# Citrix Secure Developer Spaces™

# Contents



# Citrix Secure Developer Spaces™

November 8, 2025

Citrix Secure Developer Spaces, formerly known as Strong Network™ is a secure, cloud-based development environment (CDE) platform that enhances developer productivity while maintaining enterprise-grade security. It provides fast onboarding through preconfigured workspaces that are accessible from anywhere—ideal for hybrid and remote teams.

The platform helps protect source code, credentials, and data by eliminating local dependencies and enforcing strong access controls. Its container-based environments integrate with common DevOps tools, CI/CD workflows, and security models such as Zero Trust. Organizations can reduce costs associated with laptops, maintenance, and security software, while gaining real-time visibility and governance over the development lifecycle.

# What's new in Citrix Secure Developer Spaces™

November 18, 2025

Citrix continuously delivers updates to enhance your Citrix Secure Developer Spaces™ 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.

To ensure compatibility and optimal performance, review the latest Technical Requirements for Citrix Secure Developer Spaces.

## Citrix Secure Developer Spaces 2025.10

This release contains the following new features:

### Renewal warning for CA certificates

SDS 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

SDS now provides historic insights into workspace CPU and RAM consumption. This data is automatically collected and stored in the SDS 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

### Enhanced idle detection for SSH sessions

When users connect to an SDS workspace via SSH with the SDS/Strong Network plugin, SDS 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 Citrix SDS/Strong Network plugin installed in their local IDE will be asked to install it via a notification within the SDS console. Administrators can use the SDS 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 SDS 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

SDS 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

### Template duplication in the SDS console

Project Owners can now duplicate existing Workspace templates directly within the SDS 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 SDS 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.

### Interactive onboarding guides

When accessing the SDS 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

SDS 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

SDS 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 SDS 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.

### Backstage plugin for SDS

SDS 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 Citrix SDS Workspaces Plugin for Backstage

### HashiCorp Vault integration for secret management

SDS now integrates with HashiCorp Vault, the leading secret management solution. When enabled, all secrets previously stored in the SDS 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

SDS 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

SDS 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

## Fixed issues

December 11, 2025

Citrix Secure Developer Spaces™ includes the following fixed issues:

### 2025.10.4

#### General

- Fixed a permission issue that prevented users with the Security Officer role from successfully disabling analytics features on the platform configuration settings page.
- 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. This provides more granular control in specific deployment environments. To use this option, add the following setting to your cluster configuration:

```
region:
  clusterConfig:
    disableAutoModeCheck: true
```

- Automation Introduced Terraform support for managing user groups. This allows administrators to provision, update, and manage workspace user groups using standard Infrastructure-as-Code practices.

### 2025.10.3

#### General

- Fixed an issue where some Visual Studio Code (VSCode) dependencies failed due to an underlying C standard library requirement. The minimum required glibc version is now 2.28 to ensure full stability and compatibility with remote VSCode functionality on supported Linux distributions.

#### Security

- Updated the Go language runtime to version 1.25.4, which addresses and patches known security vulnerabilities.

### 2025.10.2

#### General

- Resolved Slow Database Migration Performance. An optimization was applied to the database migration engine. This fix significantly reduces the time required to run database updates during product rollouts and version upgrades.

### 2025.10.1

#### General

- Removed Dependency on the C Standard Library (libc). The core workspace components have been updated to remove the explicit runtime dependency on libc.

## Getting Started

September 29, 2025

### Overview

Citrix Secure Developer Spaces (SDS), formerly known as the Strong Network™ platform, is a secure, cloud-based development environment (CDE) designed to enhance developer productivity while

maintaining enterprise-grade security. The platform's primary's purpose is to streamline the provisioning and management of coding environments, allowing organizations to 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 ideal for hybrid and remote teams. The platform can be deployed flexibly on public or private clouds and self-hosted servers, and it even supports fully air-gapped modes for high-security settings.

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

Ultimately, the platform helps organizations reduce costs associated with high-spec laptops, maintenance, and security software, while gaining real-time visibility and governance over the development lifecycle. Its environments seamlessly integrate with common DevOps tools and CI/CD workflows to improve IT efficiency, developer productivity, and overall governance.

## Tech Brief: Citrix Secure Developer Spaces

October 17, 2025

### What is a Cloud Development Environment?

Today, modern application developers are the driving force behind innovation. However, equipping them with the necessary tools and access while maintaining stringent security and compliance poses a significant challenge for IT departments. This results in inconsistent local setups, slow and error-prone onboarding, dependency conflicts, limited compute resources, and inadequate collaboration tools.

A **Cloud Development Environment (CDE)** is a purpose-built, centrally managed workspace that provides developers with all the necessary tools, libraries, dependencies, and access to source code and internal systems, all within a highly controlled and isolated security perimeter. Unlike traditional setups built on physical workstations or general-purpose virtual desktops, a CDE is specifically engineered to address the unique needs of software development while mitigating the inherent risks associated with intellectual property, sensitive data, and supply chain vulnerabilities.

For Citrix and End-User Computing (EUC) administrators, understanding CDEs is crucial. Traditional Citrix deployments excel at delivering standardized applications and desktops. Still, they often fall short when it comes to the dynamic, high-privilege, and usually volatile nature of a developer's workflow. CDEs, such as Citrix Developer Spaces (SDS), in contrast, offer:

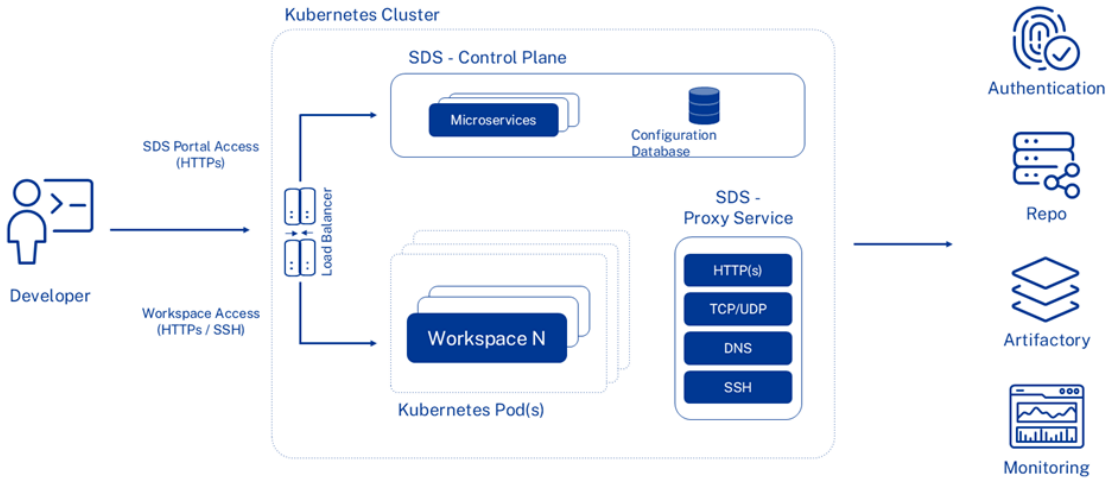
- Enhanced Security Posture
- Streamlined Compliance
- Improved Developer Experience & Productivity
- Cost Efficiency & Scalability
- Mitigation of Supply Chain Risk

In essence, a Cloud Development Environment moves beyond simply providing a remote desktop; it's a strategic shift towards a more secure, efficient, and compliant model for modern application development, perfectly complementing and enhancing your existing EUC strategy.

### What is Citrix Secure Developer Spaces?

Citrix Secure Developer Spaces (SDS), formerly known as Strong Network, offers a secure and productive CDE that can be deployed in private clouds (Azure, AWS, or GCP) or self-hosted on-premises on Kubernetes platforms. SDS also works in a full air-gapped mode for high-security environments. The SDS platform enhances developer productivity while ensuring enterprise-level security. It enables organizations to streamline the provisioning and management of modern application developer 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.

