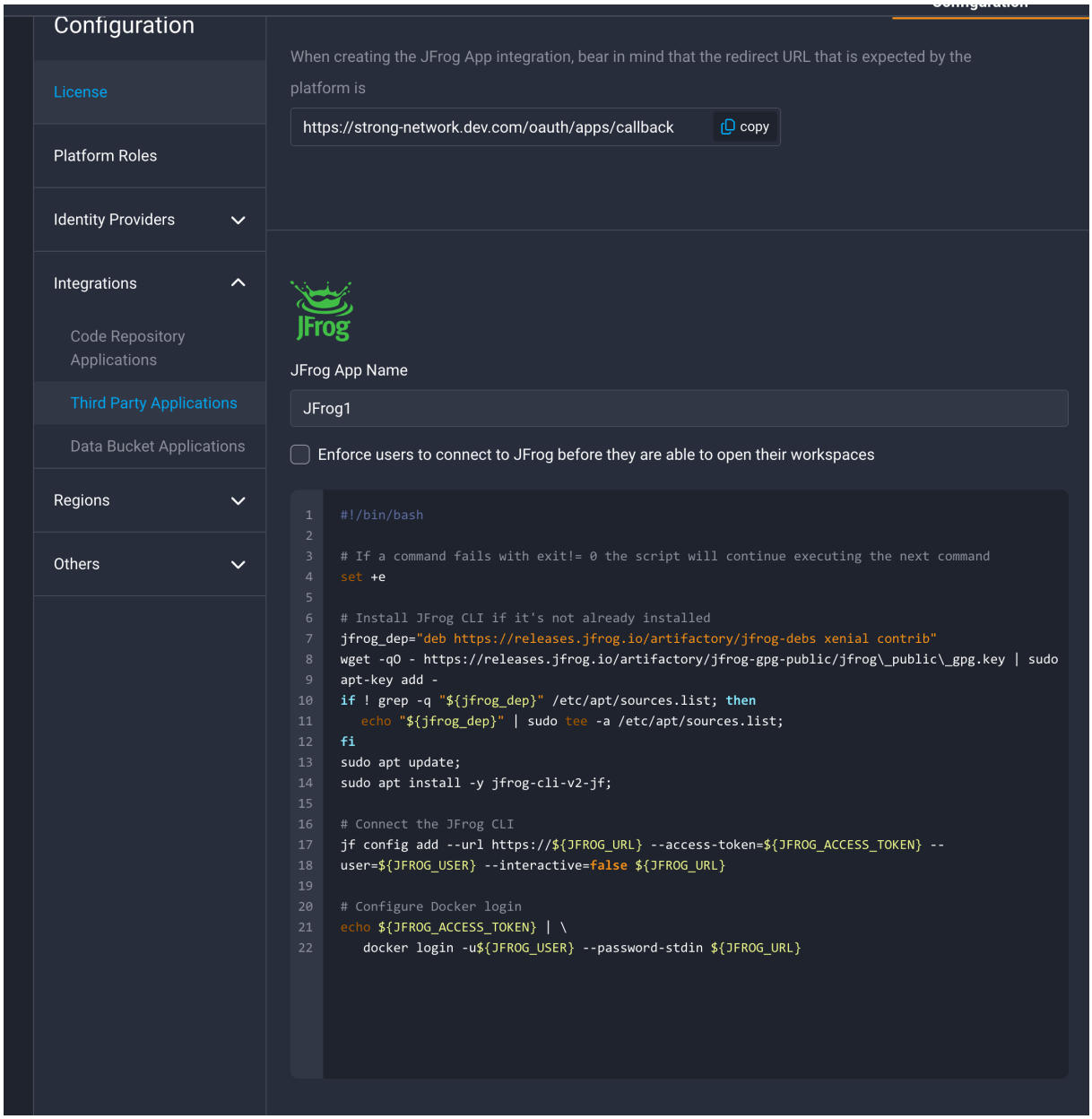


You will see a menu in which you can change some settings. In said menu, you may change the default JFrog startup script. This is a script that will run in every workspace that is owned by a user who has connected their JFrog account. It can be useful to set up specific configurations in all workspaces,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

for example, to configure the different programming languages to fetch the dependencies from your JFrog platform. Each user can build on top of this script, to customize it to their own needs.

If this default script is updated it will be automatically changed for users who haven't defined their custom script.



Finally, if you want to save the changes click on "Save".
Now JFrog is configured across the SecurSpaces platform, ready to be used seamlessly by the users.

Related information

- AI Gateway

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Backstage plugin
- Citrix DaaS
- Publish JetBrains Gateway
- HashiCorp Vault as secret manager
- Integrations

Configure AI Gateway routing in SecurSpaces

August 13, 2026

You can route selected AI service traffic from Citrix SecurSpaces™ workspaces through an internal AI Gateway by configuring Helm values. Use this option
when your organization wants centralized routing or policy control for AI provider endpoints.

Routing AI traffic through a gateway also gives you a single place to measure AI usage. SecurSpaces tags each forwarded request with the user,
project, and workspace it came from, so the gateway can report which teams and developers are consuming tokens. See
Report on AI usage.

What SecurSpaces does, and what the gateway does

The two halves are worth separating, because the split decides which questions this page answers.

SecurSpaces provides the interception and the attribution. The SecurSpaces proxy service already terminates and
inspects workspace traffic. It recognizes requests bound for a configured AI endpoint and tags each one with
the user, project, and workspace that produced it. That is a platform capability and it does not depend on
which gateway you run.

The gateway provides routing and policy. Where a request goes, which credential is used, rate limits,
and the reporting surface are the gateway's job.

Which gateway

SecurSpaces forwards to whatever service address you configure in `platform.aiGateway.address`, so the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

gateway is your choice:

Gateway	Notes
Envoy AI Gateway	What this page covers, including installation
NetScaler AI Gateway	An alternative if you already run NetScaler. See AI gateway in the NetScaler documentation

A note on terminology

The industry has not settled on one name for this component. **AI Bridge** is used by some vendors for the
same concept. Across Citrix products it is called an **AI Gateway**, and that is the term used throughout this documentation.

Model Context Protocol governance

Governing MCP traffic —which tools an agent may reach and what it may do through them —is outside what
SecurSpaces provides. SecurSpaces governs the workspace an agent runs in and attributes what it consumes;
it does not inspect or police MCP calls.

Where MCP governance is required, NetScaler provides it. See
MCP gateway in the
NetScaler documentation.

How AI traffic is routed

AI traffic flows from the workspace to the SecurSpaces proxy service, then to the AI Gateway, and finally to the LLM endpoint.

The SecurSpaces proxy service already terminates, inspects, and re-encrypts HTTPS traffic. AI Gateway routing reuses that capability to identify requests
that are destined for a configured LLM endpoint, add SecurSpaces metadata to them, and forward them to the gateway. Traffic that does not match a
configured endpoint is unaffected.

For running agents inside workspaces —isolation, credential scope, and cost —see
Run AI coding agents in workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Prerequisites

- An internal AI Gateway service that is reachable from the SecurSpaces proxy service pods. If you do not have one,
  see Install Envoy AI Gateway.
- The DNS endpoint names that must be routed through the AI Gateway
- Access to update the SecurSpaces Helm values and run `helm upgrade`

Install Envoy AI Gateway

SecurSpaces works with any gateway that accepts the forwarded traffic. Envoy AI Gateway is the reference
deployment, and this section covers what is specific to using it with SecurSpaces. For the installation itself, follow the
Envoy AI Gateway documentation, which stays current
with the project.

The installation has two parts, both by Helm chart:

Component	Namespace
Envoy Gateway, the control and data plane it builds on	`envoy-gateway-system`
Envoy AI Gateway, its CRDs and controller	`envoy-ai-gateway-system`

> **Note:**
>
> Installing Envoy Gateway does **not** interfere with your existing ingress controller. It creates its own
> proxy pods and services in its own namespace, and leaves NGINX, NetScaler, HAProxy, or anything else
> serving the platform untouched. See Ingress.

Envoy AI Gateway requires Kubernetes 1.32 or later, which any cluster running SecurSpaces already exceeds.
See System requirements.

Pin the chart versions rather than tracking the latest tag. The container tags for in-development versions are
overwritten, so an unpinned install can change under you between deployments.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

When the gateway is running, set `platform.aiGateway.address` to its internal service address, as described in
Configure platform-wide AI Gateway routing.

Increase the client buffer limit

AI requests and responses are considerably larger than typical web traffic, and the Envoy default buffer is
not sized for them. Without this, large prompts and long completions are truncated:

```
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: ClientTrafficPolicy
metadata:
  name: client-buffer-limit
spec:
  targetRefs:
    - group: gateway.networking.k8s.io
      kind: Gateway
      name: <your-gateway-name>
  connection:
    bufferLimit: 50Mi
```

Set explicit CPU and memory on the Envoy proxy deployment as well. The reference configuration leaves them
empty, which is workable in a lab and not in production.

Rate limiting

Envoy AI Gateway can apply token-based rate limits per model, backed by Redis. This is configured entirely on
the gateway and needs nothing from SecurSpaces. See the
Envoy AI Gateway documentation for the `BackendTrafficPolicy` and
token-tracking configuration.

Configure platform-wide AI Gateway routing

Add the `platform.aiGateway` block to your Helm values:

```
platform:
  aiGateway:
    address: "http://envoy-default-envoy-ai-gateway-07856df5.envoy-
       gateway-system.svc.cluster.local"
    headers:
      userIdHeader: "x-user-id"
      projectIdHeader: "x-project-id"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
      workspaceIdHeader: "x-workspace-id"
      hideProjectName: false
    healthCheck:
      enabled: false
      port: "80"
      failOpen: false
    dnsEndpoints:
      - "api.business.githubcopilot.com."
      - "copilot-proxy.githubusercontent.com."
      - "api.openai.com."
      - "chat.openai.com."
      - "chatgpt.com."
      - "api.anthropic.com."
      - "claude.ai."
      - "generativelanguage.googleapis.com."
      - "gemini.google.com."
      - "bard.google.com."
```

The `address` value is the internal service address for the AI Gateway, and can include a scheme. The `dnsEndpoints` list defines the AI provider domains that SecurSpaces intercepts and routes through that gateway. Each entry is a fully qualified domain name with a trailing dot, and wildcard entries such as `"*.openai.azure.com."` are supported.

SecurSpaces ships a default endpoint list that covers the most common public LLM providers, including GitHub Copilot, OpenAI, Anthropic, Google, Azure OpenAI, Amazon Bedrock, Mistral, and Hugging Face. The list is refreshed as part of the normal release cycle. Between releases, keep the list current yourself, and add any private or self-hosted LLM endpoints your organization uses.

Identify AI usage by user, project, and workspace

SecurSpaces adds metadata to every request it forwards to the AI Gateway, so that AI usage can be attributed to the workspace that produced it. The gateway and your monitoring stack use these headers to break down usage by user, project, and workspace.

Configure the header names under `platform.aiGateway.headers`:

Value	Default	Description
`userIdHeader`	`x-user-id`	Header carrying the ID of the user who made the request.
`projectIdHeader`	`x-project-id`	Header carrying the project the workspace belongs to.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Value	Default	Description
`workspaceIdHeader`	`x-workspace-id`	Header carrying the ID of the originating workspace.
`hideProjectName`	**`false`**	When **`false`**, the project header carries the project name. Set to **`true`** to send the numeric project ID instead.

Leave a header name empty to stop SecurSpaces from sending that value.

> **Note**
>
> Keep `hideProjectName` set to **`false`** if you want reports to show readable project names. Set it to **`true`** when project names are themselves sensitive and must not leave the cluster.

Route GitHub Copilot traffic

GitHub Copilot needs extra handling, because Envoy AI Gateway does not recognize it as a provider on its own.
SecurSpaces bridges that gap for `api.githubcopilot.com` and `api.business.githubcopilot.com`: it presents each request in a form the gateway's processors recognize, and adds headers your route rules can select on.

This covers OpenAI-compatible chat completions, the OpenAI responses API, and Anthropic models such as Claude. You configure the routes; the rest is automatic.

Headers to match in your routes

Route rules on the gateway select a backend using the headers SecurSpaces adds:

Header	Value	Match it to
`x-ai-process`	**`true`**	Select AI traffic that the gateway should process
`x-original-host`	The host the workspace requested	Choose the right Copilot backend

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Header	Value	Match it to
`x-ai-provider`	`anthropic`	Route Anthropic requests, when your teams use Claude through Copilot

Configure the gateway

Define a `Backend` for each Copilot host, a `BackendTLSPolicy` so the gateway originates TLS to it, and route
rules that match on the headers above. For example:

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: Backend
metadata:
  name: github-copilot-chat
spec:
  endpoints:
    - fqdn:
        hostname: api.business.githubcopilot.com
        port: 443
---
apiVersion: gateway.networking.k8s.io/v1
kind: BackendTLSPolicy
metadata:
  name: github-copilot-chat-tls
spec:
  targetRefs:
    - group: gateway.envoyproxy.io
      kind: Backend
      name: github-copilot-chat
  validation:
    hostname: api.business.githubcopilot.com
    wellKnownCACertificates: System
```

Then match the headers in your route, and allow a generous request timeout because model responses are slow:

```yaml
rules:
  - timeouts:
      request: 300s
    matches:
      - headers:
          - type: Exact
            name: x-ai-process
            value: "true"
          - type: Exact
            name: x-original-host
            value: api.business.githubcopilot.com
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    backendRefs:
      - name: github-copilot-chat
```

Repeat for the other Copilot hosts your developers reach.

Claude through Copilot

If your teams use Claude, add an `AIServiceBackend` with the Anthropic schema and route `x-ai-provider: anthropic` to it.

Set `prefix: "/v1"` on the schema. SecurSpaces presents Anthropic requests as `/anthropic/v1/messages` so that the gateway selects its Anthropic processor. The prefix is what restores
the real upstream path afterwards. Without it the request reaches Anthropic with the routing prefix still
attached, and fails.

```
apiVersion: aigateway.envoyproxy.io/v1beta1
kind: AIServiceBackend
metadata:
  name: github-copilot-chat-anthropic
spec:
  schema:
    name: Anthropic
    prefix: "/v1"
  backendRef:
    name: github-copilot-chat
    kind: Backend
    group: gateway.envoyproxy.io
```

> **Note:**
>
> Copilot's MCP endpoint can also be published through the gateway with an `MCPRoute`, forwarding the
> caller's `Authorization` header so each developer authenticates as themselves.

Tool types on the responses API

The gateway's external processor rejects a `/v1/responses` request that carries a tool type it cannot parse,
with `400 unknown tool type`. To keep those requests working, SecurSpaces removes unrecognized tool types
before forwarding.

The list is held by SecurSpaces, not by your gateway, so it changes with a SecurSpaces release rather than

© 1997–2026 Citrix Systems, Inc. All rights reserved.

when you upgrade Envoy AI Gateway. It currently allows `function`, `file_search`, `computer_use_preview`, `web_search`, `web_search_2025_08_26`, `mcp`, `code_interpreter`, `image_generation`, `custom`, `local_shell`, `shell`, `web_search_preview`, `web_search_preview_2025_03_11`, and `apply_patch`.

> **Important:**
>
> A tool outside this set is dropped silently. The request succeeds and its token usage is still reported, but that tool never reaches the provider. These are generally client-side tools that the upstream API does not act on, so the effect is usually invisible —but if a developer reports that a tool works outside SecurSpaces and not inside it, this is the first thing to check.

Configure AI Gateway health checking

By default, SecurSpaces does not health check the AI Gateway. Turn on health checking to control what happens to AI traffic when the gateway is unavailable.

Configure health checking under `platform.aiGateway.healthCheck`:

Value	Default	Description
`enabled`	**`false`**	When **`true`**, SecurSpaces runs periodic TCP health checks against the AI Gateway.
`port`	`80`	The port SecurSpaces dials for the health check.
`failOpen`	**`false`**	Controls the behavior when the gateway is unhealthy.

When the gateway is unhealthy:

- With `failOpen: true`, AI traffic bypasses the gateway and goes directly to the AI provider. Developers are not interrupted, but no usage data is recorded while the gateway is down.
- With `failOpen: false`, the request returns an error to the user. Use this setting when AI traffic must never bypass the gateway.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> `failOpen` defaults to **`false`**. If you enable health checking without setting `failOpen`, an AI Gateway outage stops developers from reaching AI providers. Set `failOpen: true` if you would rather lose usage data than block AI requests.

Override AI Gateway routing for a region

For an external region, add `region.aiGateway` to override the platform-wide settings:

```yaml
region:
  isExternalRegion: true
  aiGateway:
    address: "ai-gateway.region.example.svc.cluster.local"
    headers:
      userIdHeader: "x-user-id"
      projectIdHeader: "x-project-id"
      workspaceIdHeader: "x-workspace-id"
    healthCheck:
      enabled: false
      port: "80"
      failOpen: false
    dnsEndpoints:
      - "api.openai.com."
      - "api.anthropic.com."
```

The regional values apply only to that region. If you omit `region.aiGateway`, the region uses the platform-wide `platform.aiGateway` configuration.

Apply the configuration

1. Update your SecurSpaces Helm values file.
2. Run your standard `helm upgrade` command.
3. Restart any affected workspace sessions if you need existing connections to use the new routing.

After the upgrade, SecurSpaces routes requests for the configured DNS endpoints through the configured AI Gateway address.

Report on AI usage

Routing AI traffic through a gateway lets you answer questions such as which models are in use, how many tokens each project consumes, and

© 1997–2026 Citrix Systems, Inc. All rights reserved.

which users generate the most requests. This information supports cost attribution and chargeback discussions.

SecurSpaces does not parse AI traffic or store usage metrics itself. The AI Gateway extracts request counts, model names, and token counts, and exports them to your monitoring system. The accuracy and provider coverage of the resulting reports depend on the parsing support of the gateway you deploy.

Reference deployment

The following stack is validated for use with SecurSpaces:

- **Envoy AI Gateway** —receives AI traffic from the SecurSpaces proxy service, forwards it to the LLM provider, and exposes metrics such as request counts, model names, and input and output token counts.
- **Prometheus** —scrapes and stores the metrics that Envoy AI Gateway exposes.
- **Grafana** —queries Prometheus and presents the dashboards.

Deploy all three in the same Kubernetes cluster as SecurSpaces, then set `platform.aiGateway.address` to the internal service address of Envoy AI Gateway.

This is a reference path, not a requirement. Envoy AI Gateway supports OpenTelemetry and a broad range of monitoring integrations, so you can send the same data to your existing monitoring infrastructure instead.

Sample Grafana dashboard

SecurSpaces provides a sample Grafana dashboard that you can import and adapt. The dashboard reads the SecurSpaces metadata headers, so usage can be filtered and grouped by model, project, and user.

The dashboard includes:

- **Summary totals** —total input tokens, total output tokens, combined token count, and total LLM request count.
- **Token usage rate** —tokens per minute, broken down into input, output, cached input, cache creation, and reasoning tokens.
- **Request rate and response time** —requests per second, and LLM response time at the 50th, 95th, and 99th percentiles.
- **Average tokens per request** —average input and output tokens per request.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Rate limiting** —rate limited requests, requests over the global rate limit, and users over their rate limit.
- **Upstream response time** —response time of the LLM provider at the 50th and 95th percentiles.
- **By Project ID** —token usage, request rate, and total tokens per project.
- **By User ID** —token usage per user over time.
- **By Model** —token usage, request rate, and total tokens per model.

Use the **Model**, **Project ID**, and **User ID** selectors at the top of the dashboard to filter every panel.

> **Note**
>
> Panels that break usage down by project and user rely on the metadata headers described in Identify AI usage by user, project, and workspace.
> If those headers are not configured, the corresponding panels are empty.

Related information

- Backstage plugin
- Citrix DaaS
- JFrog
- Publish JetBrains Gateway
- HashiCorp Vault as secret manager
- Integrations

Publish JetBrains Gateway

This guide describes how to publish JetBrains IntelliJ Gateway as a virtual application in Citrix DaaS. By pre-configuring the Citrix SecurSpaces™ - Remote SSH plugin, you can provide a seamless remote development experience.

Overview

Publishing JetBrains IntelliJ Gateway through Citrix DaaS provides a standardized, secure entry point for developers. This configuration allows developers to use the native performance of JetBrains IDEs while keeping source code and compute resources within the secure SecurSpaces workspace perimeter.

Prerequisites

Before you begin, ensure you have the following:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- SecurSpaces environment: SSH access must be enabled at the platform, organization, and project levels.
- VDA image: A Windows VDA with the JetBrains IntelliJ Gateway installer.
- SecurSpaces plugin: The network.strong.remotessh plugin.

Administrator Guide: Prepare and publish the application

To provide a "zero-install"experience, administrators must include the SecurSpaces plugin in the Gateway installation on the VDA master image.

Install the SecurSpaces plugin (CLI method)

Use the command-line interface to install the plugin for VDA automation and image consistency.

1. Open a command prompt.
2. Navigate to the bin directory of your JetBrains IntelliJ Gateway installation.
3. Run the following command:

```
.\gateway64.exe installPlugins network.strong.remotessh --give-consent-to-use-third-party-plugins
```

> **Note:**
>
> Automation: Include this command in your VDA sealing script or App Layering recipe to ensure every session contains the required plugin.

Publish JetBrains IntelliJ Gateway in Citrix DaaS

1. Open **Citrix Web Studio** and navigate to **Applications**.
2. Select **Add Applications** and choose the **Delivery Group** that contains your prepared VDAs.
3. In **Application Settings**, enter the following:
   - **Path to executable**: C:\Program Files\JetBrains\JetBrains Gateway\bin\gateway64.exe (or your custom path)
   - **Application name**: JetBrains IntelliJ Gateway
4. In **Assignment**, assign the application to the appropriate developer user groups.
5. Select **Finish** to make the app available in the Citrix Workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Publish SecurSpaces Platform URL with Google Chrome in Citrix DaaS

1. Open **Citrix Web Studio** and navigate to **Applications**.
2. Select **Add Applications** and choose the **Delivery Group** that contains your prepared VDAs.
3. In **Application Settings**, enter the following:
    - **Path to executable**: C:\Program Files (x86)\Google\Chrome\Application\chrome.exe (or your custom path)
    - **Command Line Arguments**: {SDS_PLATFORM_URL}
    - **Application name**: SecurSpaces
4. In **Assignment**, assign the application to the appropriate developer user groups.
5. Select **Finish** to make the app available in the Citrix Workspace.

End-User Guide: Connect to SecurSpaces workspaces

After the applications are published, developers can connect to remote environments using JetBrains IntelliJ Gateway.

Set up for the first time

Follow these steps to authorize and configure your first connection between SecurSpaces and the JetBrains IntelliJ Gateway.

1. Open the **Citrix Workspace app** or go to your organization's **Citrix Workspace URL**.
2. Select the **SecurSpaces** icon.
3. In your SecurSpaces platform dashboard, locate your workspace and select **SSH** to initiate the connection.
4. In the **Connect Via SSH** dialog box, select **JetBrains Gateway**.
5. Select **Authorize** to give JetBrains Gateway permission to connect to SecurSpaces.
6. Select your **SecurSpaces workspace** from the list.
7. Select your preferred **IDE** version (such as IntelliJ IDEA, PyCharm, or GoLand).
8. Select **Connect**. The Gateway configures the SSH tunnel and opens your remote project in the JetBrains IntelliJ Gateway.

Launch the application

After the initial setup is complete, you can launch your development environment directly from JetBrains IntelliJ Gateway.

1. Open the **Citrix Workspace app** or go to your organization's **Citrix Workspace URL**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Select the **JetBrains IntelliJ Gateway** icon.
3. Select your **IDE** version.
4. Select **Connect**. The Gateway configures the SSH tunnel and opens your remote project in JetBrains IntelliJ Gateway.

Related information

- AI Gateway
- Backstage plugin
- Citrix DaaS
- JFrog
- HashiCorp Vault as secret manager
- Integrations

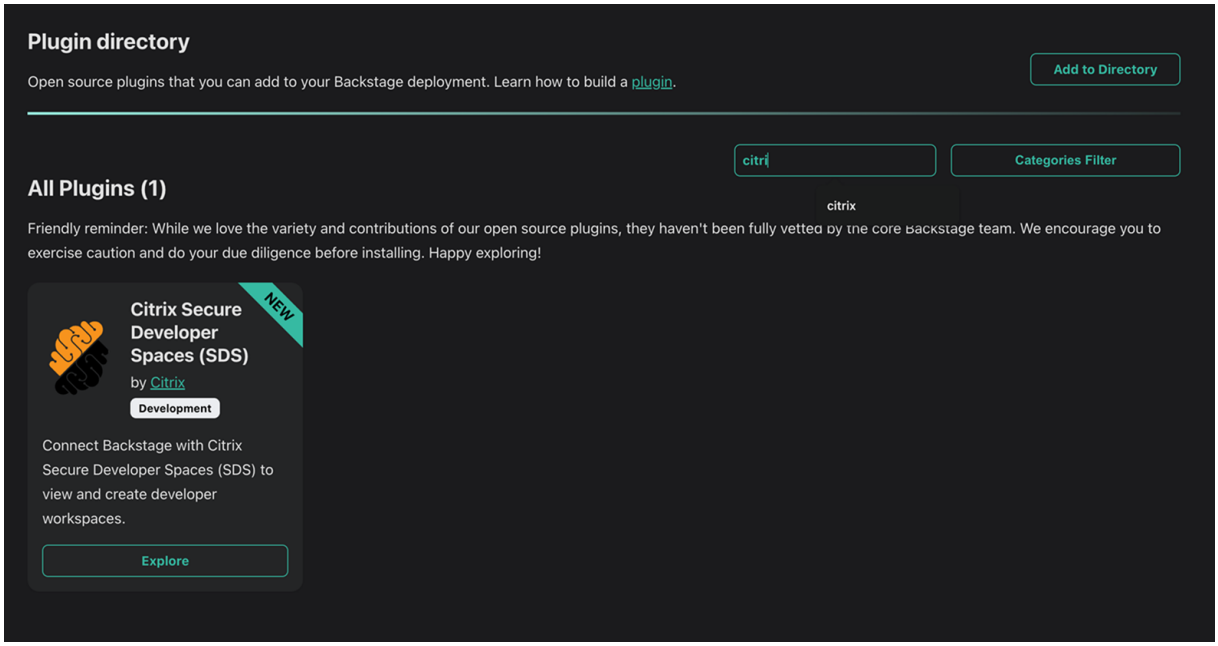
SecurSpaces workspaces plugin for Backstage

Integrate Citrix SecurSpaces™ Workspaces into your Backstage developer portal to direcly manage secure Workspaces.

Overview

Backstage is an open-source developer portal framework that centralizes software components, infrastructure tools, and documentation into a unified interface. It supports a plugin-based architecture, enabling extensibility across both frontend and backend layers.
The SecurSpaces Workspaces Plugin allows developers to view and manage SecurSpaces Workspaces directly from Backstage entity pages. This integration enhances developer productivity by embedding workspace operations into the tools they already use.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

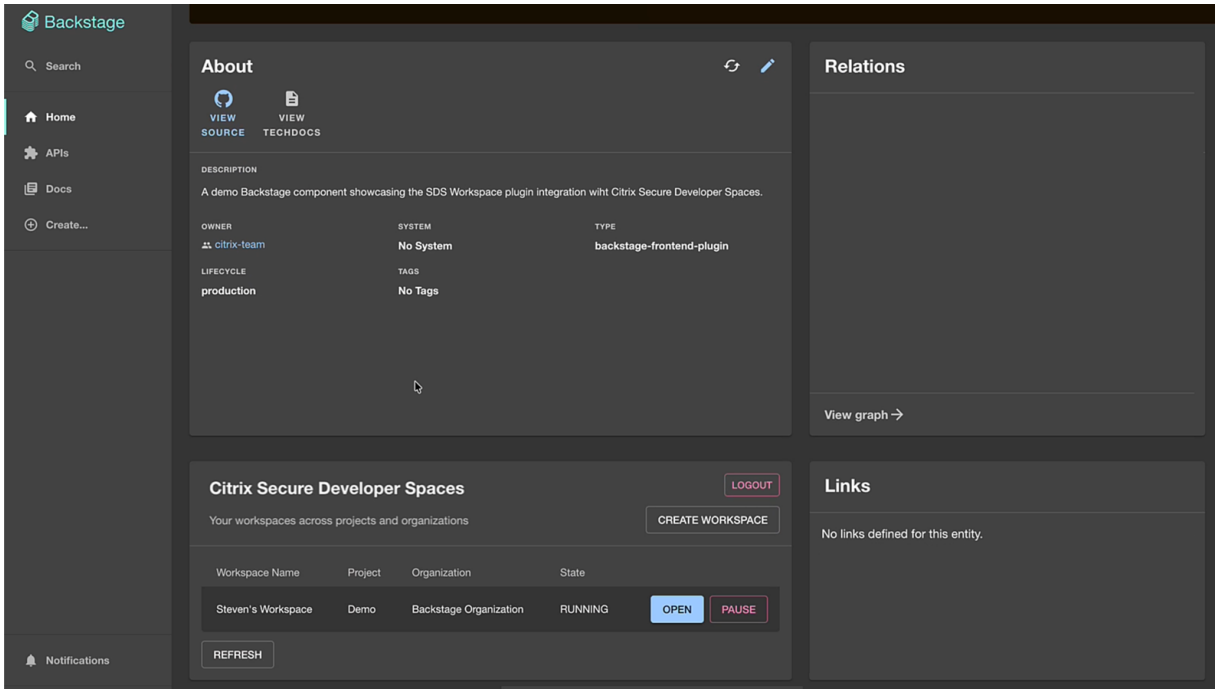


> Note:
>
> The SecurSpaces Workspaces integration requires both frontend and backend plugins to function correctly.

Key Features

- View and manage SecurSpaces Workspaces from Backstage
- Custom SecurSpaces Workspace cards and tabs on entity pages
- Secure backend integration with SecurSpaces platform APIs
- No client-side exposure of credentials

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Prerequisites

- A running Backstage instance
- Access to SecurSpaces
- SecurSpaces platform base URL

Backstage Plugin Architecture

Backstage Plugin Name	Description
@citrixcloud/backstage-sds-workspaces	Frontend plugin for displaying SecurSpaces Workspace cards and tabs
@citrixcloud/backstage-sds-workspaces-backend	Backend plugin for secure communication with the SecurSpaces Workspaces platform

> Important:
>
> To use the SecurSpaces Workspaces frontend plugin, you must also install and configure the backstage-sds-workspaces-backend plugin in your Backstage backend project.
> The backend plugin acts as the bridge between your Backstage instance and the SecurSpaces Workspaces platform, providing all required APIs for the frontend plugin.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

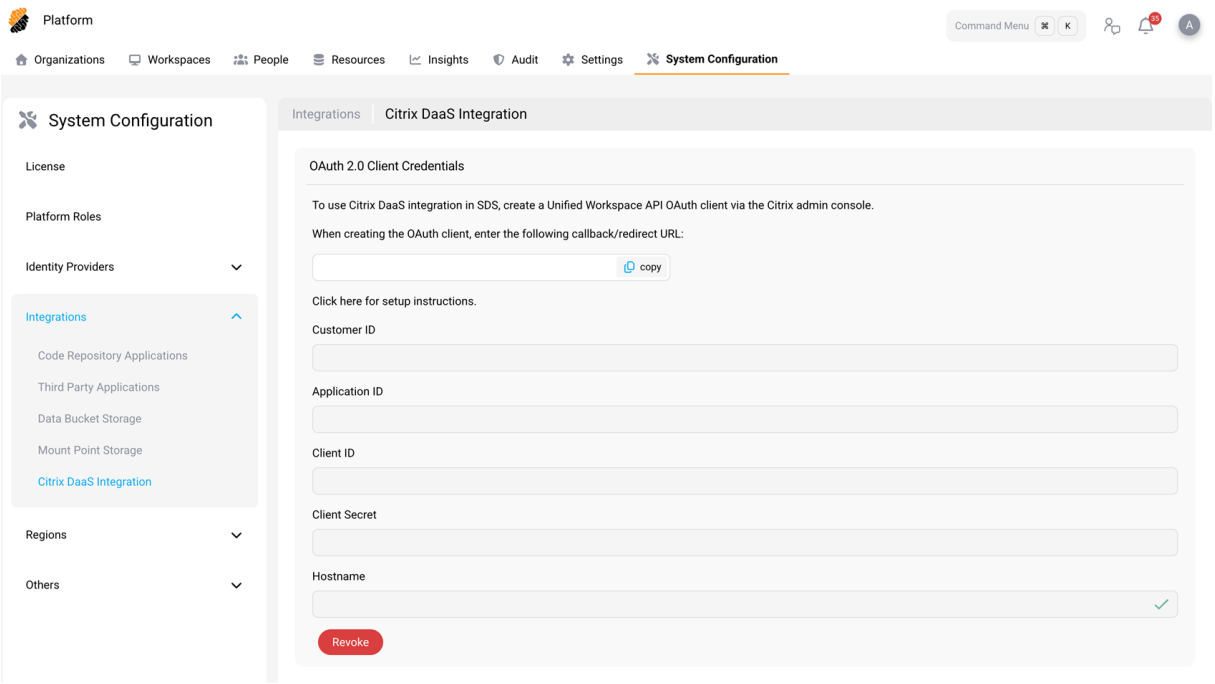
Related information

- AI Gateway
- Citrix DaaS
- JFrog
- Publish JetBrains Gateway
- HashiCorp Vault as secret manager
- Integrations

Citrix DaaS integration

The Citrix DaaS Integration connects Citrix SecurSpaces™ with Citrix DaaS and Citrix Workspace. After you configure the integration, users can sign in once in the SecurSpaces console and access both Kubernetes-based SecurSpaces workspaces and Citrix-based virtual apps and desktops without additional authentication.

This page is available at **System Configuration > Integrations > Citrix DaaS Integration**.



Prerequisites

Before you configure the integration, you need:

- A Citrix Cloud account with an active DaaS subscription.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- A Unified Workspace API OAuth client created in the Citrix admin console. For instructions, see Creating a client in the Citrix developer documentation.

When you create the OAuth client, use the following callback/redirect URL displayed on the configuration page:

```
https://<your-sds-hostname>/oauth/apps/callback
```

Configure OAuth 2.0 client credentials

Use the **OAuth 2.0 Client Credentials** section to connect SecurSpaces to your Citrix Cloud environment.

1. Go to **System Configuration > Integrations > Citrix DaaS Integration**.
2. Enter the following values from your Citrix Cloud OAuth client:

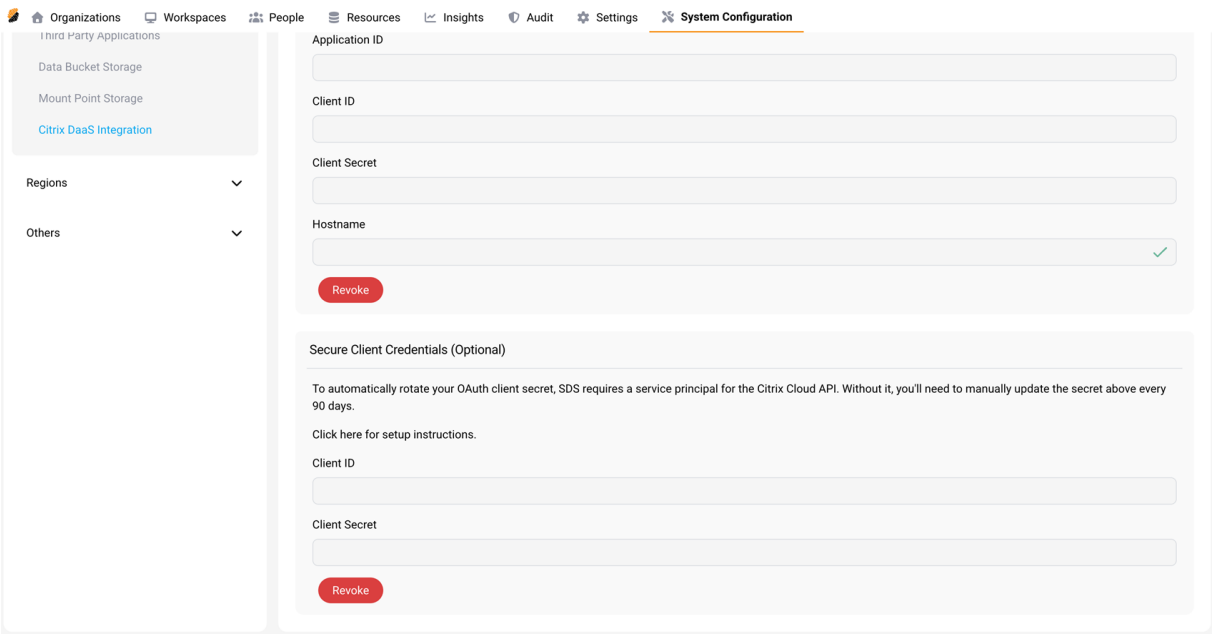
Field	Description
Customer ID	Your Citrix Cloud customer ID.
Application ID	The application ID of the Unified Workspace API OAuth client.
Client ID	The client ID generated when you created the OAuth client.
Client Secret	The client secret generated when you created the OAuth client.
Hostname	The hostname of your Citrix Workspace environment (for example, `yourcompany.cloud.com`).

To remove the stored credentials and disconnect the integration, select **Revoke**.

Configure secure client credentials (optional)

OAuth client secrets expire after 90 days. To avoid manual secret rotation, you can configure a service principal that allows SecurSpaces to rotate the secret automatically.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



1. In the **Secure Client Credentials** section, enter the **Client ID** and **Client Secret** for a Citrix Cloud API service principal.
2. For setup instructions, see Programmatic rotation in the Citrix developer documentation.

Without secure client credentials, you must manually update the client secret on this page every 90 days.

What happens next

After the platform integration is configured:

1. **Users** sign in to their Citrix account from **Profile > Integrations > Citrix Integration**. See Citrix Integration (user profile).
2. **Virtual apps and desktops** assigned to the user in Citrix DaaS appear in the **Citrix DaaS Resources** section on the project Overview page.

Related information

- AI Gateway
- Backstage plugin
- JFrog
- Publish JetBrains Gateway
- HashiCorp Vault as secret manager
- Integrations

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Platform resources

These pages cover the resources the platform pulls and caches for workspaces.

Page	What it covers
Registry access	Restrict which container registries workspaces can pull from, and set the destination registry for images built on the platform.
IDE versions	Learn how to manage VSCode versions for SecurSpaces workspaces.
Image caching	Cache platform container images on workspace nodes so workspaces start without waiting for an image pull.

Registry access

August 19, 2026

Two separate settings control how Citrix SecurSpaces™ deals with container registries: one restricts where
workspace images may come **from**, the other sets where images built on the platform are stored.

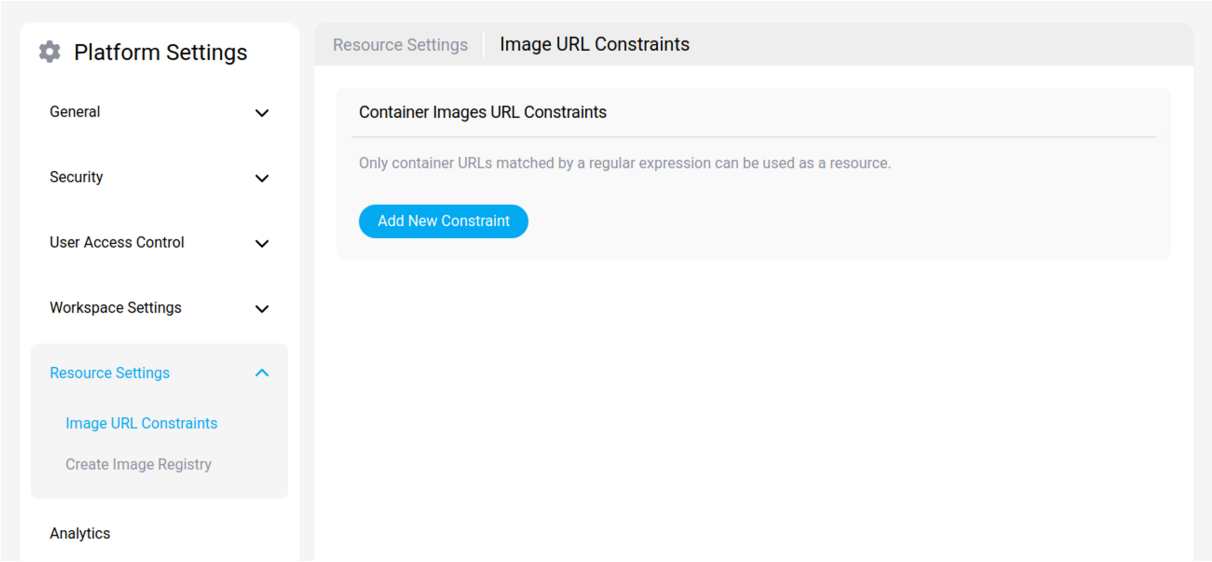
Both are platform-level and require the **Security** permission set to **Manage**. See
Roles and permissions.

Image URL constraints

By default, a project owner can import a workspace image from any registry they can reach. **Registry Access**
Management replaces that with an allowlist: when the feature is enabled, SecurSpaces blocks every registry
except those you list.

This matters because a workspace image is arbitrary code that runs with access to your source. Restricting
the registries is how you stop an image arriving from somewhere nobody has reviewed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



To restrict registries:

1. Open **System Configuration** and select **Registry Access**.
2. Enable the feature.
3. Select **Add Registry** and enter the registry.
4. Set each entry to **Allowed** or **Denied**.

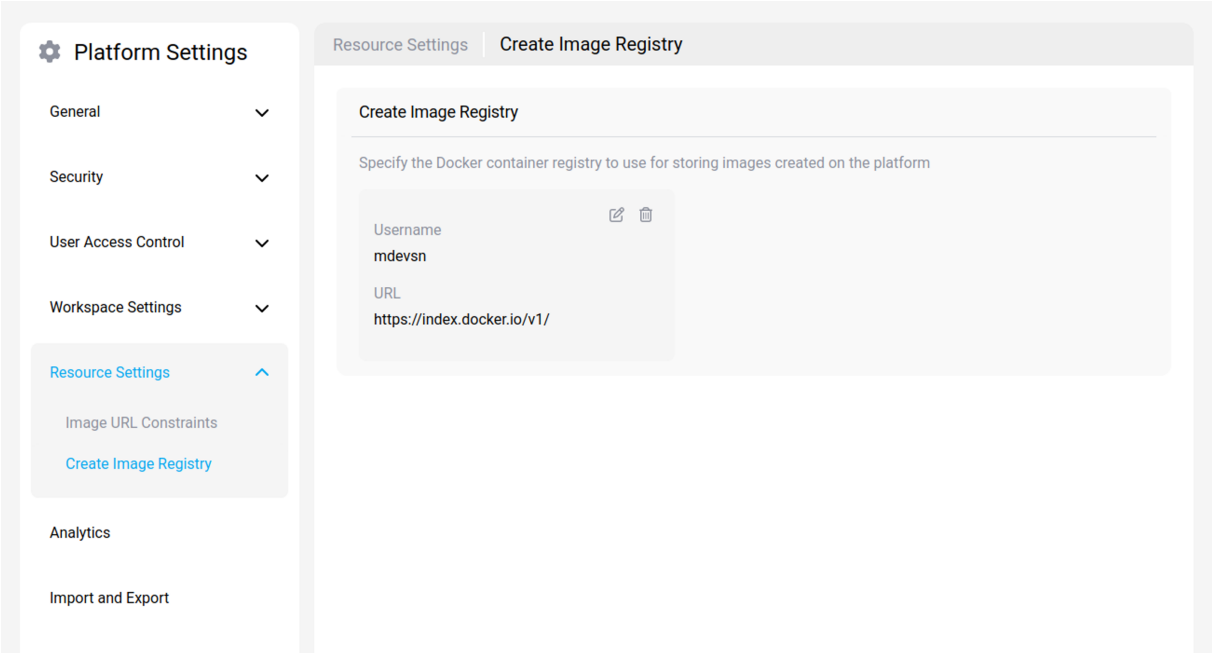
An entry can be switched between allowed and denied without removing it, so you can withdraw a registry
temporarily without losing the list.

> **Note:**
>
> Turning the feature on affects imports made from that point. Review the images already registered on the
> platform as well, since an image imported before the constraint was applied continues to be available. See
> Container images.

Create image registry

Create Image Registry specifies the Docker container registry used to store images **created on the platform**. It is the destination for images SecurSpaces builds, not a source of images to import.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Provide the registry endpoint and the credentials SecurSpaces should use to push to it. Any OCI-compatible
registry works —for example Amazon ECR, Azure Container Registry, Google Artifact Registry, or JFrog Artifactory.

Configuring this registry also switches on **Create Image From Existing**, which lets project members build a
derived image from an existing one by supplying a bash script. Until a destination registry is set, that option does not appear. See
Create an image from an existing one.

If you also use registry access constraints, make sure this destination registry appears in the allowlist. Otherwise the platform can push an image that workspaces are then blocked from pulling.

Related information

- Container images
- Container image requirements
- Workspace images
- System Configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

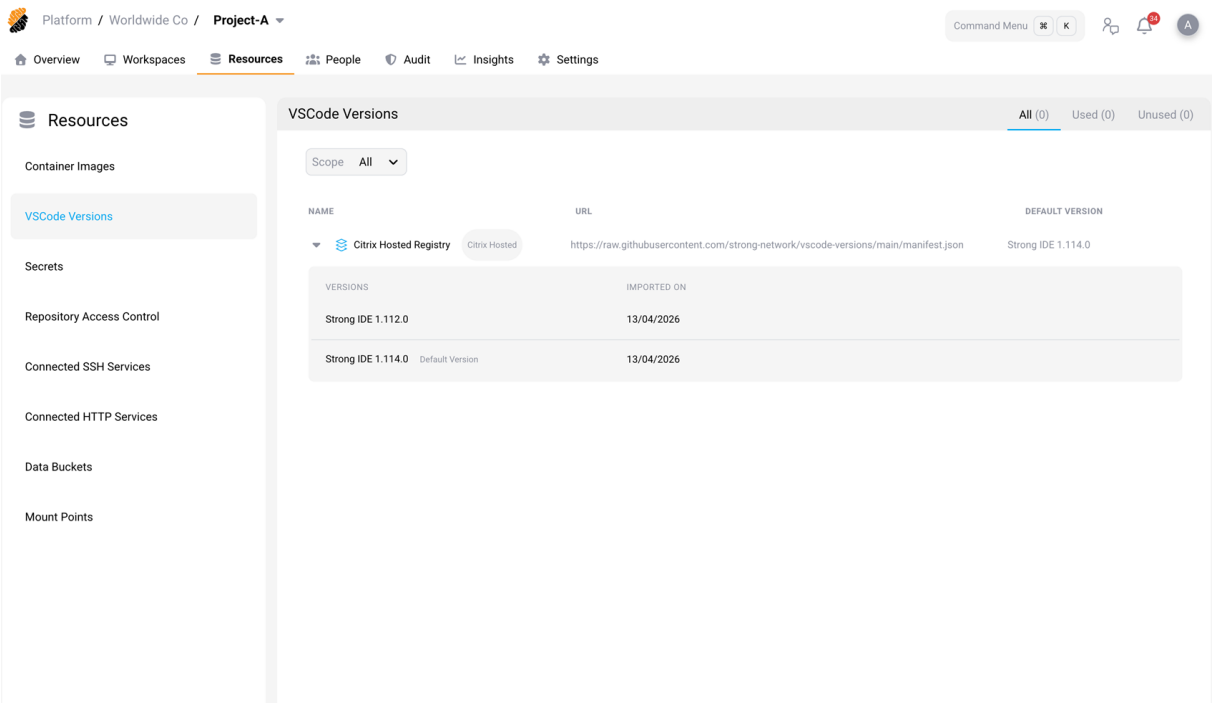
IDE versions

September 28, 2026

The **VSCode Versions** resource page manages the IDE versions available to workspaces. IDE versions are distributed
independently from Citrix SecurSpaces™ platform updates, so you can adopt new versions of the browser-based IDE (Strong IDE) without
waiting for a platform release.

This page is available at:

- **Platform level:** Platform > Resources > VSCode Versions
- **Organization level:** Organization > Resources > VSCode Versions
- **Project level:** Project > Resources > VSCode Versions



How IDE versions are distributed

By default, SecurSpaces uses the **Citrix Hosted Registry** to import IDE versions automatically. SecurSpaces checks for new versions
once an hour. When a new version is published, it is imported and a notification is sent to security officers.

The VSCode Versions page displays:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Column	Description
Name	The registry name. The default is "Citrix Hosted Registry", marked with a **Citrix Hosted** tag.
URL	The source the registry uses. For the Citrix Hosted Registry, this is the version manifest published by Citrix. For a private registry, this is your container registry URL.
Default Version	The IDE version used by workspaces that follow the default.

Expand a registry row to see the imported versions, the date each version was imported, and the version actions
available for that registry.

Network requirements

To use the Citrix Hosted Registry, your deployment must be able to reach the following destinations over HTTPS:

Destination	Used by	Purpose
`raw.githubusercontent.com`	SecurSpaces platform services	Retrieve the version manifest during the hourly check
`registry-1.docker.io`, `auth.docker.io`, `production.cloudflare.docker.com`	Kubernetes nodes	Pull the IDE container images referenced by the manifest

If outbound traffic passes through a corporate proxy, see
External proxy configuration.

If you cannot allow these destinations, use a private registry instead. See
Switch to a private registry.

How the IDE version is resolved

Each workspace stores two settings that determine its IDE version. SecurSpaces evaluates them in the following order when the
workspace starts:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. **Pinned version.** The workspace is set to a specific IDE version. That version is used.
2. **Default version.** The workspace uses **Use latest platform default version**. SecurSpaces resolves the default version
   for the workspace's scope, searching project scope first, then organization scope, then platform scope.
3. **System default.** Neither setting applies. The workspace uses the built-in IDE image that ships with the
   installed platform release.

If a pinned or default version cannot be resolved —for example, because it was deleted —the workspace falls back
to the system default image.

For how to configure these settings on a workspace, see
Create a Workspace.

Set the default version

The default version applies to every workspace in scope that uses **Use latest platform default version**.

1. Go to **Resources > VSCode Versions**.
2. Expand the registry row.
3. Point to the version you want to use, and then select **Set as Default**.

The **Default Version** column updates to the version you selected.

You need permission to manage container images to change the default version.

If no default version has been set for a scope, SecurSpaces selects one automatically after an import:

- For the Citrix Hosted Registry, SecurSpaces selects the version with the highest upstream VSCode version.
- For a private registry, SecurSpaces selects the most recently imported version.

SecurSpaces never replaces a default version that an administrator has set.

Version lifecycle

Each hourly import reconciles the imported versions against the registry contents:

- New versions are imported and added to the list.
- Metadata for versions that are already imported is refreshed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Versions that are no longer published are deleted.

Before a version is deleted, SecurSpaces moves any workspace or workspace template that uses it to the current default version.
The affected workspaces pick up the new version on their next start.

> **Note:**
>
> A workspace that is migrated this way becomes pinned to the version it was moved to. It no longer follows the default
> version. To restore automatic updates, select **Use latest platform default version** on the workspace again.

The current default version is never deleted automatically. To remove it, set another version as the default first.

Registry types

SecurSpaces supports two registry types, and the actions available on the VSCode Versions page depend on which one you use.

Action	Citrix Hosted Registry	Private registry
Set a version as default	Yes	Yes
Add a version	No	Yes
Rename a version	No	Yes
Delete a version	No	Yes

Versions in the Citrix Hosted Registry are managed by Citrix and cannot be edited.

Switch to a private registry

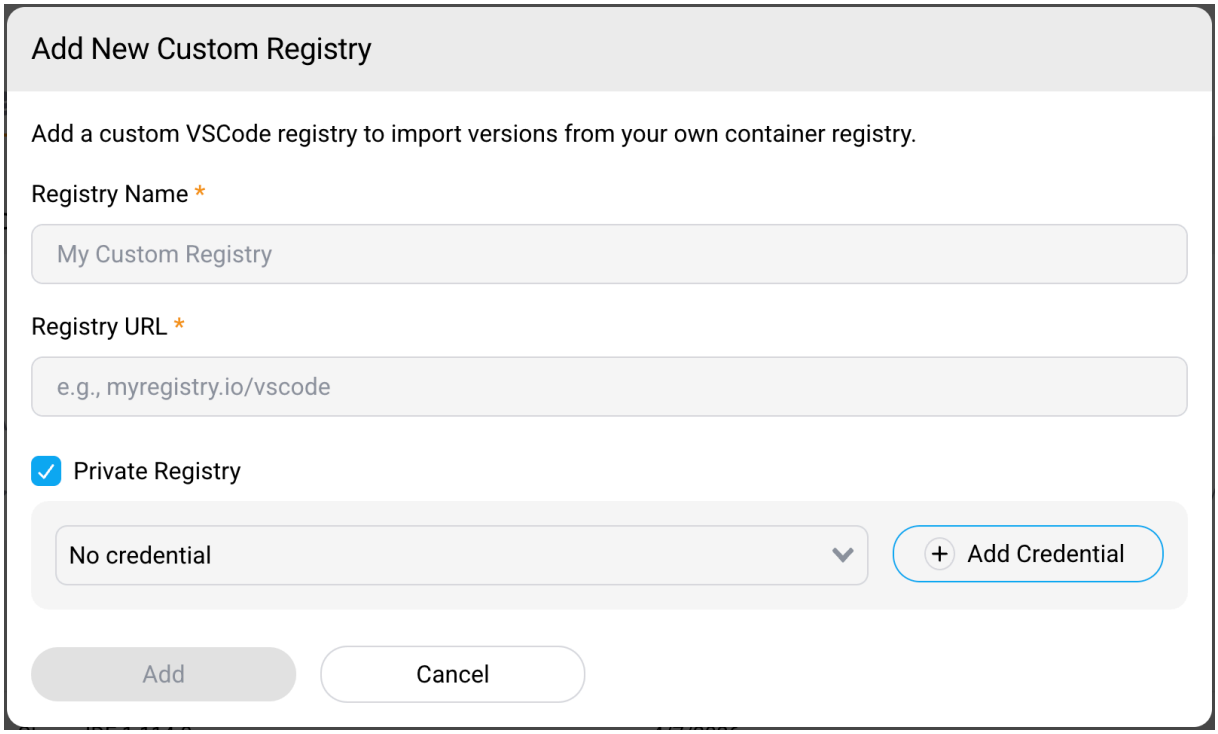
If your deployment is behind a corporate proxy or firewall and cannot reach the Citrix-hosted registry, a platform
administrator can switch to a private registry that hosts your own IDE images.

1. Go to **Platform > Resources > VSCode Versions**.
2. Select **Change to Private Registry**.
3. Enter a **Registry Name** and the **Registry URL** of your container registry, for example `myregistry.io/vscode`. Both fields are required, and the name must be unique within the scope.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

4. If the registry requires authentication, select **Private Registry** and then select a credential from the list. To
   create a credential without leaving the dialog, select **Add Credential**.
5. Select **Add**.

Add stays unavailable until the registry name and URL are filled in, and —if **Private Registry** is selected —
until a credential is chosen.



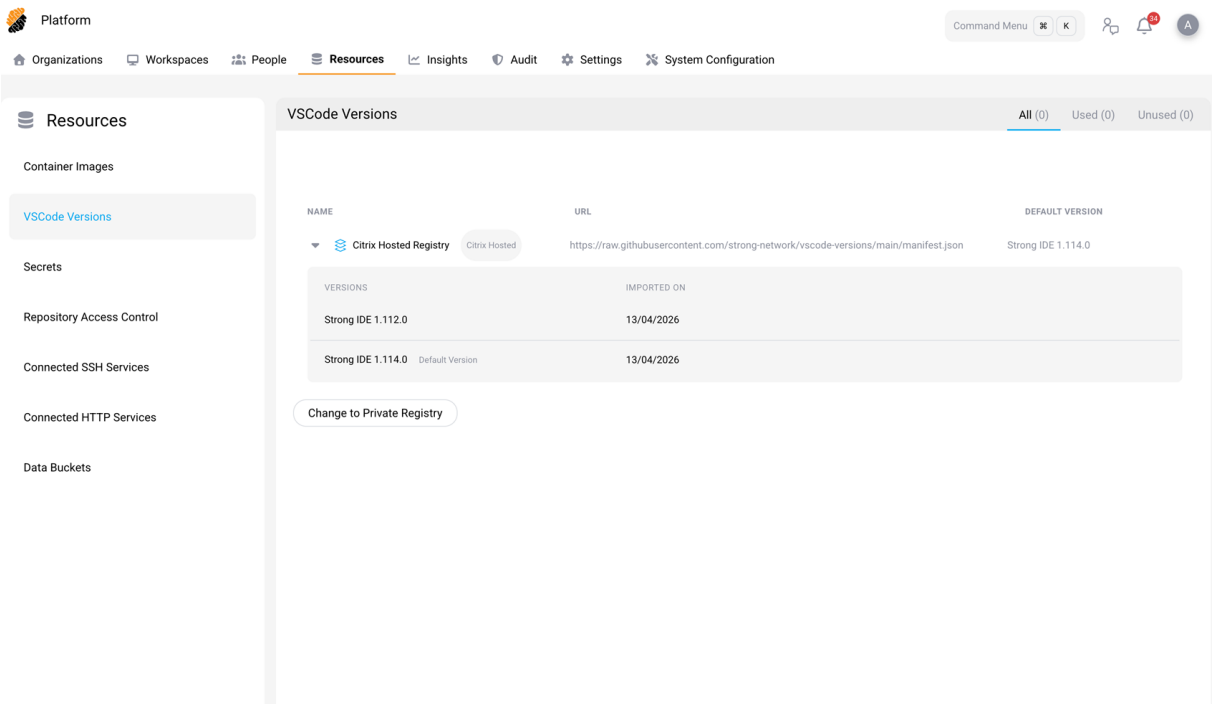
After you add a private registry, add the IDE versions you want to make available:

1. Expand the private registry row.
2. Select **Add New Version**.
3. Enter a **Version Name** and the **Image Tag**, for example `1.95.2`. SecurSpaces combines the tag with the registry URL to
   form the full image URL.
4. Optionally, select **Set as default version**.

> **Note:**
>
> A private registry replaces the Citrix Hosted Registry —it does not supplement it. While a private registry is
> configured at platform level, the Citrix Hosted Registry is hidden from the VSCode Versions page

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> and SecurSpaces stops
> importing versions from it.



The **Change to Private Registry** button is available only on the platform-level VSCode Versions page.

Switch back to the Citrix-hosted registry

1. Go to **Platform > Resources > VSCode Versions**.
2. Select **Change to Citrix-Hosted Registry**.
3. Review the list of affected workspaces.
4. Select **Migrate Workspaces & Switch**.

All versions from the private registry are deleted and the affected workspaces are moved to the default version from
the Citrix Hosted Registry. SecurSpaces resumes importing versions from the Citrix Hosted Registry.

Notifications

When a new IDE version is imported, SecurSpaces sends a notification to security officers:

> A new VSCode version “Strong IDE X.X.X”is now available.

The notification appears in the notification bell in the top navigation bar.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Manage VSCode versions through the API

The REST API can manage VSCode versions at platform, organization, and project scope. Use it when you want to
import tested IDE images, set defaults, or keep many projects aligned without using the console.

Version operations are available at these scope roots:

- Platform: `/v1/vscode-versions`
- Organization: `/v1/organizations/{ organizationId } /vscode-versions`
- Project: `/v1/projects/{ projectId } /vscode-versions`

Each scope supports listing versions, importing a version from a registry by image tag, renaming a version,
deleting a version, and setting a version as the scope default.

Platform-level custom registry management is available at `/v1/vscode-registries`. You can list custom
registries, add one, rename it, or delete it. A custom registry name must be unique within its scope, and each
scope can have one custom registry.

For endpoint details, request and response fields, and authentication, see
REST API.

View workspace VSCode versions through the API

Workspace API responses include two fields that describe the IDE version assigned to a workspace:

Field	Description
`vscode_version_id`	The ID of the version the workspace is pinned to. `0` when the workspace is not pinned.
`vscode_version_auto_update`	**`true`** when the workspace follows the default version for its scope.

The two fields combine as follows:

- `vscode_version_auto_update` is **`true`** and `vscode_version_id` is `0` —the workspace follows the default version.
- `vscode_version_auto_update` is **`false`** and `vscode_version_id` is set —the workspace is pinned to that version.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `vscode_version_auto_update` is **`false`** and `vscode_version_id` is `0` —the workspace uses the system default image.

Use these fields when you need to audit IDE versions, build integrations, or troubleshoot version-specific behavior.

For more information about available API endpoints, see REST API.

Audit VSCode version changes

Changes to VSCode versions and registries are recorded as events on the `vscodeVersions` resource type. For more information, see SIEM Integration.

Related information

- Image caching
- Registry access

Image caching

August 27, 2026

Pulling a workspace image is usually the largest single cost when a workspace starts. A workspace image is often more than a gigabyte, and a node that has never run that image has to download and unpack it before the workspace can begin.

Image caching removes that wait. Citrix SecurSpaces™ keeps the images you mark as cached present on workspace nodes, so a workspace scheduled onto one of those nodes starts without pulling anything.

What you need

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Requirement	Detail
Permission	The **Security** permission set to **Manage** —the security officer role
Scope	Platform-level container images only. Images defined at organization or project level cannot be cached.

Caching is unavailable if a platform administrator has disabled image management.

Cache an image version

Caching is set per **version**, not per image, so you can cache the tags developers actually use and leave older ones alone.

1. In the console, go to **Resources > Container Images**.
2. Select the arrow beside an image to expand its versions.
3. In the **Cached Image** column, turn on the toggle for each version you want cached.

The change takes effect immediately and applies across every region.

To stop caching a version, turn the toggle off. The image is no longer kept on nodes, and the next workspace
that needs it pulls it normally.

> **Note:**
>
> The platform default IDE image is cached automatically when it is ready and not past its deprecation date.
> You do not need to mark it. See
> IDE versions.

How images stay on the nodes

Understanding the mechanism explains both the benefit and the cost.

SecurSpaces runs a deployment whose only job is to hold the cached images. Each of its pods references every
cached image, which makes Kubernetes pull those images onto whichever node the pod lands on. The pods are
spread one per workspace node, and they do no work beyond existing —their presence is what keeps

© 1997–2026 Citrix Systems, Inc. All rights reserved.

the images
on the node.

Two consequences follow:

- **The node disk carries every cached image.** Cache the versions in active use rather than the whole
  history of an image. A workable pattern is the current default tag and the one before it, so a rollback is
  also fast, with earlier versions left uncached.
- **The cache follows the workspace nodes.** As the cluster scales, the deployment is resized so new nodes
  also receive the images.

Cached images and warm nodes

The same deployment provides warm nodes. Its size is the number of nodes currently running workspaces, plus a
buffer derived from the warm node setting for the region.

That buffer is what makes the two features work together: without it, images are cached only on nodes that
are already busy, and a workspace that triggers a scale-out still waits for the pull. Warm nodes keep spare
capacity that already holds the images, so the first workspace of the day avoids both costs at once.

Configure the buffer under **System Configuration > Regions**. See
Warm Nodes Configuration.

> **Note:**
>
> The buffer is held at zero until at least one workspace is running in a region, so an idle cluster is not
> kept warm for nobody.

Confirm caching is working

Two checks, from either side:

- **In the console**, the **Cached Image** toggle is on for the version.
- **For a developer**, the workspace startup log no longer shows a long pull for that image. Open it from
  More Actions (**…**) on the workspace, then **Logs**, and look at the **Setting up container**
  phase. See Troubleshooting.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If the pull still appears, the workspace was scheduled onto a node the cache has not reached yet, which
usually means the cluster has just scaled out.

Related information

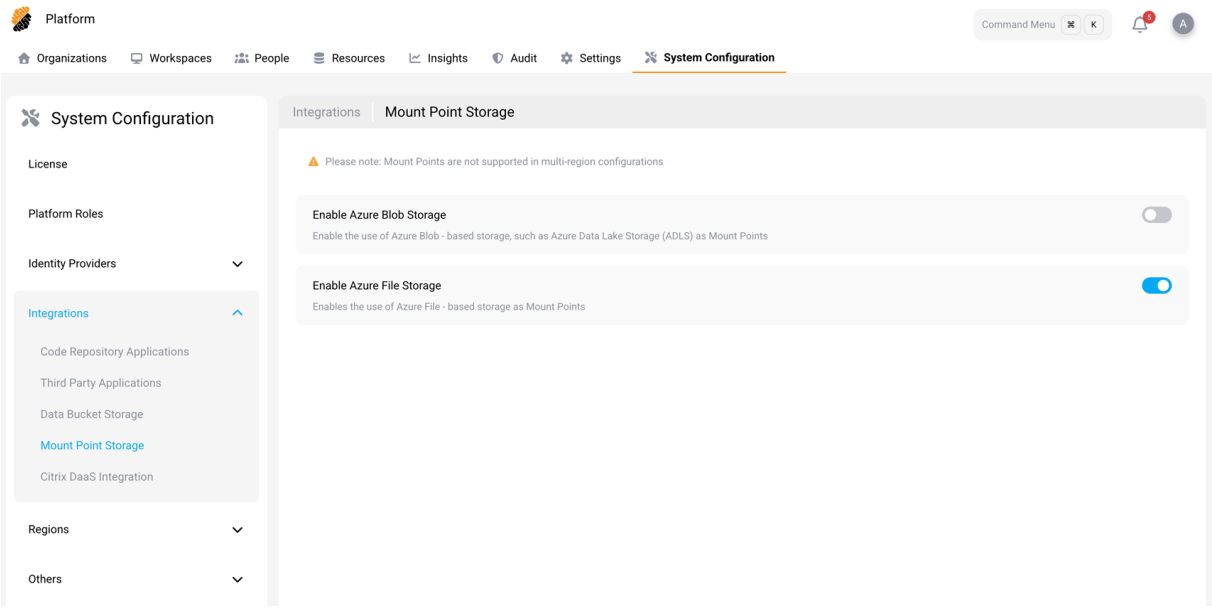
- Custom workspace images
- IDE versions
- Registry access
- Regions
- Sizing

Mount point storage

September 29, 2026

The **Mount Point Storage** section allows platform administrators to enable or disable Mount Point storage types for the platform.

Navigate to **System Configuration > Integrations > Mount Point Storage**.



The preceding screenshot is taken from an Azure cluster. An AWS cluster shows one switch, **Enable AWS File**
Storage, in place of the two Azure switches.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> Mount Points are not supported in multi-region configurations.

Which storage types appear

The switches on this page depend on the cloud that backs Mount Points, and that cloud is detected rather than
chosen. The platform reads the CSI provisioners of the Mount Point storage classes present on the cluster:
`efs.csi.aws.com` means AWS, and `file.csi.azure.com` means Azure. Nothing on this page sets the provider, and a
cluster is treated as backed by one cloud only.

- **An Azure cluster** shows **Enable Azure File Storage**, which enables Azure File-based storage as Mount
  Points. It also shows **Enable Azure Blob Storage**, for Azure Blob-based storage such as Azure Data Lake
  Storage (ADLS). That switch is not available yet. It cannot be turned on, and the platform rejects a request to
  create an Azure Blob Mount Point.
- **An AWS cluster** shows one switch, **Enable AWS File Storage**, which enables both Amazon Elastic File System
  (EFS) and Amazon S3 Files. There is no separate switch for each backend, so you cannot allow one and block the
  other from this page. The backend a Mount Point uses is decided by the `provisioningMode` parameter on the
  storage class you author —`efs-ap` for Amazon EFS, `s3files-ap` for Amazon S3 Files —so a project owner
  chooses the backend by choosing the storage class. Give each storage class a name that makes the backend
  obvious. See
  Configure SecurSpaces for AWS Mount Points.
- **A cluster holding both AWS and Azure Mount Point storage classes** shows no switches. The provider cannot be
  resolved, the feature is withdrawn from users, and the page reports the conflict and lists the providers it
  detected. Remove one cloud's Mount Point storage classes and revisit the page.
- **A cluster with no eligible Mount Point storage class** also shows no switches. The page reports that no

© 1997–2026 Citrix Systems, Inc. All rights reserved.

eligible storage class was detected, and the list of detected providers is empty. An empty list means nothing
was recognized. Two entries mean the mixed-cloud conflict described earlier.

Enable a storage type

1. Navigate to **System Configuration > Integrations > Mount Point Storage**.
2. Turn on the switch for the storage type you want.

When a storage type is enabled, project owners can create Mount Points of that type, and developers can attach them
to workspaces. When it is turned off, the **Add Mount Point** button and the rest of the create flow are hidden, and
no new Mount Points of that type can be created.

> **Warning**
>
> Turning a storage type off does not disconnect storage that is already in use. The platform checks the switch
> only when a Mount Point is created, so a workspace that already has one attached keeps mounting it, with the same
> read and write access, after the switch is off. Existing Mount Point configurations are not deleted either, and they
> become available again if you turn the switch back on. To cut off access, delete the Mount Point, or remove it
> from each workspace and workspace template that uses it.

The project and workspace views give no reason when Mount Points are unavailable. They simply hide the option. Open
this page first when a user reports that Mount Points have disappeared.

AWS storage classes

For AWS storage, an administrator must also provide eligible Kubernetes storage classes for the backing
storage. Project owners see only supported storage classes when they create or attach AWS Mount Points.

AWS Mount Points require AWS resources that Citrix SecurSpaces™ does not create, including the file system,
per-Availability-Zone mount targets, IAM roles, and the storage classes themselves. The AWS storage

© 1997–2026 Citrix Systems, Inc. All rights reserved.

type does not
appear here until an eligible AWS storage class exists on the cluster. See
Prepare AWS storage for Mount Points.

> **Note**
>
> After you add or remove a Mount Point storage class, allow about 30 seconds before this page reflects the change.
> The platform caches the detected provider and the list of eligible storage classes for that long. No restart is
> needed, so do not restart pods or force a rollout while you wait.

Related information

- Mount Points, for creating and using Mount Points within a project
- Prepare AWS storage for Mount Points
- Configure SecurSpaces for AWS Mount Points
- Choose between data buckets and mount points

Prepare AWS storage for Mount Points

September 29, 2026

AWS Mount Points give a project shared, persistent, POSIX file storage that is mounted into its workspace pods at
`/mnt/<mount-point-name>`. Every workspace that mounts it reads and writes the same files, live. It is collaborative
shared storage, not per-user storage.

Two AWS backends are supported, both through the `efs.csi.aws.com` driver:

- **Amazon EFS** —elastic NFS file storage.
- **Amazon S3 Files** —a file system built on EFS over an S3 bucket. The same bytes are reachable as objects and
  as files.

> **Important**
>
> AWS Mount Points depend on AWS resources that Citrix SecurSpaces™ does not create and does not validate. Unlike Azure file
> storage, the EFS CSI driver cannot create a file system —it can only carve access points inside a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> file system
> that already exists. You must provision the file system, mount targets, security groups, IAM roles, and
> StorageClasses yourself before anyone can use the feature.

If these prerequisites are missing, the feature usually fails silently rather than reporting an error. An empty
create dialog, a workspace that hangs while starting, or a mount that runs unencrypted are all symptoms of
incomplete AWS preparation.

Priority checklist

Review these before your first deployment. Each one either breaks the feature or weakens data security.

Requirement	What happens if you miss it
Set `tls` as a mount option on every StorageClass and every PersistentVolume you author.	NFS traffic travels unencrypted over the VPC on port 2049.
Grant the workspace service read access to StorageClasses.	The encryption check cannot run and is skipped, so an unencrypted mount can succeed.
Name an access point in `volumeHandle`. Never publish a bare file system id.	The mount exposes every other project's data on that file system.
Label every PersistentVolume you author with `strong.network/mount-point-scope`.	An unlabeled volume is offered to nobody. A wrongly scoped one is offered to the wrong projects.
Set `persistentVolumeReclaimPolicy: Retain` on volumes you author.	Deleting a Mount Point can destroy the underlying data.
Set `storageClassName: ""` on volumes you author.	A value other than the empty string never binds to the claim SecurSpaces creates, so the mount point times out into an error state.
Create one mount target per Availability Zone your node group uses, and allow NFS on port 2049.	Workspaces scheduled in an Availability Zone with no mount target hang while starting.
Create the file system with encryption at rest, and enable automatic backups.	Data is stored unencrypted, and deletions are permanent and unrecoverable.
Use one cloud provider per cluster.	Mixed AWS and Azure StorageClasses withdraw the feature from all users.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> Do not rename the technical identifiers in this article to match the product name. The volume label key
> `strong.network/mount-point-scope`, the metric names, and the Helm value paths kept their original spelling
> deliberately. A renamed label makes the volume visible to nobody, with no error message.

How to use this section

Page	What it covers
Prepare the AWS resources	Mount targets, security groups, Amazon EFS and Amazon S3 Files prerequisites, IAM scoping, instance metadata hardening
Configure SecurSpaces	The StorageClass, encryption in transit, and enabling the feature
Author a PersistentVolume	Publishing your own volume for Attach Existing
Troubleshooting	Symptoms, causes, and fixes
Mount point limits and behavior	Quotas, deletion semantics, Amazon S3 Files behavior, multi-region, and metrics

Prepare the AWS resources

September 29, 2026

Citrix SecurSpaces™ does not create or validate any of the AWS resources on this page. Provision them before
anyone uses the feature.

Prerequisites common to Amazon EFS and Amazon S3 Files

Create a mount target in every Availability Zone

Create a mount target in every Availability Zone in which the node group can place workspace nodes. Take the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Availability Zone list from the node group's autoscaling group, not from where nodes happen to run today. EFS
allows one mount target per Availability Zone, shared by all Mount Points on that file system.

A workspace scheduled into an Availability Zone with no reachable mount target hangs in `ContainerCreating`, and
Kubernetes cannot compensate. The mount target is not represented in Kubernetes, so this is the only symptom.

Mount target creation is slow. Wait until every mount target reports `available` before you create Mount Points.

Scope the NFS rule to a security group

The source of the mount target's inbound NFS rule decides who can mount the file system. If you allow the whole
VPC CIDR, every workload in the VPC can mount it, not only your workspace nodes.

Set the source of the inbound rule to the EKS worker node security group rather than an IP range:

```
aws ec2 authorize-security-group-ingress \
  --group-id <mount-target-sg> --protocol tcp --port 2049 \
  --source-group <eks-node-security-group-id>
```

Both halves of the rule are required:

- **Mount target security group, inbound** —TCP 2049, source is the node security group.
- **Node security group, outbound** —TCP 2049, destination is the mount target security group.

Most EKS node groups keep the default allow-all egress, so the outbound half is usually already satisfied. If your
account restricts node egress, add it explicitly. A missing outbound rule produces the same symptom as a missing
mount target: workspaces hang while starting, with no indication in Kubernetes.

Outbound rules on the mount target security group are not needed, because security groups are stateful.

> **Note**
>
> The source is always the node security group, including on clusters that use EKS security groups for pods. The
> mount is performed by the EFS CSI node DaemonSet, which runs with `hostNetwork`, and security groups for pods do
> not apply to `hostNetwork` pods. A `SecurityGroupPolicy` never matches the DaemonSet,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> so scoping the inbound rule
> to a pod security group leaves every workspace hanging while starting. If you must use a CIDR, scope it to the
> worker node subnet CIDRs, never the whole VPC. A security group limits reachability, not encryption. Keep `tls`
> on the mount regardless.

Attach the CSI driver policy

One CSI driver, `efs.csi.aws.com`, serves both Amazon EFS and Amazon S3 Files, so this step applies whichever
backend you use. Attach `AmazonEFSCSIDriverPolicy` to the CSI driver role. It must include `DescribeMountTargets`,
which the mount helper uses to resolve the Availability Zone local mount target.

Amazon EFS prerequisites

- **Create the file system first.** The driver cannot create it. **Create New** carves a fresh isolated access
  point on the file system you provisioned. It never creates a file system.
- **Create it as a Regional file system, not a One Zone file system.** A Regional file system allows one mount
  target in each Availability Zone, which is what the per-Availability-Zone requirement above needs. A One Zone
  file system allows a single mount target, so it cannot back a Mount Point. Workspaces scheduled into the
  Availability Zones it does not serve hang while starting, and there is no mount target you can add to fix them.
- **Enable encryption at rest when you create the file system.** Mount Points work either way, so nothing warns
  you later, and encryption cannot be enabled after creation.

Enable automatic backups

EFS has no versioning, no snapshots, and no recycle bin. Without a recovery point that already existed when data
was lost, a deletion is permanent.

A Regional file system, which is the only kind that can back a Mount Point, is created with automatic backups

© 1997–2026 Citrix Systems, Inc. All rights reserved.

disabled when you use the CLI, the API, or Terraform. Only the EFS console enables them for you. Pass
`--backup` at creation, or enable it afterwards:

```
aws efs put-backup-policy --file-system-id <fs-id> --backup-policy
   Status="ENABLED"
aws efs describe-backup-policy --file-system-id <fs-id>
```

This attaches the default AWS Backup plan for EFS: daily backups with 35-day retention. Confirm that AWS Backup is
offered in your Region, and budget for it separately from EFS.

Choose Elastic throughput

The CLI and API default to Bursting throughput, whose baseline scales with stored bytes. A Mount Point file system
starts empty, so it converges on the minimum and caps at 35,000 read and 7,000 write IOPS. Elastic throughput
raises this to approximately 250,000 and 50,000, which is what the many small-file operations in a workspace
actually need.

```
aws efs update-file-system --file-system-id <fs-id> --throughput-mode
   elastic
```

Throughput mode is a file system attribute. It cannot be set on the StorageClass, and SecurSpaces cannot see or warn about
it.

> **Note**
>
> Elastic throughput meters every gigabyte read and written, with writes charged at roughly twice the rate of
> reads. Size the change from your CloudWatch read and write metrics first, and do not switch during business
> hours. Elastic throughput does not reduce per-operation latency, so it is not a fix for slow small-file work.

Amazon S3 Files prerequisites

Amazon S3 Files requires the common prerequisites described earlier —the per-Availability-Zone mount targets, the
scoped NFS rule, and the CSI driver policy —plus the following.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Do not confuse Amazon S3 Files with Mountpoint for Amazon S3 (`s3.csi.aws.com`), which is not POSIX compliant and
> is not a supported Mount Point backend.

Prepare the bucket

The S3 bucket must have **versioning enabled** and **server-side encryption** configured before you create the file
system. Bucket versioning is also the only way to recover a deleted file on S3 Files.

The `--accept-bucket-warning` flag is not an "existing data"switch. S3 Files refuses to create a file system
scoped to a prefix that holds a large number of objects, warning that recursive renames or moves on such a file
system are slow and costly, because every file needs its own copy and delete against the bucket. Pass the flag to
acknowledge the warning and let the creation proceed. It does nothing about versioning or encryption, which must
still be set beforehand.

Create the service role

When you create the file system you pass an IAM role that S3 Files assumes to read and write your bucket and to
manage the EventBridge rules that drive synchronization. The S3 console creates this role for you. From the CLI or
Terraform you must author both the trust policy and the permissions policy.

A role with only a trust policy is accepted at creation time, but leaves synchronization unable to work.

The trust policy lets S3 Files assume the role, through the `elasticfilesystem.amazonaws.com` service principal.
Keep the account and ARN conditions. They stop another AWS account using your role against your bucket:

```
{

  "Version": "2012-10-17",
  "Statement": [{

```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    "Effect": "Allow",
    "Principal": {
  "Service": "elasticfilesystem.amazonaws.com"  }
,
    "Action": "sts:AssumeRole",
    "Condition": {

      "StringEquals": {
  "aws:SourceAccount": "<account-id>"  }
,
      "ArnLike": {
  "aws:SourceArn": "arn:aws:s3files:<region>:<account-id>:file-system/*
     "  }

    }

   }
 ]
 }
```

The permissions policy must grant the following. Scope each statement to the resources it names, and keep the conditions: without them the role holds far more than this one file system needs.

- `s3:ListBucket` and `s3:ListBucketVersions` on the bucket, conditioned on `aws:ResourceAccount` matching your account ID.
- `s3:AbortMultipartUpload`, `s3:DeleteObject*`, `s3:GetObject*`, `s3:List*`, and `s3:PutObject*` on the bucket contents, with the same `aws:ResourceAccount` condition. The wildcard actions are deliberate: S3 Files uses the version-aware S3 operations that bucket versioning enables.
- `kms:GenerateDataKey`, `kms:Encrypt`, `kms:Decrypt`, `kms:ReEncryptFrom`, and `kms:ReEncryptTo` on the ARN of the key that encrypts the bucket, conditioned on `kms:ViaService` matching `s3.<region>.amazonaws.com` and on `kms:EncryptionContext:aws:s3:arn` matching the bucket and its contents. Left without a resource and these conditions, the statement lets the role use every key in the account whose key policy delegates to IAM.
- EventBridge rule management —`events:PutRule`, `events:PutTargets`, `events:EnableRule`, `events:DisableRule`, `events:RemoveTargets`, and `events:DeleteRule` —scoped to `rule/DO-NOT-DELETE-S3-Files*` and conditioned on `events:ManagedBy` matching `elasticfilesystem.amazonaws.com`, plus Event-

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Bridge read actions.

For the full policy documents, see the AWS documentation for working with Amazon S3 Files.

> **Warning**
>
> If the bucket uses SSE-KMS, the role cannot read or write a single object without the `kms:` actions. Reads fail
> and writes never reach the bucket, both silently. Keep the KMS statement even on an SSE-S3 bucket, so the role
> still works if the bucket is switched to SSE-KMS later. If the key is a customer managed key, its key policy must
> also allow this role. An IAM policy alone is not enough.

The EventBridge rules named `DO-NOT-DELETE-S3-Files...` are how changes made directly in the bucket reach mounted
workspaces. Do not delete or disable them. Without them, cached files keep serving stale content.

Create the file system

```
aws s3files create-file-system --bucket <bucket-arn> [--prefix p/] --
    role-arn <service-role-arn>
```

The file system encrypts its own high-performance tier at rest by default. You can supply a customer managed key
for that storage instead. This is a different key from the one that encrypts the bucket. If you supply one, it must
be in the same Region as the file system, and its key policy must allow S3 Files `kms:Decrypt`, `kms:GenerateDataKeyWithoutPlaintext`, `kms:CreateGrant`, and `kms:DescribeKey`. The default key policy already
does.

> **Warning**
>
> Disabling or deleting that key makes the file system, and every Mount Point on it, inaccessible. Disabling
> takes effect after a short delay and is undone by re-enabling the key; deleting it is permanent. Treat the key
> as part of the file system rather than as a key you can retire during routine key hygiene.

Check versions and platform support

- **In the cluster** —the Amazon EFS CSI driver must be version 3.0.0 or later.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **On any EC2 host you mount from directly** —the `amazon-efs-utils` client must be version 3.0.0 or later.
- **AWS CLI** —new enough to include the `s3files` commands. Verify with `aws s3files help`.
- **In your Region** —confirm that Amazon S3 Files is offered in the Region your cluster runs in. It became
  generally available in April 2026 in 34 Regions, and coverage grows, so check the AWS regional services table
  for the current list.

Attach `AmazonS3FilesCSIDriverPolicy` and `AmazonS3FilesClientFullAccess` to the controller identity, and
`AmazonS3FilesClientFullAccess`, `AmazonElasticFileSystemsUtils`, and S3 read on the bucket to the node identity.
See Scope IAM permissions
before you attach the account-wide managed policies.

Amazon S3 Files does not support cross-account static attach: the file system and bucket must be in the same AWS
account as the cluster. It is not supported on EKS Fargate, EKS Hybrid Nodes, or Windows containers.

Scope IAM permissions

The EFS CSI driver authenticates through IAM roles for service accounts (IRSA).

> **Warning**
>
> Most AWS walkthroughs attach the node policies to the EC2 instance role of the managed node group. An instance
> role grant is not private to the CSI driver: any workload on that node that can reach the instance metadata
> service can use it. Because workspaces run customer-supplied code, treat every policy on the node role as one you
> have granted to your developers.

Give the node driver its own role bound to its service account rather than using the instance role:

1. Enable the cluster OIDC provider.
2. Create a role whose trust policy pins both the service account and the STS audience.
3. Annotate `efs-csi-node-sa` with `eks.amazonaws.com/role-arn`. Do the same for `efs-csi-controller-sa`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Use the exact `system:serviceaccount:<namespace>:<name>` value. A wildcard subject, or a missing audience
condition, lets any pod in the cluster assume the role.

Scope client access to your file systems

All AWS managed S3 Files client policies use `"Resource": "*"` and cannot be limited to a file system.
`AmazonS3FilesClientFullAccess` goes on both the controller identity and the node identity, so anything holding
either one is authorized to mount every S3 Files file system in the account and to write to them as root.

Replace that managed policy on both identities with a customer managed policy that names your file systems and
grants `s3files:ClientMount` and `s3files:ClientWrite` on them. Add `s3files:ClientRootAccess` only where a mount
genuinely needs it.

Similarly, give the node identity S3 read scoped to the buckets that back Mount Points rather than an account-wide
read-only policy such as `AmazonS3ReadOnlyAccess`. An inline policy granting `s3:GetObject`, `s3:GetObjectVersion`,
and `s3:ListBucket` on those buckets is enough.

> **Note**
>
> A file system policy that omits a role does not revoke an account-wide identity grant. An allow in either the
> identity policy or the file system policy is sufficient. Either scope the identity policy, or add an explicit
> deny.

Harden instance metadata

Require IMDSv2 and set the metadata hop limit to 1 on the node group. Apply the steps in this order. Applied out of
order, this silently breaks Mount Point provisioning.

1. **Confirm the driver uses its own role.** Role credentials are injected when a pod is created, so annotating
   the service accounts is not enough. Until the pods restart, the driver falls back to the node role

© 1997–2026 Citrix Systems, Inc. All rights reserved.

and appears
to work. Restart the driver, confirm every pod reports a role ARN, and create one Mount Point successfully.

2. **Change the launch template**, so replacement nodes are created hardened. Set `HttpTokens : required`, `HttpPutResponseHopLimit: 1`, and `HttpEndpoint: enabled`. Editing the template is not enough on its own: a
managed node group stays pinned to the launch-template version it was created with, so update the node group
to the new version as well. Otherwise every scale-out and every replaced node comes back on the pinned
version, and the hardening quietly decays on a cluster whose workspace nodes come and go.

3. **Roll the change to running nodes** one at a time, watching the `kube-system` namespace for about 15 minutes
after the first before continuing:

```
aws ec2 modify-instance-metadata-options --instance-id <id> \
  --http-tokens required --http-put-response-hop-limit 1 --http-
     endpoint enabled
```

To roll a node back, set `--http-put-response-hop-limit 2` on that instance.

4. **Check what the node group will hand out.** Read the metadata options back from the launch template, and
confirm that the node group points at the version you checked:

```
aws ec2 describe-launch-template-versions --launch-template-id <id
   > --versions '$Latest' \
  --query 'LaunchTemplateVersions[0].LaunchTemplateData.
     MetadataOptions'
```

`"HttpPutResponseHopLimit": 2` means IMDSv2 is enforced but pod access to the metadata service is not blocked.

5. **Verify from an ordinary pod, not from a workspace terminal.** Every workspace pod already drops traffic to
`169.254.169.254` inside its own network namespace, so a probe run in a workspace terminal returns nothing
whether or not the node was ever hardened. Run it from a pod that does not carry that block, and expect a
timeout with no token returned:

```
kubectl run imds-probe --rm -it --restart=Never --image=curlimages
   /curl -- \
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  sh -c 'curl -s -m2 -X PUT http://169.254.169.254/latest/api/
     token \
    -H "X-aws-ec2-metadata-token-ttl-seconds: 60"'
```

Then run the same request on the host as a positive control. It must still succeed. If it does not, you have
broken host agents rather than protected pods.

Metadata options are fixed when an instance launches. Changing the launch template does nothing to running nodes,
and changing a running node does not survive its replacement.

> **Note**
>
> With `eksctl`, use `disablePodIMDS: true` on the node group. The related `disableIMDSv1` key already defaults to
> `true`, so setting it alone changes nothing, and `eksctl` writes a hop limit of 2 without `disablePodIMDS`. That
> leaves every pod able to borrow the node role. `disablePodIMDS` cannot be combined with `iam.withAddonPolicies`, and is refused when a managed node group supplies its own launch template.

A hop limit of 1 stops ordinary pods from reaching the metadata service. Processes on the host and pods using
`hostNetwork` are unaffected, including the CSI node DaemonSet. The CSI controller runs on the pod network and is
affected: if it loses metadata access, new Mount Points fail while existing mounts keep working.

Next

- Configure SecurSpaces

Related information

- Author a PersistentVolume
- Configure SecurSpaces
- Troubleshoot AWS Mount Points
- AWS mount points

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Configure SecurSpaces for AWS Mount Points

September 29, 2026

Complete the AWS preparation first. The steps here depend on a file system, its mount targets, and the IAM roles already existing.

Create the StorageClass

You author the StorageClass by hand, one per backing file system. SecurSpaces ships none and creates none. A missing
StorageClass is the most common cause of an empty create dialog.

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sn-efs-mountpoint
provisioner: efs.csi.aws.com
mountOptions:
  - tls
parameters:
  provisioningMode: efs-ap
  fileSystemId: fs-0abc123
  basePath: "/dyn"
  directoryPerms: "700"
  gidRangeStart: "50000"
  gidRangeEnd: "51000"
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Required values:

- `provisioner: efs.csi.aws.com` for both backends.
- `parameters.provisioningMode` —`efs-ap` for Amazon EFS, `s3files-ap` for Amazon S3 Files. This parameter, not
  any option in the interface, selects the backend. Users choose the backend by choosing the StorageClass, so give
  each class a clear name.
- `parameters.fileSystemId` —the id of the file system you created. It must be present, and it must match
  exactly. SecurSpaces also uses these ids to tell the file systems it shares from external ones, so a blank or
  mistyped value costs more than the one class. Check it before anyone uses the class.
- `mountOptions: [tls]` —a top-level list, a sibling of `parameters`, not a key inside it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `volumeBindingMode: Immediate` —EFS and S3 Files are zoneless. A class set to `WaitForFirstConsumer` can never bind and the create is rejected.

Optional values include `basePath`, `subPathPattern`, `directoryPerms`, and the pair `gidRangeStart` and `gidRangeEnd`, which bound the POSIX group id the access point enforces. Set both or neither. Keep a custom range wide enough for every Mount Point planned on that file system, because each access point consumes one group id.

One file system and one StorageClass serve many Mount Points. Add more only when you exceed the access point ceiling, need throughput isolation, or need hard tenant isolation.

Check the file system id

Nothing validates `fileSystemId` against AWS when you author the class. SecurSpaces accepts whatever you write and acts on it only when a user creates or attaches a Mount Point. A blank value and a wrong value then fail in different ways.

A blank value fails closed, and the effect is not limited to the class that carries it. One Mount Point StorageClass with no `fileSystemId` makes SecurSpaces treat its whole picture of which file systems it shares as unreliable, so it refuses rather than guess: the attach mode in which a user types raw coordinates is denied for every class on the cluster, including attaches onto file systems unrelated to the incomplete one.

A wrong value does not immediately switch the protection off. SecurSpaces recognizes the file systems it shares from two independent signals: the ids declared on your Mount Point StorageClasses, and the file systems it has already provisioned Mount Points on in that region. A typo alone does not un-share the real file system. The risk is the window in which the second signal is empty —a file system SecurSpaces has never provisioned on, or one where every Mount Point it created there failed its post-bind identity check. In that window a wrong id makes the real shared file system look external, and a pre-built volume covering the whole of it becomes attachable by an ordinary user.

Read the id back from AWS before anyone uses the class:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
aws efs describe-file-systems --file-system-id <fs-id> --query '
    FileSystems[0].FileSystemId'
```

Encryption in transit

EFS and S3 Files are mounted over NFS on port 2049, and the CSI driver only encrypts that traffic when the mount
carries the `tls` option. Set it on the StorageClass and on every PersistentVolume you author. Without it, every
byte read or written travels unencrypted.

SecurSpaces keys strictly on the literal `tls` mount option. It does not accept `parameters.encryptInTransit`, a
TLS-enabled access point, or an EFS file system policy that enforces `SecureTransport` as substitutes.

How strictly this is enforced differs by mode:

- **Attach Existing fails closed.** At bind time SecurSpaces reads the volume you named and refuses the attach unless it
  is an `efs.csi.aws.com` volume whose mount options contain `tls`.
- **Create New is best-effort.** A StorageClass without `tls` is normally hidden from the **Create New** list, and
  naming it anyway is rejected. All of these checks depend on reading the StorageClass from the Kubernetes API.

> **Warning**
>
> The StorageClass read is made by the workspace service. If it cannot read StorageClasses, every Create New
> encryption check is skipped, a StorageClass without `tls` becomes usable, and the mount runs unencrypted with no
> error. This is the one failure in this article that fails in the permissive direction.

Cluster permissions

StorageClasses and PersistentVolumes are both cluster-scoped, so the grant that lets the workspace service read
them has to be a ClusterRole. The chart installs it as a ClusterRole and a ClusterRoleBinding, both named
`<release>-workspace-api-clusterwide`, bound to the workspace service account `<release>-workspace-api`. For

© 1997–2026 Citrix Systems, Inc. All rights reserved.

everything else the platform reaches on the cluster, see Trust boundaries.

Mount Points depend on these read permissions:

Resource	Verbs	What depends on it
`storageclasses`	`get`, `list`	Listing the classes offered in **Create New**, and the encryption in transit check
`persistentvolumes`	`get`, `list`	Reading back the volume a new Mount Point was given, and listing the volumes offered by **Attach Existing**

Two Helm settings take the grant away. Setting `platform.serviceKubernetesConfig.serviceAccounts.wsAPIServiceAccount` to your own service account means the chart installs none of the workspace service RBAC objects, so you must create and bind the equivalent permissions yourself. Setting `platform.serviceKubernetesConfig.rbac.createWorkspaceClusterRoles` to **`false`** removes the whole ClusterRole and its binding: not only the StorageClass read, but the PersistentVolume read, the PersistentVolume write verbs that `allowPersistentVolumeAccess` would otherwise add, and read access to nodes and node metrics. The namespace-scoped Role the chart installs alongside it is not affected by that setting.

Verify each verb separately. Kubernetes `list` does not imply `get`:

```
kubectl auth can-i get storageclasses --as=system:serviceaccount:<namespace>:<release>-workspace-api
kubectl auth can-i list storageclasses --as=system:serviceaccount:<namespace>:<release>-workspace-api
kubectl auth can-i get persistentvolumes --as=system:serviceaccount:<namespace>:<release>-workspace-api
kubectl auth can-i list persistentvolumes --as=system:serviceaccount:<namespace>:<release>-workspace-api
```

`--as` impersonates another identity, which needs cluster-admin or an `impersonate` grant of your own. On a default install you can also confirm that the objects carrying the grant are present:

```
kubectl get clusterrole,clusterrolebinding <release>-workspace-api-clusterwide
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If you supplied `wsAPIServiceAccount`, neither of those objects exists and the chart never creates `<release>-workspace-api`, so the commands above report `no` for an account nothing runs under. Substitute the
account you supplied, which you can read from the running pod:

```
kubectl -n <namespace> get pod -l app=<release>-workspace-api -o
   jsonpath='{
 .items[0].spec.serviceAccountName }
 '
```

What each missing permission costs:

- Missing `get` on `storageclasses` skips the encryption checks and allows an unencrypted mount.
- Missing `list` on `storageclasses` breaks discovery entirely: the create list is empty and the feature
  disappears.
- Missing `get` on `persistentvolumes` lets every **Create New** Mount Point reach ready without its volume being
  read back, so the Mount Point holds no record of the volume it was provisioned on.
- Missing `list` on `persistentvolumes` breaks the **Attach Existing** volume picker, which shows **Failed to
  load available volumes** and a **Retry** link. That is a different state from **No volumes available to
  attach**, which means the read succeeded and no volume matched the scope label.

Enable the feature

Mount Point Storage is enabled at **System Configuration > Integrations > Mount Point Storage**. System
Configuration is an administrator-only area. A security officer who is not also an administrator cannot enable it.

- **The cloud provider is detected, not chosen.** SecurSpaces derives whether Mount Points are backed by AWS or Azure from
  the CSI provisioners of the StorageClasses on the cluster. To expose the AWS options, create AWS Mount Point
  StorageClasses first.
- **Turn on the file storage switch.** While it is off, the create flow is hidden from users.
- **Use one cloud per cluster.** If both an AWS and an Azure Mount Point StorageClass are present, the provider
  cannot be resolved, the switches are forced off, and the feature is withdrawn. The page explains

© 1997–2026 Citrix Systems, Inc. All rights reserved.

this and lists
the providers it detected. Remove one cloud's classes and revisit the page. No restart is needed, but allow
about 30 seconds for the configuration cache.

- **`allowPersistentVolumeAccess` governs one attach mode.** The mode in which a user types raw coordinates and
  SecurSpaces builds the volume requires `platform.serviceKubernetesConfig.rbac.allowPersistentVolumeAccess`, which
  defaults to **`false`**. While it is off, that mode is hidden, but attaching a volume you authored and creating new
  Mount Points both still work. The change takes effect when `helm upgrade` completes, with no pod restart.

> **Note**
>
> Only this page explains why Mount Points are unavailable. In the project and workspace views the option is simply
> hidden. Triage any report of missing Mount Points by opening this page first.

Next

- Author a PersistentVolume, if you publish your own volumes
- Troubleshooting

Related information

- Author a PersistentVolume
- Prepare the AWS resources
- Troubleshoot AWS Mount Points
- AWS mount points
- Trust boundaries

Author a PersistentVolume for Attach Existing

September 29, 2026

SecurSpaces never creates, changes, or cleans up these volumes. It only creates a claim that binds to your volume by name.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the scope label

SecurSpaces records the owner of storage it creates. For a volume you author, it has no such record, so you must declare
the intended consumer scope with the `strong.network/mount-point-scope` label.

Label value	Who can attach it
`project.<projectId>`	Only that project.
`org.<organizationId>`	Any project in that organization.
`shared`	Any project in the whole deployment.

Use a dot, not a colon. Kubernetes forbids `:` in a label value.

A volume with no scope label, or an unrecognized value, is offered to nobody and cannot be attached. A forgotten
label can never accidentally expose a volume, but it is the most common cause of "my volume is not in the list".

Both `project.` and `org.` scopes are enforced by server-side checks, on both the picker and the bind. The `shared`
value has no boundary. Use it only for data you are content for anyone in the deployment to reach.

Find the numeric project or organization id in the browser address bar. The id is the path segment after
`/project/` or `/organization/`.

Meet the volume requirements

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: shared-dataset-alpha
  labels:
    strong.network/mount-point-scope: "project.1090137447483732"
spec:
  capacity:
    storage: 1Gi
  accessModes: ["ReadWriteMany"]
  persistentVolumeReclaimPolicy: Retain
  storageClassName: ""
  mountOptions: ["tls"]
  csi:
    driver: efs.csi.aws.com
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
16      volumeHandle: fs-0abc123::fsap-07c9def
17      # volumeHandle: s3files:fs-0def456::fsap-001d    # Amazon S3 Files
           keeps the prefix
```

SecurSpaces checks the following before it lets a volume be attached:

- `spec.csi.driver` is `efs.csi.aws.com`, for both backends.
- `spec.mountOptions` contains `tls`.
- `spec.accessModes` permits the permission the owner chose, and its first entry is one the storage driver
  accepts. See Set the access modes.
- `spec.capacity.storage` has a value, and is not smaller than the size the owner requests. Beyond that check
  the value is nominal: EFS and S3 Files ignore size and never enforce it.
- The volume is in the `Available` phase.

Write `spec.csi.volumeHandle` in the static form for your backend. For Amazon EFS it has no prefix,
`<fs-id>::<access-point-id>`; the `efs:` prefix belongs to dynamic provisioning only. For Amazon S3 Files it keeps
the prefix, `s3files:<fs-id>::<access-point-id>`. SecurSpaces does not require the handle to name an access point,
so the handle you publish is what the project gets. See
Publish access point handles only.

List `tls` in `spec.mountOptions` and nothing else. In particular, do not add `ro` there to make the data
read-only. A volume's mount options apply to every mount of it, so `ro` pins every Mount Point built on that
volume to read-only, including ones an owner created as read/write, which reads to them as data loss. Read-only
is a per-Mount-Point choice, applied at the workspace pod.

SecurSpaces reads neither of the next two fields, but Kubernetes does. Set both yourself. Each fails in a
different way, and neither failure is explained anywhere in the interface:

- `spec.persistentVolumeReclaimPolicy: Retain`. A volume set to `Delete` is offered and binds normally. When
  the Mount Point is deleted and the claim goes, the volume is left **Released**: it is not deleted and its data
  is untouched, but it cannot back another Mount Point until you
  reset it. Set `Retain` anyway. If the volume ever carries the EFS CSI driver's

© 1997–2026 Citrix Systems, Inc. All rights reserved.

`pv.kubernetes.io/provisioned-by` annotation, `Delete` hands it to the driver, which deletes the access point
the volume names.

- `spec.storageClassName: ""`, an explicit empty string. Kubernetes binds a claim to a volume only when the two
`storageClassName` values match, and the claim SecurSpaces builds always carries `""`. A volume naming a
StorageClass is offered and accepted and then never binds: the attach times out after three minutes and the
Mount Point lands in an error state. Do not omit the field either. An absent value lets the default StorageClass admission controller turn the static bind into dynamic provisioning.

No Kubernetes secret is needed. AWS access is backed by IAM and the access point.

Set the access modes

Use `accessModes: ["ReadWriteMany"]`. That single value is the right answer for every AWS Mount Point volume,
whether owners will attach it read/write or read-only.

It does not mean the volume must be writable. The read/write choice belongs to the Mount Point, not to the
volume. An owner who picks **Read Only** gets a genuinely read-only mount, because the workspace mounts the
volume read-only and the storage driver applies `ro` at mount time. The access mode plays no part in it.

> **Note**
>
> A read/write Mount Point needs `ReadWriteMany`. A read-only one accepts either `ReadWriteMany` or
> `ReadOnlyMany`. A volume listing `ReadWriteMany` therefore leaves both choices open to the owner. Where the
> volume's modes allow only one of the two, the dialog fixes the selection on **Read Only** and explains why. It
> never forces the selection to **Read/Write**.

Put `ReadWriteMany` first. Kubernetes passes only the first entry of `accessModes` to the storage driver, and the
AWS driver does not accept `ReadOnlyMany`. A volume whose first mode is `ReadOnlyMany` cannot be mounted by any
workspace: the picker lists it as non-selectable and tags it **cannot be mounted**, and an attempt to

© 1997–2026 Citrix Systems, Inc. All rights reserved.

attach it
is refused with **Selected volume cannot be mounted**. Listing `ReadOnlyMany` after `ReadWriteMany` is harmless
but pointless, because nothing reads it. A volume that lists only `ReadWriteOnce` is not offered at all, because
a Mount Point is shared across many workspace pods and needs many-node access.

`spec.accessModes` is not immutable on a PersistentVolume, so a volume authored the wrong way is corrected in
place:

```
kubectl patch pv <name> --type=merge -p '{
 "spec":{
 "accessModes":["ReadWriteMany"] }
  }
 '
```

The patch is accepted on a `Bound` volume, does not disturb the binding, and leaves the data untouched. Restart
any workspace that was already stuck on the volume, so that it gets a fresh pod and the driver is asked again.
The bound claim keeps its own original access modes, which is expected and harmless: a claim's spec really is
immutable, only the volume's first mode reaches the driver, and the claim's status follows the patch. If the
volume is reconciled from a manifest by GitOps, or guarded by an admission webhook, change it at the source
instead, or the next sync reverts the patch.

To find volumes already authored the wrong way:

```
kubectl get pv -o json | jq -r '
  .items[] | select(.spec.csi.driver=="efs.csi.aws.com")
  | select(.spec.accessModes[0]=="ReadOnlyMany")
  | .metadata.name'
```

> **Warning**
>
> Do not delete and re-create a volume to change its access modes. While a claim is still bound to it,
> `kubectl delete pv` only marks the volume `Terminating`, because Kubernetes holds the `kubernetes.io/pv-protection` finalizer until the volume leaves the `Bound` phase, so the corrected volume
> cannot be applied under the same name. Never force-remove that finalizer on a bound volume: doing so strips a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> live Mount Point's binding.

In no shape does `accessModes` publish a dataset read-only. Access modes are a hint used to match a claim to a
volume, and they enforce nothing at mount time. To publish data that a consumer genuinely cannot write, enforce
it in the storage layer: give the access point a read-only POSIX identity, or deny writes with a file system policy. That holds whichever permission the Mount Point's owner picks.

Publish access point handles only

The volume handle has three colon-separated fields, `FileSystemId:Subpath:AccessPointId`. Leave a field empty to
omit it, which is why a handle with no subpath carries two colons. Only the access point field enforces isolation.

Handle	What it mounts	Isolation
`fs-id`	The file system root.	None. The workspace sees every project's data.
`fs-id:/subdir`	A subdirectory.	None. Path-confined only, treated exactly like a bare file system id.
`fs-id::fsap-id`	An access point.	Full, but only if that access point is itself restricted.
`fs-id:/subdir:fsap-id`	A subdirectory under the access point root.	The same condition. The isolation comes from the access point, never from the subpath.

To combine a subpath with an access point, put a single colon between them. `fs-id:/subdir::fsap-id` is not that
handle. The extra colon pushes the access point id out of the third field, so SecurSpaces reads no access point id
and treats the volume as a whole file system attach.

Naming an access point is not a condition of being offered. A bare file system handle is offered and binds like
any other volume. SecurSpaces withholds one only from a user who is neither an administrator nor a security

© 1997–2026 Citrix Systems, Inc. All rights reserved.

officer, and only when the file system is one it provisions Mount Points from, or when it cannot read that set of
file systems. On any other file system, publishing a bare handle is your decision alone.

> **Warning**
>
> On the route where you publish a volume, the handle you publish is the security boundary. A bare file system
> handle mounts and works, there is no second, cluster-side check when a volume you pre-built is bound, and
> nothing warns the user. Any user who can create a Mount Point in a project your volume is scoped to can bind
> it. Publish access point handles only, unless you intend everyone in that scope to see the whole file system.

An access point id is not by itself isolation. SecurSpaces decides isolation purely from whether an access point id is
present. It never reads the access point's configuration from AWS. In AWS both the root directory and the POSIX
user are optional on an access point, and one created without them exposes the file system root while still
satisfying every check. Verify before publishing:

```
aws efs describe-access-points --access-point-id <fsap-id> \
  --query 'AccessPoints[].{
 Root:RootDirectory.Path,Uid:PosixUser.Uid,Gid:PosixUser.Gid }
 '
```

Publish only if `Root` is a real per-dataset path, not `/` and not **`null`**, and `Uid` and `Gid` are set. Access points cannot be edited after creation, so an unrestricted one must be replaced.

To sweep a file system for unrestricted access points:

```
aws efs describe-access-points --file-system-id <fs-id> \
  --query 'AccessPoints[?RootDirectory.Path==`/` || PosixUser==`null`].
     AccessPointId'
```

Never publish a subpath handle that points at your StorageClass `basePath`. That directory is the parent of every
directory created by **Create New**, including data orphaned by deleted Mount Points.

Reset a released volume

A volume you author is single-use per bind cycle. Deleting a Mount Point removes only the claim. Because of

© 1997–2026 Citrix Systems, Inc. All rights reserved.

`Retain`, Kubernetes moves the volume to the `Released` phase and keeps a stale `claimRef` that prevents rebinding.
The picker still lists it, in a separate non-selectable group marked as needing a reset.

Confirm the phase first, then dry-run the patch against the API server, then apply it:

```
kubectl get pv <pv-name> -o jsonpath='{
 .status.phase }
 '     # must read: Released
kubectl patch pv <pv-name> --type=json \
  -p='[{
 "op":"remove","path":"/spec/claimRef" }
 ]' --dry-run=server
kubectl patch pv <pv-name> --type=json -p='[{
 "op":"remove","path":"/spec/claimRef" }
 ]'
```

> **Warning**
>
> Clearing `claimRef` returns the volume to the pool with its existing data intact, and the next project that
> attaches it inherits that data. Reset only volumes you intend to hand back. The command has no safety check of
> its own: run against a bound volume it strips a live Mount Point's binding. Confirm the phase reads `Released`
> first, and use the server-side dry run to catch a mistyped volume name before the live patch changes anything.

SecurSpaces does not reset the volume for you. This flow exists for operators who do not grant SecurSpaces write access to
PersistentVolumes, so SecurSpaces has read-only access and never changes a volume you own.

Related information

- Configure SecurSpaces
- Mount point limits and behavior

Troubleshoot AWS Mount Points

September 29, 2026

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Symptom	Likely cause	Fix
A StorageClass exists but never appears in the **Create New** list.	It is an AWS class with no top-level `mountOptions: [tls]`, or it has the wrong `provisioningMode`.	Add `tls` to the class. Also check `parameters.provisioningMode`: it must be `efs-ap` for Amazon EFS or `s3files-ap` for Amazon S3 Files. A class with any other value is not recognized as a Mount Point class at all, so adding `tls` on its own does not bring it back.
Error citing missing TLS.	The StorageClass lacks `mountOptions: [tls]`.	Add `tls` to the class.
A workspace hangs while starting, although the Mount Point is ready.	No mount target in the workspace's Availability Zone, or port 2049 is blocked.	Add the mount target, or open 2049. If the file system is One Zone, no mount target can be added for a second Availability Zone: re-create it as Regional.
A Mount Point created from typed coordinates becomes ready, then workspaces hang.	A wrong or nonexistent file system or access point id. Coordinates are not validated against AWS.	Verify the ids in AWS and re-create the Mount Point.
"Bindable PV not found".	The volume is out of the caller's scope, is not `Available`, or the name is wrong.	Fix the scope label, or clear `claimRef`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Symptom	Likely cause	Fix
Attach Existing is refused with "Selected volume cannot be mounted".	The volume lists `ReadOnlyMany` first in `spec.accessModes`. Only the first entry reaches the AWS storage driver, which does not accept it, so no workspace could mount the volume.	Only an administrator can fix this. Put `ReadWriteMany` first: `kubectl patch pv <pv-name> --type=merge -p '{ "spec":{ "accessModes":["ReadWriteMany"] } } '`. The volume is changed in place and its data is not affected. If the volume is reconciled by GitOps or guarded by an admission webhook, change it at the source instead, or the next sync reverts the patch.
Attach Existing is refused with "Selected volume does not permit this access".	The volume's `spec.accessModes` do not include the mode being asked for. A read/write Mount Point needs `ReadWriteMany`; a read-only one needs `ReadWriteMany` or `ReadOnlyMany`.	Only an administrator can fix this. Add the missing mode with the same in-place patch, listing every mode the volume should support. The data is not affected.
Every **Attach Existing** from typed coordinates is refused with "Invalid storage class for mount point", although that flow has no storage class to choose. Volumes you authored with a bare file system handle also disappear from the picker.	A Mount Point StorageClass on the cluster has an empty `fileSystemId`. SecurSpaces can no longer tell which file systems it shares, so it refuses rather than guess.	Set `parameters.fileSystemId` on every Mount Point StorageClass.
The create dialog is missing entirely.	The provider is not AWS, StorageClasses are mixed, the storage switch is off, or the user's role cannot manage Mount Points.	Check the settings page.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Symptom	Likely cause	Fix
The settings page reports no eligible storage class.	No Mount Point StorageClass exists on the cluster, or the AWS classes that do exist have a missing or wrong `parameters.provisioningMode`, which stops them being counted as eligible.	Author one. If a class is already there, set `parameters.provisioningMode` to `efs-ap` or `s3files-ap` rather than authoring a second class.

> **Note**
>
> **Ready** means the volume is bound. It does not mean the volume has been mounted. Typed coordinates are never checked against AWS, so a wrong file system id produces a Mount Point that reports ready and fails only when a workspace starts.

Provisioning is subject to two fixed timeouts that cannot be changed: three minutes for the volume to bind, and five minutes for the overall provisioning call. Exceeding either moves the Mount Point to an error state, from which it can be deleted and re-created.

A Mount Point left mid-operation by a restart of the control plane recovers without any out-of-band work. Wait until more than five minutes have passed, then delete it normally.

> **Warning**
>
> Do not delete the claim, volume, or AWS access point by hand first. For a **Create New** Mount Point that destroys data which the normal delete would have reclaimed in the correct order.

Errors shown to users are sanitized, but the row is not blank. A Mount Point in error carries a short reason, shown against its state in the **Status** column on the **Resources > Mount Points** page: hover over or focus the info icon to read it. Start there. The raw AWS and Kubernetes text behind it is only in the central and workspace service logs. Enable `platform.isDeploymentDebugMode` only in test environments.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Key rotation is rejected on AWS with the message that key rotation is not supported for AWS Mount Points. AWS Mount
Points have no key: access is through IAM and the access point. Administrators migrating from Azure should not look
for a key field.

Related information

- Prepare the AWS resources
- Configure SecurSpaces
- Mount point limits and behavior

Platform settings

These pages cover the settings that apply across the whole platform.

Page	What it covers
General	Configure general platform settings.
Workspace policy	Workspace security, network, scheduling, specification, and registry settings, and how they inherit across platform, organization, and project.
Workspace lifecycle and idle detection	How the platform decides a workspace is idle before pausing it, and how to bypass the schedule.
Regions	Set up regions and warm nodes so workspaces run closer to your users.
Custom actions	Automate routine maintenance by defining rules that act on workspaces or users on a schedule.
Project labels	Define a shared taxonomy from a CSV file that everyone applies when creating or editing a project.
Onboarding emails	The automatic emails that bring new users onto the platform, and how to troubleshoot them.
Email gateway	Configure the email gateway, data breach contact, SSH server key, and related platform settings.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Page	What it covers
Information security policy	Publish your organization's security policy to users, and set how long their acceptance lasts.
Terms of service agreements	Publish documents that users must accept before they can use the platform, and reset acceptances.

General settings

September 21, 2026

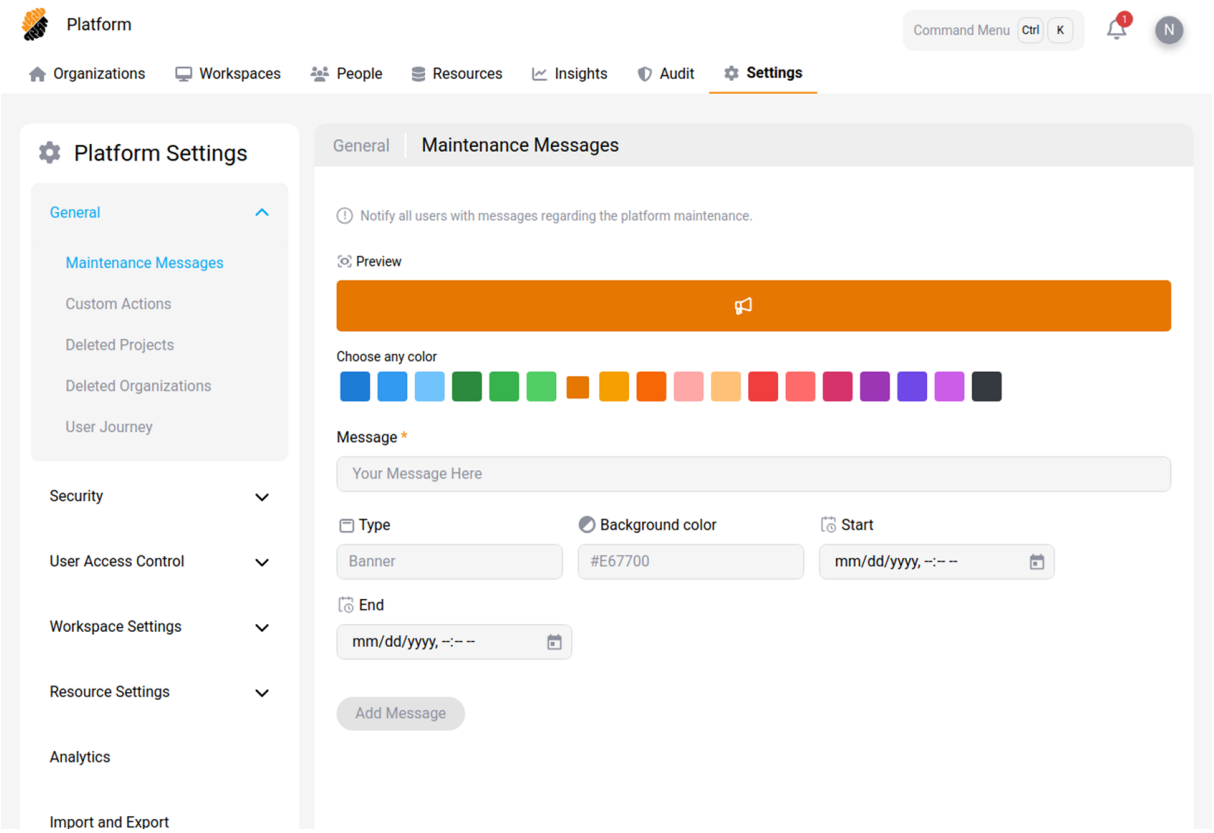
This section covers fundamental platform-wide configurations. Administrators can manage **Maintenance Messages**, configure **Custom Actions**, recover **Deleted Projects** and **Deleted Organizations**, set a **Custom Documentation URL**, and control the **Onboarding & Guidance** experiences.

- Maintenance Messages
- Custom Actions
- Deleted Projects
- Deleted Organizations
- Documentation
- OpenVSX Marketplace
- Onboarding & Guidance

Maintenance Messages

You can configure and display maintenance messages to users. These messages can inform users of scheduled downtime, ongoing maintenance, or other important platform-wide notifications.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Custom Actions

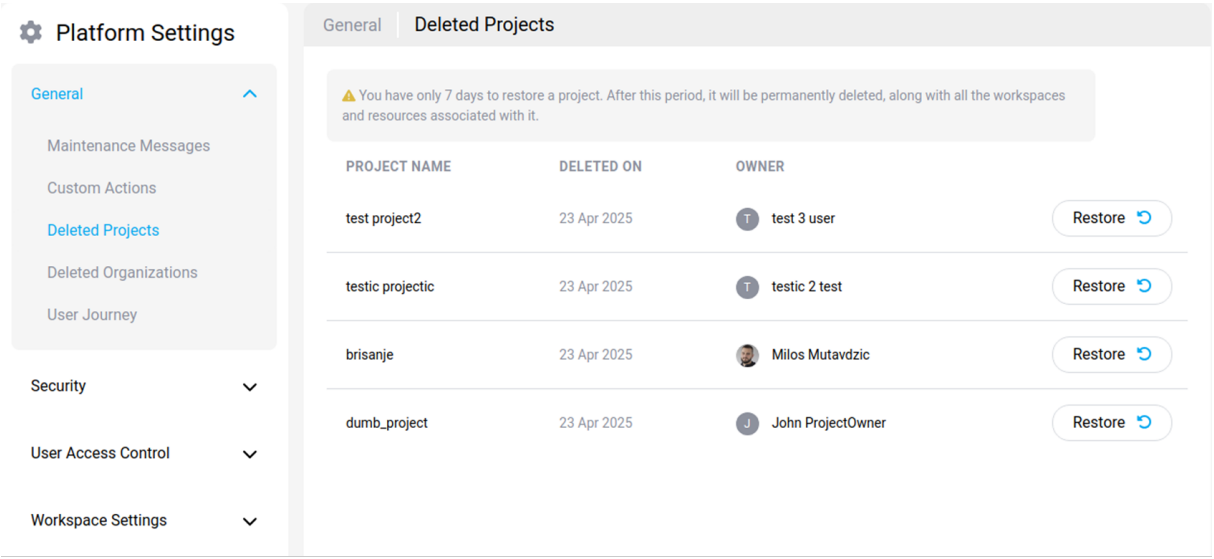
Configure custom actions that automate routine maintenance, such as deleting workspaces that have been inactive for a long time. Each action combines what runs, the conditions that decide which workspaces or users it applies to, and when it runs.

For the available actions, conditions, and scheduling options —including when to use **When Condition Met** —see Custom Actions.

Deleted Projects

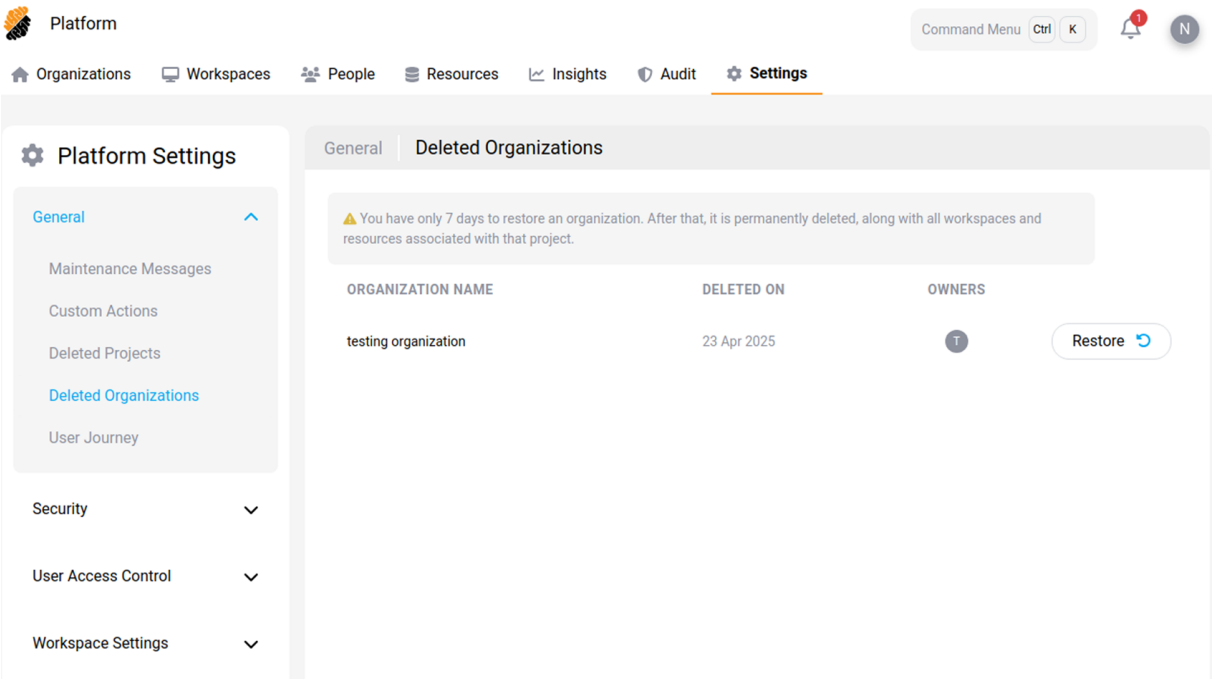
You can recover a deleted project for a period of 7 days on the **Deleted Projects** tab. Press the **Recover** button to the right of the project you want to restore.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Deleted Organizations

You can recover a deleted organization for a period of 7 days on the **Deleted Organizations** tab. Press the **Recover** button to the right of the organization you want to restore.



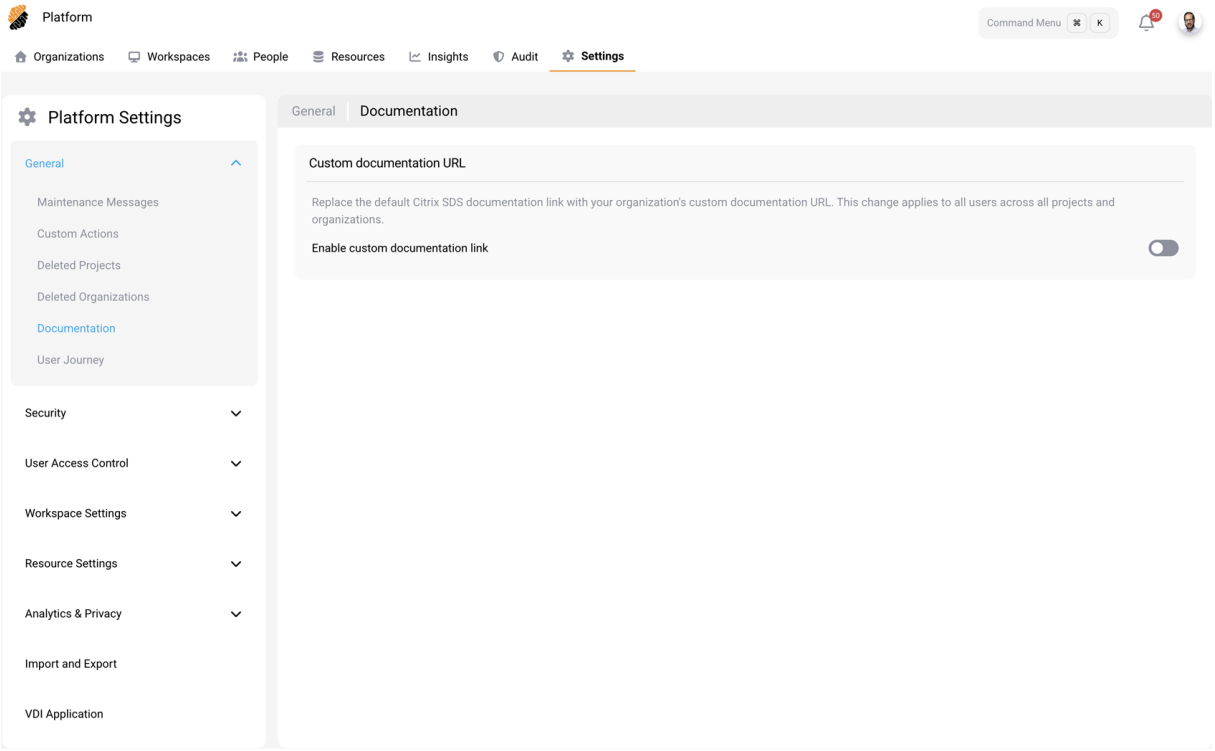
Documentation

By default, the platform links to the standard Citrix SecurSpaces™ documentation. You can replace this link

© 1997–2026 Citrix Systems, Inc. All rights reserved.

with a custom documentation URL that points to your organization's own documentation site.

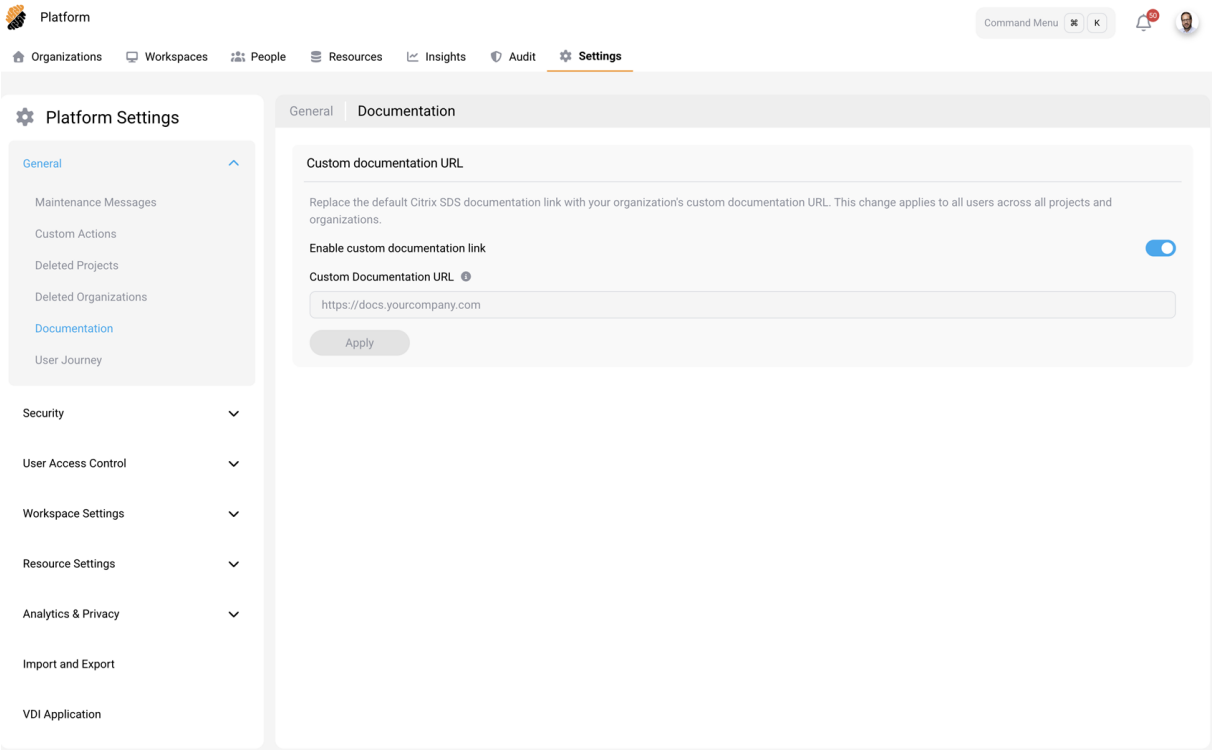
When enabled, this change applies to all users across all projects and organizations. When you turn the setting off, the link reverts to the default Citrix SecurSpaces documentation.



To set a custom documentation URL:

1. Go to **Settings > General > Documentation**.
2. Turn on **Enable custom documentation link**.
3. In the **Custom Documentation URL** field, enter the full web address of your documentation site, including `https://`.
4. Click **Apply**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



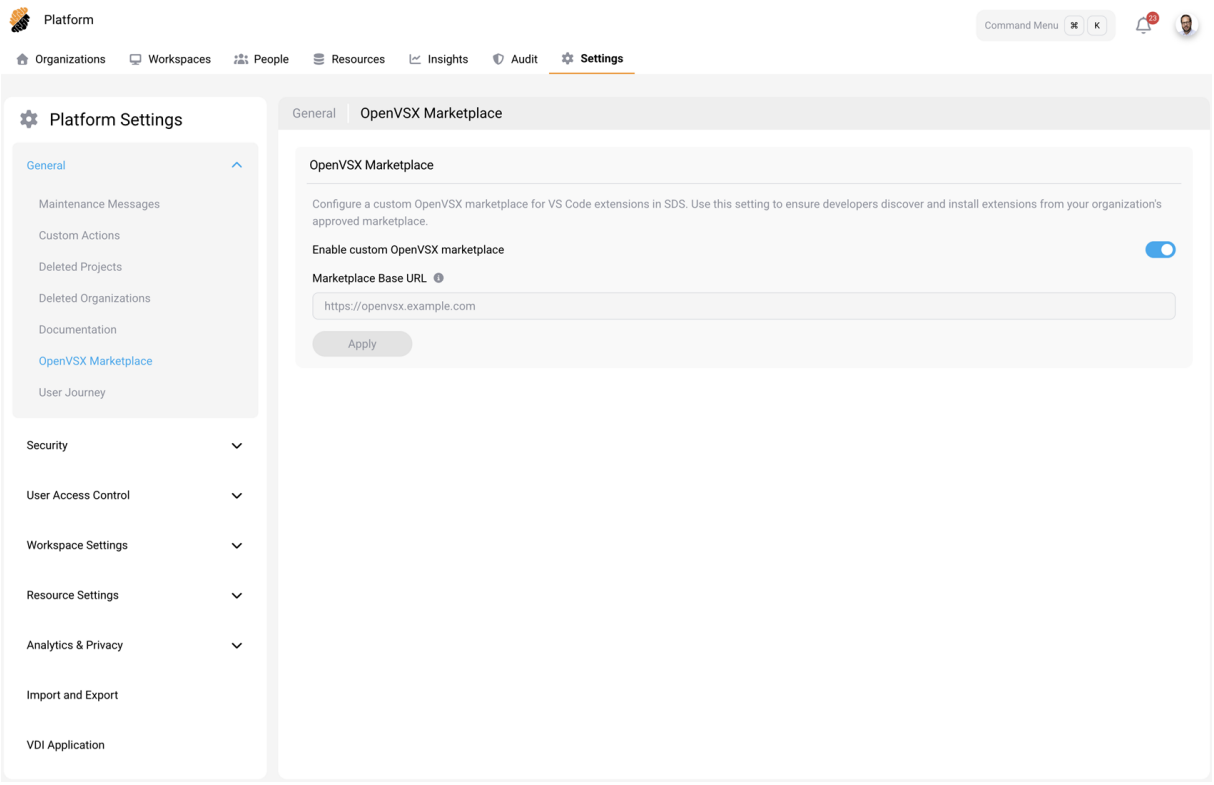
To revert to the default SecurSpaces documentation, turn off **Enable custom documentation link**.

OpenVSX Marketplace

By default, VS Code workspaces in SecurSpaces use the public Open VSX marketplace for extension discovery and installation. You can override this with a custom OpenVSX marketplace URL so that developers only discover and install extensions from your organization's approved marketplace.

This setting is configured at the platform level and is inherited by all organizations, projects, and workspaces. When enabled, new workspaces use the configured marketplace immediately. Existing workspaces use the new marketplace after their next restart.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



To configure a custom OpenVSX marketplace:

1. Go to **Settings > General > OpenVSX Marketplace**.
2. Turn on **Enable custom OpenVSX marketplace**.
3. In the **Marketplace Base URL** field, enter the base URL of your internal OpenVSX service, including `https://`.
4. Click **Apply**.

To revert to the default public marketplace, turn off **Enable custom OpenVSX marketplace**.

> **Note:**
>
> Changes take effect immediately for new workspaces. Existing workspaces continue to use their current marketplace configuration until they are restarted.

Onboarding & Guidance

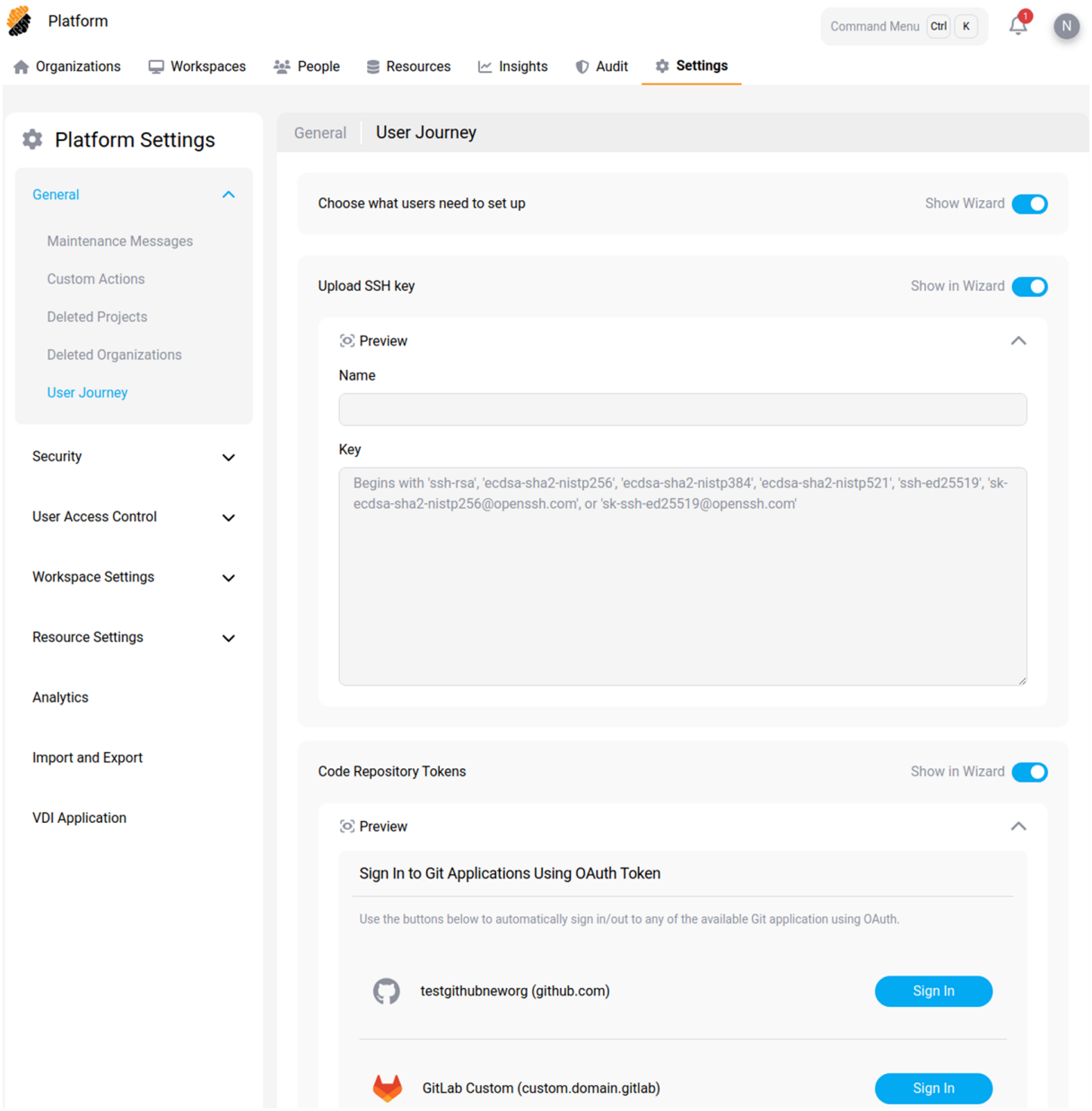
This section controls the onboarding experiences shown to new users and project owners. Three can be turned
on independently:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	What it shows
User Onboarding Checklist	A checklist on the developer's overview page
Project Scaffolding Checklist	A checklist on the project owner's overview page
Classic User Journey	The legacy welcome wizard shown at sign-in, focused on profile security settings

> **Note:**
>
> Turn on only the experiences your users need. Several overlapping flows at once can overwhelm a new user
> and dilute the value of each one.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Onboarding checklists

The two checklists give project owners and developers guided next steps from project pages. Each appears as a
collapsible side panel and keeps progress as users move through the product.

The **Project Scaffolding Checklist** guides a project owner to:

- Review container images.
- Add code repositories.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Configure secrets.
- Create workspace templates.
- Add users.

The **User Onboarding Checklist** guides a developer to:

- Review their profile.
- Configure integrations.
- Customize their configuration.
- Set up security credentials.
- Create a workspace.

To configure checklist availability:

1. Go to **Settings > General > Onboarding & Guidance**.
2. Turn **Project Scaffolding Checklist** on or off.
3. Turn **User Onboarding Checklist** on or off.
4. Select **Apply**.

Expected result: The selected checklist settings apply to project pages. Users keep their saved checklist
progress unless the feature is disabled.

Workspace policy

August 18, 2026

Workspace policy governs the workspaces created on the platform: what they may copy, how many a user may
have, whether they can be reached over SSH, and what network traffic they may generate.

The same settings appear at three scopes —**platform**, **organization**, and **project** —and are resolved from the broadest to the narrowest. This page covers all three.

Where each setting can be configured

Setting	Platform	Organization	Project
Clipboard security	yes	yes	yes
Default project limits	yes	yes	yes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Setting	Platform	Organization	Project
Remote development over SSH	—	yes	yes
Network policy	yes	yes	yes
SSH access to workspaces	yes	—	—
Schedule settings	yes	—	—
Workspace apps settings	yes	—	—
Workspace specification	yes	—	—
Registry access	yes	—	—
Deleted workspaces	yes	—	—

Platform settings are at **System Configuration**. Organization and project settings are on the settings page
for that organization or project.

How settings are resolved

Settings are read in order —defaults, then platform, then organization, then project —but two different
rules apply, and the difference matters.

Most settings: the narrowest scope wins, unless enforced

Clipboard security, default project limits, and network policy follow this rule. If a project sets a value it
overrides the organization, which overrides the platform.

A scope can also **enforce** its value. Enforcement stops the chain: the setting is locked at that scope and
no narrower scope can change it. The enforce control appears at platform and organization scope, since project
is the narrowest level.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remote development over SSH: every scope must allow it

SSH does not work that way. It is permitted only when **every** scope above allows it, and any scope that
disallows it turns it off for everything beneath. This is the default behavior, not an enforcement setting.

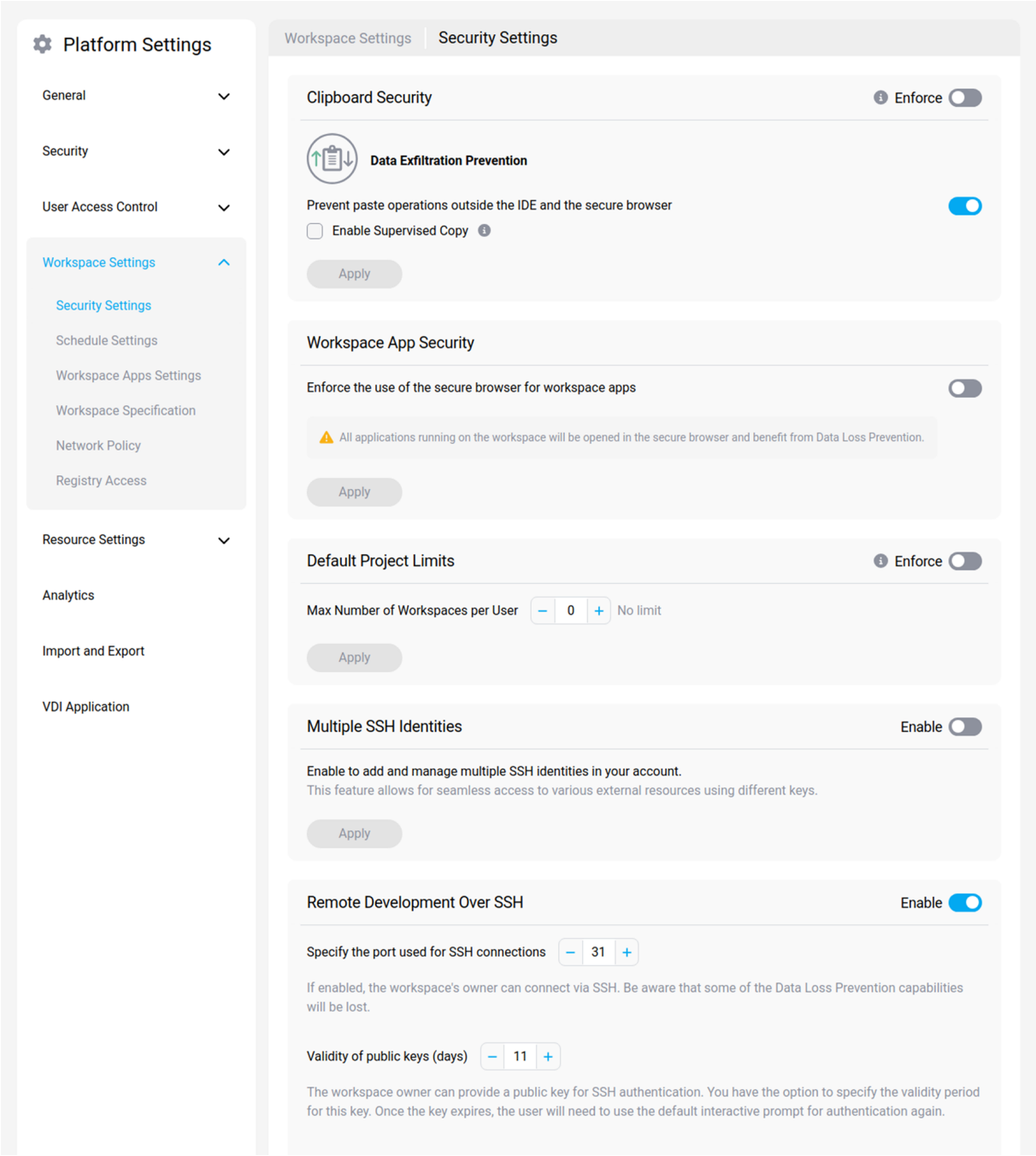
A project owner therefore cannot enable SSH when the platform has it switched off. The control is hidden
rather than shown as disabled, so if you cannot find it, check the scope above.

Security settings

Clipboard security

Prevents users from pasting content out of the IDE and the secure browser into other applications, as a data
loss prevention measure. You can restrict copying, restrict pasting, cap the number of characters, and enable
supervised copy.

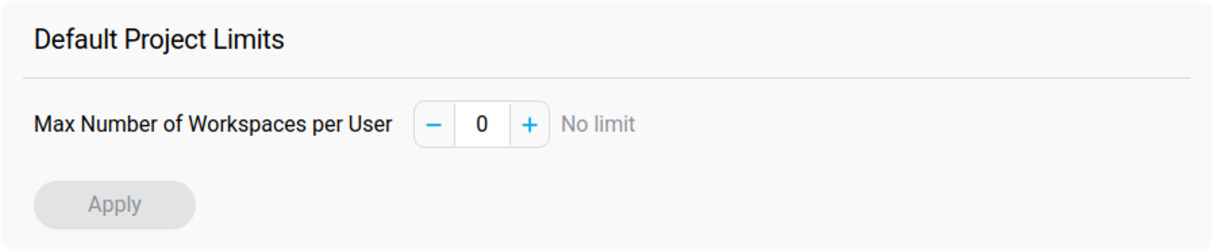
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Default project limits

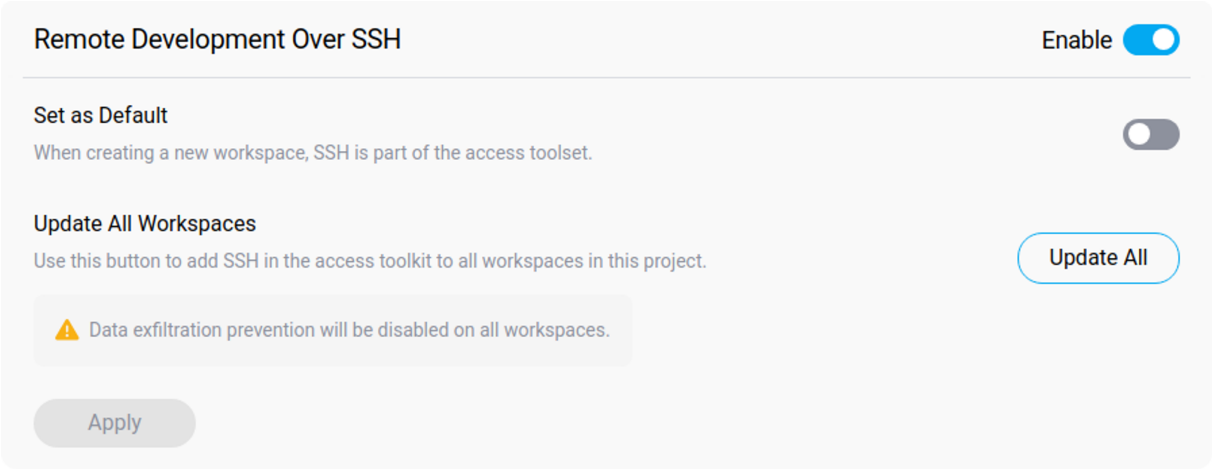
Caps the number of workspaces a user may create, which limits resource sprawl and cost. A limit of zero means
no limit.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Remote development over SSH

Allows developers to connect to their workspaces from a local IDE over SSH. Convenient, but it reduces the
effectiveness of local data loss prevention, so enable it deliberately.

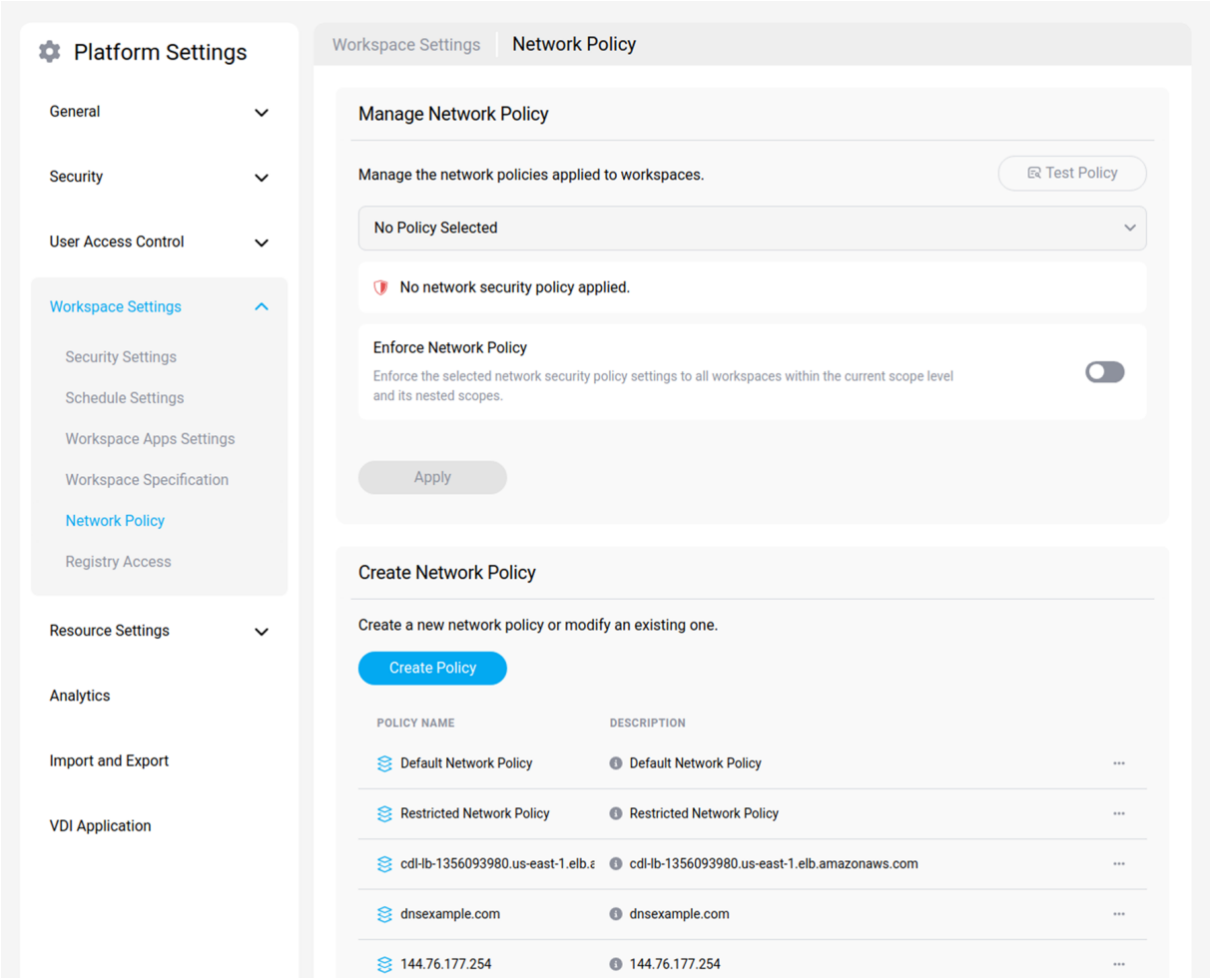


For what developers do once it is enabled, see
Connect over SSH.

Network policy

Network policies control the outbound traffic a workspace may generate, matched on domain and IP address.
Once a policy is attached, the workspace user cannot circumvent it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Three default policies are created with every project:

Policy	Behavior
Monitor Traffic (default)	Logs outgoing TCP traffic and raises audit events, without restricting it
Restrict Traffic (default, expert)	Blocks outbound traffic except attached resources and an allow list
Inspect Traffic (default, expert)	Inspects TCP content and reports it to the audit log

Network policies have their own section, covering the policy model, creating and assigning policies, the
expert-mode fields, and troubleshooting:

- Network policies
- Create a network policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Assign and enforce a policy

Platform-only settings

These exist at platform scope only.

SSH access to workspaces

Enables SSH access for the deployment, and sets the public key expiry and the port. Because SSH follows the
restrictive rule above, this switch gates every organization and project beneath it.

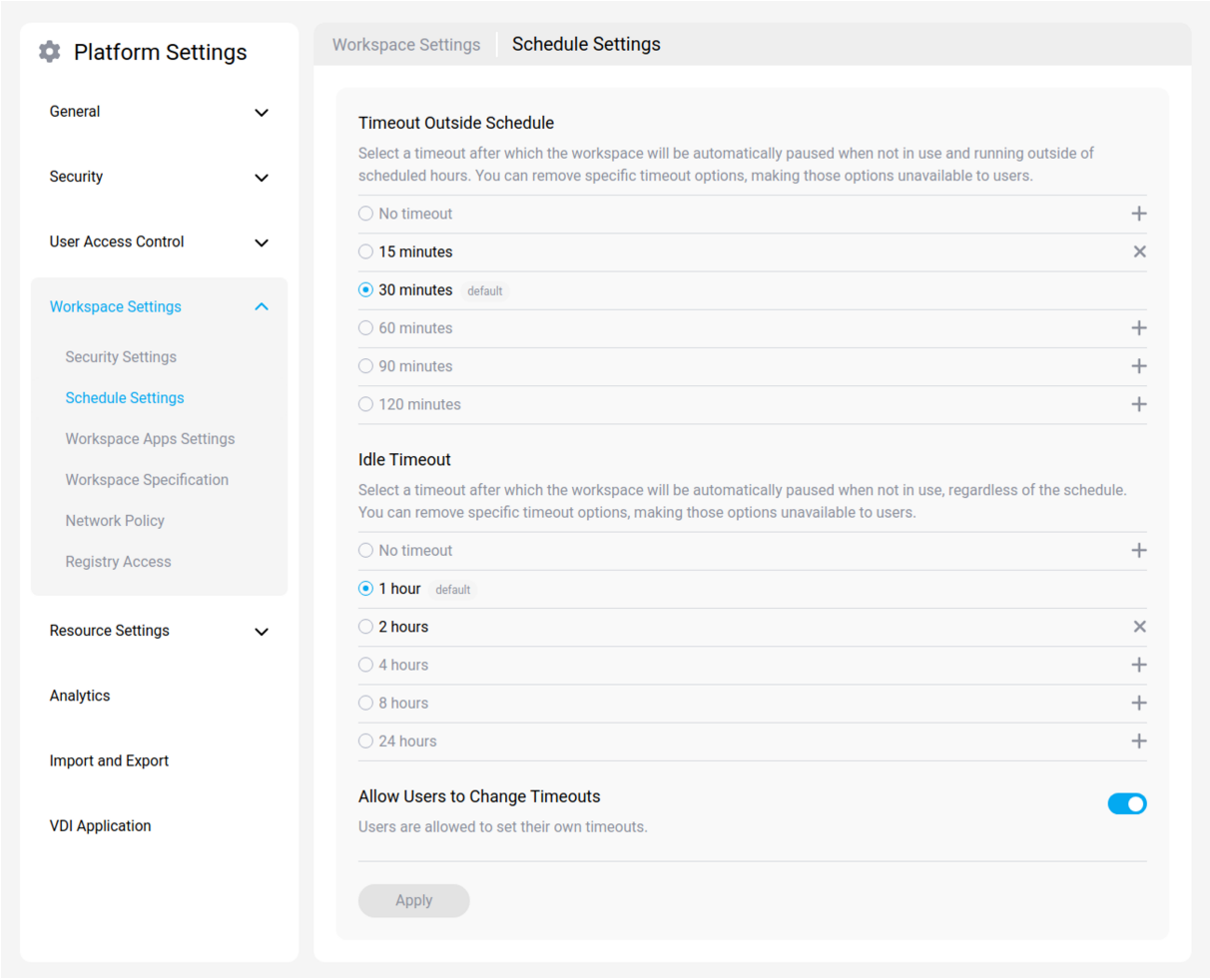
Validity of public keys (days) defaults to **7**. When a registered key reaches that age the platform stops accepting it. The SecurSpaces IDE integrations renew keys automatically, so developers using them are
unaffected; developers who connect with any other SSH client must re-register their key each time it expires.
Setting a short period therefore falls entirely on the second group. See
Key expiry and renewal.

Infrastructure configuration is separate; see
Enable SSH access to workspaces.

Schedule settings

Sets operating hours and automatic shutdown, so workspaces run only when needed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

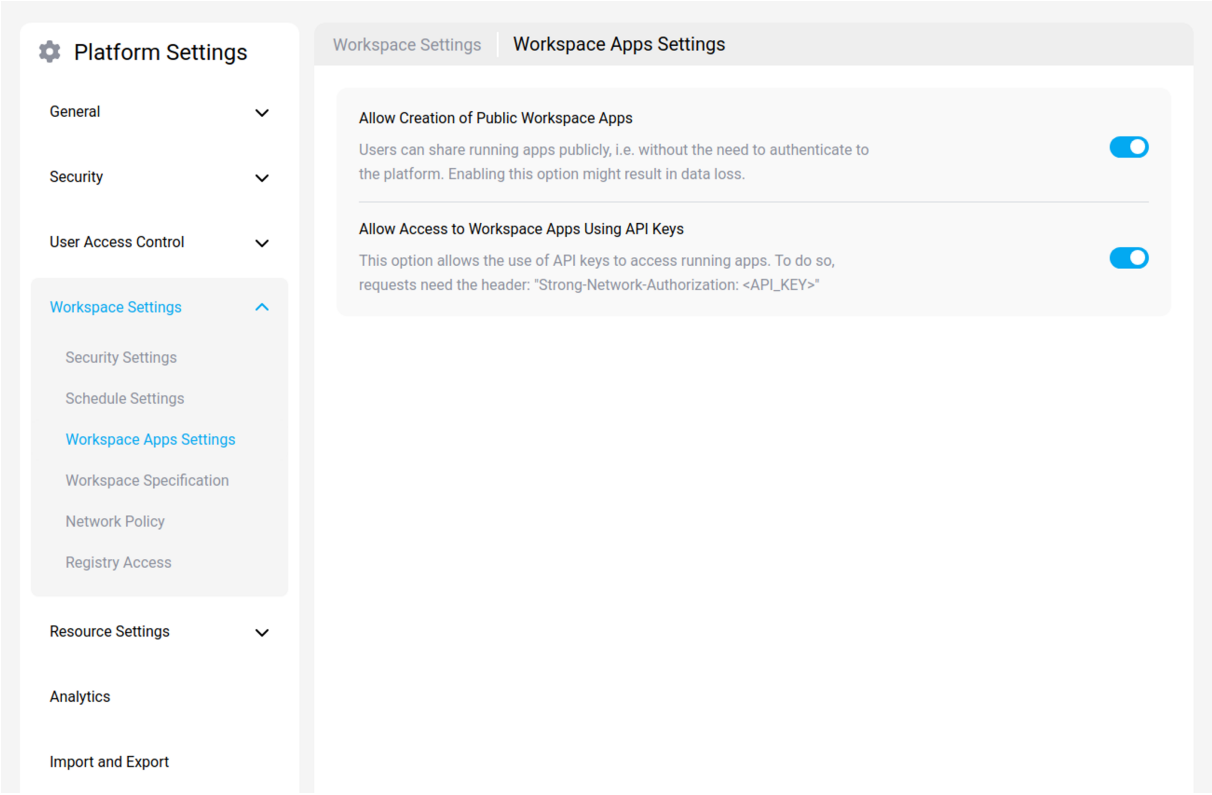


Workspace apps settings

Governs how workspace applications may be shared.

- **Allow creation of public workspace apps** —lets users share a running app without platform authentication. This can lead to data loss.
- **Allow access to workspace apps using API keys** —lets users reach a running app with an API key, sent as
  the header `Strong-Network-Authorization: <API_KEY>`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

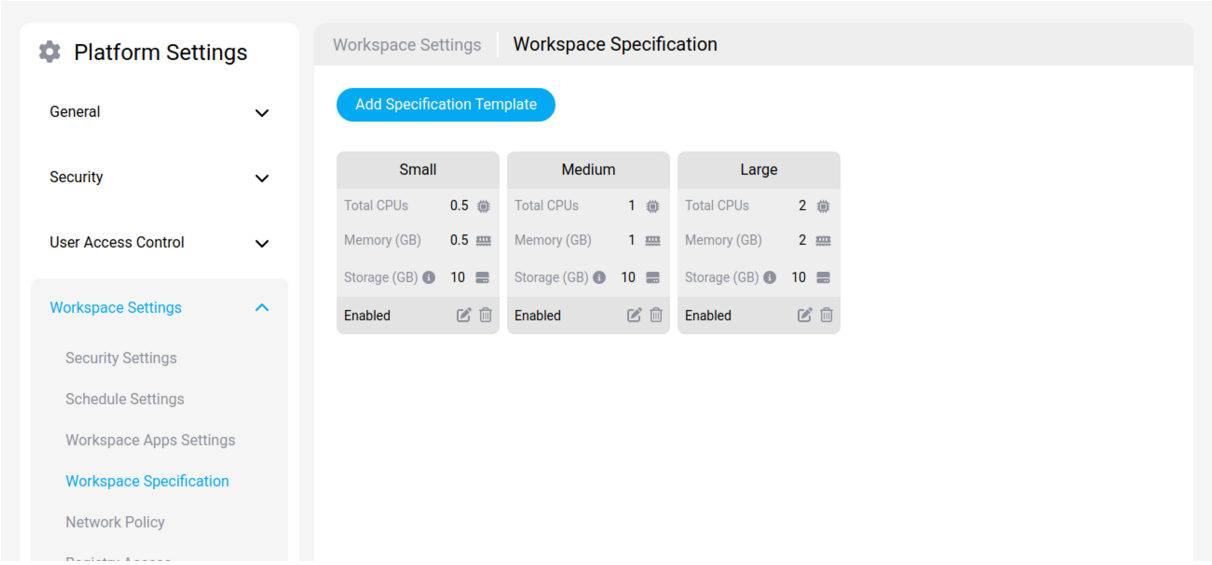


For what developers do with these, see
Workspace apps.

Workspace specification

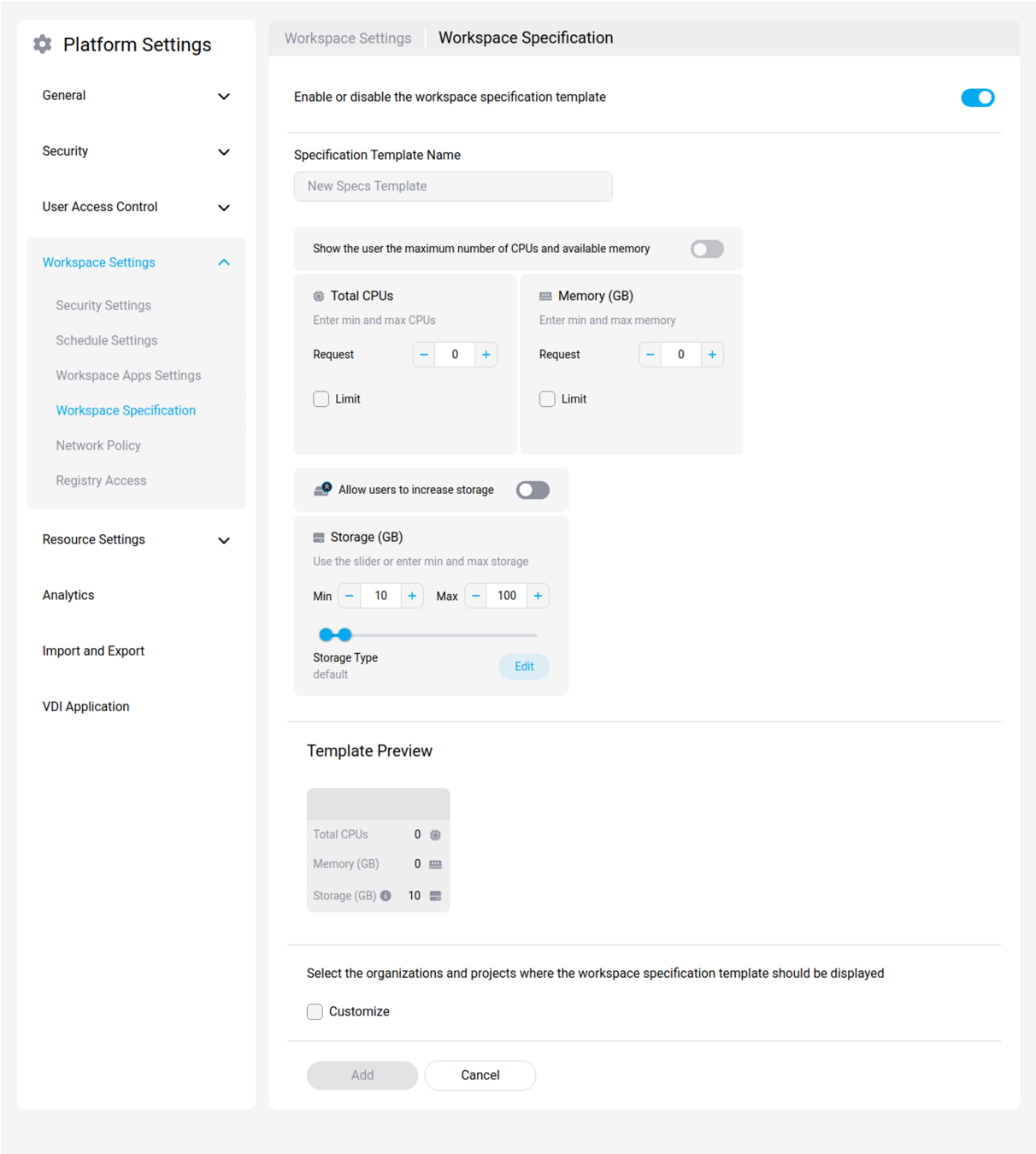
Defines the resource templates a workspace can be created from, with request and limit values for CPU, RAM,
and storage. A template can be restricted to particular organizations or projects, and users see only the
templates that apply to them.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



When you create a template, set the initial request values and the maximum limit values for CPU, RAM, and
storage.

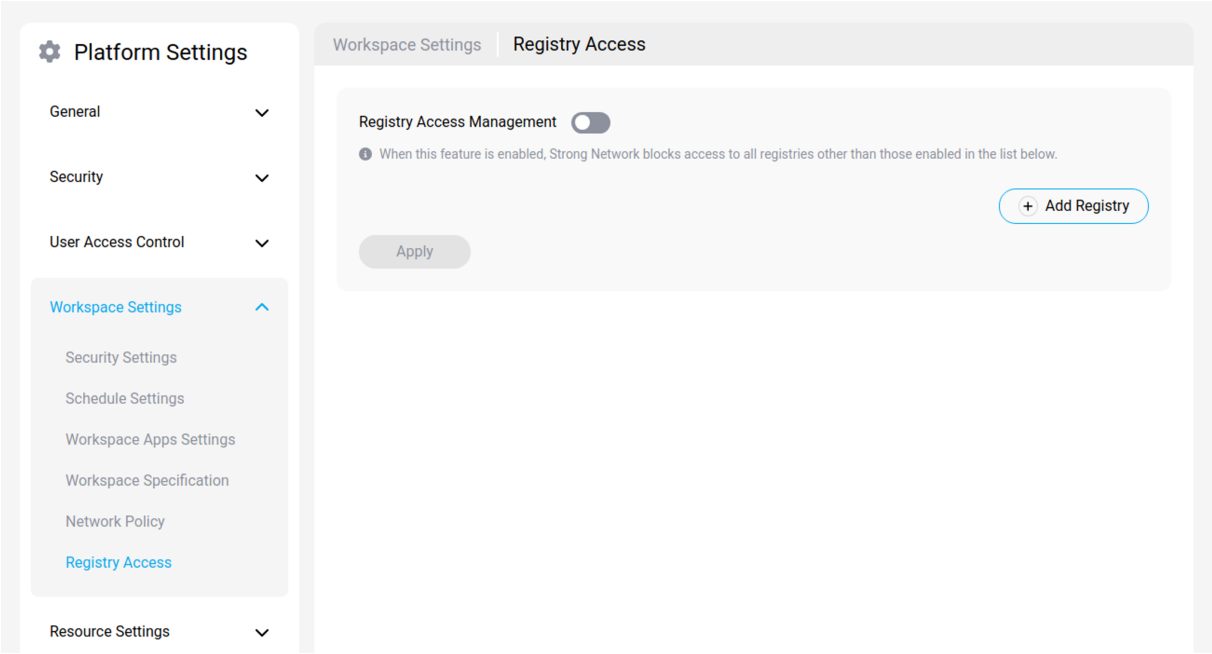
© 1997–2026 Citrix Systems, Inc. All rights reserved.



Registry access

Controls which container image registries workspaces may pull from, so only approved sources are used.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Workspaces pending deletion

Deleting a workspace does not destroy it immediately. It moves to a recycling bin, where it stays recoverable
until an expiry period elapses. **Workspaces Pending Deletion** lists every workspace across the platform that
is currently in that state, with its owner, its organization and project, and the date it was deleted.

From the list you can **Restore** a workspace, **Delete** one permanently, or use **Delete All Workspaces** to
empty the bin.

To set how long deleted workspaces stay recoverable:

1. Go to **Platform > Workspaces > Workspaces Pending Deletion**.
2. Set **Expiry in days**, from **1 to 99**. The default is **7**.
3. Select **Save**.

> **Important:**
>
> The expiry period is applied to workspaces **already in the bin**, not only to ones deleted afterwards. The
> deadline for each workspace is recalculated from the date it was deleted whenever the setting changes.
> Shortening the period can therefore bring forward the deletion of workspaces already waiting,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> and may purge
> some of them at the next cleanup. Lengthening it gives them all more time.

Workspace owners see their own pending deletions, and the emails sent about them, on their profile page. See
Workspaces pending deletion.

Deleted projects and organizations Projects and organizations have their own recycling bins, and **their retention is fixed at 7 days and cannot be configured**. Changing **Expiry in days** does not affect them.

Item	Retention	Where to restore it
Workspace	1–99 days, default 7	Platform > Workspaces > Workspaces Pending Deletion
Project	7 days, fixed	Platform or organization **Settings > Deleted Projects**
Organization	7 days, fixed	Platform **Settings > Deleted Organizations**

Deleting a project or an organization also deletes the workspaces and resources inside it, so restore the
parent first if you need something from within it. See
Project and organization settings.

Related information

- Network policies
- Project and organization settings
- Roles and permissions
- Connect over SSH

Workspace lifecycle and idle detection

Citrix SecurSpaces™ 2026.4 introduces a hybrid idle detection engine that uses multiple signals to determine whether a workspace is truly idle before pausing it. This replaces the previous approach, in which an open SSH connection alone was sufficient to keep a workspace running indefinitely.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How it works

The idle detection engine uses a priority-based approach with three layers:

IDE activity signals (primary)

When the SecurSpaces IDE extension is active in the developer's IDE, the system uses granular input signals such as keyboard and mouse activity to determine whether a user is present. An open but unused SSH connection alone no longer prevents a workspace from pausing.

This applies to IDEs that support the SecurSpaces extension, including VS Code Desktop, Cursor, Windsurf, and Kiro.

SSH fallback (secondary)

For developers using IDEs that do not support the SecurSpaces extension, the system falls back to the previous SSH-based logic. An active SSH connection continues to signal that the workspace is in use, so active work is not interrupted.

> **Note**
>
> Starting a workspace is not an activity signal. A workspace that is started and then left alone is paused
> normally once the idle timeout is reached. Starting a workspace does, however, count as use for the
> **Inactive Workspace** condition that automatically deletes unused workspaces. See
> Custom Actions.

Bypass Schedule (manual override)

Security officers and platform administrators can deactivate idle detection for individual workspaces. When Bypass Schedule is enabled for a workspace, the scheduler is bypassed, and the workspace keeps running regardless of detected user activity.

This is intended for long-running services such as shared development tooling or infrastructure components that must remain available continuously.

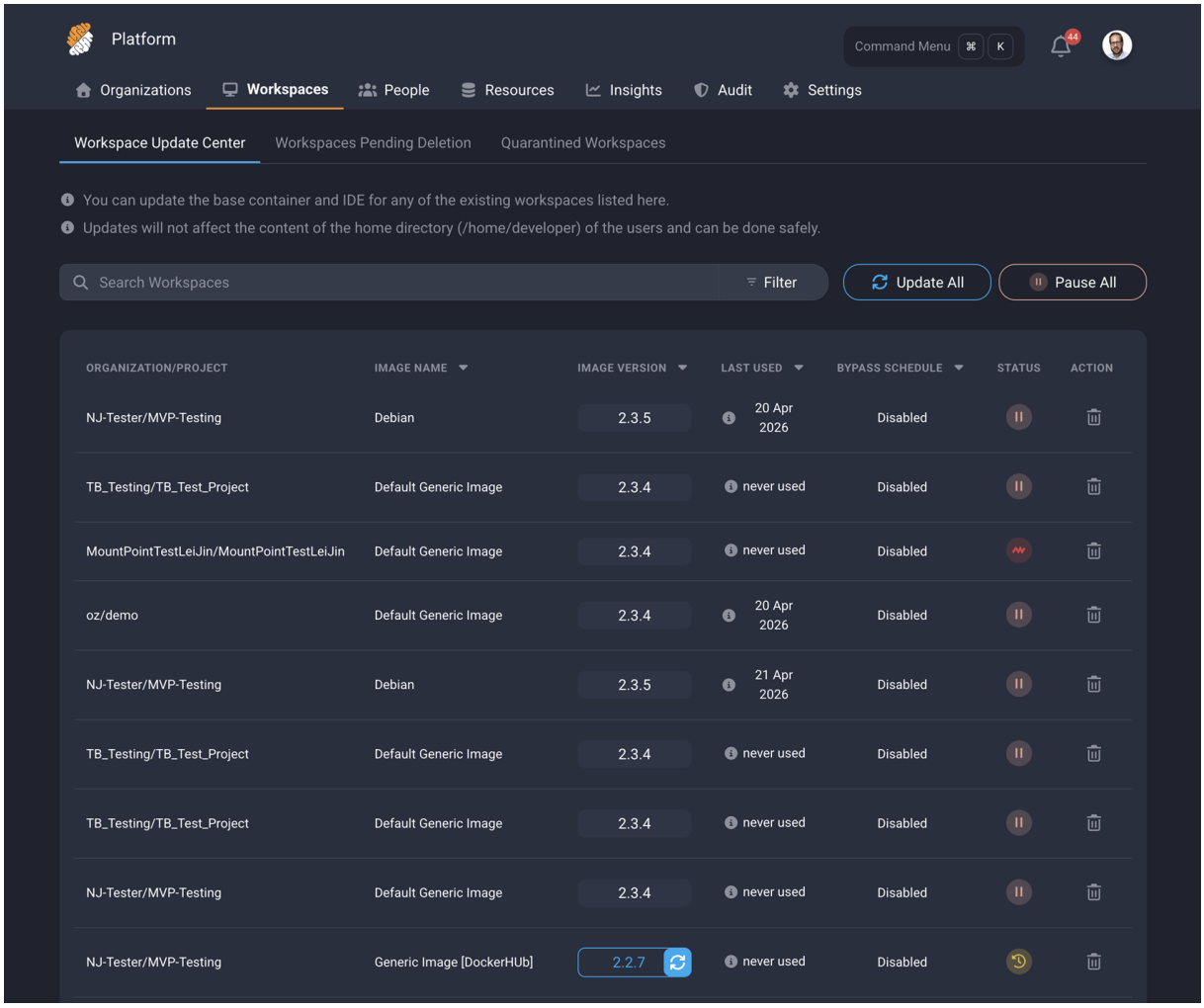
Bypass Schedule in the Workspace Update Center

The **Bypass Schedule** column is visible in the platform-level **Workspace Update Center** (Platform > Workspaces > Workspace Update Center). Each workspace row displays one of two states:

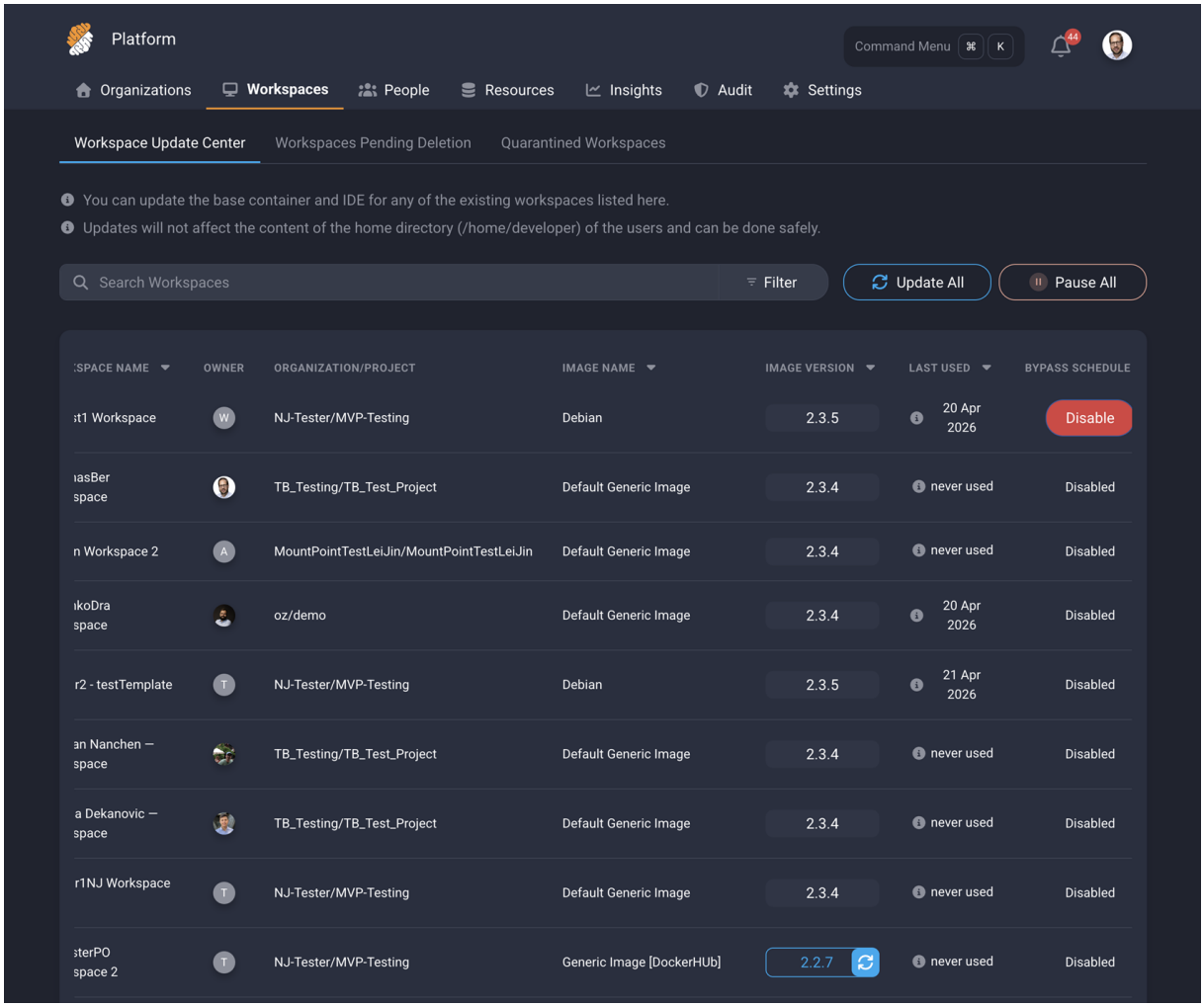
© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Disabled** (default) —The workspace follows the normal idle detection schedule. The workspace pauses automatically after the configured timeout if no activity is detected.
- **Enabled** —The workspace bypasses the schedule and keeps running. A red **Disable** button appears, allowing administrators to revert.

To enable Bypass Schedule for a workspace, select **Enable** in the Bypass Schedule column for that workspace.



© 1997–2026 Citrix Systems, Inc. All rights reserved.



Permissions

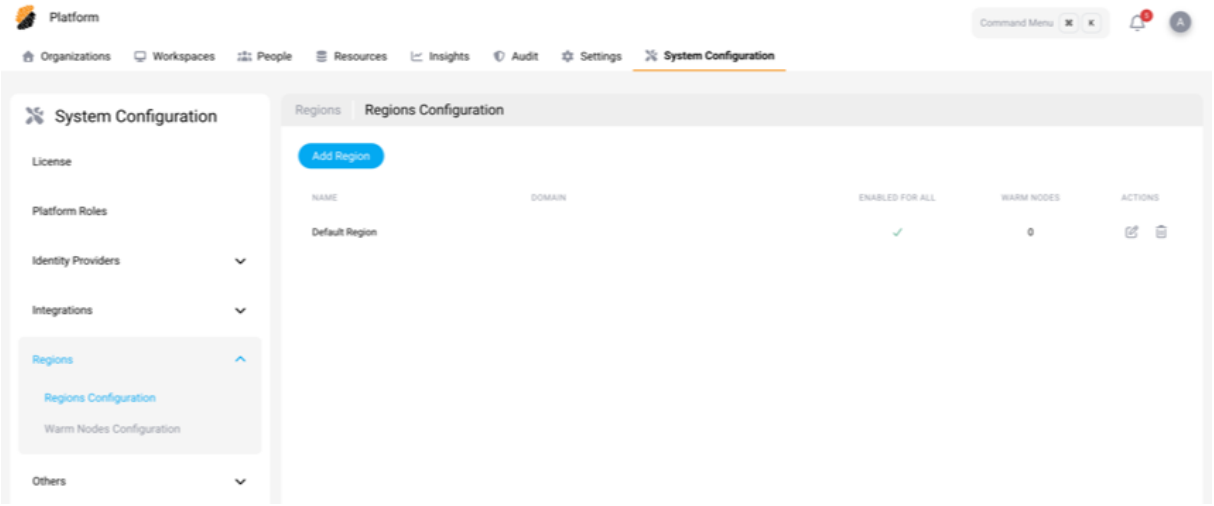
Only **security officers** and **platform administrators** can turn Bypass Schedule on or off. This restriction exists because bypassing the schedule directly impacts infrastructure costs.

Regions

Run workspaces closer to your users. In **Regions**, you set up the regions and warm nodes for a multi-region deployment of Citrix SecurSpaces™.

Find these settings at **System Configuration > Regions**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



> **Note:**
>
> Regions appears only when **Multi Region** is turned on for your license.

There are two pages:

- Regions Configuration
- Warm Nodes Configuration

Regions Configuration

Add the regions where workspaces can run. For each region, give it a unique name and domain, then decide who can use it:

- **Enable for All** —Everyone on the platform can use the region.
- Limit it to specific organizations, projects, or users.

> **Note:**
>
> If a region isn't enabled for all and no organizations, projects, or users are selected, no one can use it.

For deployment prerequisites and setup, see Deploy SecurSpaces in Multiple Regions.

> **Note:**
>
> Mount points are supported only in the primary region. A workspace that uses a mount point cannot be moved
> to another region, and the region cannot be changed while one is attached. See
> Choose between data buckets and mount points.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Warm Nodes Configuration

Keep a pool of ready-to-go nodes in a region with **warm nodes**. Because the nodes are already provisioned, workspaces start faster instead of waiting for new nodes to spin up.

Related information

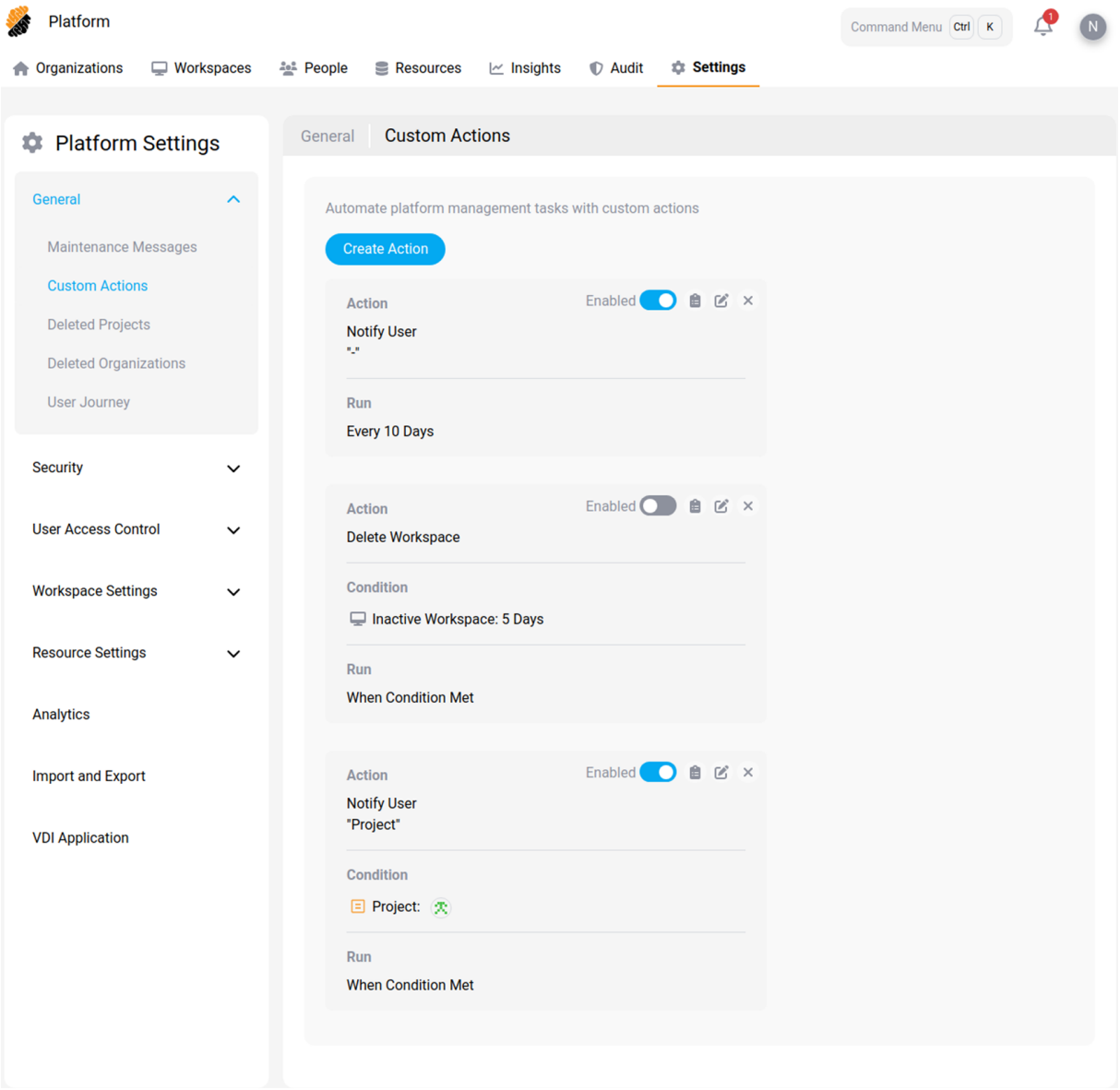
- License
- System Configuration

Custom actions

Custom Actions let you automate routine platform maintenance. You define a rule made of three parts —an action to run, the conditions that decide which workspaces or users it applies to, and when it runs. A common example is deleting workspaces that have been inactive for a long time.

Find this setting at **Settings > General > Custom Actions**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



How a custom action is built

Every custom action combines three parts.

Action —what runs

Action	What it does
Update Workspace	Applies an update to each matching workspace, such as an image update.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Action	What it does
Delete Workspace	Deletes each matching workspace.
Notify User	Sends an in-platform notification with a message you write to each matching user.

Condition —which workspaces or users it applies to

Add one or more conditions to target the action. Available conditions include:

- **Organization:** Limit to workspaces in specific organizations.
- **Project:** Limit to workspaces in specific projects.
- **User:** Limit to workspaces owned by specific users.
- **Inactive Workspace:** Match workspaces that haven't been used for a set number of days.
- **After** and **Before:** Limit the action to a time window by setting a start date, an end date, or both.

When —how the action is scheduled

The **When** field controls the schedule. It offers three options:

Option	Behavior
Once	Runs one time, at the date and time you set.
Time Period	Runs on a repeating schedule, such as every 30 days.
When Condition Met	Runs continuously in the background and applies the action whenever the conditions are met. This option appears only after you add at least one condition.

"When Condition Met"versus "Time Period"

This is the key choice for a maintenance rule like deleting inactive workspaces, so it's worth understanding how each option behaves.

When Condition Met evaluates your conditions on a recurring background cycle, roughly every 15 minutes. Each cycle, it finds everything that currently matches and applies the action. It doesn't wait for a scheduled date. As soon as a workspace crosses your inactivity threshold, it's picked up in the next cycle.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Time Period runs on the fixed interval you set —for example, every 30 days. The conditions only decide which workspaces are affected when that scheduled run happens. Anything that becomes a match between two scheduled runs waits until the next run.

For a rule like "delete a workspace when it's been inactive for 365 days,"**When Condition Met** is the right choice. The action checks continuously and deletes each workspace shortly after it passes 365 days of inactivity —not on a separate schedule you'd have to keep in sync.

> **Note:**
>
> With **When Condition Met**, the action doesn't run "once a day until the condition is met."It re-evaluates on a short background cycle (about every 15 minutes) and acts as soon as something matches. Because a deleted workspace no longer exists, it isn't matched again —each workspace is deleted once.

Example: delete workspaces inactive for 365 days

1. Go to **Settings > General > Custom Actions** and start a new action.
2. For the action, select **Delete Workspace**.
3. Add a condition, select **Inactive Workspace**, and set the period to **365** days.
4. For **When**, select **When Condition Met**.
5. Save the action.

The platform now checks continuously. Any workspace that reaches 365 days without use is deleted automatically within the next evaluation cycle.

> **Note:**
>
> Inactivity is measured from a workspace's last recorded use. A workspace that has never been opened isn't matched by the **Inactive Workspace** condition.

What counts as use

A workspace counts as used when someone works in it through an IDE, a terminal, or SSH. Starting a workspace
manually also counts, whether it is started from the console, the REST API, or the IDE extension. This means a
workspace that a developer starts regularly is not matched by the **Inactive Workspace** condition, even if
they never open a session inside it.

A workspace that the scheduler resumes automatically does not count as used. Automatic resumes are not user
activity, so they do not keep an otherwise unused workspace out of scope.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

When an administrator starts a workspace on someone else's behalf, the use is recorded against the workspace
owner.

> **Note:**
>
> Starting a workspace does not postpone the idle timeout that pauses inactive workspaces. The two are
> measured separately: starting a workspace keeps it out of scope for deletion, but only real activity inside
> the workspace prevents it from being paused. See
> Workspace idle detection.

Related information

- General Settings
- Managing workspaces

Project labels

Project Labels let you build a shared taxonomy that organizes every project on the platform. As an administrator, you define the taxonomy once by uploading a CSV file. From then on, whoever creates or edits a project chooses a value at each level of that taxonomy, so projects stay consistently categorized across organizations.

Find this setting at **System Configuration > Others > Project Labels**.

What Project Labels are used for

Project Labels give you a single, controlled vocabulary for classifying projects. Instead of letting each team invent its own naming, you publish a hierarchy that everyone selects from.

Once labels are configured, they appear in three places:

- **Create and edit project:** Users pick a value at each level of the hierarchy. If labels are configured, a selection at every level is required.
- **Projects list:** A **Project Tags** column shows the label path for each project, so you can scan how projects are categorized.
- **Usage reporting:** The selected labels are attached to workspace and active-user metrics, so you can group usage data by your business taxonomy.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The hierarchy can be up to five levels deep. Each level becomes a separate dropdown when a user creates or edits a project.

How the CSV file is structured

You define the taxonomy by uploading a CSV file. The platform reads the file as a tree: each row describes one full path from the top level down, and repeated values are merged automatically.

Part of the file	Requirement	Purpose
First row (header)	Required	Each cell names a level of the hierarchy. These names become the dropdown labels users see when they create a project.
Remaining rows	At least one required	Each row lists one value per level, from left (top level) to right.
Columns	1 to 5 used	Each column is one level of the hierarchy. The platform reads more than five columns but only the first five appear in the project form.
Empty cells	Optional	A row can stop at any column. Empty cells are skipped.

Keep these rules in mind:

- Use a standard UTF-8, comma-separated CSV file.
- List every combination you want to allow. The platform removes duplicate values at the same level automatically, so you don't need to worry about repeating a parent value across rows.
- Uploading a new file replaces the entire taxonomy. It doesn't merge with a previous upload.

Column reference

Column	Level	Example header	Data type
1	Top level	Business Unit	Text

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Column	Level	Example header	Data type
2	Second level	Product Line	Text
3	Third level	Team	Text
4	Fourth level	Sub-team	Text
5	Fifth level	Component	Text

Example CSV

Say you want to classify every project by business unit, then by product line, then by team. You'd create a file like this:

```
Business Unit,Product Line,Team
Finance,Payments,Checkout
Finance,Payments,Fraud
Finance,Billing,Invoicing
Healthcare,Patient Portal,Web
Healthcare,Patient Portal,Mobile
Healthcare,Claims,Processing
```

This file produces the following hierarchy:

- **Finance**
  - Payments > Checkout, Fraud
  - Billing > Invoicing
- **Healthcare**
  - Patient Portal > Web, Mobile
  - Claims > Processing

Notice that **Finance** and **Payments** each appear on several rows. The platform merges these repeated values automatically, so you don't create duplicates —you simply list every full path you want to allow.

What the user sees

After you upload the file, the three header names become three dropdowns on the Create Project form: **Business Unit**, **Product Line**, and **Team**. The dropdowns cascade, so each choice narrows the next.

For example, a user creating a project might:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Choose **Business Unit** > **Finance**. The **Product Line** dropdown now offers only **Payments** and **Billing**.
2. Choose **Product Line** > **Payments**. The **Team** dropdown now offers only **Checkout** and **Fraud**.
3. Choose **Team** > **Fraud**.

The project is now labeled **Finance > Payments > Fraud**. That path appears in the **Project Tags** column on the projects list and travels with the project's usage data, so you can report on all Finance projects —or just the Fraud team's —at any time.

Upload your Project Labels

1. Go to **System Configuration > Others > Project Labels**.
2. Select the file field and choose your CSV file.
3. Select **Upload**.

The new taxonomy takes effect immediately and replaces any labels uploaded before. Existing projects keep their current labels until someone edits the project and chooses new values.

Related information

- Others
- System Configuration
- Projects

Onboarding emails

Citrix SecurSpaces™ sends automatic email notifications that help new users find their way onto the platform and into their projects. These emails give recipients a direct sign-in link, so they can get started without waiting for manual instructions.

This page explains which emails SecurSpaces sends, when each one is triggered, and what recipients see. It's written for platform administrators and project owners who onboard users.

> **Note:**
>
> Email notifications are only delivered when an email gateway is configured. See Prerequisite: configure an email gateway.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How onboarding emails work

SecurSpaces generates an onboarding email whenever an event adds a user to the platform or to a project. The platform builds the message from a template, personalizes it with details such as the inviter's name and a sign-in link, and sends it through the configured email gateway.

Recipients don't need an existing account to receive these emails. Each email includes a button that takes the user straight to the sign-in page or the relevant project.

Prerequisite: configure an email gateway

Onboarding emails are sent through the email gateway defined in **System Configuration**. Until a gateway is set up, SecurSpaces generates no onboarding emails.

To set up the gateway:

1. Sign in as a platform administrator and select **System Configuration**.
2. Select **Others** > **Email Gateway Configuration**.
3. Choose a provider: **Amazon SES**, **SendGrid**, or **SMTP**.
4. Enter the required details for your provider, including the **Sender Email** address recipients see in the *From* field.
5. Select **Save**.

For the full list of provider fields, see Others.

> **Note:**
>
> The **Sender Email** address appears as the sender of every onboarding email. Use an address that recipients recognize and trust, and that your provider is authorized to send from.

Email types

SecurSpaces sends three onboarding emails.

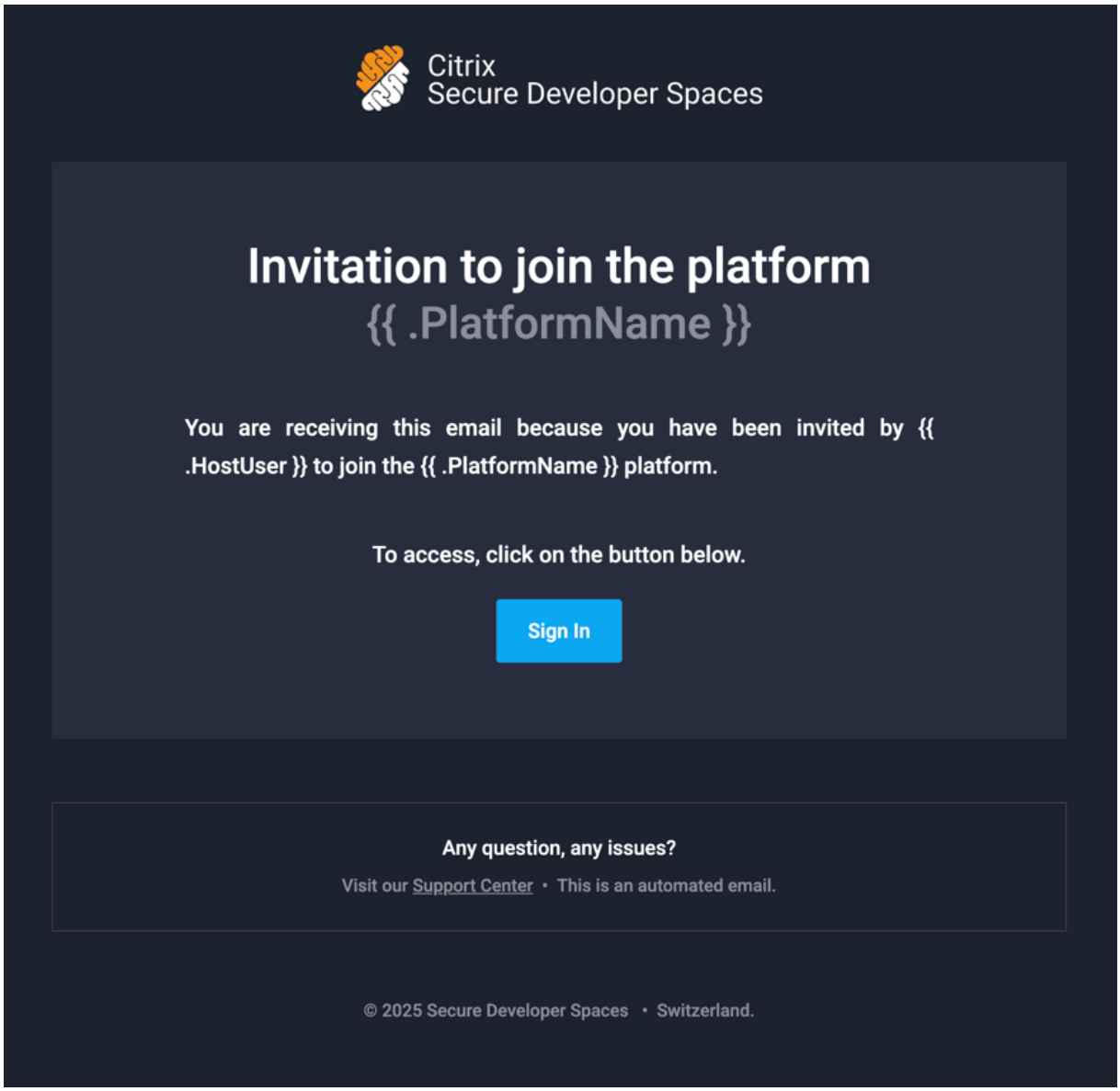
Email	Trigger	Primary action
Invitation to join the platform	A user is invited to the platform.	**Sign In** —opens the platform sign-in page.
You've joined a new project	A user is added to a project.	**Access Project** —opens the project.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Email	Trigger	Primary action
Test email	An administrator sends a test from the email gateway settings.	None. Confirms that the gateway works.

Invitation to join the platform

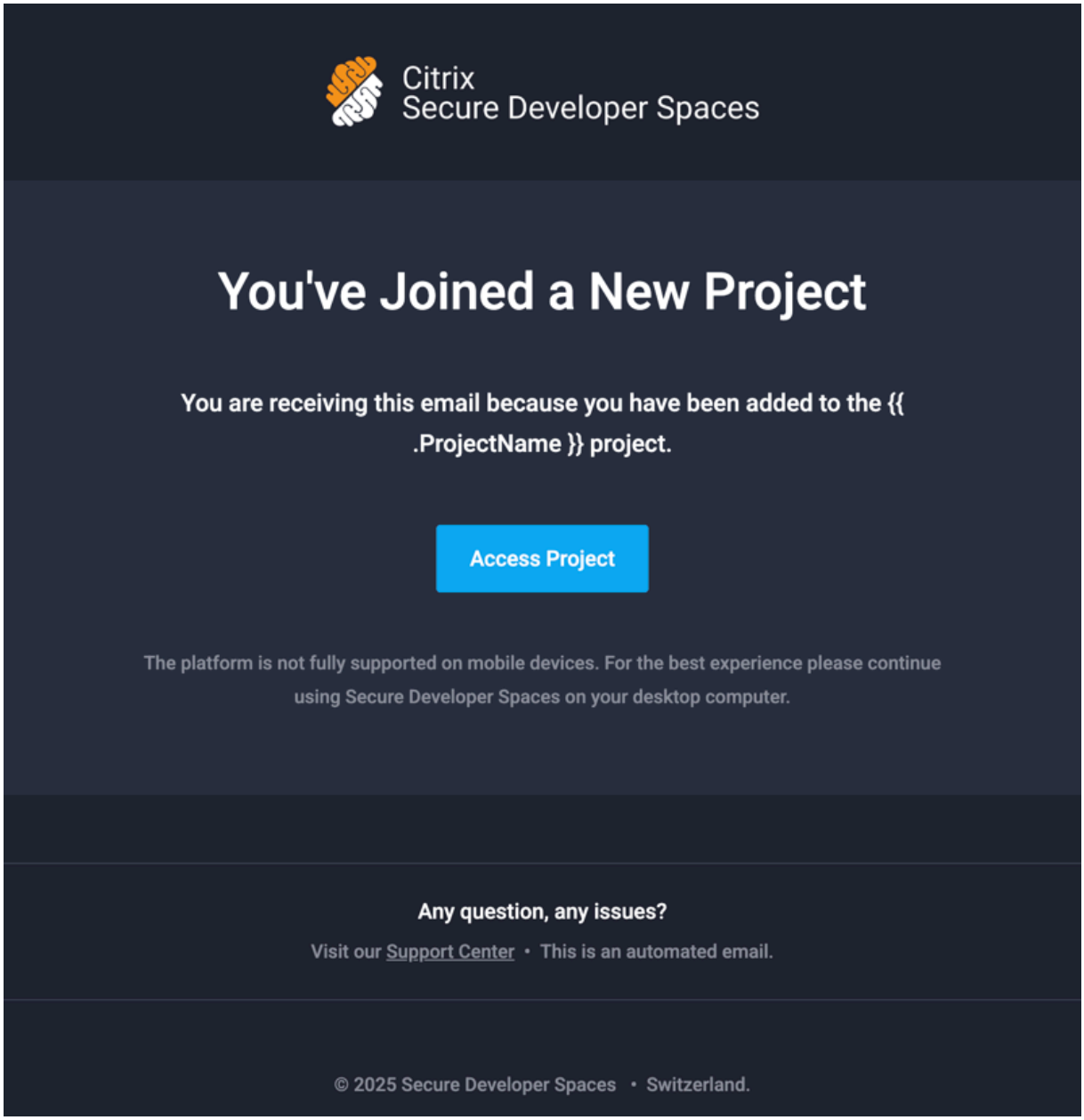
Sent when a user is invited to the platform for the first time. The email names the inviter and includes a **Sign In** button that opens the sign-in page with the recipient's email address pre-filled.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

You've joined a new project

Sent when a user is added to a project. The email names the project and includes an **Access Project** button that opens it.



Test email

Sent when an administrator uses the test option in **Email Gateway Configuration**. Use it to confirm that the gateway is set up correctly before you invite users. The test email doesn't include an action button.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Troubleshoot onboarding emails

If a user reports that they didn't receive an onboarding email:

Check	What to do
Email gateway	Confirm that a gateway is configured under **System Configuration** > **Others** > **Email Gateway Configuration**. Without it, SecurSpaces sends no emails.
Gateway health	Send a test email from the gateway settings to confirm that the provider works.
Sender address	Verify that the **Sender Email** address is valid and authorized to send through your provider.
Spam or quarantine	Ask the recipient to check their spam or quarantine folder. Automated emails are sometimes filtered.
Email address	Confirm that the address used to invite the user is correct.

If the test email arrives but onboarding emails don't, re-check the event that should have triggered the email —for example, that the user was actually added to the platform or project.

Related information

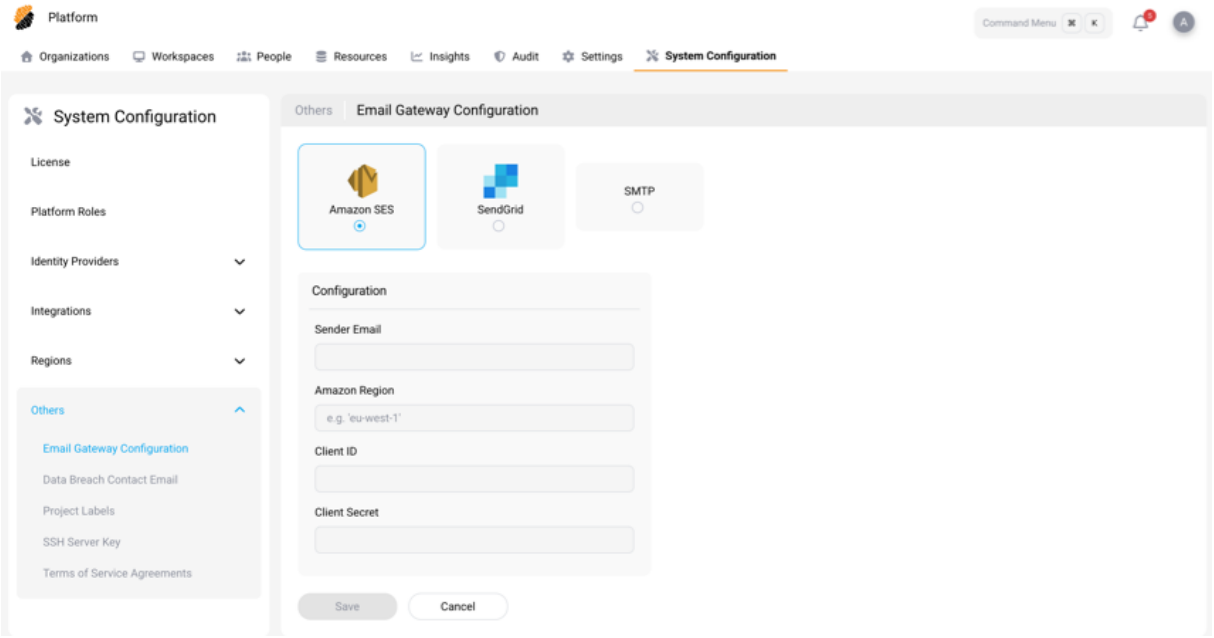
- Others
- Add and remove users and groups
- System Configuration

Email gateway

Others brings together a handful of platform-wide settings that don't fit the other areas.

Find these settings at **System Configuration > Others**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Here's what you'll find:

- Email Gateway Configuration
- Data Breach Contact Email
- SSH Server Key
- Terms of Service Agreements
- Project Labels

Email Gateway Configuration

Set up the gateway the platform uses to send email notifications, such as onboarding invitations. Choose a provider, enter its details, add the **Sender Email** address that recipients see, and send a test email to check that it works.

Choose one of the following providers:

- **Amazon SES**
- **SendGrid**
- **SMTP**

The **Sender Email** field applies to every provider. The remaining fields depend on the provider you select.

Amazon SES

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Field	Description
Sender Email	The address that appears as the sender of each email.
Amazon Region	The AWS Region for your Amazon SES setup (for example, `eu-west-1`).
Client ID	Your Amazon SES access key ID.
Client Secret	Your Amazon SES secret access key.

SendGrid

Field	Description
Sender Email	The address that appears as the sender of each email.
API Key	Your SendGrid API key.

SMTP

Field	Description
Sender Email	The address that appears as the sender of each email.
Host	Your SMTP server hostname (for example, `smtp.example.com`).
Port	Your SMTP server port.
Username	The SMTP username. Optional.
Password	The SMTP password. Optional.

To confirm the gateway works, send a test email. Until a gateway is configured, the platform sends no email notifications. For details on the emails the platform sends, see Onboarding email notifications.

Data Breach Contact Email

Give users an easy way to report a suspected data breach. Add a contact email, then select **Enable Display of Reporting Information** to show it across the platform.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SSH Server Key

Manage the server SSH key used for traffic outside the Citrix SecurSpaces™ ecosystem. Upload a new SSH server private key whenever you need to rotate it.

Terms of Service Agreements

Add the terms of service that users accept before they use the platform. You can upload more than one document, remove a document, or reset all acceptances so users accept it again.

For details on what users see, whether acceptance is required, and how to reset acceptances, see Terms of Service Agreements.

Project Labels

Manage the labels that help you sort and organize projects across the platform. You define the labels by uploading a CSV file that describes a hierarchy, and users then choose from that hierarchy when they create or edit a project.

For the CSV format, an example, and upload steps, see Project Labels.

Related information

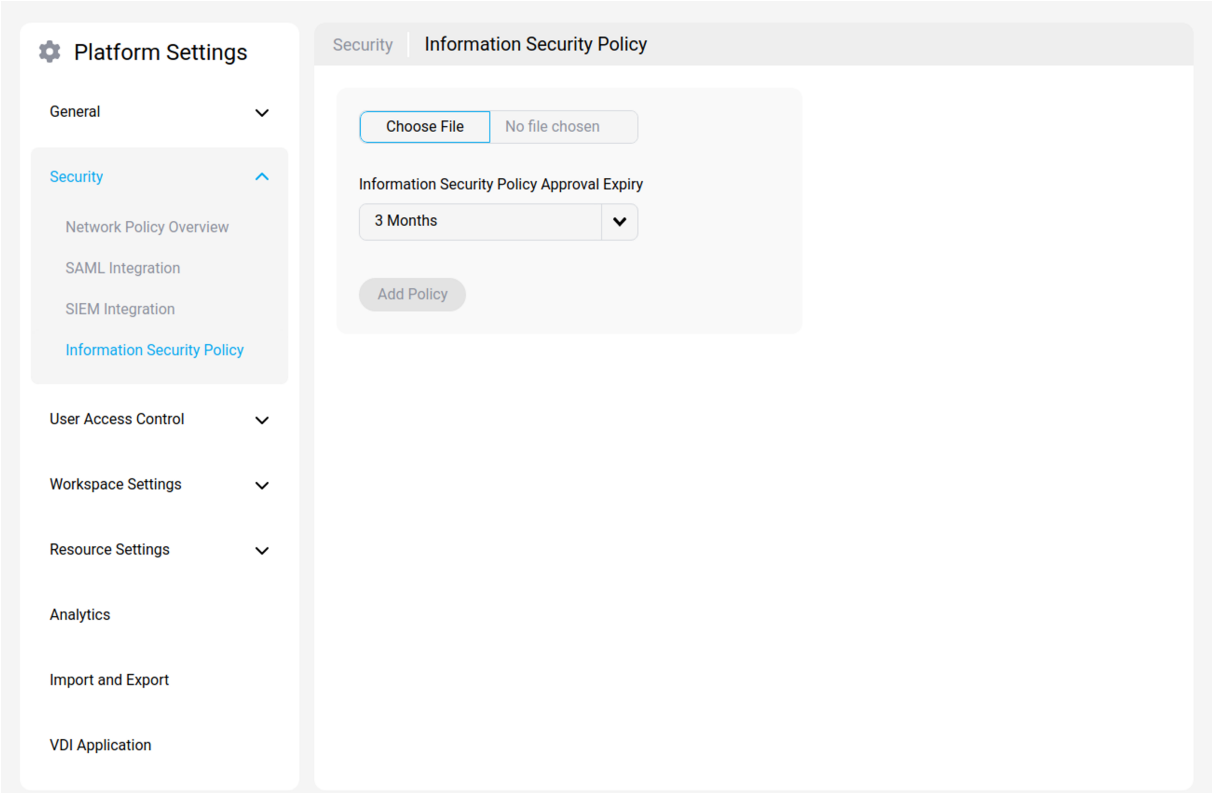
- Onboarding email notifications
- System Configuration
- SSH Access to Workspaces

Information security policy

The Information Security Policy lets you publish your organization's security policy document to everyone on the platform. As an administrator, you upload a PDF and set how long it stays valid before you renew it. Users can open and read the policy at any time.

Find this setting at **Settings > Security > Information Security Policy**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



What the Information Security Policy is used for

Use this feature to make a single, authoritative security policy available across the platform. You upload the document, and the platform stores its submission date, its most recent renewal date, and its expiry date so your policy stays current.

The document must be a PDF.

Where users see the policy

Anyone signed in to the platform can open the policy from the **Policy** link in the footer. Selecting it opens a window that shows:

- The submission date.
- The renewal date.
- The expiry date.
- The policy document itself.

Administrators also see a **Renew** button in this window.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Is acceptance required?

No. The Information Security Policy is a reference document. Users can read it whenever they want, but they aren't prompted to accept it, and they aren't blocked from using the platform if they haven't opened it.

> **Note:**
>
> If you need users to actively accept a document before they can use the platform, use Terms of Service Agreements instead. The two features are separate: Terms of Service Agreements require acceptance and block access until the user accepts, while the Information Security Policy is read-only for users and never blocks them.

How Approval Expiry works

When you upload the policy, you set an **Information Security Policy Approval Expiry** period: 3, 6, 9, 12, 18, or 24 months. The platform uses this to calculate an expiry date, measured from the upload date.

This is an administrator reminder, not a user re-acceptance schedule. Here's what expiry does and doesn't do:

- **It doesn't affect users.** Users are never re-prompted, notified, or blocked when the policy expires.
- **It reminds administrators.** Within 14 days of the expiry date, and after it passes, administrators see a warning that the policy needs attention. This keeps your published policy current.

So the **Approval Expiry** period sets how long your document is considered current before you, as an administrator, need to renew or replace it. Users aren't asked to approve the policy based on this period.

Keep the policy current

Clear an expiry warning in one of two ways:

- **Upload a new policy.** Go to **Settings > Security > Information Security Policy**, upload a new PDF, and set a fresh Approval Expiry period. This replaces the current document.
- **Renew the current policy.** Open the policy from the footer and select **Renew**. This extends the expiry date without replacing the document.

Related information

- Security Settings

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Terms of Service Agreements

Terms of service agreements

Terms of Service Agreements let you present one or more documents that users must accept before they use the platform. As an administrator, you upload the documents, and the platform blocks access for any non-administrator user until they've accepted every current document.

Find this setting at **System Configuration > Others > Terms of Service Agreements**.

What Terms of Service Agreements are used for

Use this feature to enforce acceptance of legal or policy documents, such as an acceptable use policy or a service agreement. You can:

- Upload one or more documents.
- Preview or remove a document.
- Reset acceptances for a document so that everyone accepts it again.

PDF is the recommended file format. The acceptance page also renders HTML (`.html` or `.htm`) and plain-text (`.txt`) documents.

When you upload more than one document, users accept them all together in a single step.

Where users see the Terms of Service

After a non-administrator user signs in, the platform checks whether they've accepted every current document. If any are outstanding, the platform sends them to a dedicated **Terms of Service Agreements** page before they can reach any other part of the platform.

On that page, the user:

1. Reviews each document, shown in full on the page.
2. Selects **Accept All** to accept every outstanding document at once.

After the user accepts, the platform records their acceptance and takes them into the normal experience. The page doesn't appear again unless you add a new document or reset acceptances.

Is acceptance required?

Yes. Acceptance is required and blocking for non-administrator users. Until a user accepts every current document, they can't navigate to projects, workspaces, or settings. The page shows this notice:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> You must read and accept all Terms of Service agreements before you can use the platform.

A few important details:

- **Acceptance is per user and per document.** The platform tracks which documents each user has accepted.
- **Administrators aren't blocked.** So that you can keep managing the platform, administrators aren't redirected to the acceptance page, even when documents are outstanding.
- **New documents apply to everyone.** When you add a document, every non-administrator user is prompted to accept it the next time they sign in or navigate.

> **Note:**
>
> The platform records that a user accepted a document, but it doesn't store an acceptance time-stamp. If your organization needs a dated audit record of acceptance, keep that record outside the platform.

Reset acceptances

Resetting acceptances forces users to accept a document again —for example, after you publish a new version.

1. Go to **System Configuration > Others > Terms of Service Agreements**.
2. Find the document in the list.
3. Select the reset icon next to the document.
4. Confirm the reset.

The platform clears every user's acceptance of that document. The next time each non-administrator user signs in or navigates, they're prompted to accept it again.

Related information

- Others
- Information Security Policy
- System Configuration

Security settings

Configure critical security parameters for the entire platform. This includes managing **SAML Integration** for secure web application access via RBI, setting up **SIEM Integration** for centralized logging,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

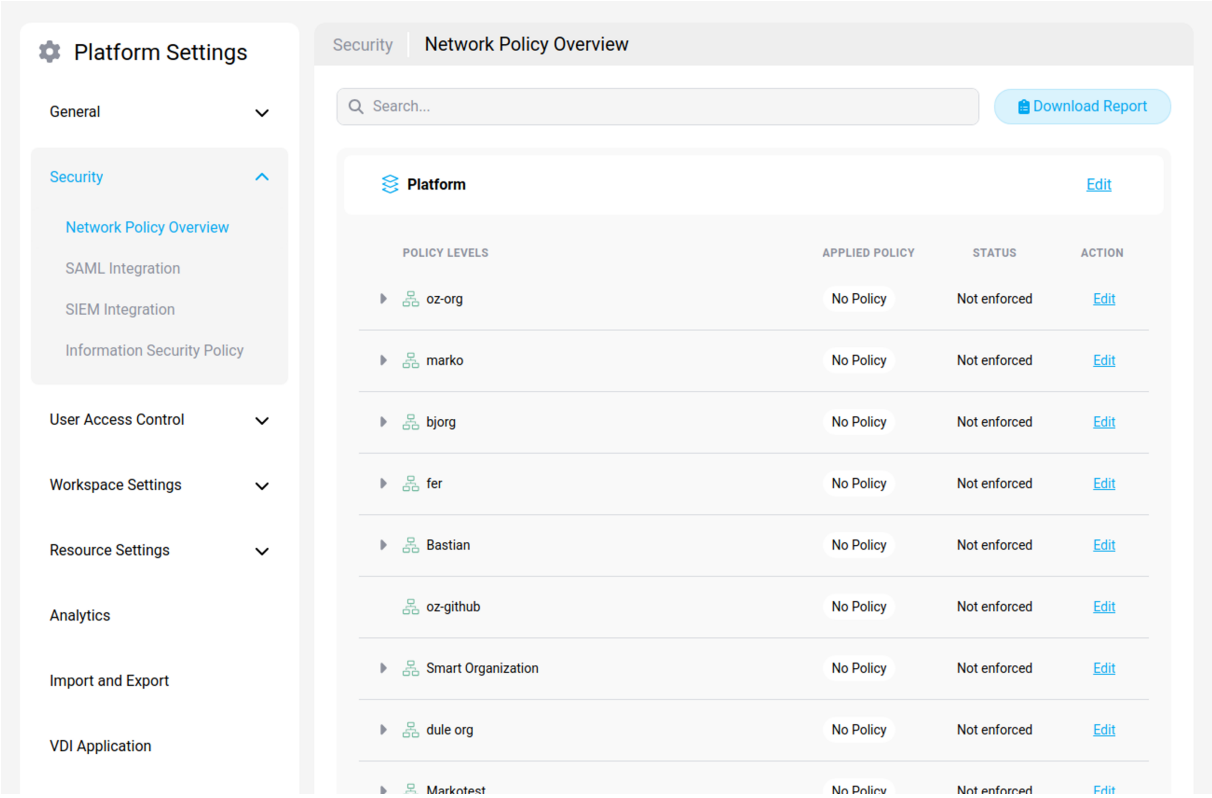
getting a **Network Policy Overview**, and establishing platform-wide **Information Security Policy** settings. These settings are essential for protecting platform resources and ensuring secure user access.

- Network Policy Overview
- SAML Integration
- SIEM Integration
- Information Security Policy

Network Policy Overview

Get a summary view of the network policies currently applied across the platform. This overview helps administrators quickly understand the existing network security configurations and rules at a high level.

For the policy model and how policies are enforced, see Network policies.



SAML Integration

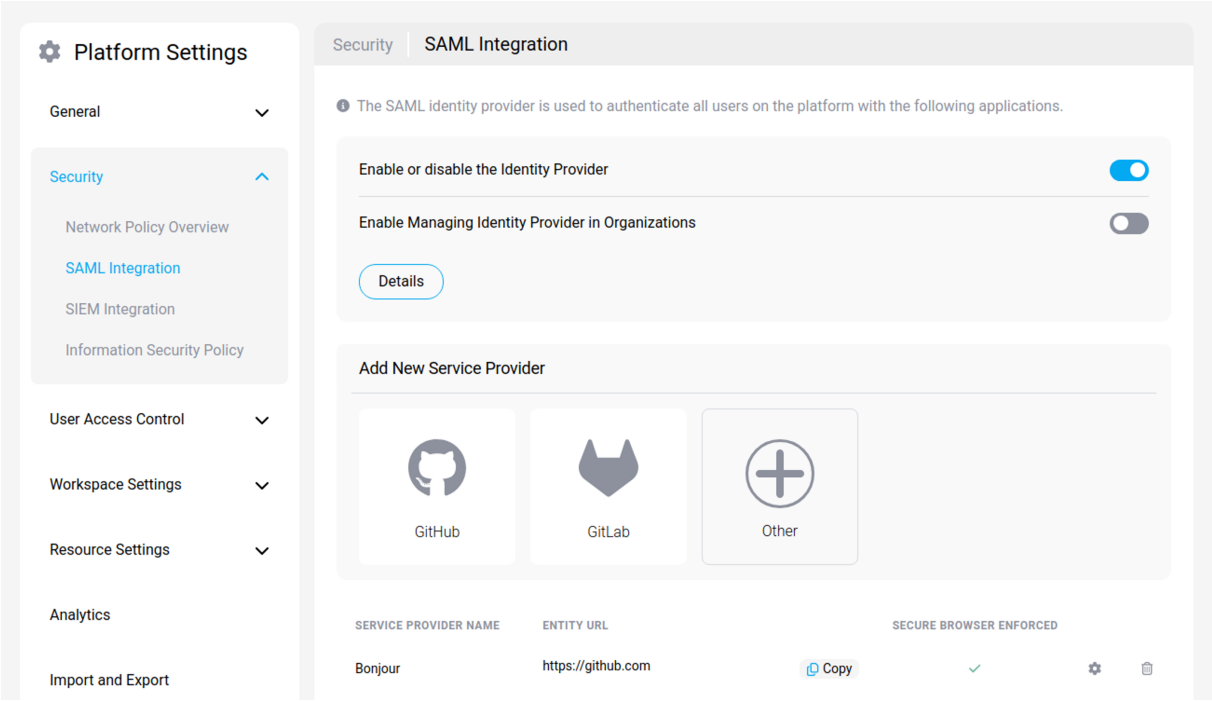
The **SAML Integration** section is responsible for authenticating all users on the platform when accessing web applications. Users access these Web Applications through Remote Browser Isolation (RBI),

© 1997–2026 Citrix Systems, Inc. All rights reserved.

known on the platform as the “Secure Browser”. The Secure Browser offers DLP-enabled access to any sensitive domains, such as GitHub, Jira, and GitLab. Users are restricted to accessing these Web Applications solely through the platform, prohibiting access via external browsers.

Administrators have the option to enable or disable a pre-configured identity provider. They can also allow organizations to oversee their own identity providers.

For how to configure a SAML identity provider, see SAML.

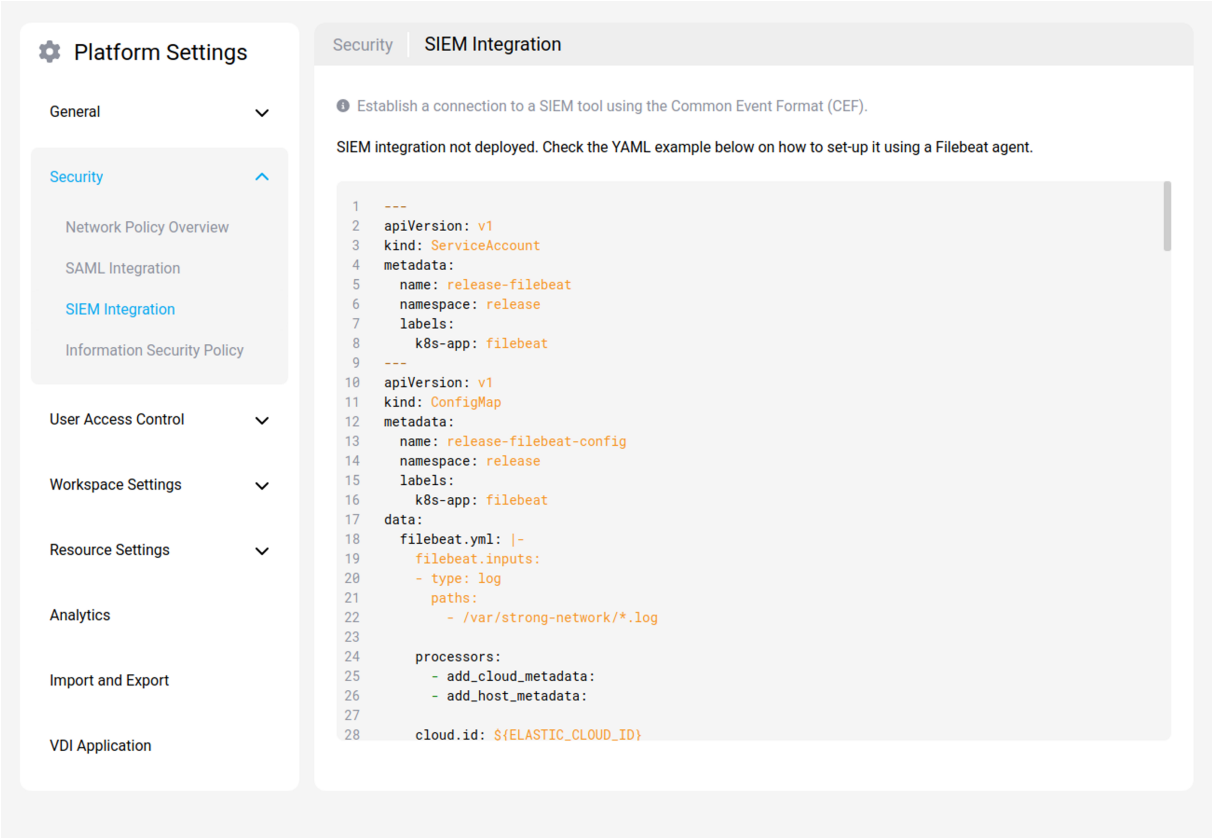


SIEM Integration

Configure the integration of the platform with your Security Information and Event Management (SIEM) system. This allows for forwarding logs and security events from the platform to your central SIEM for monitoring, analysis, and alerting.

For the setup procedure and the events that are forwarded, see SIEM integration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Information Security Policy

Publish your organization's information security policy so everyone on the platform can read it. You upload a PDF and set an approval expiry period that reminds you when to renew or replace it.

For where users see the policy, whether acceptance is required, and how the approval expiry period works, see Information Security Policy.

Operations

These pages cover running the platform day to day.

Page	What it covers
Licensing	Install a license, review what it covers, and turn licensed features on or off.
The SecurSpaces database	What the database stores, why it is critical, how it is deployed, and how to operate it safely.

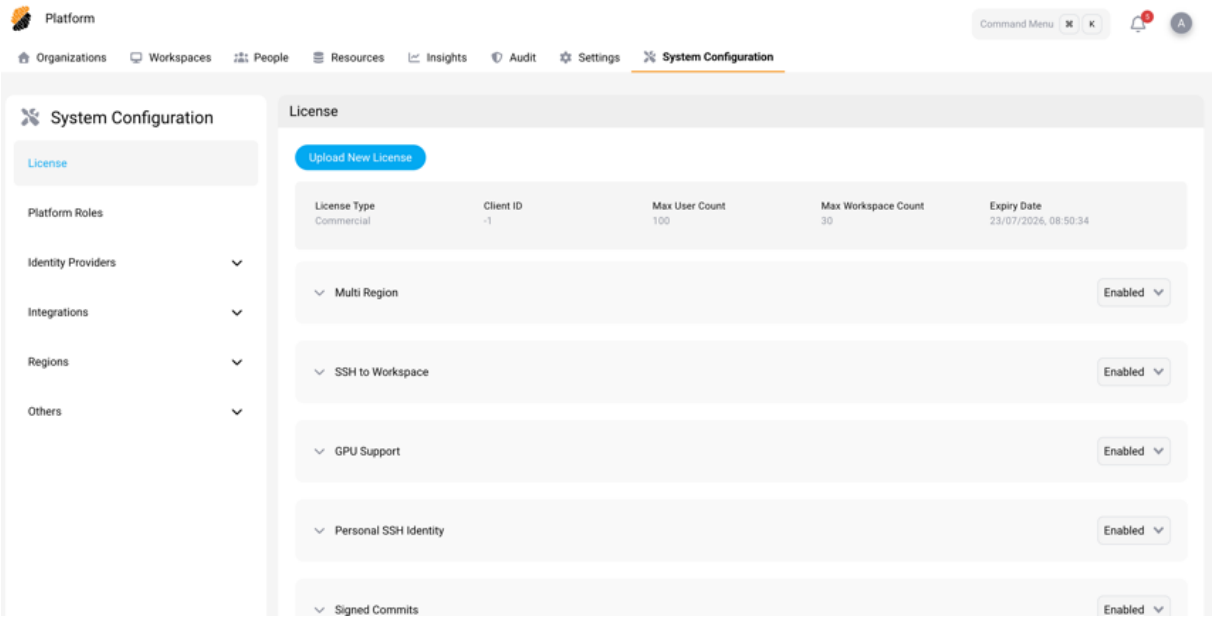
© 1997–2026 Citrix Systems, Inc. All rights reserved.

Page	What it covers
Database sizing	Expected footprint by data category, a working estimate, and provisioning guidance.
Back up the database	Configure Percona backups with Helm, keyless GCS access, mongodump snapshots, and hosted service backups.
Restore the database	Quiesce the platform, restore the data, repoint SecurSpaces, and validate.
Analytics	Download system reports and usage logs for the platform.
Workspace resource usage	Historical workspace CPU and memory data for rightsizing and trend analysis.

Licensing

Your license does two jobs: it lets your installation run, and it decides which features and limits apply. On the **License** page, you install your license and see exactly what it covers.

Find this page at **System Configuration > License**.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

Review your license

When a license is active, the summary row shows the details that matter most:

Field	Description
License Type	The kind of license, such as Internal, Commercial, Community Edition, or 1-Click VM.
Client ID	The identifier for your organization.
Max User Count	How many users you can register on the platform.
Max Workspace Count	How many workspaces the platform can hold.
Expiry Date	When the license expires.

Upload a new license

You can install your license two ways:

- **Online License** —The platform fetches and refreshes your license automatically from the Citrix License Server. Use this when the platform can reach the License Server.
- **File License** —You upload a license file yourself. Use this for air-gapped or offline installations that can't reach the License Server.

To install or replace a license:

1. Go to **System Configuration > License**.
2. Select **Upload New License**.
3. Read and accept the **Terms & Conditions**.
4. Choose how to install it:
    - For an online license, select **Online License**. The platform fetches and checks it from the License Server.
    - For a file license, select **File License** and choose your license file.
5. Confirm the upload.

The summary row updates to show your new license.

> **Note:**
>
> If you have both an online and a file license, the online one takes precedence.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Turn features on or off

The License page also lists the features your license covers. Each one shows its current status, which you can change:

- **Enabled** —Everyone can use the feature.
- **Disabled** —The feature is turned off.
- **Enabled for Selected Users** —Only the users or groups you pick can use it.
- **Enabled for Selected OS** —Only the operating systems you pick can use it, where that applies.

> **Important:**
>
> You can only manage features your license covers. Within a covered feature, you can change its status and widen the set of users or groups it applies to.

Here are the features you can manage from this page:

Feature	What it does
SSH to Workspace	Lets workspace owners connect to their workspaces over SSH. See SSH Access to Workspaces.
Multi Region	Turns on multi-region deployments. When it's on, the Regions area appears. See Deploy SecurSpaces in Multiple Regions.
Rest API	Opens access to the Citrix SecurSpaces™ REST API. See API.
Personal SSH Identity	Lets users add and manage multiple SSH identities in their account.
Signed Commits	Turns on signed Git commits from workspaces.
GPU Support	Turns on GPU-enabled workspaces. See Set up a GPU-enabled workspace on AKS.
Internal Dev User	Turns on internal developer user accounts.
Root Docker in Docker	Allows root Docker-in-Docker inside workspaces.
Persistent Container Images	Turns on persistent container images.
User Insights Monitoring	Turns on per-user insights monitoring.
Sysbox Container Runtime	Turns on the Sysbox container runtime.
VDI Application	Turns on the VDI application. See VDI Application.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Feature	What it does
Allow File Download	Lets users download files from workspaces.

Online licensing and renewal

An online license keeps itself current, so there is normally nothing to do after the first install.

Behavior	Detail
Validity period	Each online license is valid for 21 days
Automatic renewal	The platform contacts the license server several times a day to renew
Retry	A failed renewal, for example during a network outage, is retried automatically

The platform reaches the license server at `api.enterprise.strong.network` on port 443. See Network destinations.

Expiration warnings

If the license cannot be renewed, the console warns before anything stops working.

- When validity falls to 14 days without a successful renewal, a warning banner appears at the top of the
  console.
- The banner names the problem and counts down to the expiration date.

> **Note:**
>
> For air-gapped environments, either allow the SecurSpaces controller through your firewall to the license
> server, or request an offline license for fully disconnected operation.

Related information

- System Configuration
- Regions
- Network destinations

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The SecurSpaces database

October 01, 2026

Citrix SecurSpaces™ stores its platform configuration and state in a MongoDB database. This database is the
system of record for your deployment. If it is lost or corrupted, the platform cannot operate, even when every
other component is healthy.

This section is for platform administrators who deploy and operate SecurSpaces. It assumes you are familiar
with Kubernetes, Helm, and your MongoDB hosting model.

Task	Page
Estimate how large the database will be	Database sizing
Back it up, on a schedule and before changes	Back up the database
Recover from a backup	Restore the database

What the database stores

The SecurSpaces database (default name `strong-network`) holds the metadata and state that define your platform. It does **not** store the contents of developer workspaces.
Workspace file systems, Docker volumes, and source code live on Kubernetes
Persistent Volume Claims (PVCs) and in your version control system, not in MongoDB.

The database stores, among other things:

- **Organizations, projects, and settings** —organization, project, platform, and region configuration, section ordering, and setup checklists.
- **People and access** —users, groups, roles, permissions, and identity-provider mappings.
- **Workspace definitions and state** —workspace specifications, lifecycle status, schedules, SSH sessions, and the metadata that links a workspace to its project, owner, and persistent volumes.
- **Data mounts** —data buckets and workspace mount-point definitions.
- **Integrations** —connections to source control and other external services, plus their OAuth and access tokens.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Credentials and secrets** —API keys and stored credentials. By default these are **encrypted and stored in the database**. If you configure HashiCorp Vault, secrets are stored in Vault instead of the database (see the next section).
- **Audit and event data** —audit-trail and event-log records, plus usage, reporting, and compliance aggregates used by Insights and SIEM integration.

> **Note**
> The exact collection layout is an internal implementation detail and can change between releases. Treat the database as an opaque unit for backup and restore — always back up and restore the whole database, not individual collections.

Where secrets live: database vs. HashiCorp Vault

By default, SecurSpaces encrypts secrets and credentials and stores them in the database, using an internal encryption key supplied at install time.

If you configure HashiCorp Vault (`platform.hashicorpVault` in the Helm values), SecurSpaces stores secrets in Vault instead of the database.

This distinction matters for backups: when you use Vault, a database backup alone does **not** capture your secrets —you must also back up Vault. When you do not use Vault, the database backup contains the (encrypted) secrets, and you must protect both the backup and the encryption key.

Why the database is critical

The database is the single source of truth for platform configuration and state. SecurSpaces services are stateless by design and read their configuration from MongoDB on startup and during operation. As a result:

- **No database, no platform.** If MongoDB is unavailable, SecurSpaces services cannot start or serve requests, even when Kubernetes, networking, and storage are healthy.
- **Configuration loss is unrecoverable without a backup.** Organizations, projects, users, integrations, and workspace metadata exist only in the database. Losing it means rebuilding your platform configuration from scratch.
- **Workspace access depends on it.** Even though workspace file systems live on PVCs, the metadata that maps a developer to their workspace and its volumes lives in the database. Without that metadata, existing workspaces cannot be reattached.

For these reasons, treat the database as a tier-1, business-critical asset: run it with redundancy, back it up on a schedule, and test your restore procedure.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How the database is deployed

SecurSpaces supports three MongoDB hosting models. Your backup strategy depends on which one you use.

Hosting model	When it is used	Who operates MongoDB
Internal MongoDB (Percona)	Deployed by SecurSpaces into your cluster using the Percona Operator. Used for proof of concept and evaluation, and in production where a managed service is not available.	You
External / hosted MongoDB	**Recommended for production.** SecurSpaces connects to an externally managed deployment such as MongoDB Atlas.	The managed service provider
Bitnami MongoDB (legacy)	Older internal deployments before the move to Percona. Being phased out.	You

The updated chart bundles **Percona Operator for MongoDB 1.23.0**, **Percona Server for MongoDB 7.0.37-20**, and **Percona Backup for MongoDB (PBM) 2.15.0**. The operator manages the database as a
`PerconaServerMongoDB` custom resource with `crVersion: 1.23.0`.

By default the internal deployment uses:

- A single-member replica set (`rs0`, size 1). For production, increase the replica-set size to at least 3 for high availability.
- A 10 GiB persistent volume for data.
- Resource requests of 500m CPU / 1 GiB memory and limits of 2000m CPU / 4 GiB memory.

> **Note:**
>
> A managed MongoDB service is the simplest route to production, because the provider handles redundancy,
> backups, and scaling for you.
>
> The internal database is fully supported in production, and is the right choice for air-gapped deployments and environments where a managed service is not available. The difference is one

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> of ownership
> rather than capability: **you configure and operate its resilience.** The defaults are set for a quick start —a single-member replica set, with backups available but not switched on —so plan for the following before you rely on it:
>
> - A replica set with more than one member, so that the loss of a pod or node does not take the platform
>   down. See Operating principles.
> - Backups configured, scheduled, and restore-tested. See
>   Back up the database.
> - Resources sized for your own load rather than the defaults. See
>   Sizing.

Internal Percona upgrades and backups

Upgrades from SDS 2026.9 (operator 1.20.1) must proceed one minor version at a time. After `sds-cli upgrade`,
run the supplied `percona-upgrade/upgrade-percona.sh` to back up the database, upgrade through 1.21.2 and
1.22.0, and apply the 1.23 CRDs. Then run the printed Helm command to complete the upgrade to 1.23.0.
Helm 3 does not upgrade existing CRDs; the script applies them explicitly. See
Upgrade for the full procedure and recovery commands.

Backups remain disabled by default. Configure `perconaServerMongoDB.backup` and `perconaServerMongoDB.serviceAccount` in Helm values to enable backups and configure the database service
account. Operator 1.23 supports keyless GCS backups through GKE Workload Identity. Grant bucket access to
both the database and operator Kubernetes service accounts so backups, restores, and retention cleanup
work. See Back up the database.

Internal Percona MongoDB credentials

The Helm chart stores internal Percona MongoDB credentials in the
`<release>-psmdb-secrets` Kubernetes Secret. Leave the password values in
`perconaServerMongoDB.adminUsers` empty for normal operation. On the first install, the chart generates the
passwords. On later server-side Helm upgrades, the chart reuses the values from the live Secret

© 1997–2026 Citrix Systems, Inc. All rights reserved.

instead of
generating new ones.

This behavior prevents routine upgrades from rotating the database passwords unexpectedly. If you explicitly
set a password in the values file, that configured value takes precedence and rotates the corresponding
credential. When the database administrator credential changes, SecurSpaces rolls the platform services that
read it so they pick up the updated Secret.

> **Important:**
>
> `lookup` can read the existing Secret only during a server-side Helm operation against the cluster.
> Client-side rendering, including `helm template`, client-side GitOps rendering, and `--dry-run=client`,
> cannot read the live Secret. In those modes, empty password values can still render as new generated
> passwords. Use server-side Helm upgrades for the internal Percona deployment, or pin the Percona database
> passwords in your values file.

Operating principles

- Run MongoDB with redundancy in production —a managed service, or a Percona replica set with more
  than one member that you operate —never a single member.
- Enable continuous / point-in-time backups, not only periodic snapshots.
- Store backups off-cluster and in a separate failure domain.
- Encrypt backups at rest and in transit, and restrict access to backup storage.
- Back up HashiCorp Vault alongside the database when Vault stores your secrets.
- Always back up before upgrades, migrations, and major configuration changes.
- Monitor database size. In-database audit data auto-expires after 360 days; forward
  events to your SIEM if you need retention beyond that window.
- Test restores on a schedule and document the runbook.

Related information

- Architecture
- System requirements

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Configure an external database
- Upgrade

Database sizing

August 19, 2026

Use these figures to size the Citrix SecurSpaces™ database for initial deployment, then verify against actual usage during your PoC and in production.

Because the database holds metadata and not workspace contents, it stays small relative to your workspace
storage. The largest growing category is **audit and event data**, which SecurSpaces persists in the same
database.

What retention actually bounds

The database does not grow without limit. SecurSpaces applies MongoDB TTL indexes, and the retention window
differs by data type:

Data	Retention	Effect on size
Audit and security events	360 days	The dominant long-lived category
Website activity logs	360 days	Grows with user activity
Error logs	30 days	Negligible at steady state
Raw Kubernetes usage samples	14 days	Bounded regardless of workspace count
Project activity, debug logs, cron records	7 days	Negligible

This matters when sizing for a larger deployment. Per-workspace usage sampling is the data that scales
hardest with workspace count, and it is discarded after 14 days, so it does not accumulate. Network events
are also stored as summaries with a request count rather than one document per request.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The result is that the database tends toward a steady state of about one year of audit and event data, plus
your relatively static configuration.

Estimated footprint

The following estimate is for an environment of **500 developers using 500 workspaces**. For smaller deployments, including a proof of concept, see Smaller deployments.

Data category	Approximate footprint	Notes
Organizations, projects, settings	< 100 MB	Scales with project count, not workspace count.
Users, groups, roles	< 250 MB	Scales with user count.
Workspace and template metadata	< 500 MB	Scales with workspace and template-version count.
Credentials and API keys	< 250 MB	Lower if HashiCorp Vault stores your secrets.
Audit, event, usage, and reporting records	5–25 GB (steady state)	Largest category, bounded by the 360-day expiry. Varies most with how active the platform is.
Indexes and overhead	25–50% of data size	MongoDB indexes plus working-set overhead.

Working estimate: configuration and metadata for 500 developers and 500 workspaces is on the order of
1 GB. With a year of audit and event data plus indexes, plan for **10–40 GB** of actual data.

The spread is wide because activity level, not user count, drives the largest category. A platform where every workspace runs all day with network monitoring enabled sits at the top of that range; one used for a
few hours a day sits near the bottom.

Provisioning guidance

- **Storage:** Provision **at least 100 GiB** for the data volume at this scale. The headroom covers growth,
  the replication oplog, indexes, and the temporary space used during compaction and restore. The 10 GiB
  chart default suits only small or short-lived PoC environments.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Memory:** Size MongoDB so the working set —frequently accessed documents plus indexes — fits in the
  WiredTiger cache. Provision to the **4 GiB limit** rather than the 1 GiB request, and monitor cache pressure. Audit queries over a long window are what push it.
- **CPU:** Provision to the **2 core limit** rather than the 500m request. Steady-state load is low; the peaks are reporting queries, restores, and compaction.
- **Growth control:** The 360-day expiry is fixed and not configurable. For longer retention, forward events
  to your SIEM. The in-database copy still expires at 360 days. See
  SIEM integration.

> **Note**
>
> These are planning figures for a typical workload of this size. Every environment is different, and some
> have a higher footprint. Use them for initial sizing, then monitor actual database size and growth during
> your PoC and in production, and adjust accordingly.

Smaller deployments

Environment	Estimated data	Provision
50 developers, 50 workspaces (PoC)	1–3 GB	At least 20 GiB
100 developers, 100 workspaces	2–6 GB	At least 20 GiB
500 developers, 500 workspaces	10–40 GB	At least 100 GiB

The 10 GiB chart default is enough only for a short-lived proof of concept. A pilot that runs for months accumulates audit data the whole time, and none of it expires until day 360 —so a proof of concept that
starts comfortably inside 10 GiB can still outgrow it before anyone decides whether to buy. Provisioning
20 GiB from the start avoids resizing a volume mid-evaluation.

Note that storage does not scale with users in the way the data does. Most of the 20 GiB is headroom for the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

oplog, indexes, and the temporary space a restore needs, and those requirements barely change between 50 and
100 developers.

Related information

- The SecurSpaces database
- Sizing
- System requirements

Back up the database

October 01, 2026

Back up the Citrix SecurSpaces™ database before upgrades, migrations, and major configuration changes,
and on a recurring schedule in production. If you use HashiCorp Vault for secrets, back up Vault on the
same schedule.

On-premises and self-managed (internal Percona MongoDB)

> **Important:**
>
> Backups are **disabled by default** on the internal database. If you run it in production, enabling, scheduling, and restore-testing them is your responsibility. See
> The SecurSpaces database.

Use **Percona Backup for MongoDB (PBM)** as your primary, scheduled backup mechanism. Use a manual `mongodump` only as a secondary, point-in-time snapshot before a specific change.

Primary: Percona Backup for MongoDB (PBM)

PBM is the backup tool built into the Percona Operator. It supports on-demand and scheduled logical and physical backups and point-in-time recovery to an external object store (Amazon S3, Azure Blob Storage, Google Cloud Storage, or any S3-compatible store).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The chart with **Percona Operator 1.23.0** ships **PBM 2.15.0**. Backups remain disabled by default. Configure them in your Helm values with `perconaServerMongoDB.backup`, including `enabled`, storage, schedules, retention, and point-in-time recovery (PITR). Configure the database Kubernetes service account through `perconaServerMongoDB.serviceAccount`. The chart renders these settings into the Percona resources.

Keep backup settings in the values used for subsequent Helm upgrades. If you previously edited `spec.backup` directly, carry those settings into `perconaServerMongoDB.backup` before applying the new chart. Do not rely on custom-resource edits as the source of configuration for future upgrades.

For an existing deployment on operator 1.20.1, first complete the stepwise upgrade to 1.23.

Follow the Percona Operator backup documentation for the authoritative reference:

- Backups overview: https://docs.percona.com/percona-operator-for-mongodb/backups.html
- Configure storage: https://docs.percona.com/percona-operator-for-mongodb/backups-storage.html
- On-demand backups: https://docs.percona.com/percona-operator-for-mongodb/backups-ondemand.html
- Scheduled backups: https://docs.percona.com/percona-operator-for-mongodb/backups-scheduled.html

Example: enable PBM and back up to Amazon S3 This worked example follows the Percona Operator storage guide. Replace `<release>` with your SecurSpaces Helm release name and `<namespace>` with your SecurSpaces namespace. Run `kubectl get psmdb -n <namespace>` to confirm your cluster name —for SecurSpaces it is `<release>-psmdb-db`.

1. Create a Secret with your S3 credentials. Save as `backup-s3-secret.yaml` (the values are base64-encoded automatically when you use `stringData`):

```
apiVersion: v1
kind: Secret
metadata:
  name: sds-backup-s3
type: Opaque
stringData:
  AWS_ACCESS_KEY_ID: "<your-access-key-id>"
  AWS_SECRET_ACCESS_KEY: "<your-secret-access-key>"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
kubectl apply -f backup-s3-secret.yaml -n <namespace>
```

2. Enable backups and define the storage in your Helm values. Add the following to the values file used for your deployment. The storage entry references the Secret created above:

```
perconaServerMongoDB:
  backup:
    enabled: true
    storages:
      s3-primary:
        type: s3
        s3:
          bucket: <your-backup-bucket>
          region: <your-region>        # for example, eu-central-1
          prefix: sds/strong-network   # optional sub-folder in the
              bucket
          credentialsSecret: sds-backup-s3
```

Apply the values using your normal Helm upgrade command and the same chart version. For example:

```
helm upgrade <release> ./ninjahchart-<version>.tgz -n <namespace> -f <
    your-values.yaml>
```

> **Note**
> On AWS EKS you can grant bucket access with an IAM role for the service account (IRSA) and omit `credentialsSecret`. For Azure Blob or Google Cloud Storage, use the `azure` or `gcs` storage type instead of `s3`. For keyless GCS access, see GCS Workload Identity.

3. Take an on-demand backup. Create a `PerconaServerMongoDBBackup` resource that references your cluster and storage. Save as `sds-backup.yaml`:

```
apiVersion: psmdb.percona.com/v1
kind: PerconaServerMongoDBBackup
metadata:
  name: sds-backup-2026-07-02
  namespace: <namespace>
spec:
  clusterName: <release>-psmdb-db
  storageName: s3-primary
  type: logical                        # logical is the default;
     physical is also supported
```

```
kubectl apply -f sds-backup.yaml
```

4. Verify the backup completed. Track the backup resource until `STATUS` is `ready`:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
kubectl get psmdb-backup -n <namespace>
```

```
NAME                    CLUSTER              STORAGE      DESTINATION
                              TYPE      STATUS   AGE
sds-backup-2026-07-02   <release>-psmdb-db   s3-primary   s3://<bucket>/
    sds/strong-network/... logical   ready    2m
```

5. (Recommended) Add a schedule and point-in-time recovery. For ongoing protection, add a backup task and enable PITR in the same `perconaServerMongoDB.backup` section of your values file. Keep the storage definition from the previous step:

```
perconaServerMongoDB:
  backup:
    enabled: true
    pitr:
      enabled: true                    # continuous point-in-time
         recovery
    tasks:
      - name: daily-s3
        enabled: true
        schedule: "0 2 * * *"          # every day at 02:00 (cron)
        storageName: s3-primary
        retention:
          type: count
          count: 7                     # retain the 7 most recent
             scheduled backups
          deleteFromStorage: true
```

Apply the updated values with your Helm upgrade command and confirm new backups appear on schedule. As a best practice, store backups in a different failure domain from the cluster —a separate cloud region or account —so that a cluster or region failure does not also destroy your backups.

> **Note**
> Reapplying the same Helm values preserves the backup configuration. In GKE validation, a second Helm upgrade with unchanged values caused no restarts and backups continued.

GCS Workload Identity

Percona Operator **1.23.0** supports keyless backups to Google Cloud Storage (GCS) using **Workload Identity Federation for GKE**. Use the native `gcs` storage type and omit `gcs.credentialsSecret` so PBM uses Application Default Credentials. No service account JSON key or HMAC key is required.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Prepare GKE and bucket access You need a GKE cluster with Workload Identity enabled, the GKE metadata server enabled on the node pools
running the database and operator pods, a GCS backup bucket, and permission to manage that bucket's IAM
policy. See Google's Workload Identity setup guide.

Use `perconaServerMongoDB.serviceAccount` in the chart values to configure the database service account.
Inspect the target chart for its service-account options before setting them:

```
helm show values ./ninjahchart-<version>.tgz
```

Grant bucket access to **both** Kubernetes service accounts: the one used by the database pods and the one
used by the Percona operator. PBM needs access for backup and restore; the operator also needs access to
delete expired backups from GCS. Discover the accounts from the running pods rather than assuming names:

```
kubectl get pods -n <namespace> \
  -o custom-columns='POD:.metadata.name,SERVICE_ACCOUNT:.spec.
     serviceAccountName'
```

The following example grants bucket-level object access directly to the two Kubernetes service-account
principals. Replace the placeholders with the GKE cluster project's ID and number, your namespace, bucket,
and the actual account names. `roles/storage.objectUser` allows object reads, writes, listing, and deletion.

```
PROJECT_ID="<gke-project-id>"
PROJECT_NUMBER="<gke-project-number>"
NAMESPACE="<namespace>"
GCS_BUCKET="<backup-bucket>"
DATABASE_KSA="<database-service-account>"
OPERATOR_KSA="<operator-service-account>"
POOL="projects/${
 PROJECT_NUMBER }
 /locations/global/workloadIdentityPools/${
 PROJECT_ID }
 .svc.id.goog"

for KSA in "$DATABASE_KSA" "$OPERATOR_KSA"; do
  gcloud storage buckets add-iam-policy-binding "gs://${
 GCS_BUCKET }
 " \
    --member="principal://iam.googleapis.com/${
 POOL }
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
 /subject/ns/${
 NAMESPACE }
 /sa/${
 KSA }
 " \
    --role="roles/storage.objectUser"
done
```

This direct-principal configuration does not require a Google service account key or service-account impersonation. If the operator runs in a different namespace, use that namespace in its IAM principal. If you change the database service account in Helm values, grant access to the new account as well.

Configure backups in Helm values Merge this example into your deployment values alongside `perconaServerMongoDB.serviceAccount`.
It enables logical backups every 10 minutes, continuous PITR oplog capture, and retention of two scheduled
backups. Adjust the schedule and retention to your recovery requirements:

```
perconaServerMongoDB:
  backup:
    enabled: true
    storages:
      gcs-wi:
        type: gcs
        gcs:
          bucket: <backup-bucket>
          prefix: sds/mongodb
    pitr:
      enabled: true
    tasks:
      - name: gcs-every-10-minutes
        enabled: true
        schedule: "*/10 * * * *"
        storageName: gcs-wi
        type: logical
        retention:
          type: count
          count: 2
          deleteFromStorage: true
```

Apply the values with your normal Helm upgrade command after completing the Percona 1.23 upgrade.
If migrating from key-based GCS access, remove `credentialsSecret` from the storage's `gcs` settings in
your values and verify it is absent from the resulting custom resource. A configured credentials Secret takes precedence over Workload Identity.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Verify backups, retention, and restore

1. Create the on-demand `PerconaServerMongoDBBackup` shown in the S3 example above, using a unique name
   and `storageName: gcs-wi`. Wait for `ready` and confirm that the backup is present in the bucket.
2. Confirm scheduled backups complete and continuous oplog chunks appear for PITR. PITR requires a
   completed base backup and oplog coverage for the chosen recovery time.
3. Confirm retention deletes expired backups from both Kubernetes and GCS. If objects remain in GCS,
   check bucket permissions for the **operator's** service account as well as the database account.
4. Keep the same backup values for subsequent Helm upgrades and verify backups continue afterward.
5. Test restoring into a replacement deployment, including a restore to a chosen second when using
   PITR. See Restore the database.

This flow was validated on GKE with direct bucket grants to both Kubernetes service accounts, on-demand
and scheduled backups, PITR, retention count 2, and a repeated Helm upgrade. Recovery was also validated
after deleting the deployment and volumes, reinstalling, and restoring to a chosen second; the recovered
data matched and the platform was operational.

For authentication and storage details, see Percona's GCS backup guide.

Secondary: logical snapshot with `mongodump`

The Percona upgrade kit
takes a `mongodump` archive automatically before changing versions and includes matching restore and
delete scripts. Keep this pre-upgrade archive until platform validation is complete. It is separate from your scheduled PBM backups and PITR storage.

`mongodump` / `mongorestore` are the official MongoDB database tools, and SecurSpaces uses them in its own replica-set migration procedure, so this is a supported method —not
a workaround.

It is safe for **ad hoc, quiesced snapshots** (for example, immediately before an
upgrade) when you follow these rules:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Quiesce writes before you dump, or dump from a secondary member.** `mongodump` is only guaranteed consistent across collections when the database is not being written to. The simplest way to get a clean snapshot is to scale the SecurSpaces services to `0` first (the same quiesce step used in the restore procedure), then dump. `mongodump` cannot produce a cross-collection point-in-time snapshot of a single database on a live system —for online, point-in-time backups, use PBM instead.
- **Store the archive off-cluster** and protect it (it contains encrypted secrets unless you use Vault).
- **Restore only into a quiesced platform** (see the restore section), and never use destructive flags such as `--drop` against a live, in-use database.

It is **not** a replacement for scheduled, point-in-time disaster recovery —PBM (internal) or your managed service's backups remain the primary strategy. Use `mongodump` to capture a known-good snapshot before a risky change.

```
# Identify the Percona pods
kubectl get psmdb
kubectl get pods -l app.kubernetes.io/name=percona-server-mongodb

# Confirm which pod is primary (look for "[direct: primary]" in the prompt)
kubectl exec -it <pod>-rs0-0 -- mongosh \
  --authenticationDatabase admin --username <admin-user> --password <admin-password>

# Dump the strong-network database from the primary (quiesce writes first)
kubectl exec -it <primary-pod> -- mongodump \
  --db strong-network \
  --username <admin-user> --password <admin-password> \
  --authenticationDatabase admin \
  --gzip --archive=/tmp/strong-network-backup.gz

# Copy the archive off the pod
kubectl cp <primary-pod>:/tmp/strong-network-backup.gz strong-network-backup.gz
```

For tool reference, see the MongoDB documentation for `mongodump`: https://www.mongodb.com/docs/database-tools/mongodump/

Hosted MongoDB services

When you use a managed MongoDB service, the provider supplies the backup tooling. Enable and configure it according to your recovery objectives (RPO/RTO).