## High-level Architecture of SDS



### Cloud Development Environment (CDE)

At its core, SDS provides secure, fast, and highly flexible, ready-to-code development environments that are accessible online. These CDEs are architected as lightweight, containerized, and Linux-based instances, ensuring efficient and agile coding experiences. Designed for maximum deployment flexibility, they can be self-hosted on Kubernetes, allowing organizations to deploy them on-premises or within their private cloud infrastructure –Amazon AWS, Microsoft Azure, or Google Cloud (GCP).

### Access

SDS empowers developers with the freedom to work from anywhere, on virtually any device, by providing seamless access to powerful online development environments. Developers benefit from flexible access options, including a secure web browser interface or direct integration with their preferred local IDE or terminal via SSH.

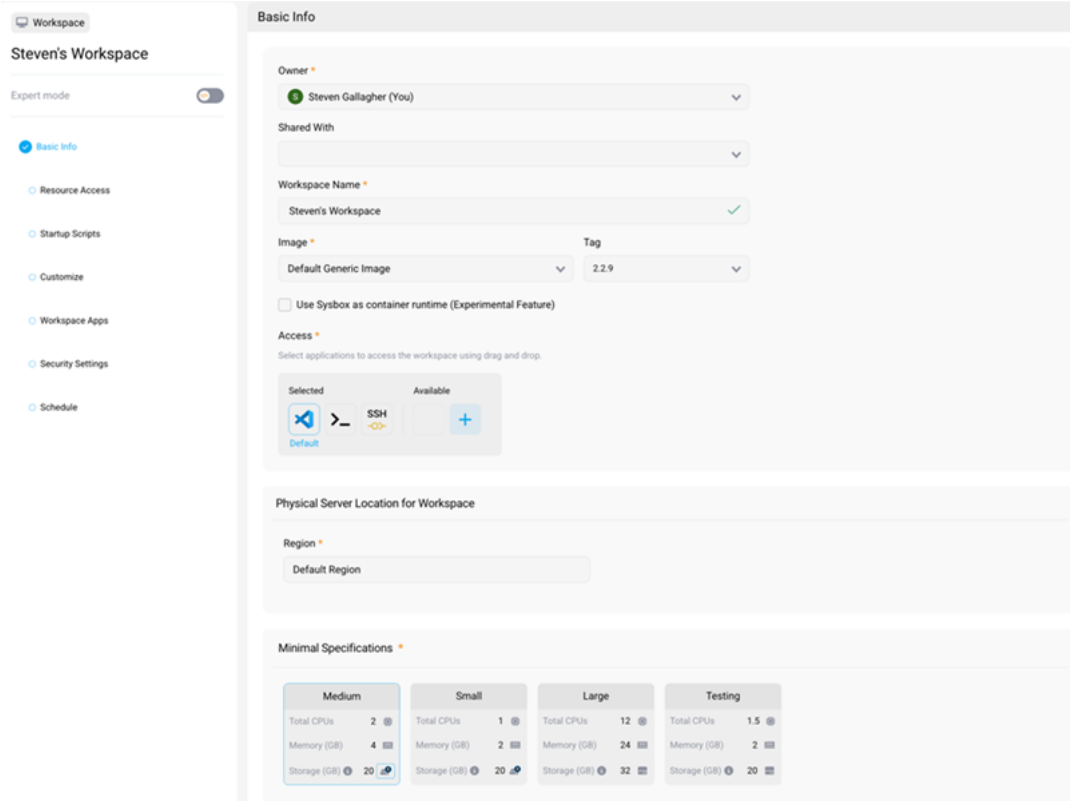
Developers may prefer a web browser interface over a local IDE for its zero-setup convenience, consistent cloud-based environment, and secure remote access. In contrast, a local IDE is often chosen for its deep customization. Browser-based tools enable instant collaboration, standardized configurations, and access from any device, eliminating the need for local installation or maintenance. This makes them ideal for onboarding, remote work, and managing secure or shared development environments.

To ensure the highest level of data security, a sophisticated front-end Data Loss Prevention (DLP) mechanism actively monitors and detects sensitive tokens, credentials, and proprietary code, preventing unauthorized exposure or exfiltration.

### Workspace

A Workspace in Citrix Secure Developer Spaces is a dedicated, preconfigured development environment provisioned in the cloud or on-premises infrastructure where a developer can securely build, test, and run code. Workspaces can be tailored with preconfigured templates, specific tools, and resource limits, ensuring consistent, compliant, and high-performance environments. This approach provides developers with flexibility while maintaining strong governance, data protection, and operational control.




### Endpoint Standardization

This solution revolutionizes endpoint management by enabling the widespread adoption of uniform, low-cost endpoints, encompassing Bring Your Own Device (BYOD), thin clients, and even low-specification Virtual Desktop Infrastructure (VDI). This is achieved by isolating and centralizing development environments remotely. This frees developers from setting up their environments, eliminating the need to install and maintain dependencies, software development tools, security patches, and plug-ins, which increasingly include AI code assistants.

## How Citrix Secure Developer Spaces empowers you to deliver a modern application developer experience

### Fully managed Cloud Development Environments

Providing performant, consistent, and secure Linux-based development environments for developers who primarily use Windows endpoints can be a monumental task. Managing WSL installations, Docker Desktop configurations, and ensuring compliance across numerous local machines is a significant drain on IT resources and introduces security vulnerabilities.

| Citrix Secure Developer Spaces | Benefits |
|---|---|
| Deliver pre-configured, fully managed Linux development environments directly from the cloud.<br>Developers access these powerful, consistent environments via a secure, browser-based interface –no complex local installations of WSL or Docker are needed on their Windows machines.<br>This simplifies management, eliminates configuration drift, and significantly reduces endpoint security risks, all while providing the best-in-class Linux experience for your developers, wherever they are.<br>It's the ideal solution for prioritizing control and ensuring compliance for your diverse developer workforce. | • **Reduced Endpoint Complexity**: No more wrestling with local WSL/Docker installations.<br>• **Enhanced Security Posture:** Centralized, managed Linux environments eliminate a wide array of endpoint vulnerabilities.<br>• **Consistent Dev Experience:** Every developer gets the exact same, pre-approved toolset, reducing "it works on my machine" issues.<br>• **Simplified Management:** Provision, manage, and secure all Linux development environments from a single pane console. |

**Developers**

Configuring external contractors, partners, or offshore teams on your own laptops, connecting via VPNs, managing complex network access, and ensuring data security are inherently risky, making it difficult to scale Linux development while maintaining control and compliance.

| Citrix Secure Developer Spaces | Benefits |
|---|---|
| Provide secure, online, always-accessible development environments that external developers can access from *any* device with a web browser.<br>These environments are isolated, secure, and pre-loaded with everything needed, ensuring developers are productive from day one without exposing your internal network or requiring complex endpoint management. | • **Rapid Onboarding:** Instant access for external teams, eliminating logistical delays.<br>• **Zero-Trust Security:** Data and code remain within the secured cloud perimeter, never residing on unmanaged external devices.<br>• **Simplified Access Management:** Granular control over what external developers can access and do, minimizing security risks. |

**Developer Onboarding & Offboarding**

The traditional process of setting up new developer workstations or managed devices, installing software, and granting access can take days or even weeks. This results in lost productivity for new hires and contractors, as well as significant administrative overhead, particularly in agile environments with high employee turnover.

- **Scalability & Flexibility:** Easily provision and tear down environments as project needs change.

| Citrix Secure Developer Spaces | Benefits |
|---|---|
| Automate the entire development environment provisioning process.<br>With Secure Developer Spaces you can provision ready-to-code, fully configured developer environments in minutes, not days.<br>Templates ensure consistency, and granular access controls mean new team members get exactly what they need, instantly.<br>Offboarding is equally swift, allowing immediate revocation of access to sensitive code and data. | • **Accelerated Productivity:** Developers start coding immediately upon joining.<br>• **Significant Time Savings:** Automate repetitive setup tasks, freeing up valuable IT resources.<br>• **Enhanced Security:** Instant offboarding minimizes the risk of lingering access for departing personnel.<br>• **Standardization:** Ensure every new environment meets your exact specifications and security policies. |

### Bring Your Own Device

Developers increasingly want to use their preferred personal devices (BYOD), but this introduces significant security and compliance challenges for IT, as it allows personal devices while ensuring intellectual property (IP) is protected and corporate data remains secure.

| Citrix Secure Developer Spaces | Benefits |
|---|---|
| Transform any modern web browser into a secure, high-performance portal to a fully functional cloud development environment. Developers can use their personal laptops, tablets, or even thin clients to access their complete development stack with no code or sensitive data ever touching their local device. This ensures a flexible work experience without compromising security. | • **Enhanced Security:** No IP or sensitive data ever resides on personal devices, preventing data leakage.<br>• **Cost Savings:** Reduce or eliminate the need to procure and manage corporate-issued developer hardware.<br>• **Developer Flexibility & Satisfaction:** Empower developers to work how and where they choose, boosting morale and productivity.<br>• **Simplified Device Management:** Focus on managing the secure, authenticated environment, not managing a diverse array of endpoints. |

### How Citrix Secure Developer Spaces integrates with DevOps Tools

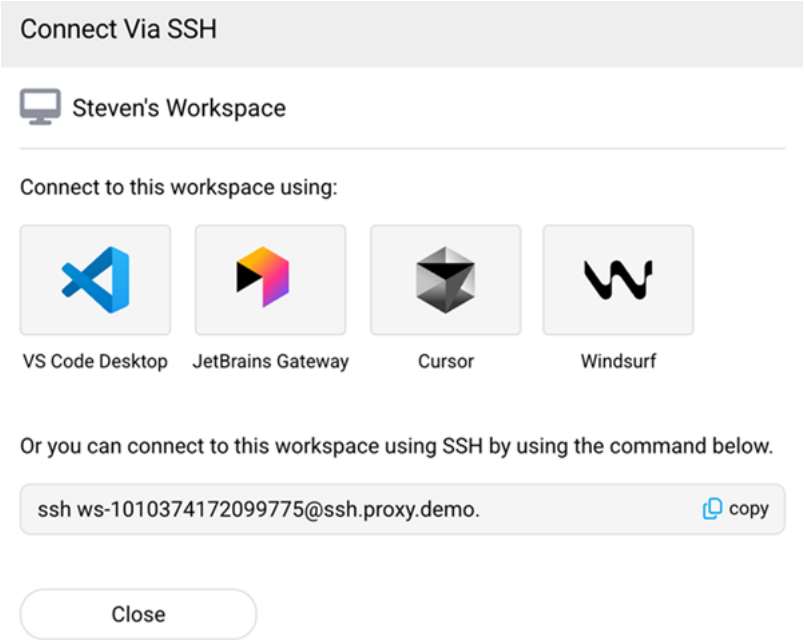
Citrix Secure Developer Spaces gives developers the freedom to work with the tools they rely on today, including popular and AI-assisted IDEs, while remaining flexible enough to support the evolving DevOps tools of tomorrow. With broad support for IDEs, management APIs, authentication standards, and DevOps integrations, SDS fits seamlessly into existing workflows without locking teams into a fixed ecosystem. Workspace Apps in SDS enable developers to share securely and test apps running within their cloud workspaces. With controlled access and port mapping, teams can collaborate, demo, and debug services without exposing the complete environment.

### Supported IDEs

The SDS platform supports a range of Integrated Development Environments (IDEs), including Microsoft Visual Studio Code Desktop, JetBrains Gateway, Cursor, and Windsurf. Notably, both Cursor and Windsurf offer AI-assisted development features to enhance productivity and code quality. By default, SDS provides Visual Studio Code for the Web, with the flexibility to manually integrate additional web-based IDEs as needed. Developers can also leverage GitHub Copilot within these cloud-based IDEs, enabling AI-powered code suggestions, completions, and contextual guidance directly in SDS workspaces, combining productivity enhancements with the platform's secure, ephemeral environment.

The platform also includes a built-in CLI terminal that supports traditional editors such as Vi, Vim, and Emacs.




**Code Repositories**

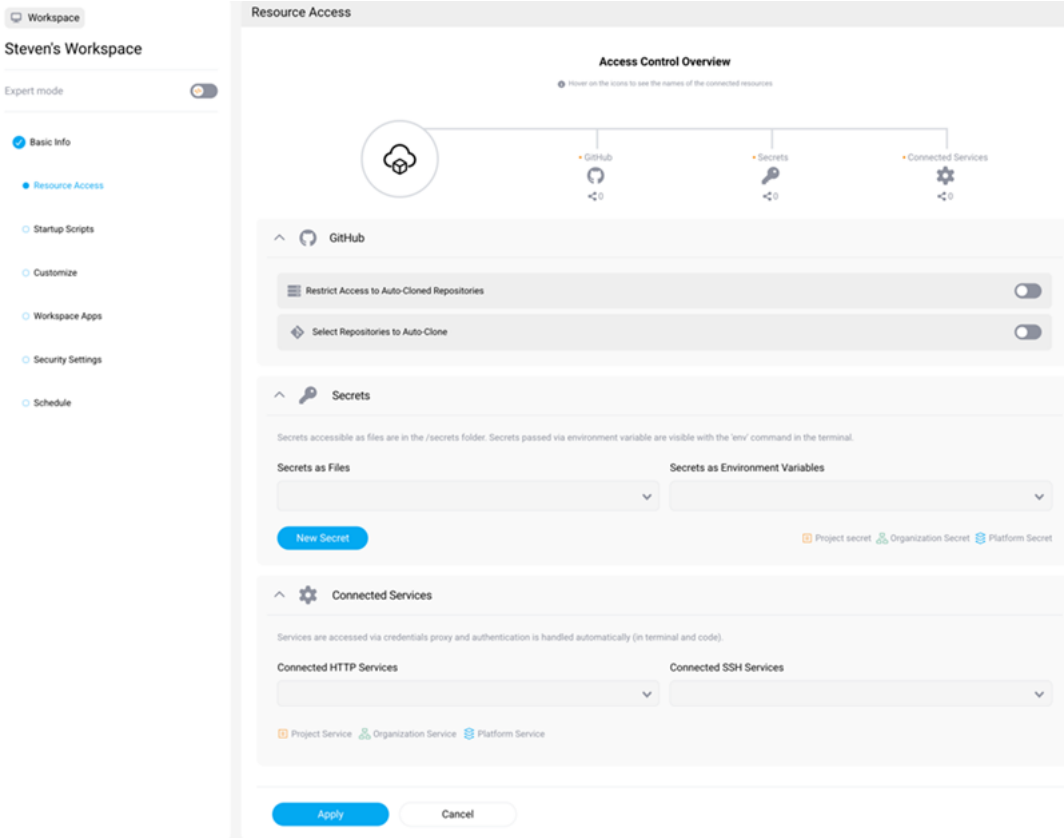
Code repositories are essential for storing, tracking, and collaborating on source code in software development projects. The SDS platform offers a unique enhancement to the developer experience by providing secure, automated single sign-on to all platform resources. This eliminates the need for developers to have explicit knowledge of resource credentials when accessing GIT applications, repositories, and HTTP/SSH services from the workspace.