SecurSpaces supports hosted MongoDB services on all three major hyperscalers (AWS, Azure, and Google Cloud). MongoDB Atlas is available on all three and is the option the SecurSpaces

© 1997–2026 Citrix Systems, Inc. All rights reserved.

deployment guides reference. Use Atlas Cloud Backups with continuous (point-in-time) backup enabled:

- Atlas Cloud Backups overview: https://www.mongodb.com/docs/atlas/backup/cloud-backup/overview/
- Atlas restore overview: https://www.mongodb.com/docs/atlas/backup/cloud-backup/restore-overview/

If you use a cloud-native, MongoDB-compatible service instead of Atlas, enable its backup feature using the provider documentation below.

Platform	Service	Official backup documentation
AWS	Amazon DocumentDB, or MongoDB Atlas on AWS	Backing up and restoring in Amazon DocumentDB ·Atlas Cloud Backups
Azure	Azure Cosmos DB for MongoDB, or MongoDB Atlas on Azure	Online backup and on-demand restore (Cosmos DB) ·Reliability in Azure Cosmos DB for MongoDB (vCore)
GCP	MongoDB Atlas on Google Cloud	Atlas Cloud Backups. Google Cloud has no first-party managed MongoDB.

General guidance for hosted services:

- Turn on **continuous / point-in-time** backup, not just daily snapshots, to minimize data loss.
- Set a retention period that meets your compliance and recovery requirements.
- Periodically test restoring a snapshot into a non-production cluster.

Related information

- Restore the database
- The SecurSpaces database
- Upgrade

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Restore the database

October 01, 2026

Restoring the Citrix SecurSpaces™ database requires a maintenance window: the platform must be quiesced
so services do not write to the database mid-restore. The procedure for step 2 depends on your hosting
model.

Step 1: Quiesce the platform

Scale the SecurSpaces services down so nothing writes to the database during the restore. The MongoDB deployment itself stays running.

```
# Stop the SecurSpaces application services (leave MongoDB running)
kubectl scale deployment \
  <release>-central-service <release>-workspace-api <release>-
     coordinator \
  <release>-frontend <release>-proxy \
  --replicas=0 -n <namespace>
```

> **Note**
> These are the five service Deployments on a primary SecurSpaces platform. An external region deployment runs only `<release>-workspace-api` and `<release>-proxy`. Confirm what is
> present with `kubectl get deploy -n <namespace>` before scaling.

Step 2: Restore the data

Internal Percona MongoDB —from a PBM backup

Use the Percona Operator restore workflow: create a `PerconaServerMongoDBRestore` resource that references the backup by name (or a point in time) and wait for the
operator to complete the restore. Save as `sds-restore.yaml`:

```
apiVersion: psmdb.percona.com/v1
kind: PerconaServerMongoDBRestore
metadata:
  name: sds-restore-2026-07-02
  namespace: <namespace>
spec:
  clusterName: <release>-psmdb-db
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  backupName: sds-backup-2026-07-02        # the
     PerconaServerMongoDBBackup to restore
```

```
kubectl apply -f sds-restore.yaml
kubectl get psmdb-restore -n <namespace>   # wait for STATUS: ready
```

For point-in-time restores and full options, see the Percona restore documentation:
https://docs.percona.com/percona-operator-for-mongodb/backups-restore.html

For GCS backups with Workload Identity, configure the replacement deployment with access to the same
bucket and prefix. Grant bucket access to both its database and operator Kubernetes service accounts,
and use the same `perconaServerMongoDB.backup` and `perconaServerMongoDB.serviceAccount` values as
appropriate for the replacement environment. See
GCS Workload Identity.

After losing the deployment and its volumes, the old Kubernetes backup resource might no longer exist.
Use Percona's restore-from-storage procedure rather than referencing a missing `backupName`. For PITR,
choose a time covered by a completed base backup and the stored oplog chunks. GKE validation included
deleting the deployment and volumes, reinstalling, and recovering to a chosen second with matching data.

Percona upgrade kit backup

For an archive taken by `percona-upgrade/upgrade-percona.sh`, use the matching restore script from the
installer kit after quiescing the platform. Run it from the shared directory where the kit and archive are available:

```
./percona-upgrade/restore-percona-backup.sh <namespace> <release> <
   backup>.archive.gz
```

For example, with namespace **`default`** and release `release`:

```
./percona-upgrade/restore-percona-backup.sh default release <backup>.
   archive.gz
```

The script restores **data only**. It does not downgrade the operator, CRDs, MongoDB, or PBM. Continue with steps 3 and 4 below to bring the platform back and validate it. Once validated, remove the up-

© 1997–2026 Citrix Systems, Inc. All rights reserved.

grade
backup from the directory where it was created:

```
./percona-upgrade/delete-percona-backup.sh
```

For the complete upgrade sequence, see Upgrade.

Internal Percona MongoDB —from a mongodump archive

Target the **primary** replica-set member:

```
# Identify the current primary
kubectl get psmdb
kubectl exec -it <pod>-rs0-0 -- mongosh \
  --authenticationDatabase admin --username <admin-user> --password <
     admin-password>

# Copy the archive to the primary pod
kubectl cp strong-network-backup.gz <primary-pod>:/tmp/strong-network-
    backup.gz

# Restore the strong-network database into the quiesced platform
kubectl exec -it <primary-pod> -- mongorestore \
  --nsInclude='strong-network.*' \
  --username <admin-user> --password <admin-password> \
  --authenticationDatabase admin \
  --gzip --archive=/tmp/strong-network-backup.gz
```

If you are restoring into a freshly provisioned database (for example, after moving from a single MongoDB pod to a Percona replica set), you may also need to recreate the application database user so SecurSpaces can connect:

```
kubectl exec -it <primary-pod> -- mongosh \
  --authenticationDatabase admin --username <admin-user> --password <
     admin-password> \
  --eval "db.createUser({
 user: '<app-user>', pwd: '<app-password>', roles: [{
  db: 'strong-network', role: 'readWrite'  }
 ], mechanisms: ['SCRAM-SHA-1'] }
 );" \
  strong-network
```

Hosted MongoDB —restore with the provider tooling

Restore the snapshot or point in time using the provider's console or CLI, then note the connection endpoint of the restored deployment (it is often a **new** host):

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **MongoDB Atlas** —restore a snapshot or point in time to the existing cluster or to a new cluster: https://www.mongodb.com/docs/atlas/backup/cloud-backup/restore-overview/
- **Amazon DocumentDB** —restore from a snapshot or point in time to a new cluster: https://docs.aws.amazon.com/documentdb/latest/developerguide/backup_restore.html
- **Azure Cosmos DB for MongoDB** —restore to a new account: https://learn.microsoft.com/en-us/azure/cosmos-db/restore-account-continuous-backup

Step 3: Point SecurSpaces at the restored database

SecurSpaces reads its MongoDB connection settings from Helm values, which the chart renders into `MONGO_DB_HOST`, `MONGO_DB_PORT`, `MONGO_DB_NAME`, `MONGO_DB_USER`, and `MONGO_DB_PASSWORD` environment variables on the service pods. You change the target database by updating those values and running `helm upgrade`.

If you restored to a hosted service or a new endpoint (external MongoDB):

1. In your Helm values file, make sure SecurSpaces is in external-MongoDB mode and update the connection details:

```yaml
mongodbAuth:
  source: "external"

platform:
  deployPerconaMongoDB: false
  deployBitnamiMongoDB: false
  externalMongodb:
    hostName: "<restored-cluster-host>"     # new endpoint from the restore
    port: "27017"
    protocol: "mongodb+srv"                 # or "mongodb"
    additionalParameters: "retryWrites=true&w=majority"
    authMechanism: "SCRAM"                  # or "MONGODB-X509"
    auth:
      database: "strong-network"
      username: "<db-user>"
      password: "<db-password>"
```

   For certificate-based (X.509) authentication, set the certificate fields instead of `username`/`password`. See Configure external MongoDB with X.509 authentication.

2. Apply the change:

```
helm upgrade <release> <chart> -n <namespace> -f <your-values.yaml>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If you restored the internal Percona database in place:

The service endpoint is unchanged (`<release>-psmdb-db-rs0.<namespace>.svc.cluster.local`), so you usually do not need
to change connection values. If credentials changed, update the `<release>-psmdb-secrets` Secret accordingly.

Step 4: Bring the platform back up and validate

```
# Scale the services back up (use your normal replica counts)
kubectl scale deployment \
  <release>-central-service <release>-workspace-api <release>-
     coordinator \
  <release>-frontend <release>-proxy \
  --replicas=1 -n <namespace>

# Or force a fresh rollout so pods pick up new connection settings
kubectl rollout restart deployment -n <namespace>
```

Then validate the platform:

1. Confirm services start and connect to the database (check pod logs for MongoDB connection errors).
2. Sign in and confirm organizations, projects, users, and workspaces are present.
3. Confirm existing workspaces can be reattached.
4. If you use HashiCorp Vault, confirm secrets resolve correctly.

> **Important**
> A backup you have never restored is not a backup. Test your full restore procedure regularly in a non-production environment so you can rely on it during an incident.

Related information

- Back up the database
- The SecurSpaces database

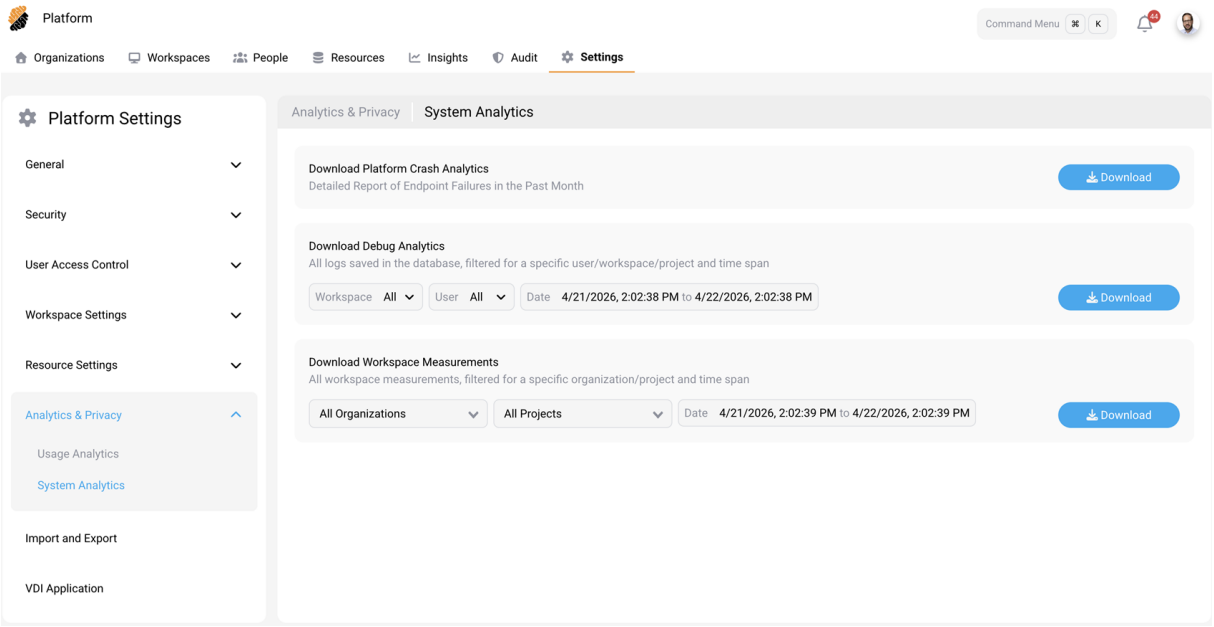
Analytics

System Analytics

Use the System Analytics section to download detailed reports and logs for the Citrix SecurSpaces™ platform. Reports are available under **Platform Settings > Analytics & Privacy > System Analyt-**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ics.



Download Platform Crash Analytics

Download a report of API and endpoint failures from the past 30 days. This report helps identify recurring errors and supports platform-level troubleshooting.

Download Debug Analytics

Download system logs filtered by workspace, user, or time range. Use this report for targeted troubleshooting and root-cause analysis.

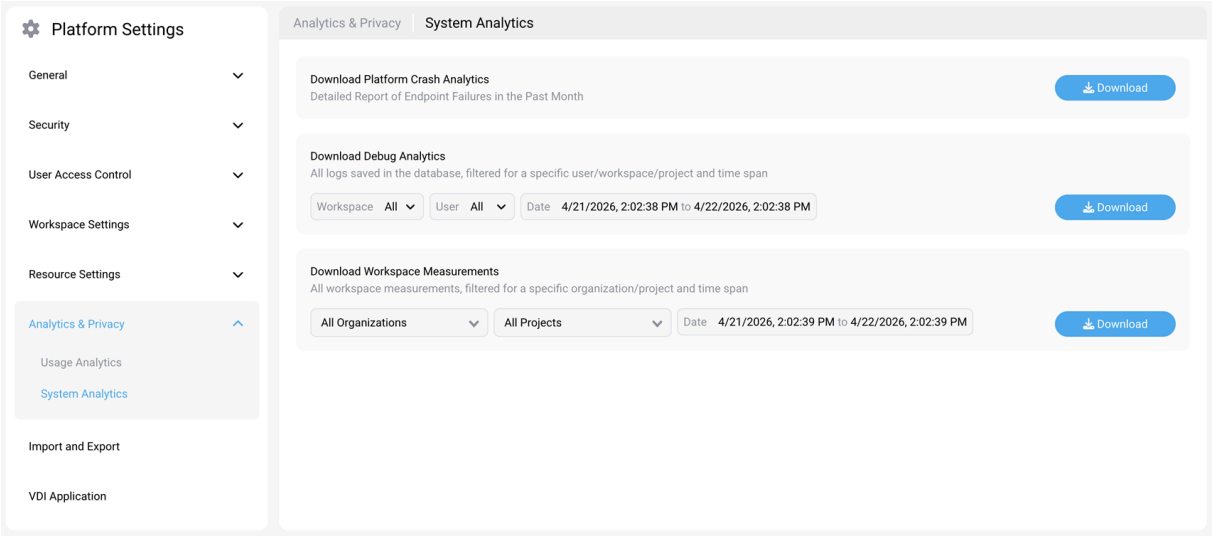
Available filters:

- **Workspace** —Select a specific workspace or choose **All** to include all workspaces.
- **User** —Select a specific user or choose **All** to include all users.
- **Date** —Specify a start and end date for the report. An **All Dates** option is also available.

Download Workspace Measurements

Download workspace resource usage measurements across the platform. This report contains workspace measurement data that can be filtered by organization, project, and time range. Because measurement files can be large, the platform uses a WebSocket connection to facilitate the download.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Available filters:

- **Organization** —Select a specific organization from the dropdown or keep **All Organizations** to include data from every organization. The dropdown supports a search to find an organization by name quickly.
- **Project** —Select a specific project or keep **All Projects** to include data from every project.
- **Date** —Specify a start and end date for the report, or select **All Dates** to download the complete measurement history.

To download workspace measurements:

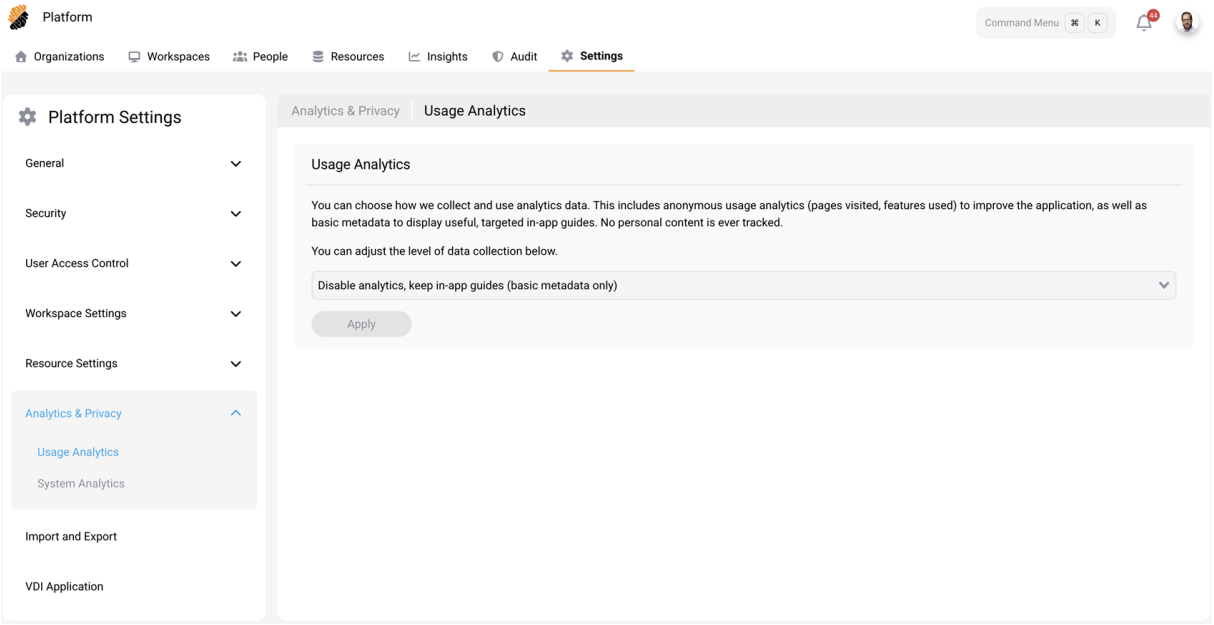
1. Navigate to **Platform Settings > Analytics & Privacy > System Analytics**.
2. In the **Download Workspace Measurements** section, select an organization, project, and date range as needed.
3. Select **Download**.

> **Note:**
>
> Only platform administrators and security officers can access the System Analytics page.

Usage Analytics

The SecurSpaces management console uses Pendo to deliver in-product notifications, feature announcements, and contextual guidance. It also collects product feedback and usage telemetry to help improve the platform experience.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Data collection preferences

You can choose how analytics data is collected and used. This includes anonymous usage data (such as pages visited and features used) to improve the application, and basic metadata to enable targeted in-app guides. No personal content is ever tracked.

To change the data collection level, navigate to **Platform Settings > Analytics & Privacy > Usage Analytics**, select an option from the dropdown, and select **Apply**.

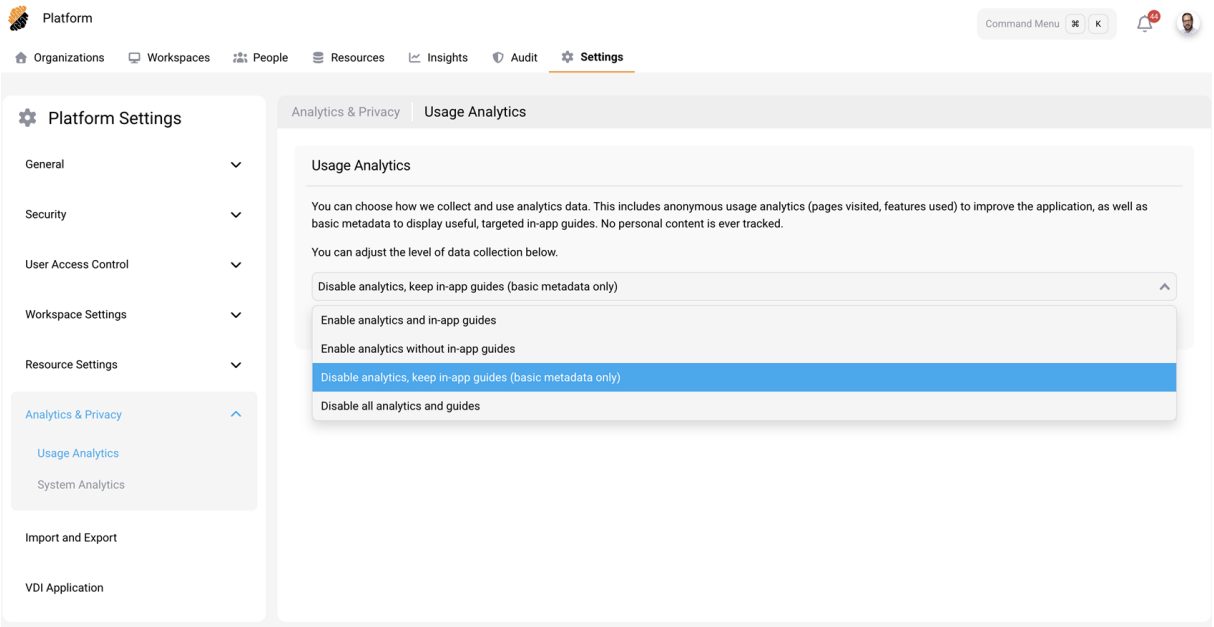
Available options:

- **Enable analytics and in-app guides** —Usage telemetry is collected, and in-app guidance is displayed.
- **Enable analytics without in-app guides** —This is the default for new deployments. Usage telemetry is
  collected, but all in-app guidance overlays, prompts, and tooltips are suppressed.
- **Disable analytics, keep in-app guides (basic metadata only)** —No usage telemetry is shared with Citrix, but in-app guidance remains available.
- **Disable all analytics and guides** —All Pendo components are disabled, and no data is shared with Citrix. In-app guidance, notifications, and the ability to submit feedback are not available.

> **Note**
>
> The default applies to new deployments only. A deployment that was installed before this default

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> changed keeps
> whatever level it was already set to, including **Enable analytics and in-app guides**.



Connectivity requirements

To view Pendo content within the management console, ensure that the address `http://citrix-sds-content.customer.pendo.io` is reachable from your network.

Pendo is a third-party sub-processor that Citrix uses to provide cloud and support services to Citrix customers. For a complete list of sub-processors, see Sub-Processors for Citrix Cloud & Support Services and Citrix Affiliates.

Workspace resource usage insights

Citrix SecurSpaces™ provides historical insights into workspace CPU and memory usage. This data is automatically collected and stored in the SecurSpaces database to support rightsizing analysis and long-term trend evaluation.

By leveraging this data, customers can:

- Analyze CPU and memory consumption for each workspace over time.
- Identify optimal resource allocation for workspaces.
- Reduce infrastructure costs while maintaining a high-quality developer experience.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Requirements

To enable the collection of workspace metrics, the **Kubernetes Metrics Server** must be installed. This component aggregates resource usage data across Kubernetes clusters and is commonly deployed in cloud-hosted environments or any setup that uses autoscaling.

For installation instructions and additional details, see the Kubernetes Metrics Server documentation

Data collection and storage

SecurSpaces automatically consolidates the raw measurement data every five minutes and provides the following data points for the previous 5-minute interval:

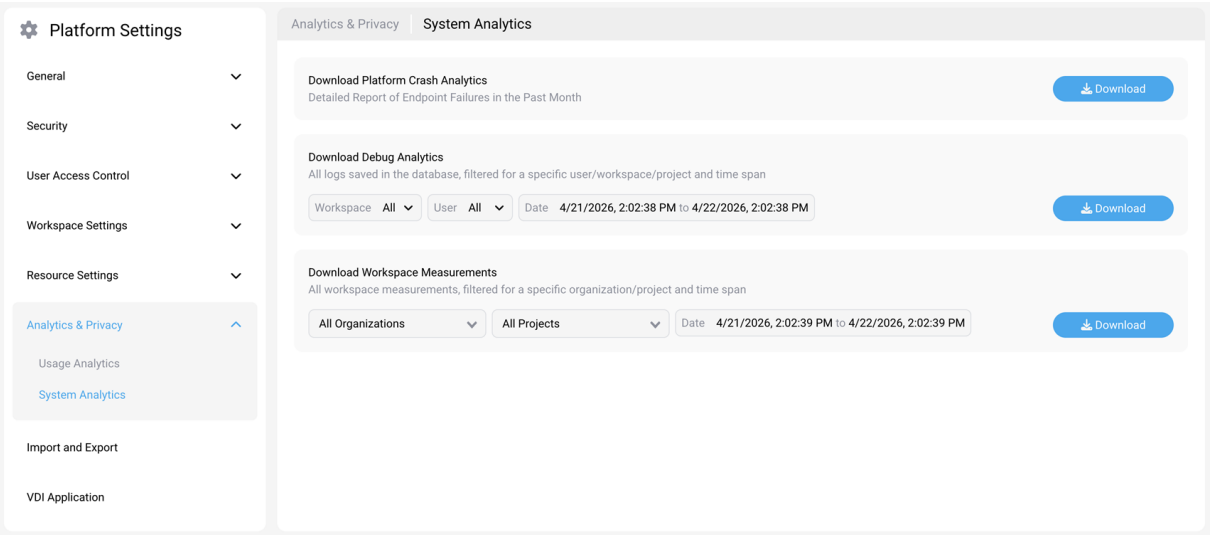
- Minimum, Maximum, Average, P50, P75, P95, and P99

Access workspace measurements

Workspace measurement data is available through the management console and via API.

Download from the management console

Platform administrators and security officers can download workspace measurements as a file from **Platform Settings > Analytics & Privacy > System Analytics**.



Available filters:

- **Organization** —Select a specific organization or keep **All Organizations** to include data from every organization.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Project** —Select a specific project or keep **All Projects** to include data from every project.
- **Date** —Specify a start and end date, or select **All Dates** to download the complete measurement history.

To download workspace measurements:

1. Navigate to **Platform Settings > Analytics & Privacy > System Analytics**.
2. In the **Download Workspace Measurements** section, select an organization, project, and date range.
3. Select **Download**.

For more information about the System Analytics page, see System Analytics.

Access via API

Data is available in both raw and aggregated formats via API. Customers can access metrics at the platform, organization, and project levels.

- For raw data, use the **workspace-measurements-samples** API (for example, `/v1/metrics/workspace-measurements-samples`).
- For aggregated data, use the **workspace-measurements** API (for example, `/v1/projects/{ projectId } /metrics/workspace-measurements`).

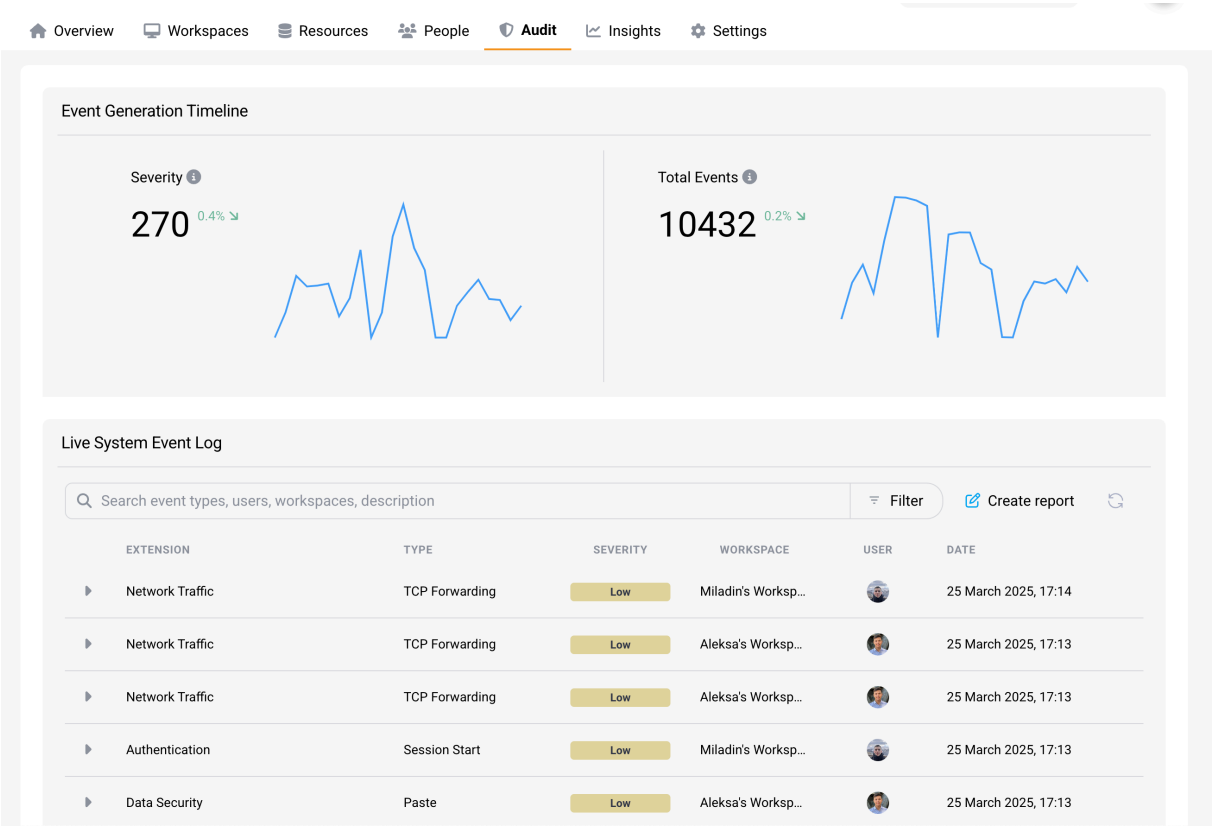
For API documentation and usage examples, see the SecurSpaces API documentation.

Audit page

Requires the Security permission set to **Access**.

The **Audit page** provides insights into the security of your **Project**, including a **Event Generation Timeline** graph that illustrates the timeline of events triggered by workspaces within the current project. Additionally, the **Live System Event Log** presents a table displaying detailed logs of each event.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



- Event Generation Timeline
- Event Logs
- Filtering Logs
- Log Display

Event Generation Timeline

The event generation timeline are two graphs, where the first one shows the severity level of triggered events over time, and the other displays the total amount of triggered events over time.

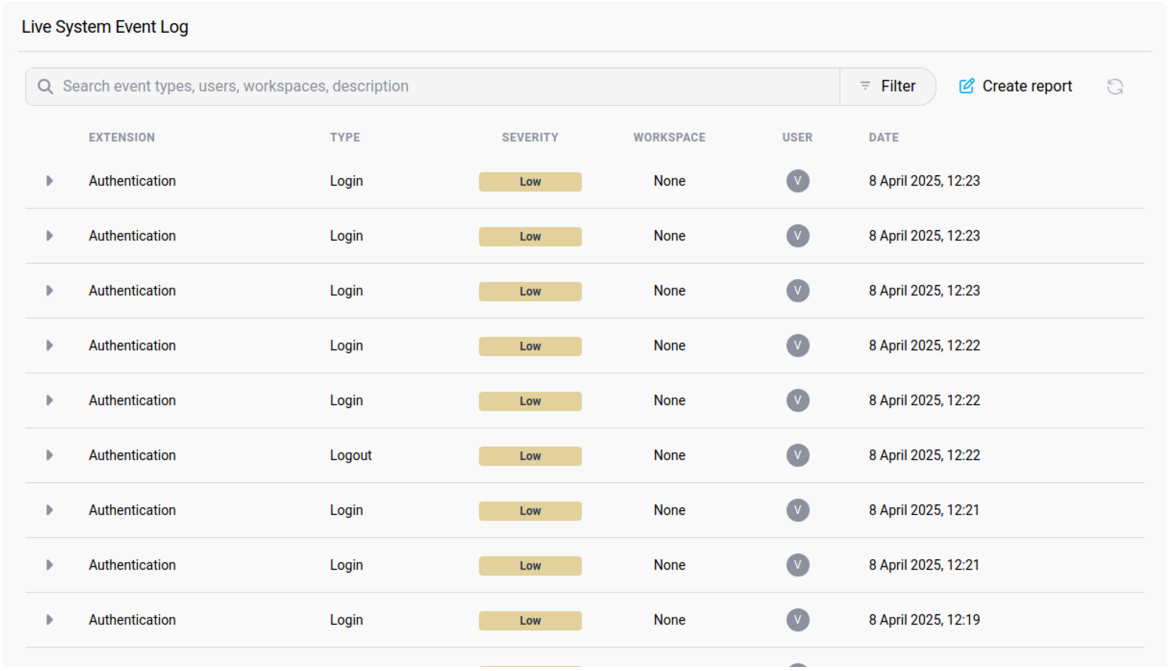


© 1997–2026 Citrix Systems, Inc. All rights reserved.

Event Logs

The **Live System Event Log** displays records of security events triggered by workspaces within a specific project. These events can take many forms, such as clipboard monitoring or network alerts, like a DNS request. These logs are significant as they have the ability to uncover potential security vulnerabilities.

> **Tip**
>
> Events are triggered once you enabled the option "Log and record outbound network traffic"for the associated Network Policy.



Filtering Logs

The log view allows users to easily filter and search through the system's event logs.
This feature makes it very convenient to identify possible issues, troubleshoot and also to monitor the usage of the system in a more granular level. To display filter options, press the "**Filter**"button located at the top right of the **Live System Event Log** panel.
Filter logs by:

- **Type** of the event,
- **Severity** level,
- **Workspace** from where the event was triggered,
- **User** that triggered the event,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Date** and time at which the event was triggered.

In addition to filtering logs, you can search through them by typing key words in the search bar below the date range (e.g. search for a specific user).

Log Display

The log view provides detailed information about each event that occurs within the system. For each log, you can view the following information:

1. **Type**: What kind of event was triggered,
2. **Severity**: Severity level of the event,
3. **Workspace**: Workspace from where event was triggered,
4. **User**: User who triggered the event,
5. **Date**: Date and time at which the event was triggered,
6. **Description**: Describes action that triggered the event.

To view more details about an event, press the dropdown menu button to the left of the event's log.

SIEM integration

SecurSpaces can forward platform audit events to an external Security Information and Event Management (SIEM) system using the **Common Event Format (CEF)**. You can use any log-shipping tool that supports CEF to collect and forward these events. The walkthrough below uses Filebeat deployed as a Kubernetes DaemonSet as an example.

Setting up the SIEM integration requires two phases: a backend deployment performed by the platform operator, followed by activation in the management console by a platform administrator.

Prerequisites

- Access to the Citrix SecurSpaces™ Helm chart and `values.yaml` configuration file
- Permission to deploy updates to the Kubernetes cluster hosting SecurSpaces
- Platform administrator or security officer role in the SecurSpaces management console
- A target SIEM system that accepts CEF-formatted events (for example, Elastic SIEM)

Phase 1: Configure the backend

Complete these steps in your deployment environment before activating the integration in the management console.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Step 1 —Enable syslog in the Helm values

Open the `values.yaml` file for your SecurSpaces deployment and uncomment the configuration block under `ninja.clusterConfig.syslog`. Update the values to match your environment:

```
syslog:
  volumeName: "syslog-volume"
  volume:
    name: "syslog-volume"
    hostPath:
      path: "/strong-network/siem"
      type: DirectoryOrCreate
  mountPath: "/var/strong-network/"
  hostPath: "/strong-network/siem"
```

Parameter	Description
`volumeName`	Name identifier for the Kubernetes volume.
`volume`	Kubernetes volume definition. The `hostPath` type maps a directory on the node's filesystem into the pod. `DirectoryOrCreate` tells Kubernetes to create the directory if it does not already exist.
`mountPath`	Path inside the platform service containers where the volume is mounted. The application writes `.log` files here (for example, `/var/strong-network/<name>.log`).
`hostPath`	Path on the host node where log files are persisted. Your log-shipping tool (for example, Filebeat) reads from this same path.

The net effect: platform pods mount the host directory `/strong-network/siem` at `/var/strong-network/` inside the container and write `.log` files there. Filebeat (or your chosen tool) reads those same files via the host path.

Step 2 —Prepare the Filebeat YAML

1. In the SecurSpaces management console, go to **Platform > Settings > Security > SIEM Integration**.
2. Copy the YAML example displayed on the page. This YAML defines a Filebeat deployment comprising a `ServiceAccount`, a `ConfigMap` (containing the Filebeat pipeline configuration), and a `DaemonSet`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

3. In the copied YAML, replace every instance of `release` with your actual deployment namespace.
4. Fill in the Elasticsearch connection credentials in the environment variables section of the DaemonSet spec. For production deployments, store these credentials in a Kubernetes Secret and reference the Secret in the env vars rather than placing plain-text values in the YAML.

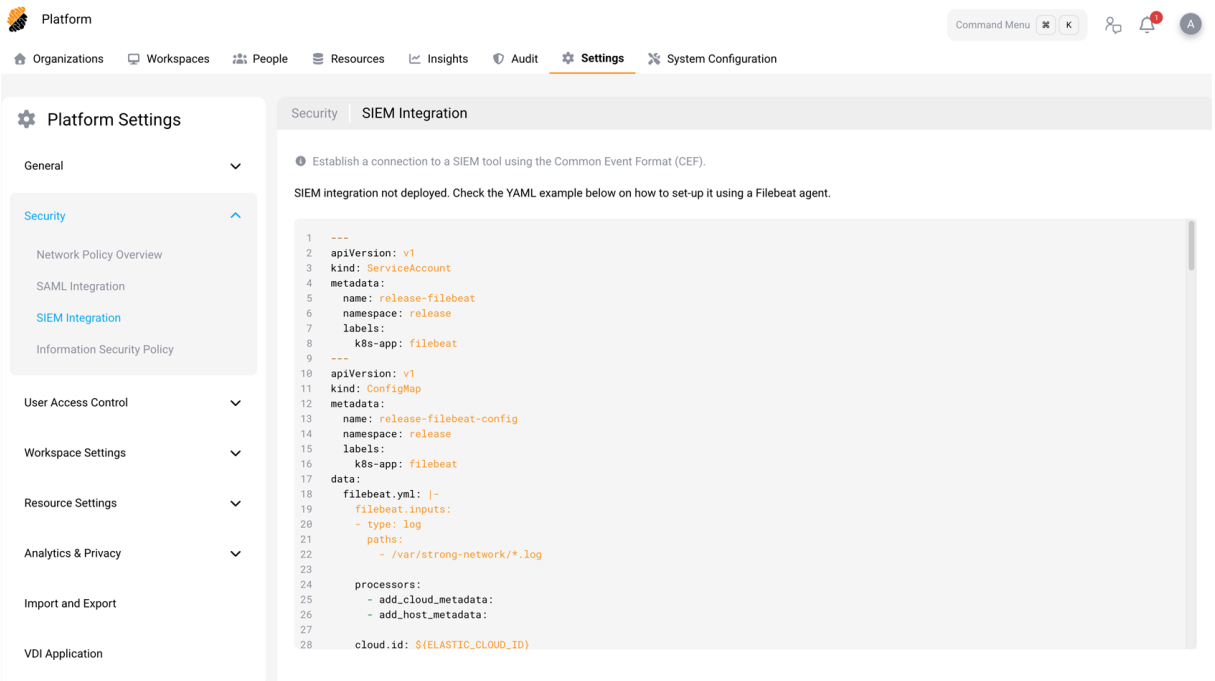
Step 3 —Deploy

Place the updated YAML into your Helm charts and run a fresh deployment to apply the updated Helm configuration and the Filebeat DaemonSet to the cluster.

Phase 2: Activate in the management console

1. Sign in to the SecurSpaces management console as a platform administrator or security officer.
2. Go to **Platform > Settings > Security > SIEM Integration**.
3. If the DaemonSet is healthy and the volume is correctly mounted, a toggle appears to activate the integration. Enable the toggle.

Once activated, the platform begins forwarding audit events to the configured SIEM target.



© 1997–2026 Citrix Systems, Inc. All rights reserved.

How it works

1. SecurSpaces platform services write audit events as `.log` files in CEF format to the container mount path (`/var/strong-network/`), which maps to `/strong-network/siem` on the host node.
2. The Filebeat DaemonSet running on each cluster node reads log files from the host path (`/strong-network/siem`).
3. Filebeat ships the events to the configured SIEM target (for example, Elasticsearch).

Audit events

The platform forwards events across eight categories. Every event is assigned a **ClassID** that identifies its category in the CEF payload.

For every event and its attributes, see Audit event catalog.

Common extensions

All events include the following CEF extensions:

Extension	Description
`timestamp`	Time the event was generated.
`suid`	User ID of the actor.
`suser`	Username of the actor.
`wsid`	Workspace ID (when applicable).
`wsname`	Workspace name (when applicable).
`projectId`	Project ID.
`projectName`	Project name.
`eventId`	Unique identifier for the event.

Event categories

Authentication (ClassID 1) Login, Logout, SessionStart, SessionEnd, SessionInterrupt

Authorization (ClassID 2)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **User:** UserBlocked, UserUnblocked, ProjectManagerSeat
- **Workspace:** SharedWithUser, UnsharedWithUser

Data Security (ClassID 3) Copy, Paste, Cut, Upload, UploadLargeFile, Download, DownloadLargeFile, SupervisedCopy

Data security events include additional extensions: `data`, `isSecret`, `isCode`.

System (ClassID 4) WorkspaceSpecsUpdated

User Activity (ClassID 5) VSCodeExtensionInstalled

Extension-install events include additional extensions: `extensionName`, `extensionId`, `extensionUUID`.

Account Management (ClassID 6) UserAddedToProject, UserRemovedFromProject, RoleChanged, UserCreated, UserDeleted, UserAddedAsSecurityOfficer, UserRemovedAsSecurityOfficer, UserAddedAsOrganizationOwner, UserRemovedAsOrganizationOwner, UserPasswordUpdated

Network Traffic (ClassID 7) SSH Commands, External SSH Commands, HTTP Requests, Git-over-HTTP, TCP Forwarding, DNS Resolution

Resource Access (ClassID 8) Actions: Created, Imported, ManuallyImported, Updated, SharedWithUsers, SharedPublicly, WorkspaceAttached, WorkspaceDetached, Deleted

Resources: repos, buckets, secrets, connectedServices, securityPolicy, images, credentials, mountPoints, workspaceApp, startupScript, workspaces, apiToken, personalSSH, vscodeVersions

Severity mapping

Each event carries a CEF severity value (0–10). The platform maps severity to syslog priority as follows:

CEF severity	Syslog priority
0–3	`LOG_INFO`
4–6	`LOG_ALERT`
7–10	`LOG_CRIT`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Event filtering

You can control which events the platform forwards using two filters on the SIEM Integration settings page:

- **Severity filter** —Sets a minimum severity threshold. Events below the threshold are not forwarded.
- **Event type filter** —Selects specific event categories or individual event types to forward.

Verify the integration

After deployment, confirm that the pipeline is working end-to-end:

1. Verify that the Filebeat pods are running:

```
kubectl get pods -n <namespace>
```

1. Confirm that `.log` files exist on the host path:

```
ls /strong-network/siem/
```

1. Check the SIEM Integration page in the management console —the status should indicate that the integration is active.
2. Perform an auditable action in SecurSpaces (for example, sign in or create a workspace) and verify that the corresponding event appears in your SIEM tool.

Insights page

The **Insights Page** displays information about the activity of the **Project**'s members, resource allocation and container process'metrics. The information displayed on this page depends on the implementation of the platform in your organization. This section provides a general view of the information commonly found across deployments.

> **Info**
>
> Depending on your permissions within the project, some of this information may not be available.

Content

- **Resource Allocation**
- **Container Process Metrics**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Resource allocation

August 19, 2026

Requires the Metrics permission set to **Access**
Project.

The **Resource Allocation** tab plots how much of your cluster workspaces are actually consuming, over time.
Because a workspace holds its allocation only while it is running, the curve tracks how many workspaces are
running at any moment.

The charts

Two charts are shown, one for CPUs and one for memory, with time along the horizontal axis. In a multi-region
deployment each region is plotted as its own line, so you can compare them directly.

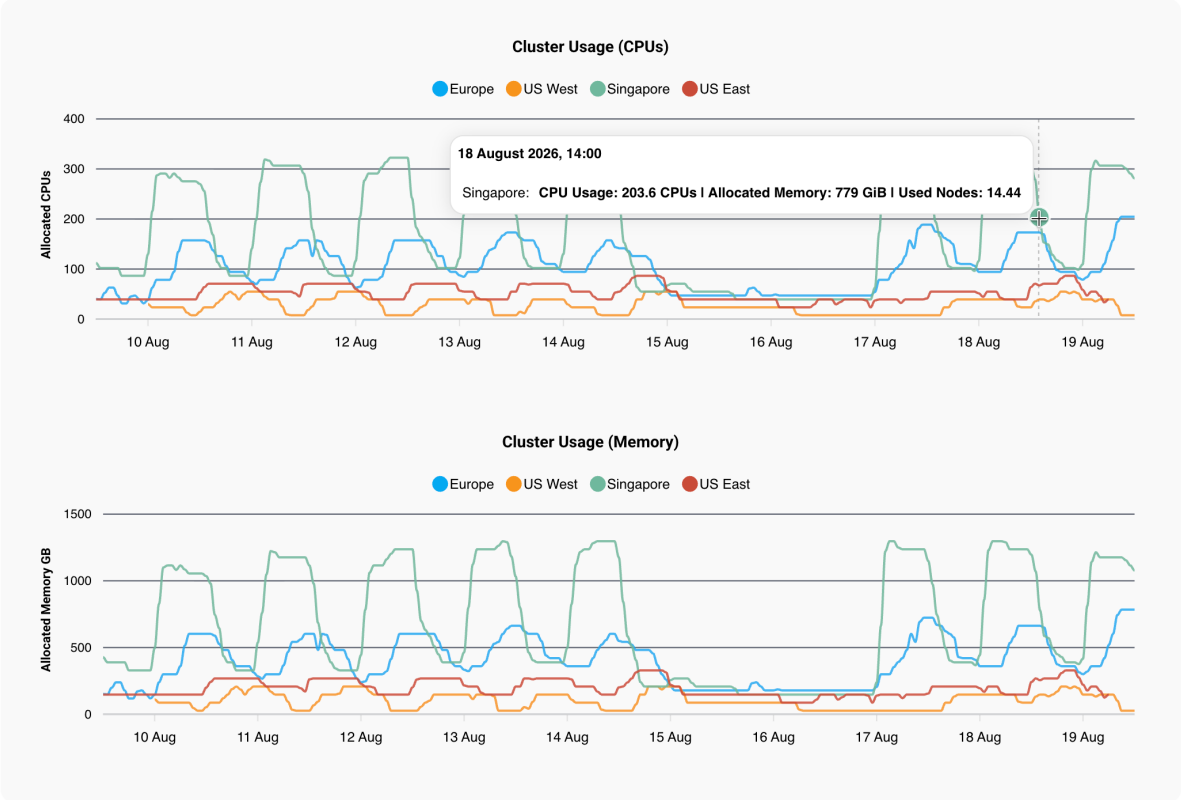
The chart titles change with where you are. The card is the same in all three places, but it is scoped to
what you are looking at:

Where you open it	Chart titles
Platform	Cluster Usage (CPUs) and Cluster Usage (Memory)
Organization	Organization Resource Allocation (CPUs) and (Memory)
Project	Project Resource Allocation (CPUs) and (Memory)

So a project owner sees their project's consumption, not the cluster's. Only the platform view covers the
whole deployment, which is the one to use for the capacity questions in
Sizing.

Use the time range list above the charts to change the window. It defaults to **Last 10 Days**, and also offers Last 7 Days, Last 7 Weeks, Last 30 Days, Last 90 Days, and Last Year.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Hovering a point reports the values at that moment for that region:

Value	Meaning
CPU Usage	Cores allocated to running workspaces
Allocated Memory	Memory allocated to running workspaces
Used Nodes	Cluster nodes in use, as an average across the interval

Reading the shape

The shape of the curve is more informative than any single number.

A daily rise and fall is what a healthy deployment looks like. Workspaces start as developers begin work,
and pause when they go idle, so usage climbs through the working day and falls away overnight. Flat stretches
across weekends and holidays are the same effect over a longer period.

A curve that does not fall is the one to investigate. If usage stays high overnight, workspaces are

© 1997–2026 Citrix Systems, Inc. All rights reserved.

running when nobody is using them, and you are paying for the nodes that host them. The usual causes are:

- Idle detection is not configured, or the timeout is too long. See Workspace idle detection.
- **Bypass Schedule** is enabled on more workspaces than intended. It is meant for unattended jobs, and it exempts a workspace from pausing entirely.
- Long-running processes are keeping sessions active.

Comparing the amplitude between regions is a quick way to see this: a region whose line barely moves, while the others cycle, is usually a configuration difference rather than a difference in how people work.

A rising baseline over weeks means the floor is climbing —workspaces that never stop, or growth in the number of workspaces. Either way it is the signal to plan capacity before it becomes urgent.

Workspace consumption

Below the charts, a sortable list shows total consumption per workspace in the project, so you can attribute a peak to the workspaces that caused it.

Each workspace is created against a specification that fixes its CPU, memory, and disk. What the list shows is consumption against that specification, which is how you find workspaces sized far above what they use.

Related information

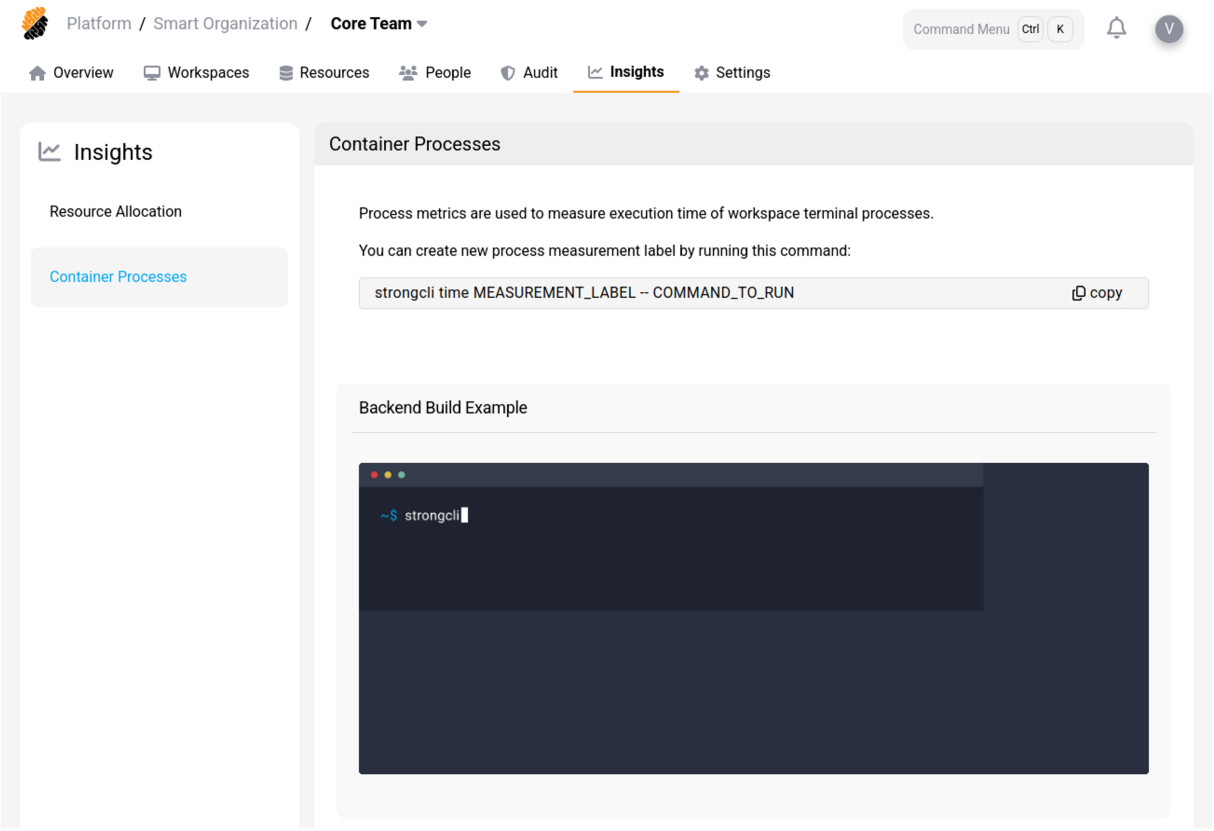
- Workspace resource usage
- Workspace idle detection
- Container process metrics
- System requirements

Container process metrics

The section **Container Process** displays time metrics registered using the platform Command Line Interface (CLI) **strongcli** available in developers workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Metrics are registered using the ‘time’option and become available in the Insight dashboard’s section **Container Process**. This CLI is typically used in scripts
embedded in the project containers such that, at startup a selection of processes can be registered for performance assessment. Once registered in a fleet of
workspaces, metrics are aggregated and eventually displayed in the Insights page.



- Track a Container Process
- Insights’Period
- Container Process Insights
    - Average
    - Total

Track a Container Process

You can track the execution time of container processes in workspaces using the platform’s Command Line Interface (CLI) **strongcli**.
Use the following command to do so:

```
1  > strongcli time LABEL -- COMMAND_TO_RUN
```

Where:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

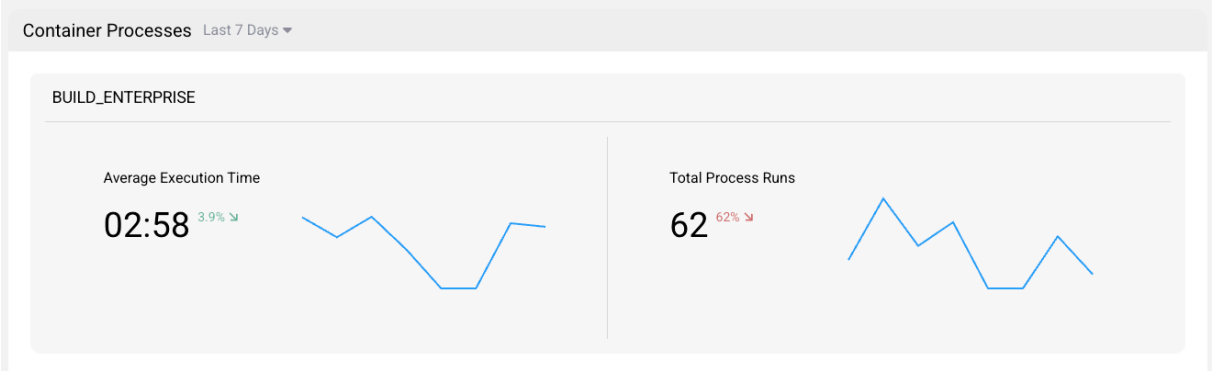
- **LABEL**: This allows setting a label to identify the process in the Insight dashboard,
- **COMMAND_TO_RUN**: The terminal command for which you would like to measure the execution time.

This registers a new process for your workspace among the **container processes** and measures its execution time.

Insights'Period

After selecting a container process, you can vary the span of the statistics from a 7-day execution average to a yearly average.

- Click on the drop-down menu to the right of "**Last 7 days**"to change the evaluation period.



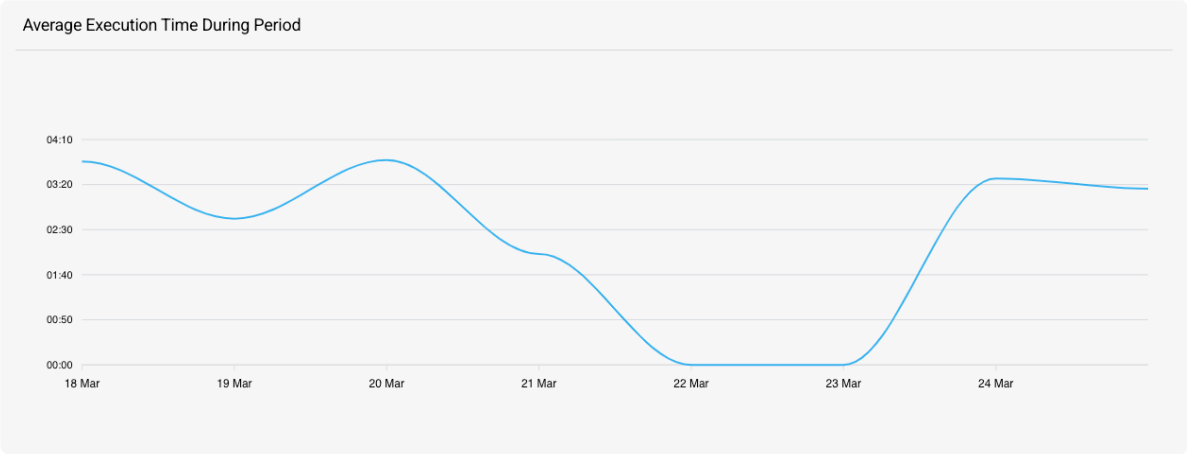
Based on selected period, the graph scale will be adapted accordingly.

Container Process Insights

Average

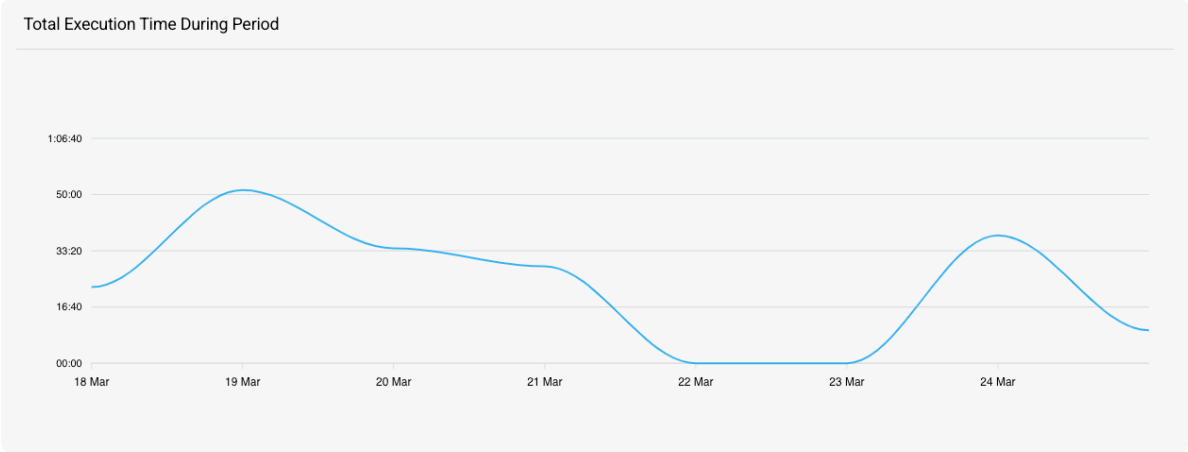
The "average execution time"graph in the container Process section of the Insight dashboard shows the average amount of time it took for a command to be executed within a developer's workspace, as recorded by the platform's Command Line Interface (CLI). The period of time displayed on the average execution time graph can be adjusted, allowing you to view metrics for a specific date range.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Total

The "total execution time"graph in the **container process** section of the Insight dashboard shows the total amount of time the command has been executed in a developer's workspace. The period of time displayed on the total execution time graph can be adjusted, allowing you to view metrics for a specific date range.



Security and compliance

August 19, 2026

This section is written for security reviewers, whether you are assessing SecurSpaces before adoption or
answering questions about a deployment you already run.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SecurSpaces is customer-hosted, so responsibility is shared. The platform provides controls; your cluster,
ingress, and infrastructure encryption are yours.
Trust boundaries states where the line falls.

Assessing the platform

Question	Page
What controls exist, and how do I cite one?	Control catalogue
Where does the deployment separate trust, and who owns each side?	Trust boundaries
How does data move between components?	Data flows
What ports must be open, and what encrypts what?	Ports and protocols, Cryptography

Controlling workspace traffic

Task	Page
Understand the policy model and the defaults	Network policies
Create a policy	Create a network policy
Apply one across projects	Assign and enforce a policy
Work out why traffic was blocked	Monitor and troubleshoot

SecurSpaces Flex

Flex is the managed offering, with a different responsibility split. See
SecurSpaces Flex security.

Related information

- Audit dashboard
- Audit event catalog
- Fixed issues —includes security fixes by release

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Security control catalogue

Introduction

This document provides an overview of the security controls available in Citrix SecurSpaces™. It supports security assessments, risk evaluations, and third-party due diligence reviews.

The controls described in this document reflect the SecurSpaces security architecture, operational practices, and shared responsibility model as a customer-hosted solution. They are designed to align with enterprise security requirements and industry best practices.

Additional documentation and supporting evidence can be provided on request to address specific security, compliance, or audit requirements.

About these identifiers

Each control below has a stable identifier, such as SDS-NET-01. These identifiers are assigned by Citrix so
that a specific control can be cited in an assessment, a questionnaire response, or a support case.

They are **not** a mapping to ISO 27001, SOC 2, NIST, or any other framework, and they do not imply certification against one. For a mapping to a specific framework, or for supporting evidence, contact Citrix
and specify what your review requires.

Identifiers are stable. A control that is withdrawn keeps its identifier, which is not reused.

Identity and access

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-IAM-01	Single sign-on and multi-factor authentication	SecurSpaces supports enterprise single sign-on (SSO) through standards-based identity federation using SAML 2.0 or OpenID Connect (OIDC). Multi-factor authentication (MFA) can be enforced by the identity provider as part of the authentication flow. SecurSpaces also includes native MFA capabilities.
SDS-IAM-02	Identity and authentication	SecurSpaces supports SAML 2.0 and OIDC for SSO and MFA enforcement. SecurSpaces also supports System for Cross-domain Identity Management (SCIM) for automated user provisioning, deprovisioning, and identity lifecycle synchronization from a central identity provider.
SDS-IAM-03	Access management	SecurSpaces includes built-in role-based access control (RBAC). Administrators can map application permissions to local user accounts or to directory groups synchronized from the identity provider. For more information, see Role-based access control.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-IAM-04	Manage IT-level privileges	SecurSpaces separates infrastructure-level management from application-level administration. Because SecurSpaces is containerized and hosted on Kubernetes, the customer's infrastructure team retains control over the underlying cluster, nodes, and container orchestration layer. Application-specific IT privileges, such as user management and configuration changes, are restricted to the application layer through granular RBAC. These privileges do not require direct access to the hosting infrastructure.

Cryptography and secrets

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-CRY-01	Encrypt data in transit	SecurSpaces encrypts data in transit using secure protocols. External web traffic uses HTTPS. TLS is terminated at the ingress controller, so the permitted TLS versions and cipher suites are set by the customer's ingress configuration. Secure terminal sessions use SSH. Within the Kubernetes cluster, ingress-to-service and service-to-service traffic is encrypted with TLS. For the full list, see Cryptography.
SDS-CRY-02	Encrypt data at rest	SecurSpaces is a customer-hosted solution. The customer is responsible for provisioning and managing the underlying database and persistent storage volumes. SecurSpaces is compatible with industry-standard encryption-at-rest technologies, including cloud-managed storage encryption, such as AWS KMS or Azure Key Vault, and encrypted Kubernetes Persistent Volumes (PVs). SecurSpaces also applies application-layer protection. When the external Vault feature is not enabled, developer secrets are encrypted in the database using AES-128-GCM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-CRY-03	Manage certificates and certificate lifecycle	SecurSpaces relies on standard X.509 certificates for TLS encryption. For external traffic, SecurSpaces supports automated certificate lifecycle management tools, such as cert-manager, Let's Encrypt, or internal PKI, to support certificate renewal without application downtime. For internal cluster traffic, SecurSpaces automatically manages the certificates required to maintain TLS between the ingress controller and backend services.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-CRY-04	Secrets management	SecurSpaces separates application and infrastructure secret handling. Application and developer secrets are managed at the application layer. By default, these secrets are encrypted before being written to the database using AES-128-GCM. SecurSpaces also supports integration with an external HashiCorp Vault instance. Secrets are injected securely into workspace runtimes through environment variables or volume-mounted configuration files. Infrastructure secrets, such as platform bootstrap credentials and database connections, use standard Kubernetes Secrets. Because SecurSpaces is customer-hosted, customers can add infrastructure-level protections, such as cloud provider KMS envelope encryption for `etcd` or a Secrets Store CSI Driver to inject infrastructure keys from an enterprise vault.

Network security

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-NET-01	Restrict network port usage to essential services	SecurSpaces operates with a minimal network footprint and exposes only the essential ports required for primary services: HTTPS on port 443, and SSH on an external port that you choose when you configure the ingress controller, commonly port 22. SecurSpaces also supports Kubernetes Network Policies, allowing network security teams to isolate pod-to-pod communication within the cluster and restrict traffic to essential paths. For the full list, see Ports and protocols.
SDS-NET-02	Network policies	SecurSpaces is containerized and requires a Kubernetes environment. It can be deployed in an isolated virtual private cloud (VPC) or subnet. SecurSpaces includes preconfigured Kubernetes Network Policies to restrict internal pod-to-pod communication. Customers can add or customize these policies to match their cluster security requirements.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-NET-03	Protect exposed applications	SecurSpaces is designed to run behind the customer's preferred web application firewalls (WAFs), reverse proxies, and API gateways. These controls can help filter malicious web traffic and protect against common web application risks, including OWASP Top 10 threats.
SDS-NET-04	Zero-trust data protection	SecurSpaces applies zero-trust principles across the application and network layers. At the application layer, every API call and data request requires explicit authentication and authorization. SecurSpaces verifies cryptographic tokens before granting data access. At the network layer, traffic from the ingress controller to backend services is encrypted with TLS. Within the cluster, SecurSpaces supports Kubernetes Network Policies to isolate and restrict pod-to-pod communication so that only explicitly authorized microservices can exchange network traffic.

Platform and container hardening

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-HRD-01	Hardening baseline	SecurSpaces uses a hardened container baseline. Core SecurSpaces service container images are built on the Debian-based Distroless image `gcr.io/distroless/static-debian12`, which remove unnecessary packages, shells, and system utilities to minimize the exploitable attack surface. A dedicated Ubuntu-based image is used only where specific SecurSpaces workspace features require standard OS utilities.
SDS-HRD-02	Container security	Core backend services use minimal Debian Distroless images to reduce the attack surface. Where specific workspace features require standard OS utilities, SecurSpaces uses dedicated Ubuntu images. Because SecurSpaces is customer-hosted, customers are responsible for implementing and operating runtime container scanning to validate images against corporate baselines. SecurSpaces is compatible with these scanning controls.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-HRD-03	Image build isolation	Images built on the platform with Create Image From Existing run in a Kubernetes Job using rootless BuildKit inside a user namespace. The build container is not privileged and runs as a non-root user, all Linux capabilities are dropped, and the user-supplied build process itself holds none. CPU and memory ceilings bound the build so a runaway script cannot exhaust the node, a wall-clock deadline stops a hung build, and the Job is never retried. The feature is inactive until a platform administrator configures a destination registry.

Data loss prevention

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-DLP-01	Data loss prevention	SecurSpaces provides native controls that support a data loss prevention (DLP) strategy. Clipboard security can be enforced to prevent users from pasting content from the IDE or secure browser into external applications. Workspace app security can require workspace applications to open only in a secure browser, helping reduce data exfiltration risk when developers share internal apps. Outbound network policies can restrict workspace traffic to approved repositories and domains. Audit events provide visibility into monitored or blocked activity. Where remote development over SSH is enabled, DLP effectiveness might be reduced and should be governed by customer policy. See Layered controls with the Citrix platform.

Logging and monitoring

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-LOG-01	Apply logging policy	SecurSpaces provides structured platform logging and real-time audit events. Platform logs capture application and service behavior to support operational troubleshooting, platform monitoring, and security investigations. The audit capability records security-relevant events, including authentication and session activity, user and workspace authorization actions, and data security events such as clipboard, upload, download, and monitored outbound network activity. These events are available on the SecurSpaces **Audit** page, where security teams can filter them by event type, severity, workspace, user, and date and time. This provides an operational audit trail for investigation and review. Platform audit events can also be exported in standard formats for downstream monitoring and SIEM integration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-LOG-02	Send logs to a Security Information and Event Management system	SecurSpaces supports Security Information and Event Management (SIEM) integration by writing security logs in Common Event Format (CEF) to the underlying host nodes. These logs can be collected and streamed to the corporate SIEM by configuring a standard log forwarder, such as Filebeat or Fluent Bit, to monitor the configured directory. For more information, see SIEM integration.
SDS-LOG-03	Establish detection use cases with the CSIRT	The SIEM integration supports filtering logs by severity threshold and event type. This allows the Cyber Security Incident Response Team (CSIRT) to create alerts for critical events, such as privilege changes or unauthorized access attempts. For more information, see Event categories and attributes.

Operations and resilience

ID	Control	Implementation
SDS-OPS-01	Vulnerability management	Citrix continuously scans SecurSpaces source code and software dependencies for vulnerabilities. Security patches and minor updates are released periodically.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-OPS-02	Patch management	SecurSpaces deployments use standard container tags and package management mechanisms. Updates can be rolled out with minimal disruption, using approaches such as rolling updates or blue-green deployments, depending on the customer's infrastructure configuration.
SDS-OPS-03	Apply backup policy	SecurSpaces separates stateless application logic from stateful data stores. Customers can apply standard snapshotting, database backup scripts, and backup retention policies to the underlying storage volumes.
SDS-OPS-04	Segregate production and non-production environments	The SecurSpaces deployment model supports isolated instances. Customers can deploy separate clusters or servers for development, staging, and production environments using separate environment configurations.

Governance and assurance

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-GOV-01	Governance and change management	SecurSpaces supports standard DevOps and GitOps deployment pipelines. Application configuration changes can be tracked through version control and infrastructure as code (IaC). Upgrades and configuration changes are documented in the release notes. For more information, see What's new in SecurSpaces.
SDS-GOV-02	Provide architecture documentation	SecurSpaces architecture documentation is available in the product documentation. A data flow diagram can also be provided as an annex when required. For more information, see Architecture.
SDS-GOV-03	Application inventory	Citrix can provide an application and component inventory for the SecurSpaces platform on request. Customers should specify the required level of detail for their assessment.
SDS-GOV-04	Infrastructure and access	SecurSpaces is customer-hosted. Physical, operating system, and infrastructure-level access are controlled and managed by the customer. SecurSpaces does not require external inbound vendor connectivity to function.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Control	Implementation
SDS-GOV-05	Security stack on the SecurSpaces platform	SecurSpaces deployment configurations, such as Helm charts or Kubernetes manifests, and delivered container images are compatible with validation through the customer security stack. Although proprietary source code is not provided, SecurSpaces supports automated infrastructure-as-code linting, configuration compliance checks, and container image vulnerability scanning, including software composition analysis (SCA), before deployment.
SDS-GOV-06	Third-party due diligence	Citrix can provide standard third-party assurance documentation on request. Customers should specify the artifacts required for their review.
SDS-GOV-07	Security annex	A data flow diagram can be provided as part of the security annex when required for assessment or audit purposes.
SDS-GOV-08	Non-disclosure agreement	Non-disclosure agreements are covered as part of the global contractual framework with Citrix.

Related information

- Security data flows
- Trust boundaries
- Ports and protocols

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Cryptography
- Network policies

Trust boundaries

September 29, 2026

Where a Citrix SecurSpaces™ deployment separates one level of trust from another, what enforces each
separation, and which side of it you own. This page supports network security review and architecture assessment.

It describes the boundaries the platform implements. It is not a threat model: it does not enumerate adversaries, abuse cases, or risk ratings.

Summary

Boundary	Enforced by	Owner
Your network to the cluster	Ingress controller, TLS termination	You
Ingress to SecurSpaces services	Kubernetes NetworkPolicy, TLS	SecurSpaces
SecurSpaces services to the database	Kubernetes NetworkPolicy	SecurSpaces
Workspace to workspace	Kubernetes NetworkPolicy	SecurSpaces
Workspace to the internet	A Kubernetes NetworkPolicy confines the workspace to the proxy service, which then enforces the SecurSpaces network policy attached to it	SecurSpaces, configured by you
Platform to the Kubernetes API	Kubernetes RBAC	SecurSpaces
Cluster to the nodes and the host	Your cluster configuration	You

> **Important:**
>
> Every in-cluster boundary on this page is a Kubernetes NetworkPolicy, so it holds only if your cluster runs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> a network plugin that implements the `networking.k8s.io/v1` NetworkPolicy API. If it does not, these
> policies are accepted by the API server and silently do nothing. See
> Requirements.

The perimeter

Traffic reaches the deployment through the ingress controller. TLS terminates there, so the permitted TLS
versions and cipher suites are part of your ingress configuration rather than something SecurSpaces sets.

The load balancer in front of the ingress controller operates at Layer 4 and does not terminate TLS.

For the ports that must be reachable, see
Ports and protocols.

Inside the cluster

SecurSpaces labels its pods by type and ships NetworkPolicies that act on those labels.

Pod type	Label
Platform services	`strong.network/type: service`
Database	`strong.network/type: database`
Workspaces and secure browsers	`strong.network/type: workspace`, `strong.network/type: browser`
Workspaces with a network policy attached	`strong.network/hardened: no_egress`

> **Important:**
>
> Do not rename these labels. The NetworkPolicies select on them, and a renamed label produces a policy that
> matches nothing, with no error.

Service pods accept traffic from anywhere

The NetworkPolicy for `service` pods accepts ingress from any source. This is deliberate —they are the entry

© 1997–2026 Citrix Systems, Inc. All rights reserved.

point for the deployment —but it means the perimeter around the platform services is your ingress configuration, not a NetworkPolicy.

The database policy

SecurSpaces ships a NetworkPolicy that restricts ingress to pods labeled `database`, accepting connections
only from pods labeled `service`.

> **Important:**
>
> A NetworkPolicy only covers pods that carry the label it selects on. Whether your database pods carry it
> depends on how the database is deployed, and a database running outside the cluster is not covered at all.
> Confirm the label is present on your database pods before treating this policy as the control that isolates
> them, and apply your own network controls where it is not.

Workspaces accept traffic only from the workspace API

Pods labeled `workspace` or `browser` accept ingress only from the workspace API pod.

This is what isolates workspaces from each other. One workspace cannot open a connection to another, and this
holds whether or not a network policy is attached, because it is enforced on the receiving side.

Workspace egress depends on the attached policy

Two different mechanisms share the name *network policy*, and both are involved here.

A workspace carries the label `strong.network/hardened: no_egress` only while a **SecurSpaces network
policy** is attached to it. A **Kubernetes NetworkPolicy** selects pods with that label and permits them to
send traffic only to `service` pods, which leaves the SecurSpaces proxy service as the workspace's only
route out. The proxy service then allows or refuses each destination according to the SecurSpaces network
policy.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

So the Kubernetes NetworkPolicy decides that a workspace may talk only to the proxy service. The SecurSpaces
network policy decides where the proxy service will carry it.

> **Important:**
>
> A workspace with no network policy attached is not labeled as hardened, so this NetworkPolicy does not apply
> to it and its outbound traffic is not restricted by the platform. Egress restriction is something you turn
> on by attaching a policy. See
> Network policies.

Platform access to the Kubernetes API

The workspace API manages workspace lifecycle through the Kubernetes API, using a service account scoped to
the release namespace.

Scope	Access
Namespace	Pods, services, secrets, persistent volume claims, config maps, service accounts, events, jobs, deployments and stateful sets, pod logs, and a named lease
Cluster-wide	Storage classes and persistent volumes, read-only. Nodes and node metrics, read-only.

Write access to persistent volumes is not granted by default. It is enabled by the Helm setting `platform.serviceKubernetesConfig.rbac.allowPersistentVolumeAccess`, which defaults to **`false`**.

Grant it only for the attach modes in which SecurSpaces builds the volume itself: Azure Files Attach Existing,
and the AWS Attach Existing mode in which a user supplies the file system coordinates and SecurSpaces builds a
static volume from them. Attaching a PersistentVolume you authored does not need it—SecurSpaces creates only
the claim, so the read access above is enough, and it never changes a volume you own. Creating a new mount
point does not need it either. While the setting is off, the modes that require it are hidden and rejected,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

and the other modes keep working. See
Configure SecurSpaces for AWS Mount Points.

Host-level changes on workspace nodes

SecurSpaces runs a DaemonSet on nodes that are not labeled as service nodes. Its init container runs as root
with `privileged: true`, in the host network, PID, and IPC namespaces, and mounts `/sys` from the host.

It sets these kernel parameters on those nodes:

Parameter	Value	Purpose
`kernel.unprivileged_userns_clone`	`1`	Allows unprivileged user namespaces, which rootless containers in workspaces require
`user.max_user_namespaces`	`49152`	Raises the user namespace limit for the same reason
`fs.inotify.max_user_watches`	`1524288`	Raises the file watch limit that the IDE needs on large repositories
`kernel.yama.ptrace_scope`	`0`	Allows a debugger to attach to a running process. Developers need this to debug compiled binaries, such as C and C++, inside a workspace. This relaxes the default restriction on most distributions.
`fs.suid_dumpable`	`0`	Prevents core dumps from setuid programs
`kernel.core_pattern`	`/dev/null`	Discards core dumps rather than writing them to disk

These are node-level settings and apply to everything scheduled on those nodes. If you run other workloads on
the same nodes, take this into account when you plan node pools.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Working from a local IDE

A developer can work in the Cloud IDE in a browser, or connect a local IDE over SSH. The second is a supported access channel rather than a workaround, but it is not equivalent: a local IDE is software running
on the endpoint, so it widens the boundary slightly. This section is what a security reviewer needs to judge by how much.

The development environment itself does not move. Code, execution, and credentials stay in the workspace
whichever channel is used; what changes is where the editor renders.

Where each artifact resides

Artifact	Resides	On the endpoint?
Source code	Workspace filesystem	No. The local IDE edits it remotely and does not write it to local disk
Repository credentials	Brokered by the SecurSpaces proxy service	No. Never exposed to the endpoint, and not readable from inside the workspace either
SSH key that opens the tunnel	The developer's endpoint	**Yes, necessarily.** This is the key that authenticates the developer to the workspace
Personal SSH identity, used for any SSH service	Held by the platform and presented by the SecurSpaces proxy service when the connection is made	No. Not written into the workspace, so a process there cannot copy it
SSH services attached as project resources	Held by the platform, injected the same way	No
Build artifacts	Workspace filesystem	No, unless a developer deliberately copies them out, which egress and transfer controls govern
Language package caches, such as npm and pip	Workspace filesystem	No

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Artifact	Resides	On the endpoint?
AI model context	Inside the workspace and the policy boundary, attributed per user, project, and workspace	No
IDE extension data	Split. Extensions that render UI run on the endpoint; extensions that execute code run in the workspace	**Partly.** The UI extension host is the one part of the IDE that is genuinely local

> **Note:**
>
> The two SSH key roles are easy to conflate and the distinction matters. The key that opens the tunnel is the developer's own and lives on their machine, like any SSH client key. The key the *workspace* presents when it connects out—its personal SSH identity, or an SSH service attached as a project resource—is held by the platform and reaches neither the endpoint nor the workspace runtime. See How SSH connections are handled.

What the endpoint can still see

Because the IDE renders locally, a compromised endpoint could still capture keystrokes, screen content, or clipboard data, and the extension host runs local code. Clipboard controls still apply, but they are enforced against the session rather than against the operating system.

This is inherent to any SSH-based remote development model rather than specific to SecurSpaces, and it is a smaller surface than a laptop or a VDI session with a local Linux subsystem, where source code, credentials, and package caches are all genuinely local. It is still larger than the Cloud IDE, which is why remote development over SSH is a setting an administrator turns on deliberately. See Workspace policy.

What you own

SecurSpaces is customer-hosted, so the following are outside the boundary the platform controls:

- The ingress controller and its TLS configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- A network plugin that enforces NetworkPolicy. Without one, the in-cluster boundaries above do not hold.
- Cluster and node hardening, and which workloads share a node pool with workspaces.
- Encryption at rest for the database, persistent volumes, and `etcd`.
- Runtime container scanning and image admission control.
- Network reachability between the cluster and any external dependency, such as the database, registries,
  repository providers, and your SIEM.

Related information

- How the SecurSpaces proxy service works
- Security data flows
- Security control catalogue
- Ports and protocols
- Cryptography
- Network policies
- Requirements

How the SecurSpaces proxy service works

August 25, 2026

Most of the security claims in this documentation resolve to one component. Credentials do not reach the
developer's machine because the SecurSpaces proxy service holds them. Egress is restricted because the proxy
service filters it. AI spend is attributable because the proxy service labels it. Repository access is auditable because the proxy service sees it.

This page explains what that component does. There is nothing to configure on it —the proxy service is not
optional and has no settings of its own. What you configure is network policy, project resources, and AI
routing, each documented separately.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Why everything passes through it

A Citrix SecurSpaces™ workspace has no direct route to anything. It cannot reach the internet, your repositories, or your internal systems on its own; every connection it opens terminates in the proxy service
first.

That single property is what makes the other controls enforceable rather than advisory. A policy a workspace
could bypass would be a recommendation. Because there is no path around the proxy service, a developer with
full control of their own workspace —including root in a container they built —still cannot reach a destination policy forbids, and still cannot read the credentials their tools are using.

What passes through it

Traffic	What the proxy service does
HTTP and HTTPS	Applies network policy, brokers credentials for repositories and services, and records the request
Git over HTTP, including Git LFS	Routes repository traffic so that access uses platform-held credentials and follows policy
DNS	Resolves names for the workspace, including the internal names used to reach workspace applications
TCP and UDP	Carries traffic for services that are not HTTP
SSH	Carries outbound SSH, when an administrator has enabled it and a method is configured for the target host

Credentials are held, not handed over

When a workspace reaches a repository or an external service, it does not present a credential, because it
does not have one. It addresses the destination by an identifier the platform assigns, and the proxy service
resolves that to the real endpoint and attaches the credential on the way through.

The practical consequences are worth stating plainly:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Nothing long-lived is written to the developer's machine, whichever IDE or connection method they use.
- Nothing readable is written inside the workspace either, so a compromised workspace does not yield a
  usable credential.
- Revoking access at the platform takes effect on the next request, without touching any endpoint or
  workspace.

Policy is applied here

Network policies attached at platform, organization, and project level are combined, and the result is applied to every connection. A destination that no policy allows is refused.

Policies are evaluated against both the address and the domain, which is why an allow list works even when a
destination resolves to a changing set of addresses. For how policies are written and assigned, see Network policies.

How HTTPS connections are handled

An HTTPS connection out of a workspace is two connections. The proxy service terminates the one from the
workspace, reads the request, decides what to do with it, then opens its own connection to the destination
and forwards the request there.

This is what makes the rest of the platform's controls possible on encrypted traffic:

The proxy service can	Because it reads the request
Apply policy to a domain rather than only an IP address	The destination is in the request, not only in the TLS handshake
Attach a repository or service credential	The credential is added to a request the workspace never sees
Attribute AI usage to a user, project, and workspace	Identifying headers are added before the request is forwarded
Route provider traffic to an AI gateway	The destination is matched against the configured endpoints

None of that is possible while the connection is opaque to the platform.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

What the workspace trusts

Because the proxy service presents its own certificate to the workspace, the workspace has to trust the
certificate authority that issued it. SecurSpaces adds its service CA certificate to the workspace's trust stores at start, which is why HTTPS tools inside a workspace work without per-tool configuration.

A consequence worth knowing: a tool that pins a certificate, or that uses its own trust store rather than the system one, may reject the connection. Some language runtimes and package managers do this. See
Container image requirements.

Where decryption happens

Decryption happens inside the deployment you host, on the proxy service pods, and the request is re-encrypted
before it leaves for the destination. Traffic is not sent to Citrix, and it is not decrypted anywhere outside
your cluster. In SecurSpaces Flex the same is true of the Citrix-managed environment your organization is
allocated.

> **Note:**
>
> **Inspect** also names a specific network policy behavior, which reports on TCP content and is applied
> through the built-in **Inspect Traffic** policy. That is a separate, optional setting. The handling described here applies to HTTPS whenever a SecurSpaces network policy is attached, whichever behavior the
> policy uses. See Network policies.

How SSH connections are handled

SSH is handled the same way structurally —an SSH connection out of a workspace is also two connections
rather than one tunnel —with one important difference. The proxy service does not read the contents of an
SSH session. It terminates the transport, decides whether the connection is allowed, and presents the right
credential, but what passes through afterwards is not examined.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

A workspace opens an SSH connection to the proxy service rather than to the destination. The proxy service
authenticates the workspace, applies the SecurSpaces network policy to the destination, then opens a
separate connection onward and authenticates to the destination itself.

The credential it presents is the developer's personal SSH identity, which the platform holds. It is read from the secret manager at the moment the connection is made and used to sign the handshake; it is not
written into the workspace, so a process inside the workspace cannot copy it.

> **Note:**
>
> Inbound SSH —a developer connecting from a local IDE into their workspace —is terminated by the
> workspace service rather than the proxy service. That connection is authenticated against the developer's
> registered public key, and the session is recorded. The proxy service governs traffic leaving a workspace.

What this means for an assessment:

- A workspace cannot open an SSH connection to a destination the SecurSpaces network policy does not allow.
- The private key used toward external systems is never in the workspace.
- Session contents are not decrypted or logged. Commands typed in an SSH session are not recorded by the
  proxy service, so SSH does not provide the request-level visibility that HTTPS does.

What it does not do

- **It is not your organization's own outbound proxy.** If your organization requires outbound traffic to pass through
  an existing proxy of your own, that is configured separately and sits downstream. See External proxy configuration.
- **It is not a web application firewall.** It governs where a workspace may connect and what identity is
  used, not the content of application payloads.
- **It does not inspect SSH session contents.** See How SSH connections are handled.
- **It does not govern Model Context Protocol traffic.** See
  AI Gateway.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Trust boundaries
- Security data flows
- Network policies
- Configure AI Gateway routing
- Ports and protocols
- External proxy configuration

Layered controls with the Citrix platform

August 21, 2026

SecurSpaces enforces its own controls on the development environment. It also runs inside the wider Citrix
platform, so the access layer in front of a workspace can carry controls of its own.

This page separates the two, so you can see which control comes from where when you are assessing coverage or
closing a specific gap.

What SecurSpaces enforces

These are properties of the platform itself and apply however a developer reaches a workspace.

Control	What it enforces
Source code isolation	Code stays in the workspace and is never cloned to the endpoint
Credential brokering	Repository and service credentials are held by the platform and injected at runtime, rather than stored on the developer's machine
Clipboard control	Copying out of the IDE or the secure browser can be blocked
Workspace app confinement	An application published from a workspace can be required to open only in a secure browser
Network egress policy	Outbound traffic is restricted to approved repositories and domains, enforced at the proxy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Control	What it enforces
Audit	Every action is recorded and can be exported to a SIEM

See the control catalogue for the full set
with identifiers.

What the access layer adds

SecurSpaces integrates with the Citrix components you may already run, rather than introducing a second
access stack. Each governs the path to the workspace, not the workspace itself.

Component	What it governs
Citrix DaaS	Delivery of desktops and published applications, including reaching a workspace from a VDA
Secure Private Access	Zero-trust access to private applications, including device posture and conditional access before a session starts
Chrome Enterprise Premium	Browser-level controls on the session itself, such as restrictions on printing, screenshots, and file transfer
Citrix Workspace	A single entry point to all of the above

For what each of these enforces and how to configure it, see that product's own documentation. This page
describes only how they relate to SecurSpaces.

Where the layers matter most

Two cases where an access-layer control closes something SecurSpaces cannot close on its own.

Remote development over SSH. When a developer connects a local IDE over SSH, the workspace controls still
apply to the environment, but the session terminates on a machine SecurSpaces does not govern. Clipboard and
file-transfer restrictions that depend on the browser do not apply. Where that matters, either restrict

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SSH
access by role, or govern the endpoint through the access layer. This limitation is stated in the control catalogue as SDS-DLP-01.

Unmanaged and contractor devices. SecurSpaces keeps source code off the device, which is the larger part
of the risk. Controls on what the *user* can do with the rendered session —screenshot, print, copy to a local
application —belong to the browser or the access layer rather than to the workspace.

Licensing

SecurSpaces is included in the Citrix Platform License, with no new SKU and no user limit. Entitlement for
the other components above depends on your agreement —confirm what your organization holds with your Citrix
account team before you plan a design around them.

Related information

- Security control catalogue
- Trust boundaries
- Security data flows
- Network policies
- Work from a VDA

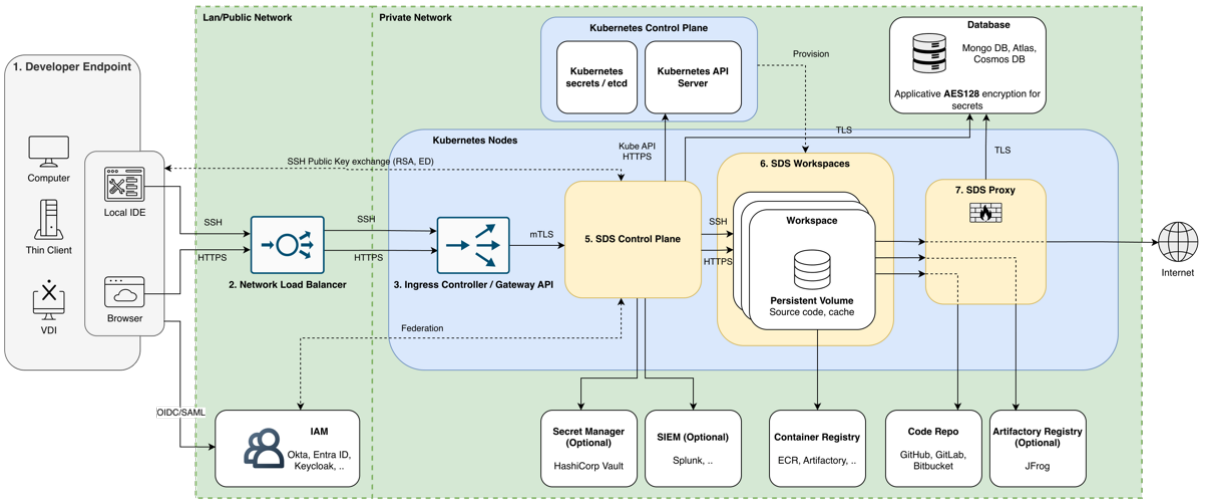
Security data flows

Introduction

This document provides a security-focused overview of the Citrix SecurSpaces™ architecture and the main data flows between core
components and external dependencies. It explains how user access, traffic handling, identity federation, workspace isolation, secrets management, and external integrations help protect sensitive development activity and maintain a secure operating model.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Data flow summary



Flow #	Component	Security data flow
1.	Developer endpoint	The developer endpoint represents the user's device, such as a physical computer, thin client, or virtual desktop infrastructure (VDI) session. Users access SecurSpaces through a local IDE over SSH or through a web browser over HTTPS. Development activity runs remotely, helping prevent sensitive source code from being stored on the local endpoint.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flow #	Component	Security data flow
2.	Network load balancer	The network load balancer provides Layer 4 connectivity between the public or enterprise network and the private SecurSpaces environment. It does not terminate TLS. Instead, it forwards raw TCP sessions and preserves end-to-end encryption until traffic reaches the ingress controller.
3.	Ingress controller or Gateway API	The Kubernetes ingress controller or Gateway API provides Layer 7 traffic routing for SecurSpaces services and acts as the TLS termination point for HTTPS traffic. By terminating TLS at this layer, it can inspect request metadata and apply routing rules that direct traffic to the appropriate internal SecurSpaces services. Downstream communication between the ingress layer and SecurSpaces internal services is encrypted with TLS. The permitted TLS versions and cipher suites for external traffic are set by the customer's ingress configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flow #	Component	Security data flow
4.	Identity and access management	SecurSpaces can integrate with an enterprise identity provider (IdP) to support federated authentication through corporate single sign-on (SSO) using standards such as OpenID Connect (OIDC) or SAML. When this integration is used, user credentials remain managed by the customer's identity system instead of being stored in SecurSpaces. SecurSpaces can also operate as the identity provider and manage user credentials directly.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flow #	Component	Security data flow
5.	SecurSpaces control plane	The SecurSpaces control plane contains the core application logic and acts as the primary security and management hub. It performs SSH and TLS termination for traffic routing and proxying, which enables traffic inspection and detailed auditing of user sessions. It also provides the web administration UI and the SecurSpaces Core API, which hosts the platform's primary business logic. For workspace management, the control plane sends orchestration requests to the Kubernetes control plane to manage the workspace lifecycle, including provisioning, scaling, and decommissioning. Workspace management operations follow a Kubernetes role-based access control (RBAC) model based on least privilege.
6.	SecurSpaces workspaces and storage	Workspaces are ephemeral, containerized environments isolated within the private network. Persistent data is stored on encrypted volumes so that source code and build caches remain secure and isolated if a workspace is restarted or destroyed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flow #	Component	Security data flow
7.	SecurSpaces proxy service	The SecurSpaces proxy service functions as a dedicated workspace egress firewall to help prevent data exfiltration. For what it does and why every connection passes through it, see How the SecurSpaces proxy service works. Outbound traffic from workspaces is routed through this component for URL filtering. This helps ensure that workspaces communicate only with approved external domains and helps prevent unauthorized data transfers to the internet.
8.	External dependencies	SecurSpaces relies on external components to manage application data, container images, code access, and security telemetry. Connectivity between SecurSpaces and these dependencies is managed by the customer using private network integration mechanisms such as private routing, VPN, network peering, or private endpoints. These controls help keep traffic within a secure and controlled network boundary.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flow #	Component	Security data flow
9.	Kubernetes Secrets	SecurSpaces uses native Kubernetes Secrets to manage certificates, access keys, encryption keys, database credentials, and JSON Web Token (JWT) private keys. By default, these secrets are stored in `etcd`, which must be protected with encryption at rest. For environments with higher security requirements, SecurSpaces can integrate with advanced Kubernetes mechanisms such as the Secrets Store CSI Driver or External Secrets Operators.

External dependency data flows

Dependency	Security data flow
MongoDB database	Stores application state. SecurSpaces uses AES-128-GCM to protect sensitive data at the application layer, including developer personal secrets.
Code repositories	Git repository integrations are routed through the SecurSpaces proxy service, enabling SSO and enforcing SecurSpaces security policies for Git repository access and management.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Dependency	Security data flow
Container registry	Hosts the container images used by SecurSpaces workspaces. Images can be stored across one or more registries, depending on the customer environment. A container registry also provides images for core SecurSpaces components, such as the SecurSpaces control plane and SecurSpaces proxy service. These core component image flows are typically omitted from high-level data-flow diagrams because they are not configurable in SecurSpaces.
SIEM	Receives audit logs and security events from SecurSpaces for centralized monitoring, compliance, and alerting. Examples include SIEM platforms such as Splunk.

Kubernetes Secrets data flows

Secret type	Use
Certificates	TLS keys for the SecurSpaces control plane and internal components.
Access keys	SSH private keys for workspace authentication. Personal SSH identities use Ed25519.
Encryption keys	AES-128 keys used with AES-GCM for application-level data protection.
Database credentials	Passwords and connection credentials for MongoDB.
JWT private keys	Keys used for secure token signing.

Ports and cryptography

For the externally exposed ports and the algorithms used for each purpose, see Ports and protocols and Cryptography. For where the deployment separates one level of trust from another, see Trust boundaries.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Security considerations

SecurSpaces uses a layered security model across the endpoint, network, application, workspace, and infrastructure layers. The customer remains responsible for the security of the hosting environment, including network boundaries, encryption at rest for infrastructure services, Kubernetes cluster hardening, and protection of `etcd`.

Where the deployment requires stronger secrets protection, customers can use Kubernetes-native integrations such as the Secrets Store CSI Driver or External Secrets Operators to connect SecurSpaces with an enterprise secrets management platform.

Network policies

Network policies control the outbound network traffic that workspaces are allowed to generate. With a network
policy you can monitor where a workspace connects, restrict it to an approved set of domains and IP addresses,
or inspect its traffic —and see every one of those events in the **Audit** dashboard.

This page explains the model. For the tasks, see:

- Create a network policy
- Assign and enforce a policy
- Monitor and troubleshoot
- Network policy fields

Policy behaviors

Every policy applies one of three behaviors to outgoing traffic.

Behavior	What it does
Monitor	TCP traffic is allowed and logged without restriction. Use this to see where a workspace connects without limiting it to an allow list.
Restrict	Traffic is blocked by default. Only attached resources and the domains and IP addresses on the allow list can be reached. Blocked requests are logged.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Behavior	What it does
Inspect	TCP traffic content is inspected and reported. Use this when you need the deepest visibility into what a workspace sends.

> **Note:**
>
> You configure **monitor** and **restrict** behavior when you create or edit a policy. **Inspect** is provided
> only through the built-in **Inspect Traffic (default, expert)** policy. You can apply it to workspaces, but
> you cannot author a new inspect policy in the policy editor.

What happens when a policy is attached

Attaching any policy changes how the workspace reaches the network, regardless of which behavior the policy
applies.

A workspace with a policy attached is labeled as hardened, and a Kubernetes NetworkPolicy then permits it to
send traffic only to SecurSpaces service pods. All outbound traffic therefore goes through the SecurSpaces
proxy service. DNS continues to work because it is served by a SecurSpaces service.

> **Important:**
>
> The moment any network policy is attached to a workspace —even a **monitor** policy —the workspace's
> outbound traffic is routed through the SecurSpaces proxy service, and non-DNS UDP traffic is dropped.
> A workspace with
> **no** policy attached is not affected. Keep this in mind if your workloads rely on UDP, for example some
> VPN, QUIC, or custom UDP-based protocols. Those connections stop working once a policy is attached,
> whatever the policy's behavior.

Before you use restrict or inspect

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Note:**
>
> Restrict and inspect policies can cause some applications to stop working, because they change what the
> workspace can reach. They are recommended for experienced administrators.
>
> Under an inspect policy, an application that reads certificates from a custom location inside the container
> is likely to fail. Check with your platform administrator before applying **Inspect Traffic (default, expert)** to a workspace that depends on one.

Scopes

Network policies are defined at three scope levels. A policy defined at a higher scope can be applied —and
optionally enforced —on every scope nested beneath it.

Scope	Who manages it	Typical use
Platform	Platform administrators, security officers	Baseline policy for the whole deployment
Organization	Organization owners, security officers	Policy for all projects in an organization
Project	Project owners, security officers	Policy for all workspaces in a project

You cannot define a policy at the workspace level. A workspace can only have an existing policy *applied* to
it, selected from the policies defined at the platform, organization, or project scope.

Applied and enforced

Two concepts govern how a policy reaches a workspace.

Term	Meaning
Applied	The policy is selected as the active policy for a scope or workspace.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Term	Meaning
Enforced	The policy is locked in for the current scope and every scope nested beneath it. Lower scopes and their workspaces inherit it and cannot replace it.

Who can manage network policies

Creating, editing, selecting, and enforcing network policies requires the **Security** permission set to **Manage**. The **Security** permission has three levels: **No Access**, **Access**, and **Manage**.

Manage belongs to:

- Security officers
- Organization owners
- Project owners, and any custom role granted the **Manage** level

Developers and other project members cannot create, change, or assign network policies, even for their own
workspaces. With the **Access** level they can see which policies apply, for example through **View summary**,
but the selection is read-only for them.

When a developer creates a personal workspace, its network policies are applied automatically from the project
and platform settings. Policies marked as required are enforced and locked, and other configured policies are
applied as the workspace's selected policies. The developer does not choose them.

The default policies

When a project is created, Citrix SecurSpaces™ adds three ready-to-use policies at project scope.

Policy	Behavior	Use it when
Monitor Traffic (default)	Monitor	You want to log outgoing TCP traffic and generate audit events without restricting it to an allow list.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Policy	Behavior	Use it when
Restrict Traffic (default, expert)	Restrict	You want to block all outbound traffic except attached resources and an approved allow list.
Inspect Traffic (default, expert)	Inspect	You want to inspect TCP content and report it to the audit log. This behavior is available only as this built-in policy.

Restrict Traffic (default, expert) ships with a starter allow list covering common developer package sources, so standard tooling keeps working:

- `*.ubuntu.com`
- `*.nodesource.com`, `*.yarnpkg.com`
- `*.npmjs.org`, `*.npm.im`, `*.npm.me`, `*.npm.red`
- `python.org`, `*.pypi.org`, `*.pythonhosted.org`
- `open-vsx.org`
- `opensuse.org`, `sourceforge.net`, `virtualbox.org`
- `*.windows.net`

Use these defaults as they are, duplicate them as a starting point, or create your own policies.

Where policies are configured

Location	What you do there
Platform settings > Security settings > Network policy overview	Review which policy is applied at each scope and whether it is enforced, and download a report. Read-only.
Workspace settings > Network policy, at platform, organization, or project level	Create, edit, select, enforce, and test the policy for that scope.
A workspace's **Security settings > Network security**	Select the policy for an individual workspace. Read-only without the **Manage** level.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Create a network policy
- Assign and enforce a policy
- Monitor and troubleshoot
- Network policy fields
- Audit dashboard

Create a network policy

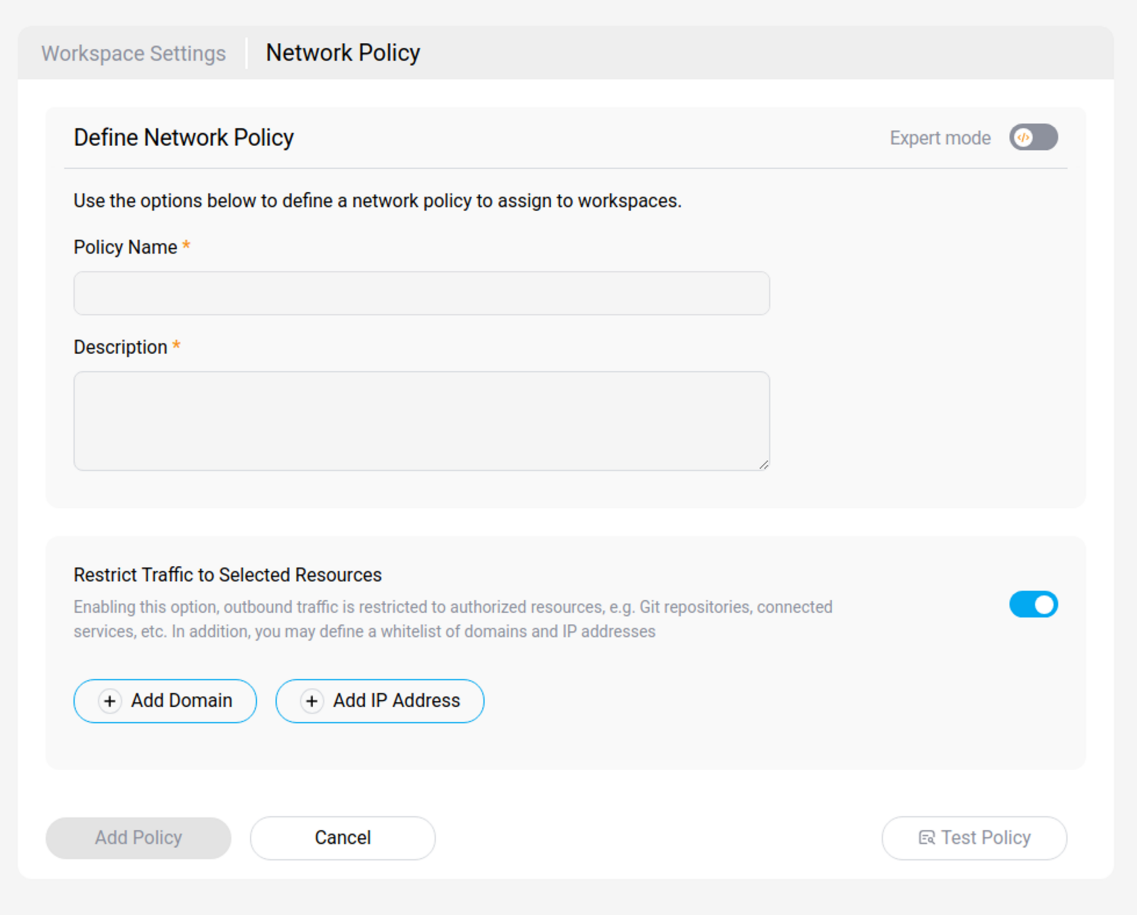
You create and edit policies from a **Workspace settings > Network policy** page at the scope you manage:
platform, organization, or project.

You need the **Security** permission set to **Manage**. See
Who can manage network policies.

Create the policy

1. Go to **Workspace settings > Network policy** for your scope.
2. Select **Create policy**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



3. Enter a **Policy name** and a **Description**. Both are required. Use a description that explains what the
   policy allows or blocks, so other administrators understand its intent.
4. Choose how to configure the rest of the policy:
   - Leave the **Expert mode** toggle off to use the guided form. See Basic mode.
   - Turn the **Expert mode** toggle on to edit the policy as YAML. See Expert mode.
5. (Optional) Select **Test policy** to validate the policy before you save it. See
   Test a policy.
6. Save the policy.

The basic form and the expert YAML editor describe the same policy. You can switch between them with the
Expert mode toggle while you work.

Basic mode

Basic mode gives you a guided form for the most common settings.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Set **Restrict traffic to selected resources**:
   - **Off** —The policy monitors traffic. Everything is allowed and logged.
   - **On** —The policy blocks all outbound traffic except attached resources and the resources on your
     allow list.
2. If you turned restriction on, build the allow list:
   - Select **Add domain** to allow a domain. Optionally turn on **Include subdomains** to also allow every
     subdomain of that domain.
   - Select **Add IP address** to allow a single IPv4 address or a CIDR range, for example `10.0.0.0/24`.
3. If your platform uses an external proxy, a **custom endpoints via proxy** section appears. Use it to list
   the domains and IP addresses that the workspace can reach through the proxy. This section is hidden when
   no external proxy is configured.
4. Save the policy.

Expert mode

Expert mode replaces the form with a YAML editor, giving you control over every field. Turn on the **Expert**
mode toggle to use it.

For every field, its type, and its validation rules, see
Network policy fields.

```yaml
name: Backend services policy
description: Restrict workspaces to internal package mirrors and the
   corporate API.
restrictedTraffic: true
whitelistedDomains:
  - domain: pypi.internal.example.com
    includeSubdomains: false
  - domain: example.com
    includeSubdomains: true
whitelistedIps:
  - 10.0.0.0/24
  - 192.168.1.50
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Build the allow list

When a policy restricts traffic, the workspace can reach only:

- **Attached resources** —Any resource attached to the workspace, such as repositories, SSH services, and
  HTTP services, is always reachable. Attached resources keep working without extra configuration.
- **Allowed domains** —Every domain on the allow list.
- **Allowed IP addresses** —Every IPv4 address or CIDR range on the allow list.

How domains are matched

- A domain entry with **Include subdomains** turned off matches only that exact domain.
- A domain entry with **Include subdomains** turned on matches the domain itself and any subdomain. For
  example, `example.com` with subdomains enabled allows `example.com` and `*.example.com`, such as
  `api.example.com`.

How IP addresses are matched

Enter a single IPv4 address, for example `192.168.1.50`, or a CIDR range, for example `10.0.0.0/24`. The
policy editor validates the format before you can save. IPv6 addresses are not accepted.

Related information

- Network policies
- Assign and enforce a policy
- Network policy fields

Assign and enforce a network policy

Creating a policy does not change any workspace on its own. You assign it to a scope, and optionally enforce
it down the hierarchy.

You need the **Security** permission set to **Manage**. See
Who can manage network policies.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Assign a policy to a scope

1. Go to **Workspace settings > Network policy** for the scope you manage.
2. Select the policy you want to apply from the list of available policies.
3. To lock the policy in for this scope and everything nested beneath it, turn on **Enforce network policy**.
   When enforcement is on:
   - The policy applies to all current and future workspaces in the scope.
   - Nested scopes, and their workspaces, inherit the policy and cannot replace it.
4. Save your changes.

If you do not enforce the policy, lower scopes and individual workspaces can still choose a different policy.
Enforcement is what guarantees consistent behavior across an organization or project.

Select a policy for a workspace

A project owner or security officer can set the network policy for an individual workspace when creating or
editing the workspace. If a parent scope enforces a policy, that policy is inherited and the selection is locked.

1. In the workspace creation or edit flow, open **Security settings**.
2. Under **Network security**, choose a policy. By default a workspace starts with **No policy selected**.
3. Select **View summary** to review exactly what the policy allows or blocks before you apply it.
4. Save the workspace.

> **Note:**
>
> This requires the **Security** permission set to **Manage**. Developers cannot change a workspace's policy.
> When a developer creates a personal workspace, its policies are applied automatically from the project and
> platform settings: required policies are enforced, and other configured policies are applied.

Test a policy

Before you rely on a policy, test it to confirm it allows what you expect and blocks what you do not.

1. Open the policy in **Workspace settings > Network policy**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

2. Select **Test policy**.
3. In the test dialog, provide the destinations you want to check.
4. Review the result to confirm whether each destination is allowed or blocked under the policy.

Testing is especially useful for restrict and inspect policies, where an incomplete allow list can break tooling inside the workspace.

Review the network policy overview

To audit your configuration across the whole deployment, go to **Platform settings > Security settings >**
Network policy overview.

The overview shows, for the platform and each organization:

- The **applied policy** at that scope.
- The **status**, either **Enforced** or **Not enforced**.

Select **Download report** to export the overview for record-keeping or review.

Related information

- Network policies
- Create a network policy
- Monitor and troubleshoot

Monitor and troubleshoot network policies

Every network policy behavior feeds the **Audit** dashboard, which is your primary tool for understanding and
fixing policy effects.

- **Monitored traffic** generates log events for all outgoing connections.
- **Restricted traffic** logs every blocked request, so you can see exactly which destination was denied.
- **Inspected traffic** reports inspected TCP content and dropped UDP traffic.

Find the traffic logs

Network events appear in the **Audit** dashboard for the project, in the **Live system event log**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. Open **Audit** for the project. You need the **Security** permission set to **Access** or **Manage**.
2. Find the network events, for example DNS requests showing the domain a workspace tried to reach.
3. Select **Filter** to narrow the log by event type, severity, workspace, user, or date, or use the search bar to find a specific destination.
4. Expand a row to see the event's full details, including the description of what triggered it.

Use a monitor policy to build an allow list

A monitor-only policy is the recommended way to discover what a workspace needs before you lock it down.

1. Apply the **Monitor Traffic (default)** policy, or any policy with restriction turned off, to the workspace.
2. Use the workspace normally so its tools generate their usual network calls.
3. In the audit log, review the network events to see which domains and IP addresses the workspace actually
   reached.
4. Create a restrict policy and add those destinations to the allow list. See
   Build the allow list.
5. Switch the workspace to the restrict policy. Any destination you missed now appears as a blocked request
   in the same audit log, so you can refine the allow list and repeat.

> **Tip:**
>
> Attached resources, such as repositories, SSH services, and HTTP services, are always reachable under a
> restrict policy. You do not need to add them to the allow list.

Common issues

An application cannot reach a service it needs

The destination is probably missing from the allow list. Check the Audit dashboard for the blocked request,
then add the domain or IP address to the policy. If the service uses subdomains, turn on **Include subdomains**.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

A tool that uses UDP stops working after a policy is applied

Any attached network policy —monitor, restrict, or inspect —drops non-DNS UDP traffic by design. DNS still
works, but other UDP-based protocols do not. If a workload depends on UDP, it cannot run in a workspace that
has a network policy attached.

See
What happens when a policy is attached.

A workspace ignores the policy you selected

A parent scope is probably enforcing a different policy, or the policy was applied automatically from project
settings. Check the network policy overview to see which scope enforces the inherited policy. Only security
officers, organization owners, and project owners can change a workspace's policy.

Related information

- Network policies
- Assign and enforce a policy
- Audit dashboard

Technical security overview for SecurSpaces Flex

September 23, 2026

This article describes the security architecture, shared responsibility model, data handling practices, and compliance posture of Citrix SecurSpaces™ Flex. It is intended for security teams, procurement reviewers, and CISOs evaluating the service.

Architecture overview

Citrix SecurSpaces Flex is built on three layers:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Layer	Description	Operated by
Control plane	Manages workspace provisioning, access, policies, and lifecycle. Built on Azure Kubernetes Service (AKS) and Azure Cosmos DB.	Citrix
Execution plane	Runs isolated, sandboxed Linux workspaces. Each workspace is a containerized environment with defined vCPU, RAM, and persistent storage.	Citrix
Integration layer	Connects workspaces to customer-owned systems: identity providers, Git repositories, CI/CD pipelines, secrets managers, and data sources.	Customer + Citrix (connectivity infrastructure)

All components run inside Citrix-managed Azure subscriptions in the Azure regions agreed for your deployment. The customer's application data, container images, secrets, and identity configuration remain under customer ownership and control.

Shared responsibility model

Citrix SecurSpaces Flex uses a clear division of responsibilities between Citrix and the customer across configuration and management, infrastructure, and ongoing support.

Configuration and management

Customer	Citrix
Workspace image hosting and management	Citrix SecurSpaces platform configuration and updates
Workspace and template management	Citrix SecurSpaces region management (upon request)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Customer	Citrix
User accounts, access, and permissions	Platform-wide Citrix SecurSpaces configurations
Organizations, projects, and secrets	Workspace specifications
Network policies and security monitoring	Backups, DR planning, and availability monitoring

Infrastructure

Customer	Citrix
Authentication (IdP)	Azure AKS, Cosmos DB, and Azure tenant
Source code repositories	Landing zone VM
Artifactories	Citrix SecurSpaces control plane scaling (up/down)
Azure vNET peering / VPN gateway (shared)	Azure vNET peering / VPN gateway (shared)

Ongoing support

Customer	Citrix
End-user support	Citrix SecurSpaces and Azure infrastructure support Citrix SecurSpaces product support

Availability commitments for the service, and the remedy if Citrix misses them, are set out in the Platform Flex SLA.

> **Note:** Azure vNET peering and the VPN gateway are a shared responsibility. Citrix provisions and operates the Citrix-managed side of the connection; the customer configures and maintains their side. See Network connections for details.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Workspace isolation

Each workspace is an isolated, sandboxed Linux container. Isolation is enforced at the Kubernetes runtime level.

- **No shared file systems between workspaces.** Each workspace has its own Persistent Volume Claim (PVC) for persistent state.
- **No lateral movement.** A compromised workspace cannot access other workspaces or the control plane.
- **Controlled ingress and egress.** Workspace network traffic is subject to policy-defined rules. Egress to the internet and to customer networks is controlled; unrestricted outbound access is not the default.
- **AI agent containment.** AI coding agents run inside the same workspace sandbox as human sessions. Agent processes are subject to the same egress controls, DLP rules, and audit logging as any other workspace activity.

Data loss prevention (DLP)

Citrix SecurSpaces Flex includes built-in DLP controls.

- **Clipboard DLP** —Clipboard operations between the workspace and the user's local device are policy-controlled. Administrators can restrict or monitor copy/paste behavior.
- **Session monitoring** —Workspace activity can be subject to session recording and behavioral monitoring for compliance and forensic purposes.
- **Audit logging** —All administrative and developer actions generate audit log entries. Logs are retained and exportable for SIEM integration.

Data residency and encryption

Data residency

Workspace execution runs in the Azure region designated as your primary region, and in any secondary regions provisioned for your deployment. Citrix does not transfer workspace data across regions outside of the regions agreed for your deployment.

Customer-owned assets —container images, secrets, source code, identity configurations —remain in customer-owned infrastructure and are not replicated into Citrix-managed systems except as required for workspace execution (for example, pulling a container image at workspace launch).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Encryption

- **In transit** —All communication between workspace clients and the platform uses TLS 1.2 or higher. Communication between internal platform components is encrypted.
- **At rest** —Persistent workspace storage (PVCs) is encrypted at rest using Azure-managed encryption keys. Control plane configuration data stored in Cosmos DB is encrypted at rest.

Role-based access control (RBAC)

Citrix SecurSpaces Flex enforces role-based access at the platform level.

- **Administrators** manage deployments, workspace templates, user access, and policy configuration.
- **Developers** access workspaces assigned to them. They cannot access other developers' workspaces or administrative functions.
- Roles and permissions are managed in the Citrix SecurSpaces console and integrated with the customer's IdP.

More information

- SecurSpaces Flex overview —the starting point for all Flex documentation
- Network connections
- Workspace templates
- Platform Flex SLA
- Citrix Trust Center

Network connections for SecurSpaces Flex

This article describes how Citrix SecurSpaces™ Flex workspaces connect to customer-owned systems, the available connectivity options, and what each option enables.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview

Citrix SecurSpaces Flex workspaces run in a Citrix-managed Azure tenant. To enable developer workflows, workspaces typically need access to customer-owned systems —such as identity providers, Git repositories, CI/CD pipelines, artifact stores, and secrets managers.

Connectivity between the Citrix-managed execution plane and customer-owned systems is established through an integration layer. This layer uses standard Azure networking primitives. The customer is responsible for configuring their side of the connection.

Connectivity options

Citrix SecurSpaces Flex supports two connectivity options for connecting workspaces to customer-owned infrastructure.

Azure virtual network (vNET) peering

Azure vNET peering creates a private, low-latency network path between the Citrix-managed Azure virtual network and a virtual network in the customer's Azure subscription.

Use this option when:

- Your customer-owned systems (IdP, Git, CI/CD, secrets manager) are hosted in Azure.
- You want private, high-performance connectivity without traffic routing over the public internet.

What the customer configures:

- A virtual network in the customer's Azure subscription.
- A peering request from the customer vNET to the Citrix-managed vNET (Citrix provides the resource identifier during setup).
- Network security group (NSG) rules and routing to allow workspace traffic to reach the target systems.

VPN gateway

A VPN gateway connects the Citrix-managed Azure vNET to the customer network using an IPsec/IKE VPN tunnel.

Use this option when:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Your customer-owned systems are on-premises or in a non-Azure environment.
- Azure vNET peering is not available or is not suitable for your network architecture.

What the customer configures:

- A VPN device or gateway at the customer network boundary.
- An IPsec/IKE tunnel configuration (Citrix provides the remote gateway endpoint during setup).
- Routing rules to direct workspace traffic to the appropriate on-premises systems.

What workspaces can reach

Once connectivity is established, workspaces can reach any system accessible within the connected network, subject to the routing and security group rules you configure. Common integration targets include:

System type	Examples
Identity providers	Azure Active Directory, Okta, any SAML 2.0 or OpenID Connect provider
Source code repositories	GitHub Enterprise, GitLab self-managed, Bitbucket Server, Azure Repos
Artifact repositories	Artifactory, Nexus, Azure Artifacts
CI/CD systems	Jenkins, Azure DevOps, CircleCI
Secrets managers	HashiCorp Vault, Azure Key Vault
Internal data sources	Databases, data lakes, internal APIs

No connectivity (default state)

By default, before any connectivity option is configured, workspaces have access to:

- The public internet (subject to egress policy rules you define in the Citrix SecurSpaces console).
- Citrix-managed platform services (required for workspace operation).

Workspaces do **not** have default access to customer-owned private networks. Connectivity must be explicitly configured using one of the options above.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Responsibility summary

Task	Responsible party
Providing the Citrix-managed vNET resource identifier or VPN endpoint	Citrix
Configuring the customer-side peering or VPN tunnel	Customer
Configuring NSG rules and routing on the customer network	Customer
Managing access to customer-owned systems once connected	Customer
Enforcing egress policy within workspaces	Citrix (platform) + Customer (policy configuration)

More information

- SecurSpaces Flex overview —the starting point for all Flex documentation
- Technical security overview
- Workspace templates
- Azure vNET peering documentation —Microsoft Azure
- Azure VPN Gateway documentation —Microsoft Azure

Reference

August 19, 2026

Look-up material: exact names, values, and limits. These pages state what is, without explaining how to use
it —the task sections do that and link here.

Ordered alphabetically, because reference is searched rather than read.

Page	What it covers
Audit event catalog	Every audit event, by category, with its attributes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Page	What it covers
Container image requirements	What a workspace image must provide
Cryptography	Algorithms in use, and where each applies
Mount point limits and behavior	Quotas, deletion semantics, metrics
Network destinations	External addresses needed to install and license
Network policy fields	Expert-mode YAML fields and their validation
Ports and protocols	Ports exposed, and the SSH forwarding model
REST API	The public API surface
SDS CLI	Every installer command and flag
Supported IDEs	Which editors work, and how each connects
System requirements	Kubernetes, networking, storage, tooling

Audit event catalog

September 29, 2026

The tables below offers a quick reference to events monitored in real time on the Citrix SecurSpaces™ platform. These events are systematically captured using standardized methods and are available in the audit section. They can be easily exported in common formats for integration with Security Information and Event Management (SIEM) systems, supporting comprehensive monitoring and analysis.

All events

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
	All	Attributes shared by all events		id, timestamp, user_id, user_name, session_id, project_id, project_name, workspace_id, work-space_name, severity

Authentication

ID	Category	Event Type	Event Description	Attributes
1	Authentication	Login	The user logged on to the platform	
2	Authentication	Logout	The user logged out of the platform	
3	Authentication	SessionStart	The user started a workspace session	
4	Authentication	SessionEnd	The user ended a workspace session	
5	Authentication	SessionInterrupt	The user workspace session has been interrupted	

Authorization

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
6	User Authorization	UserBlocked	The user has been blocked	user_id, user_name, role_name
7	User Authorization	UserUnblocked	The user has been unblocked	user_id, user_name, role_name
9	Workspace Authorization	SharedWithUser	User shares workspace with another user	user_id, user_name
10	Workspace Authorization	UnsharedWithUser	User revokes previously shared workspace access.	user_id, user_name

Data Security

ID	Category	Event Type	Event Description	Attributes
11	Data Security	Copy	In the workspace, the user copies data to the clipboard	data, is_secret, is_code
12	Data Security	Paste	In the workspace, the user pastes copied data into a new location	data, is_secret, is_code
13	Data Security	Cut	In the workspace, the user cuts selected data for potential relocation	data, is_secret, is_code
14	Data Security	Clipboard	In the secure browser, data is copied, cut, or pasted	data, is_secret, is_code

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
15	Data Security	ShareClipboardUrl	In the secure browser, the user shares a URL or link stored in the clipboard	data, is_secret, is_code
16	Data Security	Upload	Sends a file or data from a local device to a remote environment	data, is_secret, is_code
17	Data Security	UploadLargeFile	Sends large-sized files from a local device to a remote environment	data, is_secret, is_code
18	Data Security	Download	Retrieves a file or data from a remote environment to a local device	data, is_secret, is_code
19	Data Security	DownloadLargeFile	Retrieves large-sized files from a remote environment to a local device	data, is_secret, is_code
20	Data Security	SupervisedCopy	In the workspace, the copy action under supervision or monitoring	data, is_secret, is_code

System

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
21	System	WorkspaceSpecsUpdated	Modifications or updates made to the specifications of a workspace	
60	System	BulkWorkspaceUpdate	A bulk update of workspaces started from the Update Center	
61	System	BulkWorkspacePause	A bulk pause of workspaces started from the Update Center	

SecurSpaces Flex managed tier events

In Citrix SecurSpaces Flex, managed workspace size tiers also generate ResourceAccess events. SecurSpaces
records successful tier availability changes and bulk tier-version updates, and records refused attempts to
modify Citrix-managed tier definitions. The event includes the acting user, timestamp, managed tier version,
and result.

Data Security

ID	Category	Event Type	Event Description	Attributes
22	SecureBrowserNavigation	SecureBrowserNavigation	Ensures secure browsing practices during navigation	url, title allowed
23	VSCodeExtensionInstalled	VSCodeExtensionInstalled	Installation of an extension within Visual Studio Code	extension_name, extension_id extension_uuid

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
24	AccountManagement	UserAddedToProject	Addition of a user to a specific project	user_id, user_name, role_name
25	AccountManagement	UserRemovedFromProject	Removal of a user from a specific project	
26	AccountManagement	RoleChanged	Modification or alteration of a user roles and permissions	
27	AccountManagement	UserCreated	Creation of a new user profile or account	
28	AccountManagement	UserDeleted	Deletion or removal of a user profile or account	

Network Traffic

ID	Category	Event Type	Event Description	Attributes
29	SSHCommand	SSHCommand	Execution of a command via Secure Shell (SSH)	issuer, command, type, destination, commit, request, git_branch
30	ExternalSSHCommand	ExternalSSHCommand	Execution of an external command through Secure Shell (SSH)	service_id, command, destination, type
31	HTTPRequest	HTTPRequest	Transmission of a request using Hypertext Transfer Protocol (HTTP)	issuer, destination, request_type, blocked, status_code, browser_id

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
32	GitOverHTTP	GitOverHTTP	Git operations performed over HTTP protocol	issuer, command, destination, request
33	TCPForwarding	TCPForwarding	Forwarding of Transmission Control Protocol (TCP) traffic	destination_address
34	DNS	DNS	Domain Name System (DNS) operations or requests	domain, address, inspected
59	AIRequest	AIRequest	A request from a workspace to an AI service, recorded by the AI Gateway	host, url, method, status_code, blocked, model
35	ResourceAccess	Created	A resource is newly created within the system	resource_name, resource_id, action_type, resource_type, o_auth_app
36	ResourceAccess	Imported	Data or information is brought in from an external source	resource_name, resource_id, action_type, resource_type, o_auth_app
37	ResourceAccess	ManuallyImported	Specific data is manually transferred or imported into the system	resource_name, resource_id, action_type, resource_type, o_auth_app
38	ResourceAccess	Updated	Existing data or information undergoes modification or refresh within the system	resource_name, resource_id, action_type, resource_type, o_auth_app

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
39	ResourceAccess	SharedWithUsers	Resource is shared with multiple users within the system	resource_name, resource_id, action_type, resource_type, o_auth_app
40	ResourceAccess	SharedPublicly	Resource is made accessible to the public users	resource_name, resource_id, action_type, resource_type, o_auth_app
41	ResourceAccess	WorkspaceAttached	Resource is attached to a workspace	resource_name, resource_id, action_type, resource_type, o_auth_app
42	ResourceAccess	WorkspaceDetached	Removal of resource from a workspace	resource_name, resource_id, action_type, resource_type, o_auth_app
43	ResourceAccess	Deleted	A resource is removed or deleted from the system	resource_name, resource_id, action_type, resource_type, o_auth_app
44	ResourceAccess	Repository	Management of a Git application used for code or data storage	resource_name, resource_id, action_type, resource_type, o_auth_app
45	ResourceAccess	Bucket	Container utilized for data storage, commonly used in cloud computing	resource_name, resource_id, action_type, resource_type, o_auth_app

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
46	ResourceAccess	Secret	Sensitive data such as passwords, keys, or tokens	resource_name, resource_id, action_type, resource_type, o_auth_app
47	ResourceAccess	Connected_service	Establishment or utilization of an external service or integration within the system	resource_name, resource_id, action_type, resource_type, o_auth_app
48	ResourceAccess	Network_policy	Setting rules or configurations governing network behavior or access	resource_name, resource_id, action_type, resource_type, o_auth_app
49	ResourceAccess	Image	Handling representations or snapshots of data, often used in computing environments	resource_name, resource_id, action_type, resource_type, o_auth_app
50	ResourceAccess	Credential	Management of information used for authentication or access control	resource_name, resource_id, action_type, resource_type, o_auth_app
51	ResourceAccess	Workspace_app	Utilization or management of a workspace application	resource_name, resource_id, action_type, resource_type, o_auth_app
52	ResourceAccess	Startup_script	Execution or management of scripts or instructions during system startup	resource_name, resource_id, action_type, resource_type, o_auth_app

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ID	Category	Event Type	Event Description	Attributes
53	ResourceAccess	Workspace	Management or utilization of a coding environment for collaborative work	resource_name, resource_id, action_type, resource_type, o_auth_app
54	ResourceAccess	GitHub	Utilization or interaction with the GitHub OAuth application for various purpose	resource_name, resource_id, action_type, resource_type, o_auth_app
55	ResourceAccess	GitLab	Utilization or interaction with the GitLab OAuth application for various purposes	resource_name, resource_id, action_type, resource_type, o_auth_app
56	ResourceAccess	Bitbucket	Utilization or interaction with the Bitbucket OAuth application for various purposes	resource_name, resource_id, action_type, resource_type, o_auth_app
57	ResourceAccess	AzureDevOps	Utilization or interaction with the AzureDevOps OAuth application for various purposes	resource_name, resource_id, action_type, resource_type, o_auth_app
58	ResourceAccess	JFrog	Utilization or interaction with the JFrog OAuth application for various purposes	resource_name, resource_id, action_type, resource_type, o_auth_app

Attributes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Attributes	Attribute Description
action_type	Action type
address	DNS address
allowed	Flag indicating whether navigation is allowed
blocked	Flag indicating whether the request is blocked
browser_id	Browser ID
command	The SSH command executed
commit	The related commit hash
data	Clipboard data, if applied
destination	The git service name
destination	The external service name
destination	The destination name
destination_address	Destination address
domain	Domain name
extension_id	ID of the Visual Studio Code extension
extension_name	Name of the Visual Studio Code extension
extension_uuid	UUID of the Visual Studio Code extension
git_branch	The git branch name, if applied
host	The AI service host contacted
id	Event ID
inspected	Flag indicating whether it request has been inspected
is_code	Code detection flag
is_secret	Secret detection flag
issuer	Email or user ID of the issuer
method	The HTTP method used
model	The AI model named in the request, if present
o_auth_app	Third party app name, if applied
project_id	Project ID
project_name	Project name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Attributes	Attribute Description
request	The type of request
request_type	Request type
resource_id	Resource ID
resource_name	Resource name
resource_type	Resource type
role_name	The user role on the platform
role_name	The rolename in the project, if applied
service_id	The service ID
session_id	IDE session ID
severity	Severity 0-3 = Low - 4-6 = Medium - 7-8 = High - 9-10 = Critical
status_code	HTTP status code
timestamp	Date on which the event was recorded
title	Title of the webpage
type	Push or pull
url	URL of the webpage
user_id	The user id on the platform
user_name	The username on the platform
workspace_id	Workspace ID
workspace_name	Workspace name

Container image requirements

Every container image used for a Citrix SecurSpaces™ workspace must meet the requirements on this page.
An image that does not meet them may fail to start, or start without the Cloud IDE working.