Code repositories are fundamental to storing, tracking, and collaborating on source code in modern software development. The SDS platform enhances the developer experience by providing secure, automated single sign-on (SSO) to all platform-integrated resources. This streamlined access removes the need for developers to manage or be aware of individual credentials when connecting to GIT applications, repositories, or HTTP/SSH services directly from their workspace.

**The SDS platform currently supports direct integration with the following Git-based repository providers**

- GitHub
- GitLab
- Bitbucket
- Azure Repos

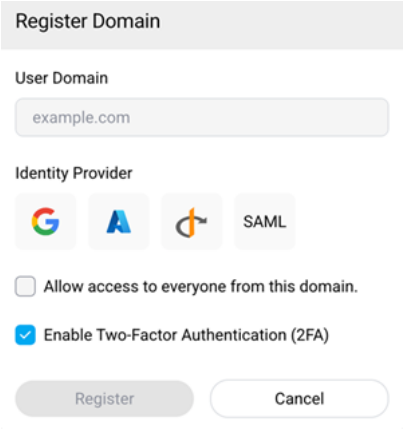



### Authentication

SDS offers a range of authentication mechanisms designed to ensure secure access to its Cloud Development Environments (CDEs). Key mechanisms include:

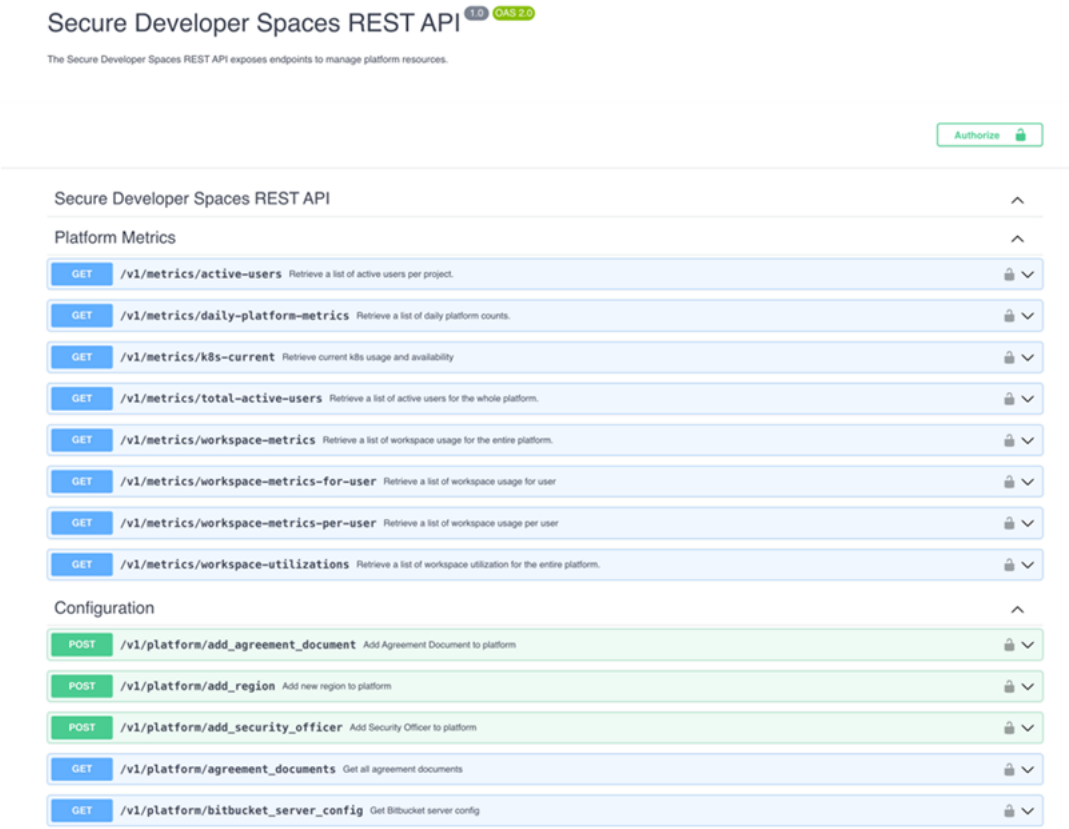
- Single Sign-On (SSO): Integration with identity providers like Azure AD and Okta to streamline and secure the authentication process.
- Multi-Factor Authentication (MFA): Adds a layer of security by requiring multiple forms of verification.
- OAuth, SAML, and OpenID Connect: Standards for token-based authentication to enhance security across applications and services.




## REST API

The SDS platform provides comprehensive control and integration through its REST API, which features over 150 endpoints (detailed on the platform's REST API page). This enables the complete management of enterprise applications and seamless integration with security and analytics tools, such as Splunk and Sumo Logic.



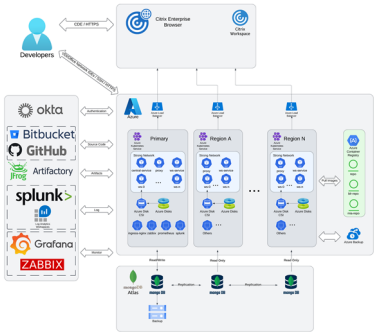

 

### Summary

By leveraging Citrix Secure Developer Spaces, Citrix EUC Administrators can transform their approach to modern application development, moving from managing individual, high-maintenance workstations to orchestrating a highly secure, scalable, and cost-efficient cloud-native developer platform.

## Architecture Diagram

October 2, 2025




The architectural diagram of a CDE has the following components:

- One Kubernetes cluster with auto-scaling node and storage Container Storage Interface(CSI) driver capacity to host the SDS platform and workspace.
- A container registry
- MongoDB database
- Code repositories, for example, Bitbucket or GitHub
- Optional: Additional Kubernetes clusters set up in different regions to ensure global access with optimized network latency.
- Optional: An identity provider (SAML), such as Okta
- Optional: Observability
- Optional: Private access using Citrix Workspace™, Enterprise Browser, or SPA

The key components of SDS - the Cloud Development Environment (CDE) Platform include Kubernetes clusters, a container registry, and a MongoDB database. You can leverage resources from any cloud service provider, use your hardware in a data center, or even use a hybrid.

The core components of the Azure-based sample deployment depicted in the architecture diagram above are:

- Azure Kubernetes Services for platform and regions
- Service node pool with two Standard_D8as_v5 VMs

- Workspace node pool with Standard_D16as_v5 VMs with auto-scaling
- Azure Container Registry
- Premium Tier
- Geo Replication peers
- MongoDB Atlas cluster
- M10 (2 GB RAM, 8 GB Storage) with auto-scaling
- Read-only nodes for regions

If you are not using Azure, you can choose from the following alternatives:

- Kubernetes Cluster:
  - Amazon Elastic Kubernetes Service
  - Google Kubernetes Engine
- Container Registry
  - Amazon Elastic Container Registry
  - Google Container Registry

> **Note:**
>
> Further deployment guidance and best practices can be found on Citrix Tech Zone

## Technical requirements for deploying Citrix Secure Developer Spaces™

December 4, 2025

This guide defines the essential platform and operating system prerequisites for running Citrix Secure Developer Spaces™ (SDS).

- **Deployment Options:** Choose between a cloud account (AWS, Azure, GCP) or an on-premises Kubernetes deployment (Red Hat OpenShift, VMware Tanzu). Make sure your account has the necessary cloud infrastructure permissions.
- **Kubernetes Cluster**: Use a dedicated Kubernetes cluster running version 1.20 or higher. Do not share the cluster with other applications.
- **Kubernetes Node OS (AWS-specific)**: Use Amazon Linux as the Kubernetes node operating system when deploying on AWS.
- **Kubernetes Node Architecture**: Ensure all nodes run on the amd64 architecture, as arm64 is not supported.

### Networking Requirements

These specifications ensure that the platform can reliably route, secure, and expose services across environments.

- **Ingress gateway:** Use Citrix NetScaler® as the recommended ingress controller. Nginx and Istio gateways are also supported.
- **Network Policy API:** Use the `networking.k8s.io/v1` API. If unavailable, install Calico or Cilium to enable network policy support.
- **DNS & SSL:** Configure two DNS domains and apply valid SSL certificates. For proof-of-concept (PoC) deployments, certificates are optional but strongly recommended. The second domain must be a wildcard subdomain of the first domain. For instance:
  - `example.com`
  - `*.proxy.example.com`

### Storage Requirements

These requirements define the persistent data capabilities needed for workspace and service storage.

- **Persistent Volume Claims API:** Provide persistent storage using the Kubernetes Persistent Volume Claim API.

### Deployment Tooling

These specify the tools necessary to install and configure Secure Developer Spaces components in Kubernetes.

- **Helm CLI tool:** Install the Helm CLI to deploy Secure Developer Spaces using the provided Helm chart.

### Enterprise-Grade Service Recommendations

We strongly recommend these configurations for production environments to ensure scalability, reliability, and enterprise-grade security, although they are optional for PoC deployments.

- **Database:** Use a MongoDB Atlas subscription for database management in production deployments. For PoC environments, the system deploys an internal MongoDB container by default.
- **Identity & Access Management:** In production, configure an identity provider (SAML or OIDC), such as Okta, for managing user identity and access. The system provides basic email/password authentication by default.

### Connectivity for Installation & Licensing

This section outlines the external URLs your environment must access to download the required installation components and validate your license.
During installation, the system connects to the Citrix Secure Developer Spaces license server to validate the license and generate a temporary token for accessing the container image artifactory. If your environment is air-gapped, request an offline license.

Here is a specific list of the required packages and images, along with their locations:

- **License Server**
  - **URL**: `api.enterprise.strong.network`
  - **Purpose**: Used for online license verification.
- **Installer Image**
  - **Source**: Docker Hub
  - **Image**: `strongnetwork/strong_installer:2025.10.4`
- **Helm Chart Package**
  - **Source**: Google Cloud Storage
  - **URL**: `https://storage.cloud.google.com/strong-network-helm-chart/ninjahchart-2025.10.4.tgz`
- **Container Images (Artifactory)**
  - **Source**: Google Artifact Registry (GCP)
  - **Primary URL**: `europe-docker.pkg.dev/strong-network-release/images`
  - **Mirrors**: For improved performance, regional mirrors are available at `us-docker.pkg.dev` and `asia-docker.pkg.dev`.
  - **Required Service Images**:
    - `browser_in_browser:2025.10.4`
    - `cloud_editor_sidecar_proxy:2025.10.4`
    - `frontend:2025.10.4`
    - `sn_enterprise_bundle:2025.10.4`
  - **Required Workspace Image**:
    - `ws-images/cloud_editor_generic:2.3.1`
  - **Optional Workspace Images** (not required for the default installation):
    - `ws-images/android_studio:2.2.5`
    - `ws-images/goland_go:2.2.5`
    - `ws-images/intellij_java:2.2.5`
    - `ws-images/intellij_ultimate:2.2.5`

- `ws-images/phpstorm_php:2.2.5`
- `ws-images/pycharm_python:2.2.5`
- `ws-images/webstorm_image:2.2.5`

## Deploy Secure Developer Spaces™ in multiple regions

October 10, 2025

This guide describes how to deploy Citrix Secure Developer Spaces™ 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:

- 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. |

### 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.

## 1-Click VM for deploying Citrix Secure Developer Spaces™

December 4, 2025

Use this guide to deploy a virtual machine (VM) running the Citrix Secure Developer Spaces™ (SDS) platform using the automated installer. The installer provisions infrastructure with Terraform, installs a lightweight Kubernetes cluster (K3S), and deploys the platform. It also configures DNS and manages TLS certificates.

> **Note:**
>
> The 1-click VM is purpose-built for proof-of-concept (POC) and demo environments. It has been optimized for implementation simplicity and provides the same functional capabilities as a standard deployment. However, it is not designed for scalability and cannot be converted into a production-grade installation. There is no upgrade path from a 1-click VM to a full production deployment.

### Prerequisites

- Docker installed on your local machine.
- Cloud provider credentials (AWS, Azure, or GCP).
- Admin email and password for platform access.

### Run the installer container

Pull and run the installer from Docker Hub. This command mounts your current directory into the container to share configuration files.
The installer uses the current working directory to download and install SDS. It's recommended that you create and use a dedicated folder for this 1-click deployment before running the installer.

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared strongnetwork/strong_installer:2025.10.4
```

### Deploy the platform

Once inside the container shell, start the deployment process:

```
./strong-cli deploy-demo
```

Follow the on-screen prompts to configure your deployment.

- **Admin Credentials:** Provide an admin email and create a secure password.

```
root@117bbcfbf7f1:/strong-network# ./strong-cli deploy-demo
Set the email of the platform admin: admin@strong-network.com
Set password for admin (leave empty to autogenerate):
```

- **VM Size:** Select a VM size. The size determines the maximum number of concurrently active workspaces.

```
Choose a VM size (default: Medium):
        [1] Small  (4 CPU,  16GB RAM) - up to 2 concurrent workspaces
        [2] Medium (8 CPU,  32GB RAM) - up to 5 concurrent workspaces
        [3] Large  (16 CPU, 64GB RAM) - up to 12 concurrent workspaces
Please enter your numeric choice:
```

> **Information:**
>
> You can resize the VM later if needed.

- **Cloud Provider:** Choose where to deploy: AWS, Azure, or GCP.

```
Pick cloud provider:
        [1] Azure - requires Contributor role
        [2] AWS   - requires IAM permissions
        [3] GCP   - requires Editor role
Please enter your numeric choice: 3
```

- **Cloud Credentials:** Provide your cloud identity. The specific steps will vary by provider.

```
Go to the following link in your browser, and complete the sign-in prompts:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.g
.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww
%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fw
counts.reauth&state=5ou5CTtnQYERYWIxaOJfaHYKp5uCkG&prompt=consent&token_usage=remote&access_t
llenge_method=S256

Once finished, enter the verification code provided in your browser:
```

For example, GCP will list available projects for selection.

```
You are now logged in as [                                  ].
Your current project is [None].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID
select GCP project to use:
        [5] staging-306409
Please enter your numeric choice: 5
```

- **Region:** Select the deployment region. Choose a predefined region (US, EU, ASIA) or select **specific datacenter** to enter a custom datacenter location.