At minimum, an image must provide:

1. An SSH client.
2. Git and Git LFS.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

3. A user named `developer` with UID 1000.

The rest of this page states the full requirements. To add an image to a project once it meets them, see
Container images. To build one, see
Workspace images.

Core packages

Required on all distributions, for compatibility with the platform and with the IDE server that runs inside
the workspace.

Package	Purpose
`glibc >= 2.28`	C standard library required by the IDE server. On Alpine, use `musl` with `gcompat` instead.
`libstdc++ >= 3.4.25`	C++ standard library
`libgcc`	GCC runtime
`bash`	Default shell required by the IDE server
`curl` or `wget`	Downloads the server and external dependencies
`git`	Source control integration and workspace initialization
`git-lfs`	Handling of large files and binary assets
`openssh-client` / `openssh-clients`	SSH-based Git authentication and remote access

User account

The image must be provisioned with this account and home directory. The values are not configurable. The
workspace container runs as UID 1000, and the persistent volume is mounted at `/home/developer`, so an image
that uses a different user or home directory cannot write to it.

Property	Value
User name	`developer`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Property	Value
Home directory	`/home/developer`
UID	`1000`
GID	`1000`
Login shell	`/bin/bash`

End the Dockerfile with `USER 1000` and `WORKDIR /home/developer` so the workspace starts as the right user
in the right directory.

How the platform checks an image

When an administrator adds an image, or adds a tag to an existing image, SecurSpaces tests it before anyone
can use it. It starts a temporary **ghost workspace** from the image and runs a check inside it. The image is
marked ready only if the check passes.

The check tests four things, all by running the command or looking the user up inside the container:

Checked	Failure message
`ssh` is on the `PATH`	`SSH is not present on the image`
`git` is on the `PATH`	`Git is not present on the image`
`git-lfs` is on the `PATH`	`Git-lfs is not present on the image`
A user named `developer` exists	`User(ID 1000)could not be accessed: the user developer is not present on the image`

If any fail, the image is flagged **The image is not compatible with our workspace infrastructure**, followed
by **The following requirements are not met:** and the specific failures. The image cannot be used until it
is rebuilt and re-added.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Important:**
>
> The check covers those four items only. Everything else on this page —the UID and GID, the home directory,
> the login shell, the startup script directory, the architecture, and the `glibc` version —is still required, but an image can pass the check and fail later at runtime. Passing is a floor, not a certification.

Read the result

The compatibility check appears in the image's log as **Checking image compatibility with our workspace**
infrastructure, in the **Setting up the workspace** phase. Open it from the image entry in **Resources > Container Images**.

The log also shows the ghost workspace pulling your image, so it is a useful place to confirm the image was
found and how large it is.

Distribution-specific packages

Ubuntu and Debian

Ubuntu 20.04 or later, or Debian 11 or later, is recommended for `glibc` compatibility.

Package	Purpose
`tar`	Unpacks the server bundle
`ca-certificates`	Secure HTTPS connections when downloading extensions

Red Hat Enterprise Linux

RHEL 8.x and 9.x meet the core requirements.

Package	Purpose
`tar`	Unpacks the server bundle

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CentOS Stream and RHEL-compatible distributions

CentOS Stream 8 and 9, Rocky Linux 8 and 9, and AlmaLinux 8 and 9 meet the core requirements.

CentOS 7 does not meet the requirements, because it ships with `glibc 2.17`.

Package	Purpose
`tar`	Unpacks the server bundle

Alpine Linux 3.16 or later

Package	Purpose
`musl >= 1.2.3`	Native C standard library for Alpine
`gcompat`	Provides the glibc compatibility layer

> **Note:**
>
> Alpine uses `musl` rather than `glibc`. Core features work through compatibility layers, but Ubuntu and
> RHEL-based images give the most stable experience with the IDE server.

Architecture

SecurSpaces runs on `linux` / `amd64` (x86_64). Workspace images must be built for that architecture. Other
architectures, including ARM, are not supported.

Troubleshooting

`GLIBC_2.xx not found` when the terminal starts

The Cloud IDE opens but the terminal fails immediately, with a message similar to:

```
The terminal process failed to launch: A native exception occurred
    during launch
(/lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found
(required by /strong/strong-ide/node_modules/node-pty/build/Release/pty
    .node)).
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The image is based on a distribution whose `glibc` is older than the IDE server requires. CentOS 7 is the
usual cause, at `glibc 2.17`. Rebuild on a newer base —Ubuntu 20.04 or later, Debian 11 or later, or RHEL 8
or later —or contact Citrix support to ask whether your distribution can be supported.

The image is flagged as not compatible

The image was added but is not usable, and shows
The image is not compatible with our workspace infrastructure.

One of the four checked requirements is missing. The message lists which. In practice it is nearly always a
base image that was never intended for interactive development —an application image such as a database or
a dashboard has no reason to carry `git`, `git-lfs`, or an SSH client, and often has no `developer` user.

Add the missing packages and the user, rebuild, and add the image again. See
How the platform checks an image, and
Build your first base image for a working starting
point.

The workspace starts but files written during the build are missing

Anything written to `/home/developer` in the Dockerfile is hidden once the persistent volume mounts over it.
Install to a system directory instead, or write the files from a startup script. See
Understanding persistence.

Related information

- Workspace images —building an image
- Container images —adding one to a project
- What persists in a SecurSpaces Workspace
- strong-network/images on GitHub and
  strongnetwork on Docker Hub —example and prebuilt images

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Cryptography

The cryptographic algorithms Citrix SecurSpaces™ uses, and where each one applies. This page supports network
security review and third-party assessment.

Data in transit

Traffic	Protection
Browser and API traffic from outside the cluster	HTTPS. TLS is terminated at the ingress controller.
Ingress controller to SecurSpaces services	TLS
SecurSpaces service to SecurSpaces service	TLS
Remote development sessions	SSH

> **Note:**
>
> Because TLS is terminated at the ingress controller, the permitted TLS versions and cipher suites for
> external traffic are determined by your ingress configuration, not by SecurSpaces. Set them to meet your own
> policy.

Data at rest

SecurSpaces is customer-hosted, so encryption at rest for the database, persistent volumes, and `etcd` is your
responsibility. SecurSpaces adds application-layer encryption on top of it.

Data	Protection
Developer and application secrets stored in the database	AES-128-GCM, when the external Vault integration is not in use
Database, persistent volumes, `etcd`	Your infrastructure encryption, such as cloud provider KMS or encrypted Kubernetes persistent volumes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Keys and certificates

Purpose	Algorithm
Application secret encryption	AES-128-GCM
API tokens	AES-GCM
Personal SSH identities	Ed25519
SAML signing certificate, when SecurSpaces acts as the identity provider	RSA 2048
Token signing	JSON Web Token signing keys, held as Kubernetes Secrets
TLS	X.509 certificates

SecurSpaces supports automated certificate lifecycle management for external traffic, such as cert-manager,
Let's Encrypt, or an internal PKI. It manages the certificates for internal cluster traffic itself.

Secret storage

Secret	Where it is held
Developer and application secrets	Encrypted in the database, or in an external HashiCorp Vault instance when that integration is enabled
Certificates, access keys, encryption keys, database credentials, token signing keys	Kubernetes Secrets, stored in `etcd`

Kubernetes Secrets are stored in `etcd`, which you must protect with encryption at rest. For stronger separation, SecurSpaces works with the Secrets Store CSI Driver and External Secrets Operators.

Related information

- Ports and protocols
- Security data flows
- Security control catalogue

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Mount point limits and behavior

September 29, 2026

Capacity, quotas, and scale

EFS and S3 Files are elastic. There is no provisioned size. The storage value on the volume is required by
Kubernetes but ignored, so do not promise users a per-Mount-Point size quota.

Limit	Value
Access points per EFS file system	About 10,000. A default that AWS Support can increase.
Access points per S3 Files file system	25,000. Cannot be increased.
Mount targets per Availability Zone	1 per Availability Zone. Hard limit.
Mount Points created per file system, per region	1,000. Enforced by SecurSpaces and not configurable.

Every **Create New** Mount Point consumes one access point, and the ceiling is shared across all projects on that
file system. An access point you create by hand occupies a slot from the moment you create it, so count
pre-created access points into the same budget.

When the 1,000 limit is reached, new **Create New** requests are rejected with a message that the file system
storage limit has been reached. The gate runs only for Create New. An **Attach Existing** Mount Point is neither
counted against the 1,000 nor blocked by it, because it consumes no new access point slot.

The limit is compiled into the product. It is not a settings, Helm, or environment value. It is also approximate,
and there are three ways a file system goes past it:

- A burst of simultaneous creates can push a file system slightly over.
- During a datastore outage the count cannot be read, and the create is allowed through unchecked.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- When the StorageClass list cannot be read, the file system id cannot be resolved, so the gate is skipped
  entirely and the create proceeds unguarded. This is the same missing permission that switches off the Create
  New encryption checks, described in
  Configure SecurSpaces.

Treat the limit as a leading indicator and the AWS ceiling as the real backstop. Both of the last two cases are
counted, so you can alert on them. See Metrics.

Prefer a file system dedicated to SecurSpaces Mount Points. On a shared file system SecurSpaces counts only its own access points,
so the real total can approach the AWS ceiling while SecurSpaces still believes there is room.

Isolation between Mount Points is directory and POSIX identity on a shared file system. Access points on one file
system share one encryption key and one throughput pool. Put a tenant that needs its own key, throughput pool, and
failure domain on a separate file system.

> **Note**
>
> A separate file system is not by itself an access boundary. Who may mount is decided by IAM and the network. To
> make the split real, scope the node role's client permissions to named file systems and keep the file system's
> mount targets closed to other clusters' nodes.

Deletion and data protection

What a delete removes depends on how the Mount Point was created:

Mode	What SecurSpaces removes	Backing data
Create New	The claim. The StorageClass reclaim policy then removes the volume.	Destroyed. The access point is removed.
Attach Existing, SecurSpaces-built volume	The claim and the volume.	Retained.
Attach Existing, volume you authored	Only the claim.	Retained.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Deleting a **Create New** Mount Point removes the access point but leaves its directory and files on the file
system. Plan periodic cleanup of these orphaned directories, and never attach a file-system-root volume over such a
file system.

A Mount Point cannot be deleted while a live workspace or workspace template uses it. It also cannot be deleted
while a deleted workspace in the recycling bin still uses it, because that workspace can be restored. Detaching does
not help in that case. Either permanently delete the named workspaces from **Deleted Workspaces**, wait for the
retention period to expire, or restore the workspace and remove the Mount Point from it.

Updates are metadata-only. Only the name and mount path can be changed. The file system id, access point, and
subpath are fixed when the Mount Point is created, AWS Mount Points cannot be resized, and the read-write or
read-only setting cannot be changed after creation.

> **Warning**
>
> A Mount Point is collaborative read-write storage with no undo. Every workspace that mounts it can delete any
> file, and the access point squashes all activity to a single POSIX identity, so file ownership gives no per-user
> protection. SecurSpaces does not create, schedule, verify, or monitor backups. Backup and recovery of Mount Point data
> are entirely the responsibility of the AWS account owner.

For Amazon EFS, the only recovery path is an AWS Backup recovery point that existed before the loss. Backups are
taken per file system, not per Mount Point. A full restore returns every project's directory on that file system
together, and whoever can run it can read all of it. An item-level restore can target individual files and directories, so recovering one deleted folder does not mean bringing back everything. Prefer an item-level restore
when only one folder is being recovered.

For Amazon S3 Files, bucket versioning is the recovery path: a file deleted through the file system becomes a
non-current object version. Pair it with a lifecycle rule that keeps non-current versions long enough to be useful.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> Never let live data move into an archive tier. Objects in Glacier Flexible Retrieval or Deep Archive, and
> Intelligent-Tiering objects that have fallen into an Archive Access tier, cannot be read through the file system
> at all. They must be restored through the S3 API before the file system can read them again, and nothing in
> SecurSpaces or in the mount warns you: the files simply become unreadable to every workspace. Apply transition
> rules only to non-current versions, and keep every current object in a directly readable class: Standard,
> Standard-IA, One Zone-IA, or Intelligent-Tiering with the archive tiers left off. A transition rule is only one
> of the two ways in. The archive tiers of Intelligent-Tiering are a bucket setting, not a lifecycle rule.

A restore never puts files back where they were. AWS Backup writes into a new directory off the file system root,
named `aws-backup-restore_<datetime>`, and every attempt creates another one. Because every Mount Point is rooted
inside an access point directory, that directory sits above the mount root and is invisible from every workspace.
Budget for a manual copy-back as part of every restore:

1. Run the restore. Restore into the **source** file system whenever it still exists. That keeps the file system
   id and every access point valid, so no Mount Point has to change. Restore into a new file system only if the
   original is gone.
2. From an administrative host in the same VPC, not a developer workspace, mount the file system without an
   access point: `sudo mount -t efs -o tls <file-system-id>:/ /mnt/recovery`.
3. Copy the wanted files from `/mnt/recovery/aws-backup-restore_<datetime>/` into the access point directory the
   Mount Point uses, preserving ownership and permissions with `cp -a` or `rsync -a`.
4. Delete the recovery directory once you have confirmed the copy-back, because it keeps billing as file system
   storage. Then unmount `/mnt/recovery`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> That mount has no access point, so it exposes every project's data on the file system. Run it under change
> control and unmount as soon as the copy-back is finished. Keep the `tls` option: the SecurSpaces requirement that
> every mount carries `tls` applies to the volumes the product creates, not to a mount you run by hand, so without
> it the copy-back crosses the VPC unencrypted on port 2049.

If the original file system is gone, a restore recovers bytes, not Mount Points. A new file system has a new file
system id and no access points, and a Mount Point's file system id, access point, and subpath are fixed at
creation. Every affected Mount Point must be deleted and recreated against the restored file system, with its data
copied into the newly carved directory.

Rehearse the whole procedure, not just the restore job. A restore that reports success proves nothing about whether
developers get their files back. Confirm from inside a workspace that the files and their ownership appear at the
mount path.

> **Note**
>
> Keep your StorageClass and volume manifests in source control. If you lose the cluster, the volumes, claims, and
> StorageClasses are gone, while the Mount Point records survive and still report as ready.

Amazon S3 Files behavior to plan around

- **No hardlinks.** One file is one S3 object key, and an attempt returns `Too many links`. Tools that deduplicate
  using hardlinks fail or fall back. Symlinks work.
- **Export is asynchronous**, taking roughly 60 to 72 seconds. A file written in a workspace appears as an object
  after a sync, not instantly. Within the file system, reads, writes, and locking are immediately consistent.
- **Conflicting writes discard the workspace's version.** If the same file changes both in a workspace and
  directly in S3, the bucket wins. No error is returned to whoever wrote the file, and SecurSpaces

© 1997–2026 Citrix Systems, Inc. All rights reserved.

does not
surface the conflict either, so the only detection channel is AWS. Alarm on `LostAndFoundFiles`, the
per-file-system count Amazon S3 Files publishes in the `AWS/S3/Files` namespace under the `FileSystemId`
dimension. It counts the files currently in the lost-and-found directory, so alarm on any increase. Partition the bucket by writer, or keep external processes read-only.

- **Displaced files land where workspaces cannot see them**, in a directory named `.s3files-lost+found-<file-system-id>` in the file system root. The name is dot-prefixed, so a plain `ls` does
not show it, and it sits above every mount root, so only an operator can recover the files. Those copies are
never exported, so no object version ever exists for them: bucket versioning does not cover them, and recovery
from that directory is the only path. AWS keeps and bills for them indefinitely.
- **A path longer than the 1,024-byte S3 key limit never reaches the bucket.** Export fails terminally, so no
object version exists to restore and bucket versioning does not cover it. Alarm on the CloudWatch
`ExportFailures` metric in the `AWS/S3/Files` namespace.

Use Amazon EFS for latency-sensitive work such as git trees and build caches. Use Amazon S3 Files for large, mostly
immutable datasets, where it is substantially cheaper.

Files are owned by the access point identity rather than the workspace user, so ownership-sensitive tools such as
git's `safe.directory` check may warn. This is normal access point behavior.

Recover a displaced file

Reaching the lost-and-found directory means mounting the file system with no access point, which exposes every
project's data on that file system. Treat the whole procedure as a change-controlled operation on an administrative host in the same VPC, never something run from a developer workspace. If you prefer not to mount
without an access point, create a temporary access point with no root directory restriction and a fixed POSIX
user, allow it only from the recovery host, and delete it in the same change window. An access point rooted at the
lost-and-found directory is not a workaround: files cannot be moved or renamed inside it, and the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

copy-back writes
to the file's original path, which lies outside it.

Mount the file system and list the directory:

```
sudo mount -t s3files <file-system-id>:/ /mnt/recovery
ls -la /mnt/recovery/.s3files-lost+found-<file-system-id>
```

The names there are not the original ones. A hexadecimal id is prepended so that repeated conflicts on one file
stay distinct, names longer than 100 characters are truncated, and the original directory path is not kept. The
path survives only in an extended attribute. Read it with `getfattr`. The timestamp in the attribute name is
required, because it forces a fresh status read:

```
getfattr -n "user.s3files.status;$(date -u +%s)" \
  /mnt/recovery/.s3files-lost+found-<file-system-id>/<hex-id>_<name> --
     only-values
```

`S3Key` is the object key, and is empty if the object was deleted in the bucket. `FilePath` is the path the file
had before the conflict.

To keep the workspace's version, copy the file back to `FilePath` under the mount. S3 Files exports it again as a
new object version. To keep the bucket's version, delete the displaced copy instead. You can copy out of this
directory and delete inside it, but you cannot move or rename within it, and you cannot delete the directory
itself.

Finish in the same change window: unmount, and delete any temporary access point you created. Anything left in the
directory keeps billing as file system storage.

Multi-region

Mount Points are not supported in multi-region configurations. The create dialog offers no region choice, so a
Mount Point created there always belongs to the primary region, and the workspace **Resources** step prevents
adding a Mount Point to a workspace in any other region.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> That restriction is enforced in the create dialog, not on the REST create endpoint. On a deployment with a
> secondary region configured, an API caller that sets `region_id` to that region creates a Mount Point there. The
> record is not filtered out afterwards, so it stays in the project's **Mount Points** list and can be attached to
> a primary-region workspace, where the volume does not exist. Leave `region_id` unset when you create a Mount
> Point through the API.
>
> Workspace templates are not covered by that block. The template editor allows a Mount Point to be attached
> regardless of region, so a template can be saved carrying storage that workspaces outside the primary region
> cannot use. Do not treat this as a supported way to use Mount Points in a secondary region.

Timeouts

Provisioning is subject to two fixed timeouts. Neither is configurable.

Timeout	Value
Volume bind	3 minutes
Overall provisioning call	5 minutes

Exceeding either moves the Mount Point to an error state, from which it can be deleted and re-created.

Metrics

SecurSpaces publishes Prometheus metrics for Mount Points on two endpoints, and only one of them is reachable as
shipped:

- **The control plane** serves `/metrics` over **HTTPS**, on port 8080, alongside the rest of its API.
- **The workspace service** serves `/metrics` over **plain HTTP**, on port 2112. The chart does not put that port
  in a Kubernetes Service and does not declare it as a container port, so nothing can scrape it until

© 1997–2026 Citrix Systems, Inc. All rights reserved.

you expose
it yourself.

The chart also ships no ServiceMonitor, PodMonitor, PrometheusRule, scrape annotations, or dashboards. Building
your own alerting starts with building the scrape configuration. Because the control-plane endpoint is HTTPS, a job
pointed at `http://` fails the TLS handshake, with nothing in the product to explain it:

```
scrape_configs:
  - job_name: securspaces-central
    scheme: https
    metrics_path: /metrics
    tls_config:
      # The CA that signed the control plane's certificate.
      ca_file: /etc/prometheus/certs/ca.crt
    static_configs:
      - targets: ["<release>-central-service:8080"]
```

To reach the workspace service metrics, add port 2112 to the workspace service container ports and to a Service in
front of it, then add a second job for that Service with `scheme: http`.

Published metrics

Metric	Labels	What it records
`sds_mountpoint_create_total`	`type`, `result`, `reason`	Terminal outcome of a mount point create
`sds_mountpoint_create_rejected_total`	`reason`	Creates rejected before provisioning starts, excluding the soft-quota rejection counted separately
`sds_mountpoint_quota_gate_total`	`outcome`: `allowed`, `rejected`, `failed_open`, `skipped_no_fsid`	Decisions of the per-file-system soft quota gate on Create New
`sds_mountpoint_attach_admit_total`	`decision`, `reason`	Attach Existing admission decisions
`sds_mountpoint_readback_total`	`provider`, `mount_point_type`, `result`	Outcome of the volume handle read-back after bind

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Metric	Labels	What it records
`sds_mountpoint_state_write_total`	`outcome`: `persisted`, `write_error`, `lost_race`. `state`: `MOUNT_POINT_PROVISIONING`, `MOUNT_POINT_READY`, `MOUNT_POINT_ERROR`	Every Mount Point state transition except the two claim writes that begin a delete and an update
`sds_mountpoint_stranded_recovered_total`	`state`: `MOUNT_POINT_CREATING`, `MOUNT_POINT_PROVISIONING`, `MOUNT_POINT_DELETING`	Rows re-admitted after being stranded in a transient state
`sds_mountpoint_enable_flag_failopen_total`	—	Times the enable-flag check failed open, allowing a create despite an unreadable or absent configuration
`sds_mountpoint_picker_class_hidden`	`reason`	StorageClasses currently hidden from the Create New picker for missing the `tls` mount option

Every metric in this table is published by the control plane except `sds_mountpoint_readback_total`, which is published by the workspace service. An alert on the read-back counter stays permanently empty until port 2112 is exposed and scraped.

Query the label values exactly as they are written above. The `state` label carries the protocol buffer enum names, in upper case, while every other label value is a lower-case token, so a query for `state="ready"` matches nothing and reads as healthy. Note also that a counter's series does not exist until it is first incremented, so you cannot discover these values by scraping a deployment that has not yet hit the condition.

> **Warning**
>
> Do not rename these metrics to match the product name. They kept their original spelling delib-

© 1997–2026 Citrix Systems, Inc. All rights reserved.

> erately, and
> renaming one silently breaks every dashboard and alert built on it.

Alerts worth building

No alerts ship with the product. These are the ones worth building first.

- **`sds_mountpoint_quota_gate_total`** with an `outcome` of `failed_open` or `skipped_no_fsid`. Either value means a Create New was allowed through with the per-file-system count unchecked. Alert on any non-zero value of either.
- **`sds_mountpoint_attach_admit_total{ decision="deny"}`**. A denied attach, including a cross-project attempt. These denials exist as control-plane log lines and this counter only. They are not written as audit events, so they do not appear on the **Audit** page and do not reach a CEF or SIEM export. If your security team expects attach denials in the SIEM, this counter is the only channel.
- **`sds_mountpoint_readback_total{ mount_point_type="create_new", result="no_access_point"}`**. Scope the alert to `create_new`. A Create New whose volume handle resolves without an access point is refused: SecurSpaces does not mark it ready and the record goes to the error state, so no such Mount Point is ever usable. A non-zero count means a Create New was accepted although its StorageClass should have been refused. It does not mean data has been exposed. Do not alert on every `result` other than `resolved`: an Attach Existing that deliberately names no access point reads back `no_access_point`, and that is a supported configuration. This counter is the one published by the workspace service, so the alert needs the port 2112 scrape described above.
- **`sds_mountpoint_state_write_total`** with an `outcome` of `write_error`, and `lost_race` with `state="MOUNT_POINT_READY"`. A lost READY write can leave a record that looks correct in the product but never truly reached ready.
- **`sds_mountpoint_enable_flag_failopen_total`**. Any non-zero value means a create was allowed while the configuration could not be read.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **`sds_mountpoint_picker_classes_hidden`** above zero. That means at least one StorageClass is hidden from the
  Create New list. Zero does not mean there is no misconfiguration: a class you deliberately keep visible is not
  counted, and a probe that fails also reports zero, including when it fails because the StorageClass read
  permission is missing. Scope the alert to the regions still in service, because the per-region series is never
  deleted and a decommissioned region keeps reporting its last value.

The one failure with no counter

If the workspace service cannot read StorageClasses, the Create New encryption checks switch off and creates
proceed without them. That failure increments no counter. It appears only as a log line, and only in the
workspace service logs, not the control plane's.

Build a log alert on the literal text `RBAC forbidden` in the workspace service logs. Match that string rather
than a longer phrase: the permission failure is reported from several different checks, and a narrower match
catches only one of them.

Configure SecurSpaces covers
the same permission, the one-time `kubectl auth can-i` check, and the two Helm settings that remove the grant.

Related information

- Prepare AWS storage for Mount Points
- Troubleshoot AWS Mount Points

Network destinations

September 29, 2026

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The external addresses a Citrix SecurSpaces™ deployment must reach, for firewall rules and air-gapped
planning.

Version numbers on this page are for the 2026.9.4 release. Substitute the version you are installing.

Summary

Destination	Purpose	Needed
`api.enterprise.strong.network:443`	License verification and renewal	Continuously, unless you use an offline license
Docker Hub	The Strong Installer image	At install and upgrade time
`europe-docker.pkg.dev` and regional mirrors	Helm chart and container images	At install and upgrade time

License server

Item	Value
Address	`api.enterprise.strong.network`
Port	443
Purpose	Online license verification and renewal

During installation the platform contacts the license server to validate the license and to issue a temporary
token for the container image registry.

You can supply a signed license instead, by setting `platform.license` in the Helm values to a base64-encoded
license. It is applied automatically on deploy and upgrade.

> **Note:**
>
> For air-gapped environments, either allow the SecurSpaces controller to reach the license server through
> your firewall, or request an offline license for fully disconnected operation. See
> License.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Installer image

Item	Value
Source	Docker Hub
Image	`strongnetwork/strong_installer:2026.9.4`

Helm chart

Item	Value
Source	Google Artifact Registry
Primary	`europe-docker.pkg.dev/strong-network-release/charts/ninjahchart:2026.9.4`
Mirrors	`us-docker.pkg.dev`, `asia-docker.pkg.dev`

Container images

Item	Value
Source	Google Artifact Registry
Primary	`europe-docker.pkg.dev/strong-network-release/images`
Mirrors	`us-docker.pkg.dev`, `asia-docker.pkg.dev`

Choose the mirror closest to your cluster with the `--source-registry-region` flag. See SDS CLI reference.

Service images

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Image
`cloud_editor_sidecar_proxy:2026.9.4`
`sn_enterprise_bundle:2026.9.4`

Workspace image

Image
`ws-images/cloud_editor_generic:2.3.9`

This is the only workspace image the installer pushes. The tag comes from the installer's built-in default,
which you can override with the `-w` or `--workspace-image-tag` flag.

Additional workspace images

Citrix publishes further workspace images and example image definitions, covering IDEs and language toolchains
beyond the default image.

Source	Address
Example image definitions	github.com/strong-network/images
Prebuilt images	hub.docker.com/u/strongnetwork

> **Important:**
>
> The installer does not push these images. It pushes only the service images and the single workspace image
> listed above. To use an additional image in an air-gapped deployment, pull it and push it to your own
> registry yourself, then add it as a container image.

See Container images for adding an image
to a project, and
Container image requirements for
what an image must provide.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- System requirements
- Ports and protocols
- SDS CLI reference
- Install with the Strong Installer CLI

Network policy fields

The fields available in the expert-mode YAML editor for a Citrix SecurSpaces™ network policy. For how to reach
the editor, see
Create a network policy.

Policy fields

Field	Type	Required	Description
`name`	string	Yes	The policy name. Must not be empty.
`description`	string	Yes	What the policy does. Must not be empty.
`restrictedTraffic`	boolean	Yes	**`true`** restricts outbound traffic to attached resources and the allow list. **`false`** monitors traffic without blocking.
`customEndpointsEnabled`	boolean	No	Enables the custom-endpoints-via-proxy allow lists. Relevant only when the platform uses an external proxy.
`whitelistedDomains`	list of domain entries	No	Domains the workspace is allowed to reach.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Field	Type	Required	Description
`whitelistedIps`	list of strings	No	IPv4 addresses or CIDR ranges the workspace is allowed to reach.
`customEndpointsViaProxyDomains`	list of domain entries	No	Domains reachable through the external proxy. Used only when `customEndpointsEnabled` is **`true`**.
`customEndpointsViaProxyIps`	list of strings	No	IPv4 addresses or CIDR ranges reachable through the external proxy. Used only when `customEndpointsEnabled` is **`true`**.

> **Important:**
>
> The editor rejects any field that is not in this table. A policy containing an unrecognized field does not
> validate.

Domain entries

Each entry in `whitelistedDomains` and `customEndpointsViaProxyDomains` has this shape.

Field	Type	Description
`domain`	string	The domain to allow, for example `example.com`.
`includeSubdomains`	boolean	When **`true`**, the domain and all of its subdomains are allowed.

The two lists validate `domain` differently:

- In `whitelistedDomains`, the value must be a plain domain name, such as `example.com` or

© 1997–2026 Citrix Systems, Inc. All rights reserved.

`pypi.internal.example.com`. Use `includeSubdomains` to cover subdomains rather than writing a wildcard.

- In `customEndpointsViaProxyDomains`, a leading prefix is accepted, so a value such as `*.example.com` validates.

IP address entries

`whitelistedIps` and `customEndpointsViaProxyIps` accept:

- A single IPv4 address, for example `192.168.1.50`.
- An IPv4 CIDR range with a prefix length from `/0` to `/32`, for example `10.0.0.0/24`.

IPv6 addresses are not accepted.

Example

```yaml
name: Backend services policy
description: Restrict workspaces to internal package mirrors and the
   corporate API.
restrictedTraffic: true
whitelistedDomains:
  - domain: pypi.internal.example.com
    includeSubdomains: false
  - domain: example.com
    includeSubdomains: true
whitelistedIps:
  - 10.0.0.0/24
  - 192.168.1.50
```

> **Note:**
>
> When `customEndpointsEnabled` is **`false`**, the custom-endpoints lists are ignored and cleared. Set those fields only when an external proxy is in use.

Related information

- Network policies
- Create a network policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Ports and protocols

The network ports that a Citrix SecurSpaces™ deployment exposes, for firewall rules and network security
review.

This page covers deployments to a Kubernetes cluster. For the ports a 1-Click VM evaluation environment
needs, see Deploy in the cloud.

Externally exposed ports

These are the only ports that need to reach the cluster from your network.

Port	Protocol	Purpose
443	HTTPS	The SecurSpaces console, the API, and workspace applications served through the proxy host name
80	HTTP	Redirects to HTTPS
Your choice, commonly 22	TCP, carrying SSH	Remote development over SSH. Forwarded by the ingress controller to the workspace API.

TLS is terminated at the ingress controller. The permitted TLS versions and cipher suites are therefore set by
your ingress configuration, not by SecurSpaces.

SSH port forwarding

SecurSpaces does not fix the external SSH port. You choose it when you configure TCP forwarding on the ingress
controller, and the ingress forwards it to port 2222 on the workspace API.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Side	Port
External	Whatever you configure on the ingress controller. Commonly 22.
Internal, on the workspace API	2222

> **Important:**
>
> Port 2222 is hardcoded in the workspace API. Do not change it.

For the configuration steps, see
Enable SSH access to workspaces.

Internal service ports

SecurSpaces services communicate over additional ports inside the cluster, including gRPC and HTTP ports on
the central, coordinator, monitor, workspace, and proxy services. These are cluster-internal. They do not need
to be reachable from outside the cluster, and you can restrict them with Kubernetes Network Policies.

For how SecurSpaces uses network policies to control workspace traffic, see
Network policies.

Related information

- Cryptography
- Security data flows
- Enable SSH access to workspaces

Sizing

September 24, 2026

How much infrastructure a Citrix SecurSpaces™ deployment needs, and what drives it.

For what the platform requires rather than how much of it, see
System requirements.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

How to think about it

A SecurSpaces cluster has two kinds of node, and they size differently:

Node pool	Runs	Scales with
Service	The control plane —central service, frontend, coordinator, proxy, workspace service, and the database if it is in-cluster	The size of the platform, not the number of developers. Largely fixed.
Workspace	Developer workspaces, one pod each	The number of workspaces **running at once**, not the number that exist

Sizing the service pool is capacity planning you do once. Sizing the workspace pool is arithmetic on the
workspace specifications you offer, and it is the part that changes.

This page gives figures for two sizes: a **50-developer proof of concept** and a **500-developer production
deployment**. Treat both as starting points and verify against your own usage —see
Resource allocation for
reading actual consumption.

> **Note**
>
> If you want to evaluate SecurSpaces rather than run a pilot with real users, the 1-Click VM is a faster
> route than building a cluster. It runs the whole platform on a single virtual machine. For its sizes, see
> 1-Click VM. To deploy it, see
> Deploy an evaluation environment.
> The guidance below is for a Kubernetes deployment that a group of developers will actually use.

Reference node pools

500 developers

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Cloud	Service node pool	Workspace node pool
AWS	Two `m7i.2xlarge` (8 vCPU, 32 GiB)	`m7i.4xlarge` (16 vCPU, 64 GiB), autoscaling
Azure	Two `Standard_D8as_v6` (8 vCPU, 32 GiB)	`Standard_D16as_v6` (16 vCPU, 64 GiB), autoscaling
Google Cloud	Two `n2-standard-8` (8 vCPU, 32 GiB)	`n2-standard-16` (16 vCPU, 64 GiB), autoscaling

50 developers (proof of concept)

Cloud	Service node pool	Workspace node pool
AWS	One `m7i.2xlarge` (8 vCPU, 32 GiB)	`m7i.2xlarge` (8 vCPU, 32 GiB), autoscaling 1–6
Azure	One `Standard_D8as_v6` (8 vCPU, 32 GiB)	`Standard_D8as_v6` (8 vCPU, 32 GiB), autoscaling 1–6
Google Cloud	One `n2-standard-8` (8 vCPU, 32 GiB)	`n2-standard-8` (8 vCPU, 32 GiB), autoscaling 1–6

The service node does not get smaller when the deployment does. The control plane runs the same services
either way, and an in-cluster database can still use 2 cores and 4 GiB at its limit, so 8 vCPU and 32 GiB remains the sensible floor. What changes is the **count**: one service node instead of two.

That single node is the trade-off to understand. A proof of concept with one service node has no control
plane redundancy —during a node failure or a node upgrade, the platform is unavailable and running workspaces lose their connection. That is usually acceptable while evaluating and never acceptable in
production. **Move to two service nodes before you take on real users**, whether or not the developer count
has grown.

The workspace pool is smaller in shape as well as in count, because a pool that scales in 16 vCPU steps is
a coarse fit when peak demand is 20 workspaces. Smaller nodes waste less while idle.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Both sizes

What matters is the shape —8 vCPU / 32 GiB for service nodes and 16 vCPU / 64 GiB for workspace nodes —not
the specific family. Earlier generations such as `m6i` or `Standard_D8as_v5` are equivalent, and any general-purpose instance of the same size works.

> **Important:**
>
> SecurSpaces requires `amd64` nodes. AWS Graviton instances (`m7g`, `m8g`) and Azure ARM-based sizes are
> `arm64` and are not supported. See
> System requirements.

Two service nodes is a minimum for availability rather than for capacity. Run the workspace pool with autoscaling: workspaces pause when idle, so the pool should be able to shrink overnight as well as grow
during the day.

Control plane resources

The chart requests modest resources and permits considerably more. **Size your nodes against the limits, not
the requests** —the requests are what a service needs at rest, and the limits are what it may use under load. A cluster provisioned to the requests will schedule successfully and then throttle.

Component	Request	Limit
MongoDB (Percona, in-cluster)	500m CPU, 1 GiB memory	2 CPU, 4 GiB memory
MongoDB (legacy Bitnami path)	1 CPU, 2 GiB memory	4 CPU

Two `m7i.2xlarge` service nodes give 16 vCPU and 64 GiB in total, which accommodates the platform services
alongside an in-cluster database at its limits, with room for the ingress controller and cluster add-ons. A
single node of the same size fits the same workload for a proof of concept, without the redundancy.

Using a managed database instead of an in-cluster one frees that headroom, which is one reason it is the
production recommendation. The figures above are the defaults for a single-member deployment. If you run the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

internal database in production, size it for your own load and redundancy rather than from this table. See
The SecurSpaces database.

Workspace capacity

A workspace holds its CPU, memory, and disk allocation only while it is running. Capacity is therefore driven
by how many run concurrently, which idle detection largely determines.

To estimate the workspace pool:

1. Take the workspace specifications you offer, for example 2 vCPU and 8 GiB.
2. Estimate peak concurrency. It is lower than your developer count —allow for time zones, meetings, and
   workspaces that pause when idle.
3. Multiply, then add roughly 20% for the node's own overhead and for scheduling slack.

Two worked examples, both at 2 vCPU and 8 GiB per workspace:

Deployment	Peak concurrent	Total	Workspace nodes
500 developers	200 workspaces	400 vCPU, 1,600 GiB	About 28 `m7i.4xlarge`
50 developers (PoC)	20 workspaces	40 vCPU, 160 GiB	About 6 `m7i.2xlarge`

Both are memory-bound rather than CPU-bound at this ratio, which is typical: an 8 GiB workspace on a node
with 4 GiB of RAM per core exhausts memory before CPU. If your workspace specifications are memory-heavy,
size on memory and treat the core count as a by-product.

This is also why the workspace pool autoscales and the service pool does not. In a proof of concept the
difference is stark —outside working hours a pool that scales to one node costs a sixth of one that does
not.

Idle detection is the largest single lever on this number. A deployment where workspaces never pause
needs a pool sized for every workspace that exists, rather than for those in use. See
Workspace idle detection.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Specification ceilings

The platform enforces these maximums on a single workspace specification:

Resource	Maximum
CPU	1000 cores
Memory	10,000 GiB
Disk	10,000 GiB
GPUs	1000

These are product limits, not recommendations. What you can actually request is bounded by the hardware in
your cluster.

Shared memory

By default a workspace pod gets the container runtime's own `/dev/shm`, which is small. Tools that rely on
shared memory —browser-based test runners such as Chrome or Playwright, and some data science and machine
learning libraries —fail when it runs out, usually with an error that does not name `/dev/shm`.

To give every workspace a larger shared memory segment, set the size in your Helm values:

```
workspaceShmVolumeSize: "2Gi"
```

Each workspace pod then mounts a memory-backed `emptyDir` at `/dev/shm` with that size limit. Leave the value
out, or set it empty, and no additional volume is mounted.

The volume is backed by memory, so it counts against the node's RAM rather than its disk. Add it to the
per-workspace memory figure when sizing the workspace pool.

Database

Sized separately, because it grows with retained audit data rather than with concurrency. See Database sizing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

In short: plan for 10–40 GB of data at 500 developers and provision at least 100 GiB of storage. For a 50-developer proof of concept, plan for 1–3 GB and provision at least 20 GiB. Size MongoDB to its 4 GiB
memory limit in both cases —the working set is driven by query patterns rather than by user count.

Registry and storage

Component	Guidance
Container registry	Standard tier is sufficient at any size. Enable cross-region replication for multi-region deployments.
Workspace storage	One persistent volume per workspace, at the disk size in its specification. This is usually the largest storage consumer in the deployment.

1-Click VM

The installer's machine capacity setting sets the maximum number of workspaces running at once, and tunes the
deployment for the VM. On a local VM it does not change the VM's resources; the VM must already have them.

Capacity	vCPU	Memory	Disk	Workspaces running at once
Small	4	16 GB	200 GB	Up to 3
Medium (default)	8	32 GB	500 GB	Up to 11
Large	16	64 GB	1 TB	Up to 25
Custom	User-defined	User-defined	1 TB or more	Depends on resources

Workspace images use most of the disk. On a hypervisor, allocate memory as a fixed amount rather than dynamic memory.

The workspace resource profiles, such as Minimal, are resource requests, not limits. Measured use can be higher.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- System requirements
- Database sizing
- Resource allocation
- Kubernetes deployment

REST API

September 28, 2026

The Citrix SecurSpaces™ REST API exposes the platform's resources for automation: workspaces, projects,
organizations, resources, users, and reporting. Anything the console does to a platform resource, the API can
do.

The specification is served by your own deployment, at `https://<your-sds-hostname>/api`. That copy is
the authoritative one, because it matches the version you are running. This page covers what the specification does not tell you: how to reach it, how to authenticate, and how the endpoints are arranged.

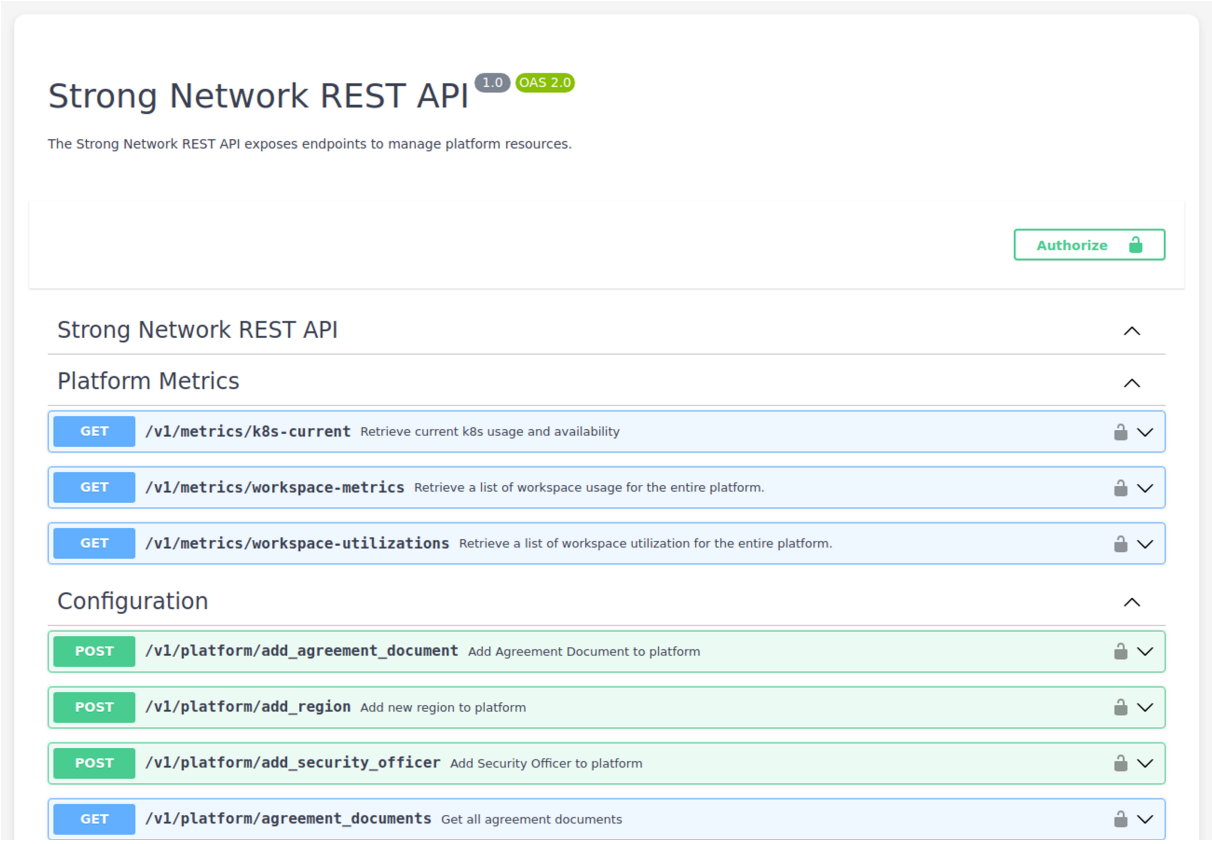
> **Note:**
>
> The REST API is controlled by a platform feature flag, which a platform administrator turns on. See
> Turn features on or off. Only users signed in to
> the platform can open the API documentation.

Open the API documentation

1. Sign in to SecurSpaces.
2. Select your profile picture at the top right of the screen to open the account menu.
3. Select **REST API**.

Or browse directly to `https://<your-sds-hostname>/api`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



The page is an interactive specification browser. You can expand any operation to see its parameters, request body, and response schema, and call it directly from the page once you have authorized.

Authenticate with an API key

The API authenticates with an **API key**, sent in the `Authorization` header:

Scheme	API key
Parameter name	`Authorization`
Location	Request header

Create a key in **Profile > Security > API Keys**. See Security.

A key carries the identity of the user who created it, so a request can only do what that user could do in
the console. Treat it like a password, and revoke it from the same page if it is exposed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

To try operations from the documentation page itself, select **Authorize** and provide your key.

How the endpoints are organized

The API mirrors the platform hierarchy. Most resources exist at more than one scope, and the path tells you
which one you are addressing:

Scope	Path prefix	What it addresses
Platform	`/v1/platform/...`	Deployment-wide configuration, regions, agreements, platform users and resources
Organization	`/v1/organizations/{ organizationId } /...`	Everything belonging to one organization
Project	`/v1/projects/{ projectId } /...`	Everything belonging to one project

So a data bucket, for example, is reachable at project, organization, and platform level, with the same shape
at each. Choosing the scope is usually the first decision when you write against this API.

Alongside those, several resource families sit at the top level:

Path	Covers
`/v1/workspaces`, `/v1/personal_workspaces`	Workspaces, and the caller's own workspaces
`/v1/workspace-images`	Container images available to workspaces
`/v1/network-policies`	Network policy definitions
`/v1/data-buckets`	Data buckets
`/v1/metrics`	Usage and reporting data
`/v1/groups`	Groups

Within the documentation page, operations are grouped by area —configuration, users, workspaces, resources,
metrics, and status —at each scope.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The specification

Format	OpenAPI 2.0 (Swagger)
Current version	1.1
Content type	`application/json`

Version 1.1 defines 207 paths and 258 operations. Those figures move with the product, so treat your own deployment's specification as authoritative rather than this page.

Select **Download OpenAPI JSON** on the API page to save the specification. Use it to generate a client, to diff versions during an upgrade, or to drive contract tests.

The API page also lists what changed in the current version, including behavior changes that do not alter any route or schema —for example, a request that was previously accepted and silently ignored now being rejected. Read that section before an upgrade: a client can break without any endpoint having changed.

Endpoints used by IDE extensions

Workspace API responses include the `vscode_version_id` and `vscode_version_auto_update` fields, so administrators and integrations can identify the VS Code version assigned to a workspace. For a description of these fields, see IDE versions.

IDE extensions can also use these personal workspace endpoints:

- `GET /v1/personal_workspaces/{ workspaceId }` returns details for a personal workspace, including attached repositories, mount points, and data buckets.
- `DELETE /v1/personal_workspaces/{ workspaceId }` deletes a personal workspace belonging to the signed-in user.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `GET /v1/personal_workspaces/{ workspaceId } /connection_details` returns what an extension needs to open a connection.

`https://<your-sds-hostname>/version.json` is served by the platform frontend rather than by the REST API. An extension can read it to check capability flags before calling version-specific endpoints, without authenticating first.

Manage VSCode versions through the API

The API includes VSCode version endpoints for automating the same version-management workflows available in the console. Use them to standardize IDE versions across scopes, import versions from a custom registry, and set a default version as part of an upgrade or environment build pipeline.

Version operations are available at these scope roots:

- Platform: `/v1/vscode-versions`
- Organization: `/v1/organizations/{ organizationId } /vscode-versions`
- Project: `/v1/projects/{ projectId } /vscode-versions`

Each scope supports:

- `GET` to list versions.
- `POST` to import a version from a registry by image tag.
- `PATCH /{ versionId } /name` to rename a version.
- `DELETE /{ versionId }` to delete a version.
- `POST /{ versionId } /default` to set the scope default.

Platform-level custom registry management is also available through `GET /v1/vscode-registries`, `POST /v1/vscode-registries`, `PATCH /v1/vscode-registries/{ registryId } /name`, and `DELETE /v1/vscode-registries/{ registryId }`.

The endpoints enforce the same scope and permission checks as the console. A custom VSCode registry name must be unique within its scope, and each scope can have one custom registry.

For background about how IDE versions are resolved and used by workspaces, see IDE versions.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Security —create and revoke API keys
- Licensing —enabling the REST API
- Roles and permissions —what a key can do
- SDS CLI reference

SDS CLI reference

October 01, 2026

`sds-cli` installs, upgrades, and manages Citrix SecurSpaces™ deployments. It runs inside the Strong Installer
container.

This page describes the commands and their flags. For the procedures that use them, see Install with the Strong Installer CLI and
Upgrade.

> **Note:**
>
> This command was previously named `strong-cli`. The binary is now `sds-cli`. A `strong-cli` symbolic link
> remains in the container, so existing scripts continue to work, but use `sds-cli` in new work.

Run the CLI

The Strong Installer is published on
Docker Hub. The image tag matches the SecurSpaces
release you are installing or upgrading to.

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:<version>
```

`${ PWD }` is mounted to `/strong-network/shared` inside the container. The CLI writes configuration files and
Helm charts to that location, so they remain on the host after the container exits.

To run a command without an interactive shell —for example in a pipeline —pass it as an argument:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared \
    strongnetwork/strong_installer:<version> \
    <command> [flags]
```

Commands

Command	Purpose
`install`	Install the platform for the first time.
`upgrade`	Upgrade the platform to a new version.
`push-images`	Push SecurSpaces images to a registry.
`create-config`	Create the initial configuration file.
`update-config`	Generate an updated configuration file from an existing one.
`fetch-access-token`	Fetch a SecurSpaces access token.
`fetch-helm-chart`	Fetch the Helm chart for a given configuration.
`helm-upgrade`	Run `helm upgrade` against an existing deployment.
`get-terraform`	Download the SecurSpaces Terraform provider.
`deploy-demo`	Deploy a demo environment to a cloud or local environment.
`stress-test`	Create workspaces for stress testing.

`install` and `upgrade` are the two commands used in a normal deployment. The others are building blocks that
those two call, and are useful when you need to run a single stage on its own —for example, to push images
from a machine that has registry access and then upgrade from elsewhere.

Common flag groups

Most commands compose the same groups of flags. This section describes each group once; the entry for each
command lists the groups that apply to it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Configuration file

Flag	Description
`-c`, `--config-file`	Last deployed configuration file, which must be in the directory you started the container from, for example `config_2026.9.0.yaml`
`--ignore-config`	Do not read a configuration file

When you use `--ignore-config`, the CLI prompts for the values it would otherwise read from the file, such as
the host domain and the registry URL.

Access token

Flag	Description
`-a`, `--access-token`	SecurSpaces access token
`--gcp-key-file-path`	Path to the SecurSpaces service account key file, used to fetch a token
`-d`, `--domain`	Domain override used when fetching the token

If you do not pass `--access-token`, the CLI fetches one using the other two flags or prompts for what it
needs.

Source registry

The source registry is the Citrix registry that images are pulled *from*.

Flag	Description
`--source-registry-region`	Region to pull images from: `US`, `EU`, or `Asia`. Not case sensitive.
`--source-registry-url`	Source registry URL, overriding the region

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Destination registry authentication

The destination registry is your own registry, that images are pushed *to*. The CLI detects the registry type
from its URL and prompts for the matching credentials. Use these flags to supply credentials non-interactively, for example in a pipeline that cannot answer prompts.

Flag	Registry	Description
`--service-account-file-path`	Google	Path to the Google Cloud service account key file
`--access-key-id`	AWS	Access key ID
`--secret-access-key`	AWS	Secret access key
`--session-token`	AWS	Session token
`--service-principal-app-id`	Azure	Service principal application ID
`--service-principal-secret`	Azure	Service principal client secret
`--service-principal-tenant-id`	Azure	Service principal tenant ID
`--use-docker-login`	Any	Authenticate with `--registry-username` and `--registry-password` instead of using cloud provider authentication
`--registry-username`	Any	Registry user name
`--registry-password`	Any	Registry password

> **Note:**
>
> `--gcp-key-file-path` and `--service-account-file-path` are different flags. The first authenticates against
> the Citrix source registry; the second authenticates against your destination registry.

Image selection

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flag	Description
`-w`, `--workspace-image-tag`	Workspace image tag
`--bitnami-mongo-image-tag`	Bitnami MongoDB image tag
`--skip-service-images`	Do not push service images
`--skip-workspace-images`	Do not push workspace images
`--skip-bitnamo-mongodb-image`	Do not push the Bitnami MongoDB image
`--ignore-image`	Image to skip. Repeat the flag to skip several images.

> **Important:**
>
> `--skip-bitnamo-mongodb-image` is spelled as shown, with `bitnamo`. This is a known misspelling in the product. `--skip-bitnami-mongodb-image` is not recognized.

Cluster access

These flags tell the CLI how to reach your cluster when it runs `kubectl` and `helm`.

Flag	Description
`--ignore-kubectl-setup`	Skip `kubectl` setup for cloud authentication
`--cluster-name`	Cluster name to use for the Helm upgrade
`--azure-subscription-id`	Azure subscription ID
`--azure-resource-group`	Azure resource group
`--gcp-project-id`	Google Cloud project ID
`--gcp-region`	Google Cloud region
`--gcp-zone`	Google Cloud zone
`--aws-region`	AWS region
`--aws-resource-group`	AWS resource group

Proxy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flag	Description
`--proxy`	HTTP or HTTPS proxy
`--proxy-https`	HTTPS proxy
`--proxy-http`	HTTP proxy
`--no-proxy`	Hosts that bypass the proxy

Output and transport

Flag	Description
`--verbose`	Enable verbose output
`--insecure`	Disable TLS certificate verification
`--no-grpc`	Use REST instead of gRPC

install

Installs the platform for the first time. The command prompts for the initial configuration values, for registry authentication, and for the destination registry URL.

```
sds-cli install [flags]
```

Flag	Description	Default
`-v`, `--version`	Platform version to install	Value from the `VERSION` file in the container

Also accepts: configuration file, access token, source registry, destination registry authentication, image
selection, cluster access, proxy, and output and transport flags.

> **Important:**
>
> After authentication, the Helm chart file is written to the shared directory. It contains your deployment
> configuration and should be secured immediately.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

upgrade

Prepares an existing deployment for a new version: downloads the chart, updates the configuration, prints the Helm upgrade command, and pushes the images. Apply the printed Helm command to update the
cluster.

```
sds-cli upgrade -c <config-file> [flags]
```

Flag	Description	Default
`-v`, `--version`	Platform version to upgrade to	Value from the `VERSION` file in the container

Also accepts: configuration file, access token, source registry, destination registry authentication, image
selection, cluster access, and output and transport flags. `upgrade` does not accept `--insecure`.

Internal Percona upgrade kit

For the chart with Percona Operator 1.23.0, if `platform.deployPerconaMongoDB` is unset or **`true`**,
`upgrade` also copies the kit to `/strong-network/shared/percona-upgrade/`. It contains `upgrade-percona.sh`, `restore-percona-backup.sh`, `delete-percona-backup.sh`, and the 1.21.2 and 1.22.0 CRDs.
Rerunning `upgrade` replaces the kit so that it matches the installer version.

This adds no CLI prompts. The output instructs you to run `upgrade-percona.sh` before Helm and supplies
the restore and delete commands. The script itself asks for confirmation. It upgrades Percona one minor
at a time to 1.22.0 and applies the 1.23 CRDs; Helm completes the move to 1.23.0. External MongoDB deployments
skip this behavior. See Upgrade for the full procedure.

Upgrading without a configuration file

If the configuration file from the previous deployment is unavailable, pass `--ignore-config`. The CLI then
prompts for each value it would have read from the file, including the host domain and the registry

© 1997–2026 Citrix Systems, Inc. All rights reserved.

URL, and
does not write a new configuration file.

> **Warning:**
>
> Without a configuration file you must supply every deployment parameter manually. Use this option only when
> the configuration file cannot be recovered.

push-images

Pushes SecurSpaces images to your registry without installing or upgrading.

```
sds-cli push-images [flags]
```

Accepts: configuration file, access token, source registry, destination registry authentication, image selection, and output and transport flags.

> **Tip:**
>
> `--use-docker-login` with `--registry-username` and `--registry-password` suppresses the authentication
> prompts entirely, which is what a non-interactive pipeline needs.

create-config

Creates the initial configuration file.

```
sds-cli create-config [flags]
```

Accepts: access token, source registry, proxy, and output and transport flags.

update-config

Generates an updated configuration file from an existing one.

```
sds-cli update-config -c <config-file> [flags]
```

Flag	Description
`-v`, `--version`	Platform version tag to write into the new configuration file

Also accepts: configuration file, access token, source registry, and output and transport flags.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

fetch-access-token

Fetches a SecurSpaces access token and prints it.

```
sds-cli fetch-access-token [flags]
```

Accepts: configuration file, access token, and output and transport flags.

fetch-helm-chart

Fetches the SecurSpaces Helm chart for the configuration you supply.

```
sds-cli fetch-helm-chart -c <config-file> [flags]
```

Accepts: configuration file, access token, source registry, and output and transport flags.

helm-upgrade

Runs `helm upgrade` against an existing deployment, using a Helm chart you already fetched.

For an existing internal Percona deployment moving to the 1.23 chart, first complete the Percona preparation step.
The chart blocks the upgrade until the database has reached 1.22 and the required 1.23 CRDs are applied;
subsequent upgrades of a deployment already on 1.23 are allowed.

```
sds-cli helm-upgrade -c <config-file> --helm-chart-file-name <chart> [
    flags]
```

Flag	Description
`-c`, `--config-file`	Last deployed configuration file
`--helm-chart-file-name`	Helm chart file to use for the upgrade
`--verbose`	Enable verbose output

Also accepts the cluster access flags.

get-terraform

Downloads the **SecurSpaces Terraform provider** into the shared directory. Despite the command name, this
does not fetch the Terraform binary itself —install Terraform separately.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

For what the provider manages and how to configure it, see Terraform provider.

```
sds-cli get-terraform [flags]
```

Flag	Description	Default
`-c`, `--config-file-name`	Last deployed configuration file	
`-v`, `--version`	Provider version to fetch	Latest version
`-d`, `--domain`	Host domain	
`--src-reg-key-file`	Path to the SecurSpaces service account key file	

Also accepts the output and transport flags.

> **Note:**
>
> This command uses `--config-file-name`, not the `--config-file` used by the other commands. The `-c` shorthand is the same.

deploy-demo

Deploys a demonstration environment. Use this for evaluation, not for production.

```
sds-cli deploy-demo [flags]
```

Flag	Description	Default
`--deployment-location`	Where to deploy: `cloud` or `local`	`cloud`
`--helm-chart-version-tag`	Version of the Helm chart archive to download	
`--helm-chart-registry`	Registry to pull the Helm chart from	
`--skip-version-check`	Do not check that the version tag exists in the registry	**`false`**
`-v`, `--version`	Platform version tag	

Also accepts: configuration file, access token, source registry, proxy, and output and transport flags.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

stress-test

Creates workspaces to load test a deployment. This is a testing tool, not part of installing or operating a
production deployment.