```
Pick where to deploy:
        [1] US   (us-south1)
        [2] EU   (europe-west3)
        [3] ASIA (asia-south2)
        [4] Specific datacenter
Please enter your numeric choice: 2
```

Terraform will now provision and configure your resources.

### What to expect

After deployment, you'll have:

- A VM running the SDS Platform
- A secure URL to access the SDS platform

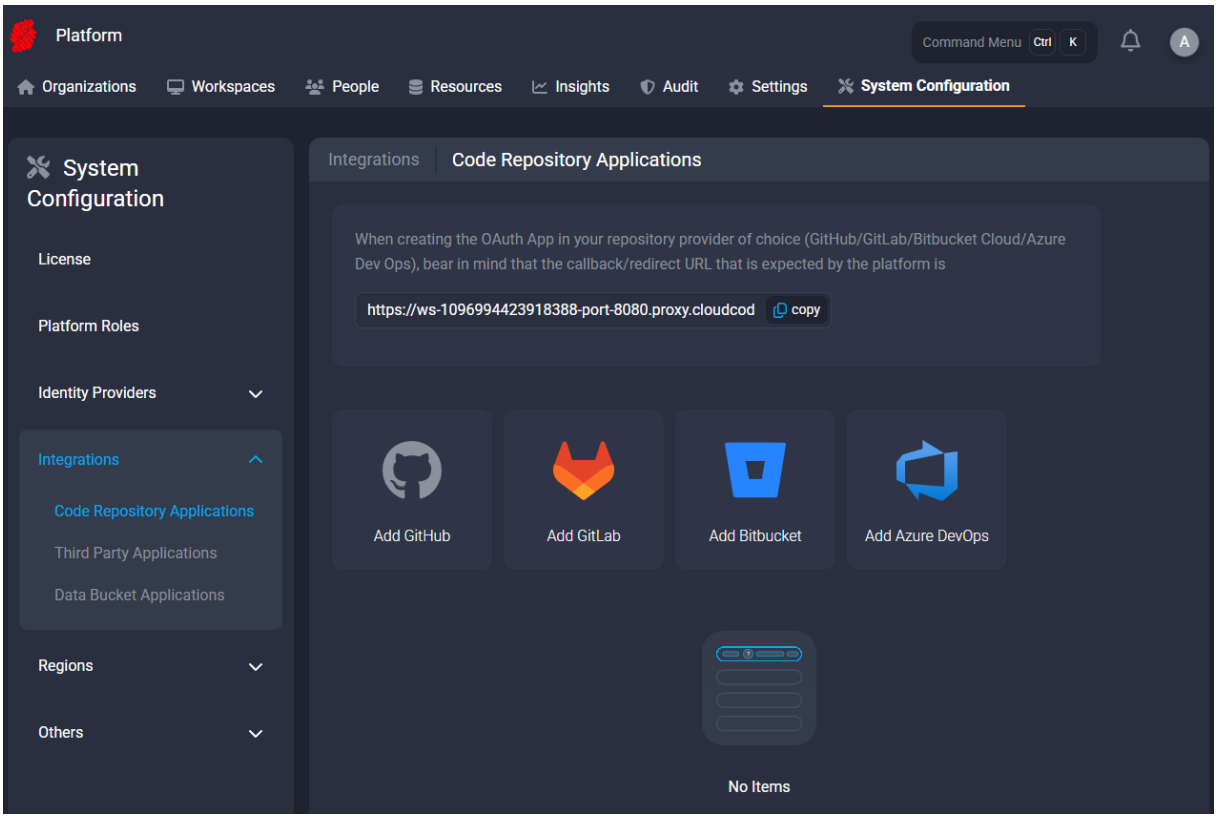
> **Warning**
>
> - **Initialization Time:** The login page may appear before all services are initialized. If you see an **invalid username or password** error, wait up to 5 minutes for the SDS platform to fully initialize before trying again.
> - **License and certificates:** The initial SDS platform license is valid for 6 months.
> - **TLS certificates:** TLS certificates are valid for 3 months.

## Setup Code Repository Applications

October 2, 2025

This folder contains a list of guides on how to set up different code repositories:

- GitHub
- GitLab
- Bitbucket
- Azure DevOps



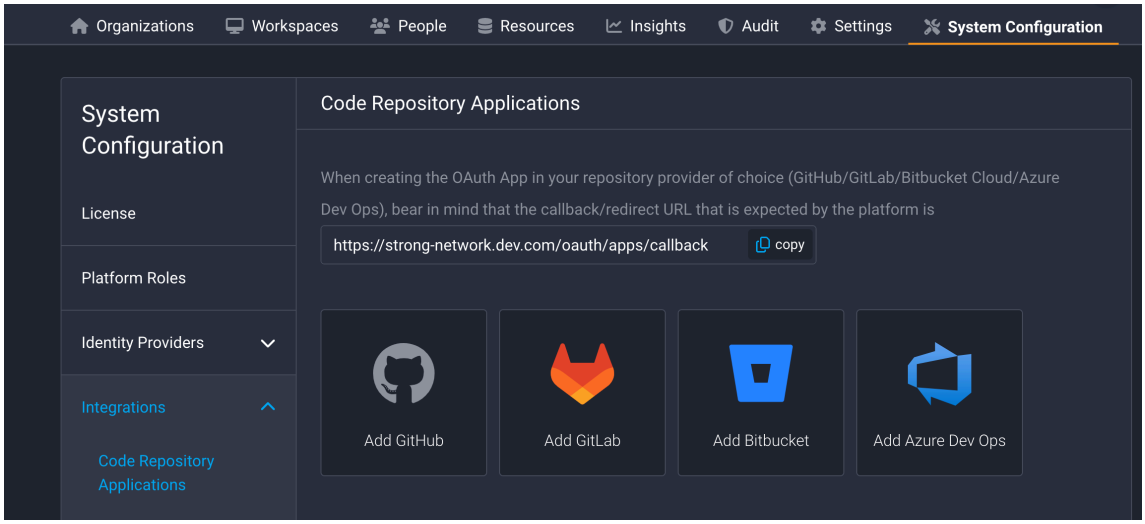
## Azure Dev Ops integration as Code Repository Provider

October 2, 2025

Follow these steps to create an OAuth App in Azure DevOps to connect it to the platform.

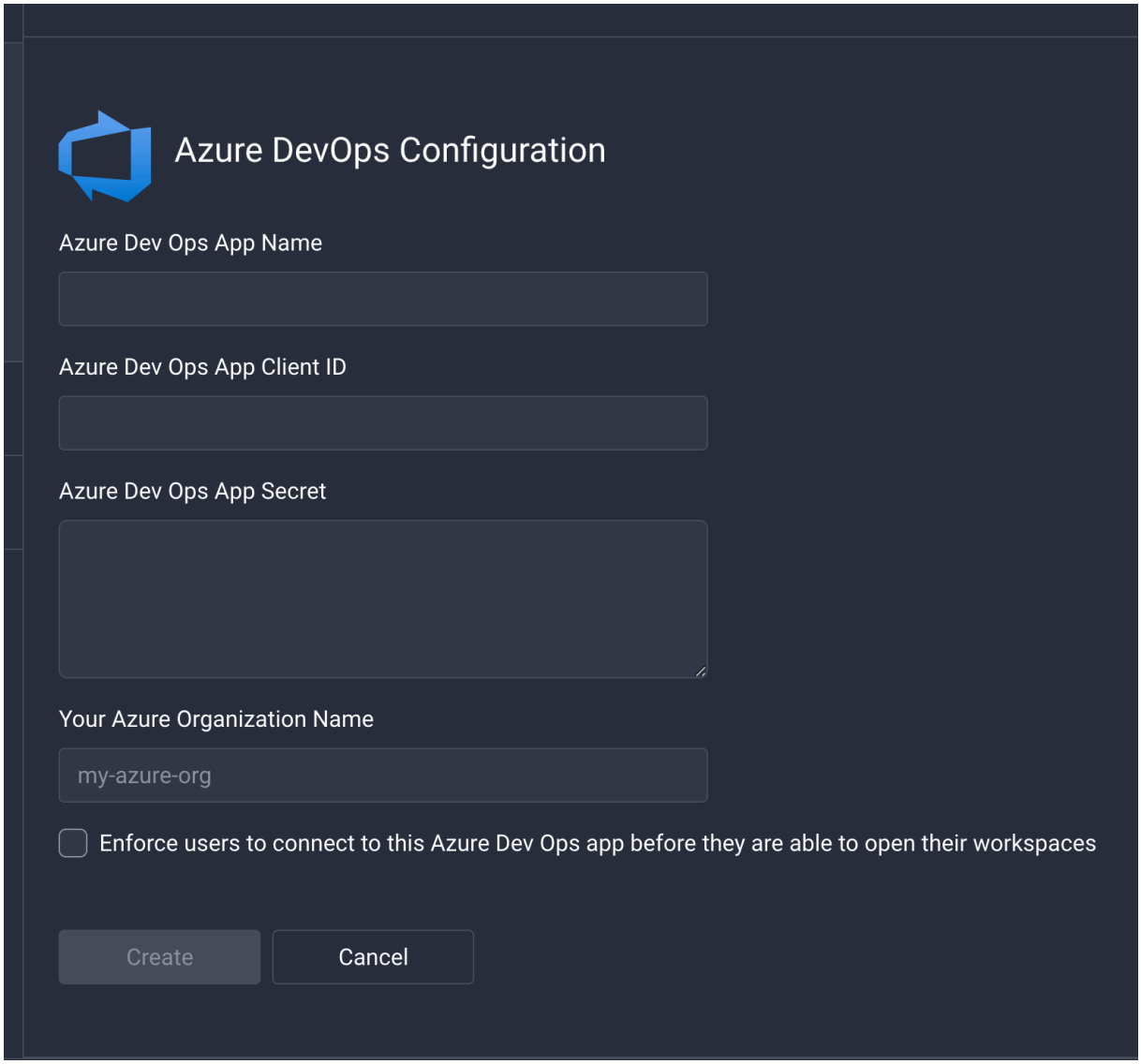
- Using an Azure DevOps account, go to the following link:
  Register an application
- Click on the "Add consumer"button and set the following fields:
  - **Company Name:** Your company"™s name.
  - **Application Name:** The name you want to give to the application. It will be public.
  - **Application Website:** Set to `https://example.com/oauth/apps/callback` (replace `example.com` with the proper domain name).
  - **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 admin panel of the Strong Network 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 Admin configuration of the Strong Network™ 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:




## Bitbucket Cloud Integration as Code Repository Provider

October 2, 2025

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.
  - **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 Strong Network 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. This can also be checked in the Strong Network™ 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

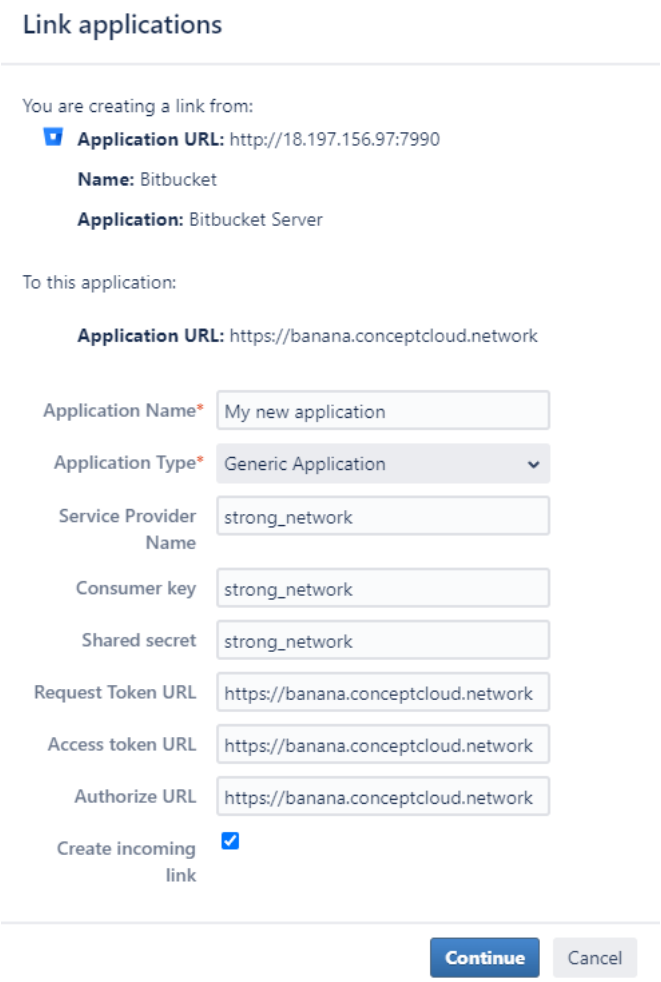
In this section, we will see how to connect the Strong Network platform to a self-hosted Bitbucket instance:

- **Configure Strong Network Platform:**
- Go to the Strong Network 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 Strong Network 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.

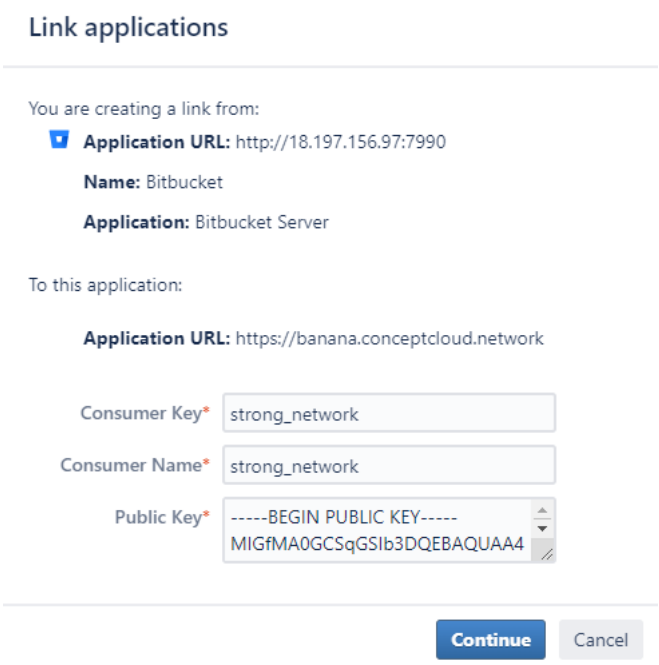