```
sds-cli stress-test --domain <domain> --test-name <name> --workspaces <n> [flags]
```

`--domain` and `--test-name` are required for the command and all its subcommands. `--workspaces` is required
for `stress-test` itself.

Flag	Description	Default
`--domain`	Host domain. Required.	
`--api-key`	Platform API key. Can also be set with the `STRONG_API_KEY` environment variable.	
`--test-name`	Base name identifying the test, organization, project, and workspaces. Required.	`stress-test`
`--concurrency`	Maximum number of concurrent workspace requests	
`--delay`	Seconds between sequential create requests when `--concurrency` is 1	`0`
`--heartbeat-interval`	Send a heartbeat every N seconds from inside the workspace after startup. `0` disables it.	`0`
`--workspaces`	Number of workspaces to create. Required.	`0`
`--image-name`	Workspace image to use	
`--region`	Region to create workspaces in	Default region
`--per-region`	Create workspaces in all available regions	**`false`**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Flag	Description	Default
`--no-cleanup`	Do not clean up resources when the test finishes or is interrupted	**`false`**
`--memory`	Memory in MB for each workspace	
`--vcpu`	vCPU in MHz for each workspace	
`--disk`	Disk size in GB for each workspace	
`--gpu`	Number of GPUs for each workspace	
`--state-on-create`	Initial workspace state: `CREATE_AS_RUNNING` or `CREATE_AS_PAUSED`	
`--users`	Number of users to create and assign workspaces across	`0`
`--user-email-prefix`	Email prefix for stress test users	Value of `--test-name`
`--user-role`	Project role assigned to created users	`developer`

stress-test subcommands

Subcommand	Purpose
`cleanup`	Clean up resources created by a stress test
`create-users`	Create users and add them to a new stress test project, without creating workspaces
`delete-users`	Delete persisted stress test users, without deleting workspaces, project, or organization
`pause`	Pause all workspaces in a project
`resume`	Resume all workspaces in a project
`pause-resume`	Pause and then resume all workspaces in a project

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Install with the Strong Installer CLI
- Upgrade
- Requirements

Supported IDEs

Which editors work with Citrix SecurSpaces™, and how each one connects.

Cloud IDE

VS Code for Web is the only Cloud IDE the platform ships. It runs in a browser tab and requires nothing installed locally.

If your team needs a different browser-based IDE, it can be built into a custom workspace image. See Container image requirements.

Local IDEs over SSH

IDE	Connection method
VS Code Desktop	SecurSpaces extension, one-click from the console
Cursor	SecurSpaces extension, one-click from the console
Windsurf	SecurSpaces extension, one-click from the console
Kiro	SecurSpaces extension, one-click from the console
JetBrains Gateway	SSH configuration through the Gateway interface
Other SSH-capable editors	Manual SSH key pair

For the connection procedure, see
Connect over SSH.

For where source code, credentials, and extension data reside when a local IDE is used, see Working from a local IDE.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Extensions

There is one extension to install. **Citrix Secure Developer Spaces - Remote SSH** is published on the VS Code Marketplace and on Open VSX. Its own documentation refers to it as **Citrix SDS - Remote SSH**.

It has two parts, and you do not need to install or manage the second one.

Part	Runs on	Provides
The extension you install	Your local machine	The sidebar, sign-in, the workspace list, SSH connections, and the status bar. Your access token stays on your machine and is never placed on the workspace.
A companion component, packaged inside it	Inside the workspace	Reports activity, so a workspace you are actively using is not paused by idle detection, and responds to network quality measurements.

The companion component is installed into the workspace automatically when you connect.

Extension sidebar

The sidebar is titled **Secure Developer Spaces**. Workspaces are grouped by project, and each project has an add button. A **Quick links** section at the bottom provides **Open SDS Console**, **Edit profile settings**, and **Troubleshoot connection**.

Workspace states

State	Meaning
Running	The workspace is running, but your IDE is not connected to it

© 1997–2026 Citrix Systems, Inc. All rights reserved.

State	Meaning
Connected	Your IDE is connected to this workspace. The row is highlighted.
Paused	The workspace is paused

Workspace row controls

Control	Available when
Connect	The workspace is running and not connected
Start (play icon)	The workspace is paused
Pause (pause icon)	The workspace is running or connected

Workspace context menu

Right-click a workspace row.

Action	Available when	Notes
Connect in current window	The workspace is running and not connected	
Connect in new window	The workspace is running and not connected	
Edit workspace	Always	Opens the workspace settings in the SecurSpaces console, in your browser
Delete workspace	Always	Permanently removes the workspace and its data

Connecting requires SSH to be enabled on the workspace. See Connect over SSH.

Workspace contents

Expand a workspace to see what it contains: its workspace apps, its cloned repositories, and its mount points. Mount points that you cannot write to are labeled read-only.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Related information

- Connect over SSH
- Develop in a workspace
- Publish JetBrains Gateway as a Citrix DaaS application

System requirements

September 24, 2026

What a Citrix SecurSpaces™ deployment needs from your infrastructure. For how much of it, see Sizing.

For the external addresses the deployment must reach, see Network destinations.

Each requirement below is marked with what happens if you do not have it:

Level	Meaning
Required	The deployment does not complete without it.
Recommended	The deployment completes. Plan for this before the environment carries real work.
Default	SecurSpaces provides this already. No action needed to start.
Conditional	Needed only in the circumstances named.

Before deployment day

Everything marked Required, in the order you will need it:

- A deployment model chosen, and an account with permissions to provision it
- A dedicated Kubernetes cluster, `amd64`, not shared with other workloads
- An ingress controller, and a cluster that enforces the Network Policy API
- Two DNS domains, the second a wildcard subdomain of the first
- Persistent storage through the Persistent Volume Claim API
- The Helm CLI, available to whoever runs the deployment
- Outbound access to the license server, Docker Hub, and Google Artifact Registry

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If the environment is air-gapped or has restricted egress, confirm that last item first. It is the one that most often stops a deployment on the day, and the fix —firewall changes or an offline license —is the one
that takes longest to arrange. See
Network destinations.

> **Note:**
>
> Workspaces are Linux containers. Development that requires macOS or Apple hardware, such as iOS or macOS
> applications, is not supported.

Kubernetes

Requirement	Value	Level
Kubernetes version	1.34 or later	Required
Cluster	Dedicated to SecurSpaces. Do not share it with other applications.	Required
Node architecture	`amd64`. `arm64` is not supported.	Required
Node operating system on AWS	Amazon Linux	Required
Cluster size	A single node is enough to start. High availability and multiple nodes are not needed for a proof of concept.	Default

> **Note:**
>
> Run a Kubernetes release that is still supported upstream rather than pinning to this minimum. A release
> that has left support stops receiving security fixes, and the supported window moves with each Kubernetes
> release cycle.

Supported platforms

Choose a cloud-native, on-premises, or air-gapped Kubernetes deployment, and make sure your account holds the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

infrastructure permissions that environment needs.

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)
- Red Hat OpenShift
- VMware Tanzu Kubernetes Grid (TKG)
- Nutanix Kubernetes Platform (NKP)

Networking

Requirement	Detail	Level
Ingress gateway	NetScaler CPX is recommended, and the SecurSpaces chart can deploy it for you. NGINX, HAProxy, and the Kubernetes Gateway API are also supported. See Ingress.	Required
Network Policy API	`networking.k8s.io/v1`. If your cluster does not implement it, install Calico or Cilium.	Required
DNS domains	Two DNS domains, the second a wildcard subdomain of the first	Required
TLS certificates	Valid certificates for both domains	Recommended

> **Important:**
>
> The Network Policy API is not optional if you rely on network policies or workspace isolation. Without a network plugin that enforces it, those policies are accepted by the API server and do nothing. See Trust boundaries.

Domains

The second domain must be a wildcard subdomain of the first. For example:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `example.com`
- `*.proxy.example.com`

Certificates are optional for proof-of-concept deployments, but strongly recommended.

Storage

Provide persistent storage through the Kubernetes Persistent Volume Claim API. This is **required**: confirm
the target cluster supports the PVC API before you deploy.

Deployment tooling

Tool	Purpose	Level
Helm CLI	Deploys SecurSpaces from the provided Helm chart. Must be installed and accessible to whoever runs the deployment.	Required
cert-manager	Needed only if the chart renders cert-manager `Certificate` resources. If your cluster does not use cert-manager, set `platform.useCertManager: false`.	Conditional

What SecurSpaces provides by default

You do not need to arrange either of these before a proof of concept. SecurSpaces deploys with both already
working, and each has a production path you can move to later.

Area	Default	Production path
Database	Percona Server for MongoDB, deployed by the chart and running in your cluster	An external MongoDB service, such as MongoDB Atlas. See Database.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Area	Default	Production path
Identity and access	Email and password authentication	An identity provider using SAML or OIDC, such as Okta. See Identity.

The internal database is supported in production, including for air-gapped and on-premises deployments. What changes is ownership: backups, high availability, and scaling become yours to run. An external service moves that work to the provider. See Database for the comparison.

1-Click VM

The 1-Click VM runs the whole platform on a single virtual machine for evaluation. It has its own requirements, which replace the Kubernetes requirements on this page.
For how much capacity it needs, see Sizing.

Target host

Requirement	Value	Level
Operating system	Ubuntu 22.04 Desktop or Ubuntu 24.04 Desktop	Required
CPU	x86_64 with the AVX instruction set, exposed to the guest. Intel Celeron processors do not support it.	Required
Software	A clean installation, with no Docker, Kubernetes, K3s or related container software	Required
Network	Outbound internet access, and an IP address that client devices can reach. A private address is valid.	Required

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Requirement	Value	Level
Access	A user account with full `sudo` permissions	Required
Hypervisor	XenServer, Hyper-V, another hypervisor, or bare metal. Cloud VMs are created by the installer.	Default

These operating systems are not supported for the target host: Ubuntu Server, Ubuntu minimal and cloud images, Debian
and other Debian-based distributions, Red Hat Enterprise Linux, CentOS, Rocky Linux and other RPM-based distributions.

Installer host

Requirement	Value	Level
Machine	Separate from the target host	Required
Docker	Installed and running, able to pull from Docker Hub. On Windows, Docker Desktop with Linux containers.	Required
Network	Outbound HTTPS access to Docker Hub and the SecurSpaces image registries	Required
Disk	20 GB free	Required

What the 1-Click VM contains

Component	Detail
Cluster	Single-node K3s
Ingress	NetScaler CPX (`netscaler-cpx-ingress`)
Platform	SecurSpaces services and database

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Component	Detail
URL	An HTTPS hostname under `try.sds.citrix.com`
License	An evaluation license, valid for 6 months
TLS certificate	Valid for 90 days
Sample data	The **Fast Cars Organization**, with sample users and a sample workspace

Limits

- Not supported for production workloads, multi-node clustering or scale testing.
- Cannot be upgraded or migrated to a production installation.
- The evaluation license expires after 6 months, and the TLS certificate after 90 days.
- Local deployments: the DNS association must be completed within 1 hour of generating the deployment script.

Related information

- Network destinations
- Ports and protocols
- Install with the Strong Installer CLI
- License
- Sizing
- Database sizing

Release notes

August 19, 2026

Page	What it covers
What's new	Features and changes by release
Fixed issues	Resolved defects by release and category, including security fixes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Page	What it covers
What's new in SecurSpaces Flex	Changes to the managed offering
Document history	Changes to this documentation

> **Note:**
>
> Security fixes are listed in Fixed issues
> under the **Security** category. If you track them for review, that page is the record.

What's new

October 06, 2026

Updates continuously enhance your Citrix SecurSpaces™ experience. Each release introduces new features,
improvements, and fixes to ensure you always have access to the latest innovations and performance enhancements.
This article highlights the new and updated capabilities available in this release, as well as resolved issues.

For compatibility and performance details, see
Technical Requirements.

SecurSpaces 2026.9

Keyless Google Cloud Storage backups for internal Percona MongoDB

Internal Percona MongoDB deployments now use Percona Operator for MongoDB 1.23.0. This version adds native
Google Cloud Storage backup support with Workload Identity Federation for Google Kubernetes Engine, so
platform teams can store backups in Google Cloud Storage without static service account keys or HMAC
credentials.

The installer includes a Percona upgrade kit for deployments that started on SecurSpaces 2026.9. Run `sds-cli upgrade` first, then run the supplied `percona-upgrade/upgrade-percona.sh`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

script before applying
the Helm command printed by the installer. The script backs up the database, upgrades through the required
intermediate Percona versions, applies the required custom resource definitions, and prepares the deployment
for the final Helm upgrade to operator 1.23.0.

Key capabilities include:

- Upgrade internal Percona MongoDB from operator 1.20.1 to 1.23.0 with an installer-provided script.
- Configure backup storage, schedules, retention, and point-in-time recovery in Helm values.
- Select the database Kubernetes service account used by Percona Backup for MongoDB agents.
- Use keyless Google Cloud Storage backups on Google Kubernetes Engine with Workload Identity.
- Restore from the upgrade backup or from Percona Backup for MongoDB backup resources.

For more information, see
Back up the database
and Upgrade.

Managed workspace sizes for SecurSpaces Flex

SecurSpaces Flex now uses a Citrix-managed workspace size catalog. Citrix publishes the available size
tiers and their versions through the platform, and Flex deployments reconcile that catalog during upgrades.
This keeps workspace sizing aligned with the service catalog while preserving the organization's availability
settings for each tier.

Administrators can still decide which organizations and projects can use each tier. The tier's resource definition is managed by Citrix, so developers choose from approved sizes instead of editing the underlying
CPU, memory, storage, and storage-class values.

Key capabilities include:

- See managed workspace size tiers with version and lifecycle status.
- Keep availability settings when Citrix publishes a newer tier version.
- Prompt affected workspaces to update when their tier version is no longer current.
- Preview and apply bulk updates that move workspaces to the current version of the same tier.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Prevent new workspaces from using retired tier versions while existing workspaces remain visible.
- Keep the same managed tier version identifier across Flex tenants when Citrix pins the identifier.
- Audit successful tier availability changes and bulk tier-version updates with the user, time, and result.

For more information, see
SecurSpaces Flex workspace templates.

Audit trail for managed Flex tier changes

SecurSpaces Flex now records successful managed tier availability changes and bulk tier-version updates in
the audit log. This gives administrators a clearer record of who changed a managed workspace size tier, when
the action happened, and what changed.

Use the audit log to review when a tier was enabled or disabled, or when a bulk update moved workspaces to the
current version of a managed tier. Refused attempts to modify Citrix-managed tier definitions continue to be
recorded.

For more information, see Audit event catalog.

REST API support for IDE version management

The REST API now includes IDE version management. Platform teams can automate VSCode version operations that
previously required the console, which helps keep SecurSpaces environments reproducible across projects and
deployment pipelines.

The API can list, import, rename, delete, and set the default VSCode version at platform, organization, and
project scope. Platform administrators can also list, add, rename, and delete custom VSCode registries at
platform scope.

Use these endpoints when you want to standardize IDE versions across many scopes, import a tested image tag
from a custom registry, or make a version the default as part of an upgrade workflow.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

For more information, see REST API.
For IDE version concepts, see IDE versions.

Safer Percona MongoDB upgrades

Helm upgrades with the internal Percona MongoDB deployment now reuse existing database passwords by default.
Previously, a render with empty password values could generate new credentials, causing MongoDB and platform
services to disagree about the active password during an upgrade.

On server-side Helm upgrades, SecurSpaces reuses the password already stored in the live `<release>-psmdb-secrets` Secret unless you explicitly set a new value. When the database administrator
credential changes, the platform services that consume it now roll so they read the updated Secret.

Client-side rendering still cannot read the live Secret, so `helm template`, client-side GitOps rendering,
and `--dry-run=client` can still produce new generated values. Use server-side Helm upgrades for the internal
Percona deployment, or pin the database passwords in your values file.

For more information, see
The SecurSpaces database.

Istio ingress support

Clusters that already use Istio as their entry point can now expose SecurSpaces through an Istio ingress
gateway. Set `platform.useIstio` and the chart renders Istio Gateway, VirtualService, and DestinationRule
resources instead of Kubernetes Ingress resources.

The Istio ingress gateway terminates public TLS and then opens a second TLS connection to the SecurSpaces
services, so traffic stays encrypted on the way through the cluster.

As with NGINX, the chart does not create an SSH ingress when Istio is enabled.

For more information, see Istio.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Clearer reporting when a workspace runs out of memory

A workspace stopped for exceeding its memory limit now reports `OOMKilled` in its failure details, with
guidance about the memory it needed. Previously the same failure read `Reason: Unknown` and suggested
updating the workspace image, which sent people to the wrong place.

The workspace error dialog now points at the task that consumed the memory, or at moving to a specification
with a higher limit, rather than at the image. Image-pull failures keep their existing guidance.

For more information, see
Troubleshooting.

Product name updated to Citrix SecurSpaces

Citrix SecurSpaces is the new product name for Citrix Secure Developer Spaces. The updated name now appears
throughout the product interface, browser title, package metadata, email templates, and generated API
documentation.

This change helps align the product experience with the current Citrix naming while preserving existing
workflows and deployment paths.

Documentation URLs have moved from `/en-us/secure-developer-spaces/` to `/en-us/securspaces/`. Every previous
address redirects to the matching page, so existing links and bookmarks continue to work, but the new form is
the one to use going forward.

For more information, see Overview.

Desired state configuration for workspace templates

Workspace templates now keep an active relationship with the workspaces created from them. Each workspace records the template version it was created from, so project owners can see which workspaces have drifted away from the current standard and bring them back up to date.

Previously, a template was applied once at creation time and the workspace kept no connection to it. Project owners had no way to tell which workspaces were still aligned with the current configuration,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

and any change to a repository, secret, extension, or policy had to be coordinated manually with each developer.

Key capabilities include:

- See how many workspaces are in testing, outdated, or unversioned directly on the Templates page.
- Open the Associated Workspaces window to review workspaces grouped by template version, and export
  the list to CSV.
- Set a default template version to signal the desired configuration for a project.
- Let workspace owners apply the update from an update window that previews the configuration changes, without losing their persistent data.
- Require an outdated workspace to be updated before its configuration can be edited.
- Prevent deletion of a template or template version while workspaces still use it.

For more information, see
Templates.

AI observability for developer AI usage

AI traffic that SecurSpaces routes through an AI Gateway is now tagged with the user, project, and workspace that produced it. Platform teams can
report on AI request volume, model usage, and token consumption across the organization, and attribute that usage to the teams and
developers responsible for it.

Enterprise adoption of AI-assisted development has made token spend difficult to account for. Provider dashboards are specific to a single
vendor and cannot show usage across the tools and models a development organization actually uses. This release gives customers a single,
provider-independent view.

Key capabilities include:

- Add user, project, and workspace metadata to AI requests forwarded to the AI Gateway.
- Report on AI usage independently of the LLM provider serving each request.
- Break down token consumption and request volume by project, by user, and by model to support chargeback.
- Health check the AI Gateway, and choose whether AI traffic bypasses it or fails when it is unavailable.
- Import a sample Grafana dashboard covering token totals, usage rates, response times, and per-project, per-user, and per-model views.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SecurSpaces documents a validated reference stack of Envoy AI Gateway, Prometheus, and Grafana. Because the gateway exports standard metrics, you
can also send this data to your existing monitoring infrastructure.

For more information, see
Configure AI Gateway routing.

Dot files for workspace personalization

Developers can now connect a personal dot files repository from **Profile > Configuration > Dot Files**. SecurSpaces clones the repository into each selected workspace and runs a supported install script so shell,
editor, and tooling preferences can be applied automatically.

Dot files are configured per user and use that user's connected Git integration. Developers can apply the
configuration to all workspaces or only selected workspaces, choose the install-script timeout, and review
Dot files output in workspace startup logs.

Key capabilities include:

- Use private GitHub, GitLab, Bitbucket, or Azure DevOps repositories for dot files.
- Apply dot files when a workspace starts or resumes with new repository commits.
- Run **Personalize Environment** to reapply dot files to an existing workspace.
- Keep workspace startup non-blocking if a dot files script fails or times out.

For more information, see Configuration.

Setup Checklist for guided project onboarding

The project console now includes a Setup Checklist that guides project owners and developers through common
first steps. The checklist appears as a collapsible side panel on project pages and keeps progress across
navigation so users can return to their next setup task.

Project owners see steps for reviewing images, adding repositories and secrets, creating templates, and adding
users. Developers see steps for reviewing their profile, configuring integrations, customizing their environment, adding credentials, and creating a workspace. Returning developers see a shorter checklist focused
on project-specific readiness.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Platform administrators can enable or disable the project-owner and developer checklists from on-boarding
settings.

For more information, see
General Settings.

AWS Mount Points for shared workspace storage

Mount Points now support Amazon Web Services storage in addition to Azure file storage. Project owners can
create or attach shared storage backed by Amazon Elastic File System (EFS) or Amazon S3 Files, then make it
available to workspaces through the existing **Resource Access** flow.

Use AWS Mount Points when teams need live shared files for build artifacts, large datasets, or collaborative
project data without copying that data into each workspace. Administrators enable the available storage types
first, and project owners choose whether to create new storage or attach storage that an infrastructure team
already prepared.

Key capabilities include:

- Create new AWS-backed Mount Points from eligible storage classes.
- Attach existing Amazon EFS or Amazon S3 Files storage to a project.
- Use read-only or read/write access when attaching Mount Points to workspaces.
- See clearer validation and delete guidance when a Mount Point is still used by live or deleted workspaces.

For more information, see Mount Points.

Amazon EFS and Amazon S3 Files require AWS resources that SecurSpaces does not create, including the file system,
per-Availability-Zone mount targets, IAM roles, and Kubernetes storage classes. For the administrator setup steps,
see Prepare AWS storage for Mount Points.

Redesigned IDE extension for VS Code and compatible IDEs

The SecurSpaces extension for VS Code, Cursor, Windsurf, and Kiro has been redesigned around a sidebar that acts

© 1997–2026 Citrix Systems, Inc. All rights reserved.

as the SecurSpaces entry point inside the IDE. Developers who start their day from a local IDE, a Progressive Web
App, or a Quickstart link no longer have to return to the console for routine actions.

Previously the extension listed workspaces and offered little beyond start, pause, and connect. Developers who
worked mainly from their IDE had no way to see what a workspace contained or to reach the console actions they
needed, which meant constant context switching.

Key capabilities include:

- Browse workspaces grouped by project, with the connected workspace highlighted and sorted to the top.
- Start, pause, connect in the current or a new window, and switch between workspaces.
- Expand a workspace to open its cloned repositories, attached mount points, and running workspace apps.
- Open a folder in a terminal, or copy a workspace app URL, from the right-click menu.
- See connection quality in the status bar, with round-trip time and estimated bandwidth.
- Jump to the console to create, edit, or delete a workspace, or to reach profile and troubleshooting pages.

The extension now runs on your local machine, and a small companion extension runs inside the workspace to report
activity and answer connection measurements. Your access token stays on your local machine.

Actions that depend on newer platform features are hidden when the extension is connected to an older
SecurSpaces platform, and the sidebar explains why.

For more information, see
Connect to a workspace via SSH.

Recover your own workspaces before they are deleted

Workspaces removed by an inactivity custom action are now visible to their owner before they are gone for good.
Profile > Overview > Workspaces separates **Active workspaces** from **Workspaces pending deletion**, and
the pending list shows the permanent deletion deadline in UTC with actions to restore or permanently delete.

SecurSpaces also emails the workspace owner when a workspace is moved to the recycling bin, and again 24 hours

© 1997–2026 Citrix Systems, Inc. All rights reserved.

before it is permanently deleted. Previously a workspace could disappear with no warning and no self-service way
to get it back, which was a common source of support requests.

For more information, see
Profile Overview.

Starting a workspace counts as use

A workspace that is started regularly is no longer treated as unused. Starting a workspace from the console, the
REST API, or the IDE extension now counts towards the **Inactive Workspace** condition, so a workspace someone
relies on is not deleted by an inactivity policy simply because they never opened a session inside it.

Starting a workspace still does not postpone the idle timeout that pauses it, so the cost savings from idle
pausing are unchanged.

For more information, see
Custom Actions.

Guided workspace creation, and a way to require templates

Creating a workspace now starts with a dialog that asks how you want to create it: from a template, from
scratch, or by copying an existing workspace. **Create from Template** is marked as recommended. This
replaces the split button, where the template and copy routes were hidden in a drop-down and easy to miss.

Platform administrators can also now require templates. A new **Require templates for workspace creation**
permission on a role restricts everyone holding that role to creating workspaces from a template; the custom
and copy options are shown but disabled, with a tooltip explaining why. Use it where every workspace in a
project has to be reproducible and consistent.

For more information, see
Create a workspace and
Roles and permissions.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Archive workspace templates

Workspace templates can now be archived and restored. Archive a template to stop it from being used for new workspaces while keeping the existing workspaces that were created from it.

An archived template cannot be selected in the create-workspace flow or through a Quickstart link, and is hidden from the Templates list unless you choose to show archived templates. Because a template cannot be deleted while workspaces still use it, archiving is the supported way to retire a template that is still in use.

For more information, see
Templates.

Deleted workspace recovery

Deleted workspaces now move to a recoverable **Deleted Workspaces** area before permanent deletion. This gives
users a safety window to restore a workspace that was deleted accidentally and helps administrators manage the
retention period.

Project owners can restore or permanently delete workspaces from the project settings view. Platform
administrators can review deleted workspaces across projects and adjust how many days deleted workspaces remain
recoverable before they are removed permanently.

For more information, see Profile Overview.

Installation and ingress configuration improvements

Administrators have more deployment options in this release. Helm values now support license configuration
during deploy or upgrade, HAProxy ingress annotations, and an option to skip rendering cert-manager Certificate
resources for clusters that do not use cert-manager.

These options reduce manual post-installation work and help fit SecurSpaces into a wider range of Kubernetes
environments.

For more information, see
Technical Requirements.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SecurSpaces 2026.4

SCIM user and group responses include membership details

SCIM responses now include more complete group membership information for identity providers that validate
membership through the SecurSpaces SCIM API. The `/Users` endpoint returns the direct groups a user belongs to,
and the `/Groups` list endpoint returns each group's members without requiring a separate filtered group lookup.

This improvement helps administrators connect SecurSpaces to SCIM providers that expect user and group membership
data to be available during synchronization checks. Users without group memberships still omit the `groups`
attribute, so empty group data is not added to their SCIM user response.

For more information, see SCIM.

Quickstart link management in the REST API

Platform teams can now manage workspace template Quickstart links through the REST API. This helps teams automate
Quickstart links for developer portals, repository README files, and other onboarding workflows without manually copying
links from the SecurSpaces console.

The API can enable, retrieve, and revoke a template's Quickstart link. Responses include the URL and ready-to-use
Markdown and HTML embed snippets, so automation can keep portal content synchronized with the current template link.

Key capabilities include:

- Enable, retrieve, and revoke Quickstart links for workspace templates
- Retrieve the raw URL, Markdown snippet, and HTML snippet for a Quickstart link
- Use light or dark embed button styles for generated snippets
- Create workspaces from Quickstart links through the same flow as UI-generated links

For more information, see REST API and
Templates.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Custom Access Items in the REST API

Custom Access Items are now available through the REST API for workspaces and workspace templates. This gives platform
teams API parity with the SecurSpaces console for adding one-click access entries to internal tools or services that run inside
a workspace.

API callers can include Custom Access Items when creating workspaces and templates, update the full set of items on an
existing workspace, and read configured items from workspace and template responses. Each item can define a display
name, port, command, optional icon, and optional health probe.

Key capabilities include:

- Add Custom Access Items when creating a workspace or workspace template
- Update or remove Custom Access Items on an existing workspace
- Read Custom Access Item status and URLs from workspace API responses
- Receive machine-readable validation errors for invalid ports, duplicate names, missing commands, or unsupported icons

For more information, see REST API and
Create a Workspace.

Self-hosted release documentation

SecurSpaces product documentation is now bundled with each release and served from the control plane. This helps air-gapped and
restricted-network deployments keep documentation available without relying on external internet access.

Users can open the bundled documentation from the SecurSpaces environment at `/docs/`. The release also includes an `llms.txt`
format so AI tools can read the same release-matched documentation when your organization permits that access.

Key capabilities include:

- Access release-matched product documentation from the deployed SecurSpaces environment
- Keep documentation available in environments without direct internet access
- Provide AI-readable documentation through `llms.txt`

For more information, see Give your AI assistant access to SecurSpaces documentation.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Workspace API updates for IDE extensions

The workspace APIs now include more information and operations for IDE extension integrations. These updates help IDE
extensions show richer workspace details and manage personal workspaces without requiring users to switch back to the
SecurSpaces console.

The new API support includes workspace details for attached repositories, mount points, and data buckets. IDE extensions
can also use capability flags from `/version.json` to enable features only when the deployed SecurSpaces version supports them.

Key capabilities include:

- Retrieve personal workspace details from `GET /v1/personal_workspaces/{ workspace_id }`
- Delete a personal workspace with `DELETE /v1/personal_workspaces/{ workspace_id }`
- Use `/version.json` capability flags for extension feature gating
- Read VSCode version fields in workspace API responses

For more information, see REST API and
VSCode versions.

AI Gateway routing from Helm values

Platform administrators can now configure AI Gateway routing directly in the SecurSpaces Helm values. This update lets deployments intercept
selected AI service domains and route that traffic through an internal AI Gateway service.

Use this configuration when your organization wants centralized control for AI provider access from workspaces. The Helm values support
both platform-wide settings and region-specific overrides for external regions.

Key capabilities include:

- Configure the internal AI Gateway service address with `platform.aiGateway.address`
- List AI provider DNS endpoints to intercept with `platform.aiGateway.dnsEndpoints`
- Override the platform-wide values per region with `region.aiGateway`

For more information, see Configure AI Gateway routing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

X.509 authentication for external MongoDB

SecurSpaces can now connect to an external MongoDB deployment by using X.509 certificate authentication. This gives organizations an alternative
to username and password authentication for external MongoDB services such as MongoDB Atlas.

Administrators can select the `MONGODB-X509` authentication mechanism in Helm values and provide the client certificate, private key,
and optional certificate authority bundle either as base64-encoded values or from an existing Kubernetes Secret.

For more information, see External MongoDB X.509.

Guidance for blocked outbound SSH connections

When a workspace tries to start an outbound SSH connection and no SSH method is configured for the target host, SecurSpaces now shows an
actionable terminal message before closing the connection. Previously, the user saw only that the SSH connection closed.

The message explains that an administrator must enable either Personal SSH Identity or a Connected SSH Service before the workspace can
connect to external hosts over SSH.

For more information, see Connect to external servers and services via SSH.

Smarter workspace idle detection

SecurSpaces now uses a hybrid approach to determine whether a workspace is truly idle before pausing it. Previously, an open SSH connection was enough to keep a workspace running indefinitely, even when no one was actively working. This led to unnecessary infrastructure costs, for example, when developers left their IDE open overnight or over the weekend.

The new idle detection engine introduces three layers of intelligence:

- **IDE activity signals take priority.** When the SecurSpaces IDE extension is active, the system uses granular input signals such as keyboard and mouse activity to determine whether a user is present. An open but unused SSH connection alone no longer prevents a workspace from pausing.
- **SSH fallback for unsupported IDEs.** For developers using IDEs that do not support the SecurSpaces extension, the system falls back to the previous SSH-based logic, so active work is not interrupted.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- **Manual override.** security officers can deactivate idle detection for select workspaces to support long-running services, such as shared development tooling or components. When activated, the workspace continues to run even without direct user input.

For more information, see Workspace idle detection for SSH connected workspaces

Shared storage mount points

Teams that work with large, shared datasets, such as ML training data, model artifacts, or experiment results, can now mount external storage directly into their workspaces. This new capability, called Mount Points, removes the need to copy data into the cluster before it becomes accessible.

Previously, SecurSpaces only offered Data Buckets, which function as point-in-time snapshots. While useful for versioning, snapshots duplicate data onto the cluster, require significant disk space, take a long time to replicate for large datasets, and prevent real-time collaboration because updates require a manual publish cycle.

With mount points:

- **Platform admins** enable the feature globally under Settings > Integrations.
- **Project owners** configure mount points by specifying the storage details, including server address, share name, default mount path, and access permissions (read-only or read-write).
- **Developers** attach available mount points when creating a workspace or workspace template. The mount path can be customized per workspace, and the system prevents path conflicts.

This initial release supports only Azure file-based storage.

For more information, see Sharing Data Between Users - Mount Points.

Unified access with Citrix DaaS (Technical Preview)

SecurSpaces now integrates with Citrix DaaS and Citrix Workspace to provide a unified access experience across Linux, Windows, and macOS development environments. Until now, SecurSpaces supported only Linux-based Kubernetes workspaces, excluding teams that depend on Windows or macOS tooling, such as Xcode, full Visual Studio, or platform-specific simulation tools.

With this integration, developers can access both traditional SecurSpaces workspaces and Citrix-based virtual desktops and applications from a single console. Users only need to sign in once in the SecurSpaces console, and all SecurSpaces- and Citrix-based resources can be accessed seamlessly without further authentication.

This feature provides:

- Unified access to Kubernetes-based and Citrix-based apps and workspaces from the SecurSpaces console

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Integrated authentication with single sign-on
- Access to virtual apps and desktops with the native Citrix Workspace app

This feature allows organizations to bring more developer teams onto one secure platform while leveraging their existing Citrix infrastructure and licensing.

For more information, see Citrix DaaS Integration.

Independent VS Code version management

VS Code updates are no longer tied to SecurSpaces platform releases. Previously, customers had to wait for a full SecurSpaces upgrade to get a newer VS Code version, which could delay access to new editor features and plugin compatibility.

With this release, VS Code is delivered and updated independently from the SecurSpaces platform. Key capabilities include:

- **Controlled updates and rollback.** Administrators can adopt new VS Code versions when ready and roll back to a previous version if issues arise.
- **Offline and air-gapped support.** Organizations that restrict internet access from the control plane or workspaces can host VS Code update files on an internal distribution point such as an artifact repository or network share.
- **Built-in notifications.** SecurSpaces admins and security officers receive notifications through the built-in notification system when a new VS Code version becomes available.
- **User-controlled upgrades.** Developers can control when the VS Code update is applied to each of their workspaces, similar to the existing workspace upgrade flow.

For more information, see VSCode Versions.

Native support for Kiro IDE

Kiro is now supported as an IDE for SSH-based workspace connections, joining VS Code Desktop, JetBrains Gateway, Cursor, and Windsurf. Previously, Kiro users had to manually configure SSH connections and extension installations to access their SecurSpaces workspaces.

With this update, the Connect via SSH modal in the SecurSpaces console includes Kiro as a launch option. Selecting it opens Kiro on the developer's local machine, automatically installs or activates the SecurSpaces extension, and establishes the remote SSH connection to the target workspace.

For more information, see Connect to a workspace via SSH

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Improved user onboarding flow

The Add User flow for projects has been redesigned to reduce errors and speed up onboarding. Project owners can now add users in bulk by pasting a list of email addresses, with inline validation and clear error feedback. Single-user search is limited to platform users not yet in the project, avoiding confusion. When adding external users, the flow guides the admin through the required additional details before completing the operation.

For more information, see Add and remove users and groups - Bulk add users

Automatic SecurSpaces extension installation for SSH connections

When connecting to a workspace via SSH from VS Code Desktop, SecurSpaces now automatically installs the required SecurSpaces extension on the remote workspace. This removes a previously manual step and reduces connection setup failures.

Configurable ghost workspace resources via Helm

Ghost workspace resource specifications can now be configured through the Helm chart, giving administrators more control over resource allocation for pre-provisioned workspaces.

Kubernetes PriorityClass support for workspace pods

Administrators can now assign a Kubernetes PriorityClass to workspace pods through a new optional Helm configuration field. When set, all newly created workspace pods are assigned the specified priority class, allowing Kubernetes schedulers and tools like Karpenter to make better scaling and scheduling decisions. When the setting is omitted, existing behavior is unchanged.

Debian desktop image with browser access

A new preconfigured Debian-based container image with NoVNC is now available, providing a full desktop environment directly accessible in the browser. This option is useful for teams that need GUI-based tooling or testing environments within their SecurSpaces workspaces.

VS Code updated to 1.114

SecurSpaces now ships with VS Code 1.114, bringing the latest editor features and plugin compatibility improvements.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Improved downloads for large analytics exports

Workspace measurement exports can now reliably handle large report files. Administrators and security officers can download workspace analytics filtered by organization, project, and date range, even when the resulting files are too large for a standard API call. The download mechanism now uses an optimized transfer path to support large file sizes.

For more information, see Workspace resource usage insights - Access workspace measurements

Preserve Host proxy option for workspace apps

Workspace apps now include a Preserve Host proxy option. When enabled, the proxy forwards the workspace app's public URL in the Host header, aligning it with the Origin and X-Forwarded-Host headers. This allows tools like Jupyter Notebook and pgAdmin to work behind the SecurSpaces proxy service without custom per-tool proxy configuration. The option is turned off by default to avoid breaking local development workflows that expect localhost.

For more information, see Workspace Apps - Override Host Header

Self-service OpenAPI specification downloads

The SecurSpaces REST API documentation page now includes a Download OpenAPI specification button. The OpenAPI spec is automatically packaged and shipped with every minor and major SecurSpaces release, so platform engineering teams can download the spec that matches their currently deployed SecurSpaces version without contacting Citrix.

Custom documentation URL

Platform admins can now replace the default SecurSpaces documentation link with a custom URL that points to their organization's internal documentation. When enabled under Settings > General > Documentation, the custom link applies globally and updates the Documentation entry in the user profile menu for all users, projects, and organizations. When disabled, the link reverts to the default SecurSpaces documentation.

For more information, see General Settings - Documentation

Custom OpenVSX marketplace for VS Code extensions

Platform administrators can now point all VS Code workspaces to a custom OpenVSX marketplace under Settings > General > OpenVSX Marketplace, replacing the previous manual `EXTENSIONS_GALLERY` environment variable approach. The setting is inherited across all

© 1997–2026 Citrix Systems, Inc. All rights reserved.

organizations, projects, and workspaces. New workspaces use the configured marketplace immediately; existing workspaces pick up the change after their next restart.

For more information, see General Settings - OpenVSX Marketplace

UI enhancements

Simpler workspace template versioning

Edit, Publish, and Set as Default actions are now directly available in the template version list, making versioning steps faster and easier to discover.

Streamlined Add Image workflow

The container image import flow now starts with the image URL, auto-fills fields where possible, supports inline credential entry, and includes clearer descriptions and tooltips.

Alphabetical sorting for workspaces and templates

Workspace and template lists are now sorted alphabetically by default.

Separate control for in-app guides

Usage Analytics settings now let administrators keep analytics enabled while turning off in-app guides independently.

Better search in dropdowns

Larger dropdowns now show a visible search field that is focused on open, with options sorted alphabetically.

Consistent select components

SecurSpaces now uses a unified select component with consistent search and sorting behavior across the console.

Cleaner Set as Default for container images

The Set as Default action is now shown more selectively in the container image version list, reducing visual clutter while keeping the action accessible.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Startup scripts description

The Startup Scripts step in the workspace and template wizard now includes a description explaining when and why to use startup scripts.

Language selector moved to user profile

Language selection has moved from the page footer to user profile settings, now labeled with language names instead of flags.

Consistent wizard exit warning

SecurSpaces now always warns users before they leave the workspace or template wizard, regardless of navigation path, and uses clearer dialog labels.

Improved empty secrets message

When no secrets exist, the wizard now shows guidance on how to add one instead of implying that a filter returned no results.

Clearer filter no-results messages

Filtered lists now show a contextual no-results message with a Clear Filters action so that users can distinguish between empty results and missing data.

Standardized UI components and tooltips

Icons, tooltips, and styling patterns are now more consistent across the SecurSpaces console.

Bug fixes

Context menus close on scroll

Context menus now close automatically when the user scrolls, preventing visual artifacts from continuous repositioning.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Maintenance banner stays aligned during scrolling

The maintenance notification banner no longer briefly slips under the menu bar when scrolling on long pages.

Validation checkmark no longer appears on empty input

The secret name input field no longer shows a success checkmark when the field is empty or invalid.

Consistent image cache scheduling across nodes

Image cache pods are now scheduled with node-level anti-affinity, ensuring one pod runs on every Karpenter-managed workspace node instead of one per availability zone. This restores predictable warm node behavior in multi-AZ clusters.

SecurSpaces 2025.10

This release contains the following new features:

Renewal warning for CA certificates

SecurSpaces now displays a warning when a CA certificate is approaching expiration, enabling administrators to take timely action to renew certificates and prevent service disruptions.

Workspace resource usage insights

SecurSpaces now provides historic insights into workspace CPU and RAM consumption. This data is automatically collected and stored in the SecurSpaces database, and can be accessed via the workspace-measurements and workspace-measurement-samples APIs to support rightsizing analysis and long-term trending insights.

The system gathers the following information:

- CPU usage over time
- RAM usage over time

The data is retained for 7 days.

For more information, see Workspace resource usage insights

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enhanced idle detection for SSH sessions

When users connect to a SecurSpaces workspace via SSH with the SecurSpaces/Strong Network plugin, SecurSpaces can now monitor activity with greater precision. This will allow the system to pause idle workspaces more reliably, improving cost efficiency without disrupting active sessions. Users without the SecurSpaces/Strong Network plugin installed in their local IDE will be asked to install it via a notification within the SecurSpaces console. Administrators can use the SecurSpaces API `/v1/metrics/ssh-workspaces-no-extension-usage` to determine a list of users connecting via SSH without the plugin installed.

> Note:
>
> The initial version of this release will not change the behavior of the SecurSpaces scheduled to minimize the disruption to existing users. A future minor version will enable the scheduler changes.

Enhanced Quickstart workspace creation

The Quickstart interface, used when creating a new workspace via a Quickstart link, has been enhanced. Before provisioning, users can now review:

- The image and template used to create the workspace
- The organizational location where the workspace will be deployed

Additionally, users can select the template version and geographic deployment location, providing greater control and transparency during workspace setup.

For more information, see Quickstart

Support for Azure Cosmos DB

SecurSpaces now supports Azure Cosmos DB as a managed database option, in addition to MongoDB Atlas. This gives teams greater flexibility in choosing the database service that best fits their workloads and cloud environment.

Workspace template flow: Add draft & promote functionality

New Workspace template versions are now created in a draft state. Drafts can be modified and tested until they are explicitly promoted to the default version. This workflow simplifies the process of iterating on templates while preventing users from inadvertently using versions that are not production-ready.

For more information, see Create a new version of a Template

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Template duplication in the SecurSpaces console

Project owners can now duplicate existing Workspace templates directly within the SecurSpaces console. This makes it easier to create new templates that share the same toolstack and integrations as existing ones, while allowing for fine-tuned configuration to meet specific developer needs.

Workspace resource visibility and sorting

The Project/Workspace view now displays the full resource configuration (CPU, RAM, and storage) for every workspace in a project. Workspaces can also be sorted by these attributes, enabling project owners to quickly identify high-resource allocations and support rightsizing activities.

New filters in Project/Workspaces view

A new filter option has been added to the Project/Workspaces view, making it easier to identify workspaces with specific characteristics within large projects. Available filter criteria include:

- Owner –workspace owner
- Image –base image used for the workspace
- Created On –creation date
- Status –current workspace status
- CPU, RAM, Storage –allocated resources

This enhancement streamlines workspace management and helps quickly locate relevant workspaces.

For more information, see Filtering Workspaces

Optimized Console Responsiveness

We have significantly optimized the way the SecurSpaces console loads data, resulting in a much more responsive and fluid user experience.

- Near-instant navigation: Actions that previously had a short delay are now almost instant. For example, navigating from the platform level into a specific project is notably faster.
- Improved workflow: This foundational enhancement minimizes wait times, improving your overall workflow and making the console feel smoother and more efficient.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Interactive onboarding guides

When accessing the SecurSpaces web console for the first time, users are now presented with interactive onboarding guides. These guides highlight key functionality and walk through important first steps, helping new users get up and running more quickly.

Updated Visual Studio Code version

SecurSpaces workspaces now include Visual Studio Code v1.105.1, providing the latest features, improvements, and fixes.

Improved workspace creation workflow

The input fields for creating a new workspace from a template have been reorganized to reduce the number of clicks required. Additionally, the proposed workspace name is now automatically generated using the format <First Name><First 3 letters of Surname>-<TemplateName>, streamlining the setup process and ensuring consistent naming.

For example: StevenGal-Frontend Workspace

Enhanced UX for workspaces without resource limits

SecurSpaces now allows customers to create workspaces without CPU or RAM limits, enabling fully elastic scalability. Workspaces configured with unlimited resources will display an infinity symbol for the affected resource, providing a clear visual indicator of this configuration.

Default selection of current user for resource ownership

Whenever SecurSpaces prompts for an owner of a newly created resource, the current user is now listed at the top of the user list. This change streamlines common workflows and speeds up the resource creation process.

Enhanced user details page

The user details page now displays user-configured workspace schedules and lists all workspaces with custom schedules. This page is also accessible to project owners, in addition to security officers. The enhanced view provides better visibility into a user's context and special configurations, aiding troubleshooting and workspace management.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Backstage plugin for SecurSpaces

SecurSpaces now offers a plugin for Backstage, enabling users to list and access all workspaces associated with a specific software project, as well as create new workspaces directly from Backstage. For organizations using a Backstage-based Integrated Developer Portal, this integration streamlines developer workflows and simplifies workspace management.

For more information, see SecurSpaces Workspaces Plugin for Backstage

HashiCorp Vault integration for secret management

SecurSpaces now integrates with HashiCorp Vault, the leading secret management solution. When enabled, all secrets previously stored in the SecurSpaces database are securely stored in Vault. This includes:

- Platform secrets: Platform SSH private key, OAuth app secrets, email gateway secrets, and workspace image registry credentials
- User secrets: User SSH personal identity, private SSH keys, and GPG keys

This integration enhances security by centralizing secret management and leveraging Vault's robust access controls and auditing capabilities.

For more information, see Use HashiCorp Vault as a Secret Manager

Usage Telemetry

SecurSpaces now collects usage telemetry to help improve the platform. This telemetry is used for understanding feature adoption and identifying areas for performance and usability improvements. No personal data is collected, and all information is handled in accordance with organizational privacy policies.

Pendo integration for in-app guidance and analytics

SecurSpaces now collects usage telemetry to help improve the platform. This telemetry is used for understanding feature adoption and identifying areas for performance and usability improvements. No personal data is collected, and all information is handled in accordance with organizational privacy policies.

`https://FQDN/platform/settings/analytics/usage_analytics`

For more information, see Usage Analytics

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fixed issues

October 06, 2026

Citrix SecurSpaces™ 2026.9 includes the following fixed issues

Release Date	Version	Category	Description
October 06, 2026	2026.9.4	New features	Added support for upgrading internal Percona MongoDB deployments from Percona Operator 1.20.1 to 1.23.0 with an installer-provided upgrade kit. The updated chart also exposes Percona backup configuration for storage targets, schedules, retention, point-in-time recovery, and the database service account used by backup agents.
		New features	Added support for keyless Google Cloud Storage backups for internal Percona MongoDB on Google Kubernetes Engine using Workload Identity Federation.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		Security	Updated the workspace sidecar image base packages and containerd to address high-severity vulnerabilities.
		Security	Updated the NetScaler CPX ingress controller chart from 4.1.17 to 4.2.26 to address known ingress controller vulnerabilities.
September 29, 2026	2026.9.3	New features	Added support for stable `specification_id` values in Citrix SecurSpaces Flex managed tier definitions. A managed tier version can now keep the same identifier across tenants when Citrix pins the value in deployment values.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where successful Citrix SecurSpaces Flex managed tier availability changes and bulk tier-version updates were missing from the audit log. The audit log now records who performed the action, when it happened, and the result.
		General	Resolved an issue where the outbound SSH guidance shown in workspaces pointed to the previous documentation URL. The guidance now opens the current SecurSpaces external services documentation.
September 28, 2026	2026.9.2	New features	Added Citrix SecurSpaces Flex support for Citrix-managed workspace specification tiers, including managed tier versions, lifecycle labels, workspace update prompts, and bulk tier-version updates.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		New features	Added REST API support for managing VSCode versions at platform, organization, and project scope. The API can list, import, rename, delete, and set defaults for versions, and manage platform-level VSCode registries.
		General	Resolved an issue where Helm upgrades with the internal Percona MongoDB deployment could regenerate database passwords and leave platform services unable to authenticate. Server-side upgrades now reuse existing passwords unless an operator sets a new value, and credential changes roll the services that consume them.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where a workspace could stay on an old sidecar image after a platform upgrade without showing an update prompt. Workspaces now show an update when the stored sidecar image differs from the version shipped by the deployment.
		General	Resolved an issue where project **People > More details** could open the wrong user details route or show the wrong organization in breadcrumbs. The project user details flow now keeps the project and organization context aligned.
		General	Resolved an issue where release service images could have different manifest digests depending on the regional registry used. Future release images now use consistent compressed layers across release regions.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where concurrent Kubernetes updates could cause workspace start, restart, or edit operations to fail with a StatefulSet conflict. StatefulSet updates now retry on conflicts.
September 21, 2026	2026.9.1	General	Resolved an issue where a workspace stopped for exceeding its memory limit reported `Reason: Unknown` and advised updating the workspace image. The failure is now reported as `OOMKilled`, with memory-specific guidance.
		General	Resolved an issue where SCIM provisioning stored a user's email address as both the given name and the family name. `name.givenName` and `name.familyName` now map to the user's name.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where marketplace extensions preconfigured in a workspace template were not installed.
		General	Resolved an issue where a searchable multi-select list cleared the search text while the user was typing.
		General	Updated the default workspace image to `2.3.9`.
August 13, 2026	2026.9.0	Security	Updated dependencies and images to address known vulnerabilities.
		Security	Resolved an issue where some role-protected API requests could bypass access checks.
		Security	Resolved an issue where some flows could accept the wrong token type.
		Security	Resolved an issue where user-management actions could affect users outside the caller's scope.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		Security	Resolved an issue where local secrets could be copied without verifying container ownership.
		Security	Resolved an issue where sidecar file streaming could expose unintended workspace files.
		Security	Resolved an issue where sidecar Git handling could accept unsafe command arguments.
		Security	Resolved an issue where workspace Docker access was broader than required.
		Security	Resolved an issue where the image builder ran with more privilege than required. It now uses rootless BuildKit.
		Security	Removed test license keys from production images.
		Security	Rotated credentials that had been committed to the repository, and added operator recovery guidance.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		Security	Hardened the AWS Mount Point lifecycle, including attach, update, and delete.
		Security	Hardened mount point handling: storage classes without TLS are hidden, deletion from the recycle bin is blocked, and error messages no longer disclose internal detail.
		Security	Hardened the documentation served from the control plane at `/docs/`.
		New features	Added AWS Mount Point support for Amazon EFS and Amazon S3 Files storage.
		New features	Added a redesigned IDE extension sidebar with repositories, mount points, and workspace apps.
		New features	Added connection quality monitoring in the IDE status bar.
		New features	Added user, project, and workspace metadata on AI requests forwarded to an AI Gateway.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		New features	Added AI Gateway health checking with a configurable fail-open behavior.
		New features	Added a sample Grafana dashboard for AI usage and token consumption reporting.
		New features	Added desired state configuration tracking of the template version used by each workspace.
		New features	Added template version status indicators and an Associated Workspaces window with CSV export.
		New features	Added a workspace update flow that aligns a workspace with the default template version.
		New features	Added archive and restore for workspace templates that still have associated workspaces.
		New features	Added dot files in **Profile > Configuration** for personal workspace setup.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		New features	Added a Setup Checklist on project pages for guided onboarding.
		New features	Added deleted workspace recovery during the configured retention period.
		New features	Added Helm support for applying a platform license during deploy or upgrade.
		New features	Added HAProxy ingress support and optional cert-manager Certificate rendering.
		New features	Added GitHub Copilot support for workspaces behind an AI Gateway.
		General	Updated the product interface and email templates to use the Citrix SecurSpaces product name.
		General	Resolved an issue where data bucket uploads could start from ineligible workspaces.
		General	Resolved an issue where data bucket upload status could be misleading.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where the data bucket upload modal could show stale bucket data.
		General	Resolved an issue where image import could use the wrong credential or image URL.
		General	Resolved an issue where a failed image import could become the default image tag.
		General	Resolved an issue where workspaces could start with the wrong container image tag.
		General	Resolved an issue where reused image tags could leave stale workspace image content.
		General	Resolved an issue where VSCode version settings could be lost in workspace flows.
		General	Resolved an issue where a private VSCode registry URL could remain stale.
		General	Resolved an issue where signed Git operations could fail with `commit.gpgsign=true`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where workspace startup could stall during system startup scripts.
		General	Resolved an issue where workspaces could stay in `Pending` because of storage settings.
		General	Resolved an issue where a stale sidecar IPC socket could cause `CrashLoopBackOff`.
		General	Resolved an issue where workspace schedules with a `00:00` start time did not save.
		General	Resolved an issue where manual **Run** actions were not counted as activity.
		General	Resolved an issue where Profile page SSH addresses could omit the workspace region.
		General	Resolved an issue where organization role checks did not apply to protected pages.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where disabled role references could disrupt project loading.
		General	Resolved an issue where the Disable Role modal showed `-1` instead of a placeholder.
		General	Resolved an issue where **Connect to GitHub** did not start OAuth for new users.
		General	Resolved an issue where workspace template version drafts could sort unpredictably.
		General	Resolved an issue where create-from-existing showed incorrect specification values.
		General	Resolved an issue where HTTP and SSH connected service selections did not persist.
		General	Resolved an issue where workspace specification template restrictions could be lost.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where workspace update actions appeared when no update was available.
		General	Resolved an issue where template lifecycle actions could leave orphaned workspace state.
		General	Resolved an issue where **Add to Favorite** stayed open after selection.
		General	Resolved an issue where the audit page **Select All** filter option did not apply.
		General	Resolved an issue where AI dashboard token totals were inconsistent.
		General	Resolved an issue where AI traffic from SSH workspaces lacked observability metadata.
		General	Resolved an issue where 1-click demo deployment could fail in common retry scenarios.
		General	Resolved an issue where read-only AWS mount points could not be mounted.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where Azure Files mount points had several defects affecting attach and update.
		General	Resolved an issue where a workspace could briefly report a `DESTROYED` state while it was being created.
		General	Resolved an issue where the workspace wizard could become stuck after **Discard**.
		General	Resolved an issue where editing a workspace specification did not reach running workspaces, pods, and templates.
		General	Resolved an issue where replacing a specification template could write outside its intended scope.
		General	Resolved an issue where Community Edition license activation could fail.
		General	Resolved an issue where the Insights resource usage chart showed gaps, unstable colors, and an inaccurate tooltip.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where updating workspace properties could cause a backend error.
		General	Resolved an issue where the workspace update dialog showed the wrong state for a template version.
		General	Resolved an issue where authorizing an SSH key did not list workspaces the user owns without the manage permission.
		General	Resolved an issue where invalid repositories were included in the duplicate path check.
		General	Resolved an issue where searching for users responded slowly.
		General	Resolved an issue where image build logs could not be streamed.
		General	Resolved an issue where image builds could fail when the cluster used an external proxy.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

		General	Resolved an issue where the platform did not recover promptly after losing its leader election lock.

Citrix SecurSpaces 2026.4 includes the following fixed issues

Release Date	Version	Category	Description
August 24, 2026	2026.4.10	Security	Updated Helm chart security contexts to disallow privilege escalation for platform service containers and the syslog permissions init container.
		General	Resolved an issue where SCIM user responses did not include group membership and SCIM group list responses omitted members unless a specific group was requested. `/Users` now returns direct group membership and `/Groups` list responses include members.
August 14, 2026	2026.4.9	New features	Added HAProxy ingress support.
		General	Resolved an issue where project creation could fail after the **Project Owner** role was disabled.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where Azure 1-click demo deployment could fail and leave incomplete network rules.
		General	Resolved an issue where NetScaler ingress could fail on Google Cloud.
		General	Resolved an issue where Google Kubernetes Engine 1.35 probes could time out too quickly.
July 20, 2026	2026.4.8	Security	Updated dependencies and images to address known vulnerabilities.
		New features	Added REST API support for managing workspace template Quickstart links.
		New features	Added REST API support for Custom Access Items on workspaces and templates.
		General	Resolved an issue where template REST API responses omitted setup scripts.
		General	Resolved an issue where VSCode registry records could be missed or left stale.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
July 2, 2026	2026.4.7	Security	Updated dependencies to address known vulnerabilities.
		New features	Added SecurSpaces documentation served from the control plane at `/docs/` for air-gapped deployments.
		New features	Added IDE extension API support for personal workspace endpoints and VSCode version fields.
		General	Resolved signed Git commit failures when `commit.gpgsign=true` was set.
		General	Resolved JFrog artifact repository access failures for users without a configured token.
		General	Resolved an issue where CPU and memory resource limits could be lost after workspace edit operations.
		General	Resolved an issue where Create Workspace API responses could miss expected fields.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where attached repositories were missing from personal workspace API responses.
		General	Resolved an issue where IDE extension quick-link sign-in did not redirect to non-project routes.
June 9, 2026	2026.4.6	Security	Updated the Go runtime and build toolchain to address known vulnerabilities.
		New features	Added AI Gateway routing support via Helm values for selected AI service domains.
		New features	Added X.509 certificate authentication for SecurSpaces connections to external MongoDB deployments.
		General	Resolved outbound SSH connections closing without explanation when no SSH method was configured.
		General	Resolved an issue where the add icon on the empty home page create card appeared flattened.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue where Pendo in-app guide sessions could continue after logout.
June 4, 2026	2026.4.5	Security	Resolved JFrog anonymous tokens accepted during OAuth sign-in.
		General	Resolved a nil panic in the OAuth client secret rotation job when no OAuth application was configured.
		General	Resolved an issue where the auto-delete action used an incorrect workspace last-used time.
		General	Resolved GPU configuration options being hidden for the security officer role in workspace templates.
		New features	Added workspace Docker registry mirror support via `workspaceDockerRegistryMi` Helm value.
May 29, 2026	2026.4.4	Security	Upgraded Go and Docker toolchain versions to address known vulnerabilities.
		Security	Updated the MongoDB container image to address known vulnerabilities.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		Security	Bumped the default workspace image to `2.3.6` to address known vulnerabilities.
		General	Resolved an issue where the “update available”icon persisted after a workspace update completed.
		General	Resolved non-deterministic color assignment in Resource Allocation charts. Colors now sort by region.
		General	Resolved an issue where Custom Actions conditions were not applied correctly.
		General	Resolved an issue where the Profile page showed an error on refresh for Developer role users.
		General	Resolved incorrect units for memory and disk when editing resource limits.
		General	Resolved workspace status records not initialized for cloned or template-created workspaces.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved the **Advanced** dropdown appearing empty for some roles in workspace settings.
May 12, 2026	2026.4.3	General	Resolved a Kubernetes cluster cost issue caused by warm node pre-provisioning.
		General	Resolved an image pod caching issue by re-aligning node anti-affinity to `kubernetes.io/hostname`.
		New features	Added a platform setting to configure a custom upstream URL for the VS Code Extensions Gallery.
		New features	Increased the maximum configurable workspace specification limits for CPU, GPU, memory, and disk.

SecurSpaces 2025.10 includes the following fixed issues

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
April 30, 2026	2025.10.21	Security	Resolved an issue where a rejected workspace SSH approval still authorized the presented public key, allowing a later connection with the same key to succeed. SSH keys are now authorized only when the approval request is accepted.
		Security	Tightened file and directory permissions in the installer. Generated directories, Terraform variable files, and the CA certificate bundle are now created with owner-only access to address Mend SAST findings (CWE-732).
		Security	Updated the MongoDB container image to address known vulnerabilities reported by Mend.
		Security	Bumped the default workspace image to `2.3.6` to address known vulnerabilities.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved slow query performance for `ip_pod_assignment` and `workspaces_v5` collections that could lead to proxy `OOMKilled` events. Added a compound index on `(ip, region_id)` for `ip_pod_assignment` and a `team_id` index on `workspaces_v5`.
		General	Resolved a delay when switching to the Workspaces tab in the user interface.
		General	Resolved an issue where the table component did not show its loading state on initial data fetch.
		General	Resolved an issue where the copy-to-clipboard tooltip did not appear on the first hover or copy action.
		General	Resolved an issue where opening a terminal session from the dropdown launched a duplicate window instead of focusing the existing one.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
April 23, 2026	2025.10.20	Security	This release includes several vulnerability fixes across the platform's core services.
		Security	Updated the Go language runtime to version 1.25.9 and resolved frontend CVEs reported by Mend.
		General	Resolved an issue where workspace setup scripts were lost when a workspace was created through the REST API. Setup scripts are now preserved on create.
		General	Resolved a UI issue where the workspace access item icon appeared taller than expected.
April 14, 2026	2025.10.19	General	Resolved an issue where 1-Click VM provisioning failed when targeting an existing Azure Resource Group.
April 13, 2026	2025.10.18	Security	This release includes several vulnerability fixes across the platform's core services.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		General	Resolved an issue in the OAuth2 authentication flow where email addresses containing uppercase letters caused authentication failures or account mismatches.
April 1, 2026	2025.10.17	Security	This release includes several vulnerability fixes across the platform's core services.
March 30, 2026	2025.10.16	General	Resolved a critical CosmosDB index error that caused backend upgrade failures. This fix ensures that database schema updates and indexing operations complete successfully during platform deployments, preventing service disruptions during the upgrade cycle.
		General	Added GPU-accelerated workspaces with hardware acceleration for machine learning, data science, and graphics-intensive workloads.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
March 17, 2026	2025.10.15	Security	This release includes several vulnerability fixes across the platform's core services.
March 9, 2026	2025.10.14	Networking	Resolved an issue where cross-workspace HTTP access was broken.
February 19, 2026	2025.10.13	General	Resolved an issue with Git Large File Storage (LFS) that affected GitLab repositories when using SSH for authentication.
February 11, 2026	2025.10.12	Security	This release includes several vulnerability fixes across the platform's core services.
February 11, 2026	2025.10.11	General	Resolved an issue where a misleading error message was generated by the measurement cron job for workspaces with no active pods, such as those in a paused state.
January 30, 2026	2025.10.10	Security	This release includes several vulnerability fixes across the platform's core services.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
January 28, 2026	2025.10.9	General	Resolved an issue with the VS Code Extensions Gallery that prevented the discovery and installation of extensions.
		New features	Upgraded the NetScaler Ingress Controller (NetScaler CPX) from version 3.2.22 to 3.3.2. This update provides improved stability and higher performance for ingress traffic management.
January 23, 2026	2025.10.8	Security	This release includes several vulnerability fixes across the platform's core services.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
January 15, 2026	2025.10.7	Security	Refactored the Helm Chart RBAC configuration to enhance the platform's security posture through least-privilege principles. The previously unified ClusterRole has been split into two specific scopes: Namespace-scoped Role manages workload-specific resources, including pods, jobs, and secrets; Minimal ClusterRole is restricted to essential cluster-wide resources, such as nodes, storage classes, and metrics.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
December 18, 2025	2025.10.6	General	Fixed a bug in 1-click VM environments using OAuth integration that prevented local IDEs from successfully connecting via SSH extensions and plugins. This resolution ensures that developers can use local IDE tools, such as VS Code Desktop or JetBrains Gateway, within the authenticated workspace session.
December 15, 2025	2025.10.5	General	Added support for F5 NGINX annotations. Platform administrators can enable dedicated controller support with `platform.useF5NginxController: true`.
		Security	Updated the Go language runtime to version 1.25.5, which addresses known security vulnerabilities.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
December 2, 2025	2025.10.4	General	Fixed a permission issue that prevented users with the security officer role from disabling analytics features on the platform configuration settings page.
		General	Resolved an issue causing the Workspace API component to enter a CrashLoopBackOff state when the system was managing a large number of active or decommissioned workspaces.
		New features	Added a new configuration option to disable Amazon EKS auto-mode detection during cluster setup. Administrators can set `region.clusterConfig.disableAutoModeCheck: true` for more granular control in specific deployment environments.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
		New features	Introduced Terraform support for managing user groups, allowing administrators to provision, update, and manage workspace user groups using Infrastructure as Code practices.
November 26, 2025	2025.10.3	General	Fixed an issue where some Visual Studio Code dependencies failed due to an underlying C standard library requirement. The minimum required `glibc` version is now 2.28 to ensure stability and compatibility with remote VS Code functionality on supported Linux distributions.
		Security	Updated the Go language runtime to version 1.25.4, which addresses known security vulnerabilities.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Release Date	Version	Category	Description
November 21, 2025	2025.10.2	General	Resolved slow database migration performance. An optimization to the database migration engine significantly reduces the time required to run database updates during product rollouts and version upgrades.
November 14, 2025	2025.10.1	General	Removed the dependency on the C standard library (`libc`). The core workspace components no longer require `libc` at runtime.

What's new in SecurSpaces Flex

September 29, 2026

This article summarizes new features, improvements, and fixes for Citrix SecurSpaces™ Flex. For the self-hosted Citrix SecurSpaces release history, see What's new in Citrix SecurSpaces.

September 2026

Managed workspace size catalog

Citrix SecurSpaces Flex now uses a Citrix-managed workspace size catalog. Citrix publishes the available

© 1997–2026 Citrix Systems, Inc. All rights reserved.

workspace size tiers and their versions, and Flex deployments reconcile that catalog during platform updates.

This release adds lifecycle status for managed tier versions. Administrators can see current, older, and
retired tier versions, keep organization and project availability settings, and update affected workspaces to
the current version of the same tier.

Key capabilities include:

- Managed Light, Medium, and Heavy workspace size tiers with version status.
- Organization and project availability controls that remain available to administrators.
- Workspace update prompts when a workspace uses an older managed tier version.
- Bulk tier-version updates that move workspaces to the current version of the same tier.
- Retired tier versions that are hidden from new workspace creation while existing workspaces remain visible.

For more information, see
SecurSpaces Flex workspace templates.

Audit and stable managed tier identifiers

Managed workspace size tiers now provide a clearer operating record. When an administrator enables or
disables a managed tier, or applies a bulk update to move workspaces to the current tier version, SecurSpaces
records the successful action in the audit log with the actor, time, and result.

Citrix can also pin the identifier for a managed tier version in deployment values. This keeps the same managed tier version identifier across Flex tenants, while existing deployments without a pinned value continue
to receive a generated identifier.

August 2026

General Availability: Citrix SecurSpaces Flex

Citrix SecurSpaces Flex is now generally available as a Citrix-managed service within Citrix Platform Flex that delivers secure Linux workspaces for development teams. Delivered through Citrix Cloud,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

this service provides hosted developer environments with no customer infrastructure to stand up or maintain. In this release, Citrix SecurSpaces Flex is applicable only to Azure environments. For more information, see Citrix SecurSpaces Flex overview.

New and enhanced features

- **Citrix-managed control plane** —Citrix operates the Azure Kubernetes Service (AKS) and Azure Cosmos DB infrastructure.
- **Workspace sizing by persona** —Light (2 vCPU / 8 GB RAM), Medium (4 vCPU / 16 GB RAM), and Heavy (8 vCPU / 32 GB RAM), each mapped to a published Platform Flex credit rate.
- **Decoupled workspace size and template** —A single workspace template can be paired with any compute profile on demand.
- **Multi-region support** —A primary Azure region is provisioned at deployment time, and secondary regions can be added later by your Citrix team.
- **Version management** —Citrix manages version updates for your environment, including scheduling of maintenance windows.
- **Network connectivity** —Connect workspaces to customer-owned systems via Azure vNET peering or VPN gateway.
- **Native DLP, RBAC, and audit logging** —Built-in controls for data loss prevention, access governance, and compliance.
- **Sandboxed AI agent support** —AI coding agents run inside workspaces under the same control plane and policy framework as human developer sessions.
- **Platform Flex credit model** —Citrix SecurSpaces Flex credits draw from the same pool as Citrix DaaS Flex, with no separate procurement cycle.

More information

- Citrix SecurSpaces Flex overview
- Licensing for Citrix Platform Flex

SecurSpaces document history

The following table describes important changes to the Citrix SecurSpaces™ product documentation. To receive notifications about these updates, subscribe to the RSS feed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Changes are listed with the most recent first.

Date	Change	Description
17 Aug 2026	Restructured	Documentation reorganized by role into About, Get started, Develop, Manage, Deploy, Administer, Security and compliance, Reference, and Release notes. Articles were split, renamed, and rewritten to match the new structure.
02 Jul 2026	Added new article	Administer the SecurSpaces database
01 Jul 2026	Added new section	System Configuration, a top-level section for platform-level settings such as license, identity providers, integrations, and regions
01 Jul 2026	Added new article	Onboarding email notifications, describing the automatic invitation and project emails, and expanded the Email Gateway Configuration provider details
08 Jul 2025	Added new sections	Organization Settings, Platform Settings and Project Settings
14 May 2025	Document migration	Published Strong Network document
29 Apr 2025	Document migration	Strong Network document
04 Mar 2025	Sample document	Sample Strong Network document

© 2025 Cloud Software Group, Inc. All rights reserved. This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc. This and other products of Cloud Software Group may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal. Citrix, the Citrix logo, NetScaler, and the NetScaler logo and other marks appearing herein are either registered trademarks or trademarks of Cloud Software Group, Inc. and/or its subsidiaries in the United States and/or other countries. Other marks are the property of their respective owner(s) and are mentioned for identification purposes only. Please refer to Cloud SG's Trademark Guidelines and Third Party Trademark Notices (https://www.cloud.com/legal) for more information.