- **Configure Bitbucket Instance (Version 7.20 or Later):**
- Go to Administration > Applications > Application Links and click on "Create link":
  - **Application Type:** External application
  - **Direction:** Incoming
  - Click on continue
  - Set a unique name
  - **Redirect URL:** Set to `https://example.com/oauth/apps/callback`, 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.




- 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.



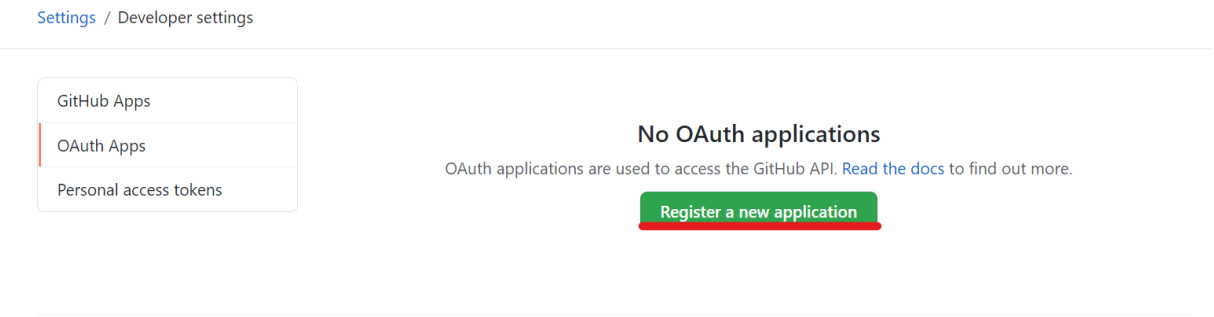

- **Complete the Configuration:**
- Click on Continue. The configuration is complete.

## GitHub Integration as Code Repository Provider

October 2, 2025

Follow these steps to create an OAuth App in GitHub to connect it to the platform:

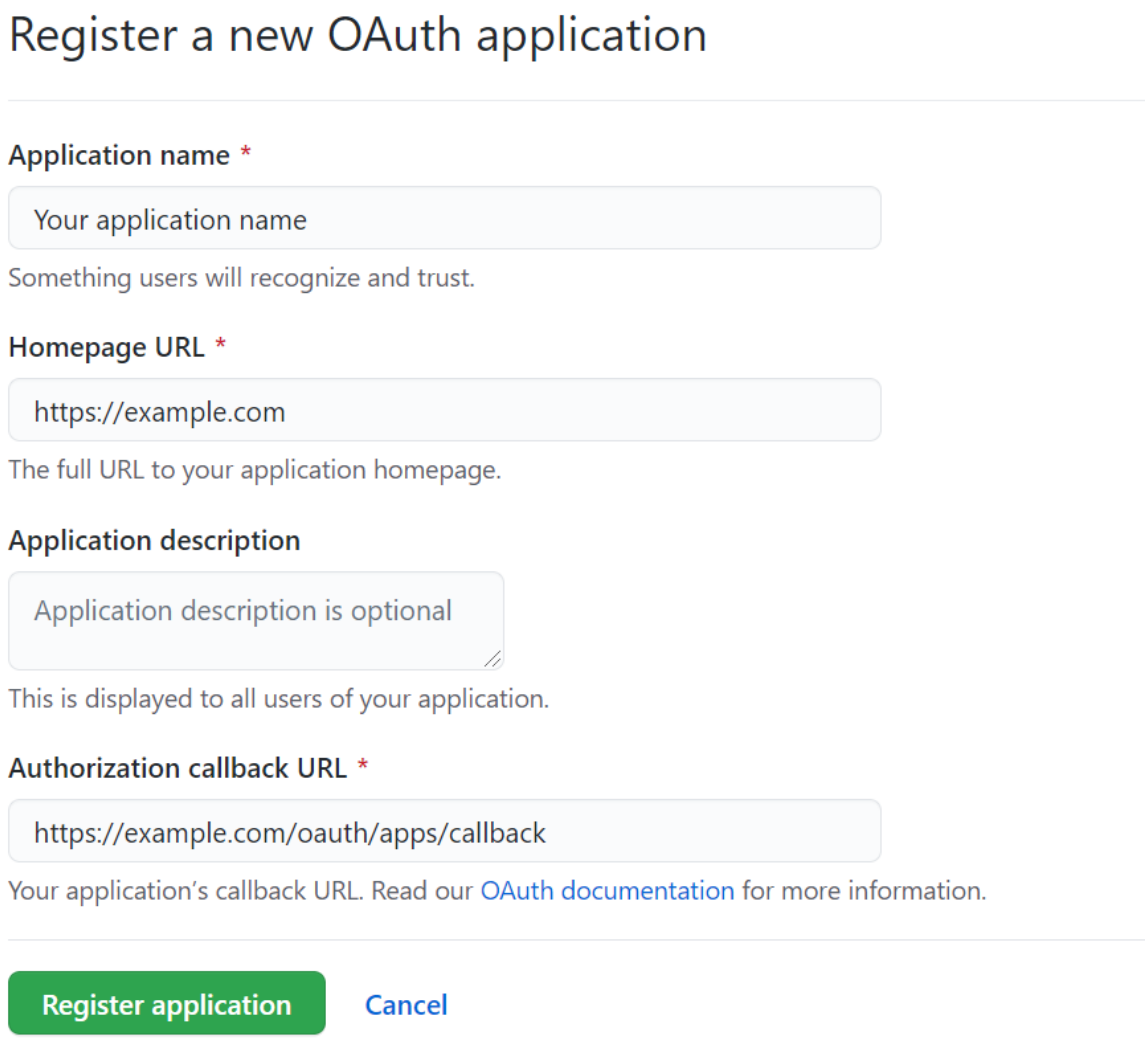
- **Navigate to Developer Settings:**
- Using a GitHub account, go to its settings and then to “Developer settings”.
- Inside this menu, click on “OAuth Apps”.
- You can follow this https://github.com/settings/developers to reach this menu directly.




- **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).




- **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.

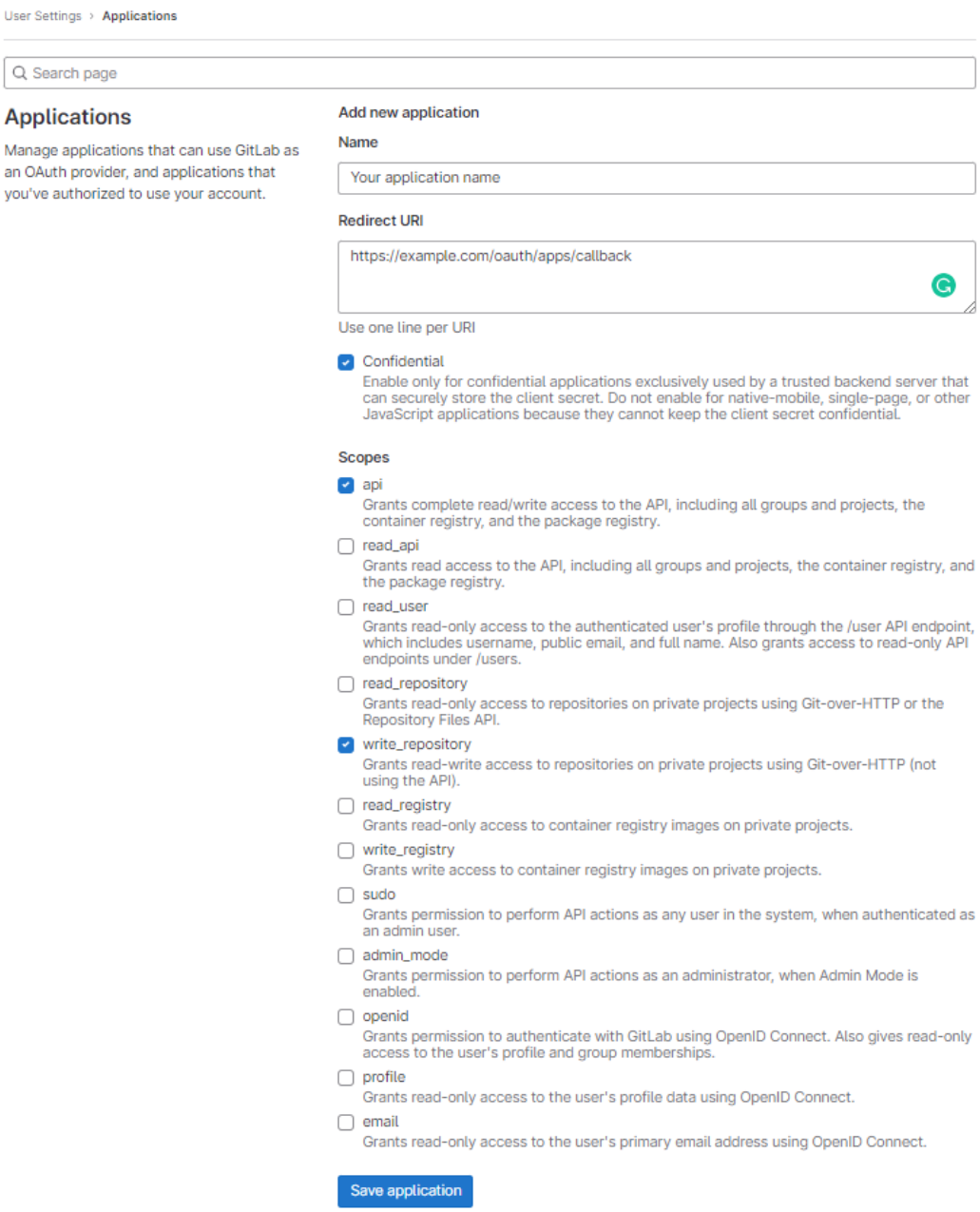
## GitLab Integration as Code Repository Provider

October 2, 2025

Follow these steps to create an OAuth App in GitLab to connect it to the platform:

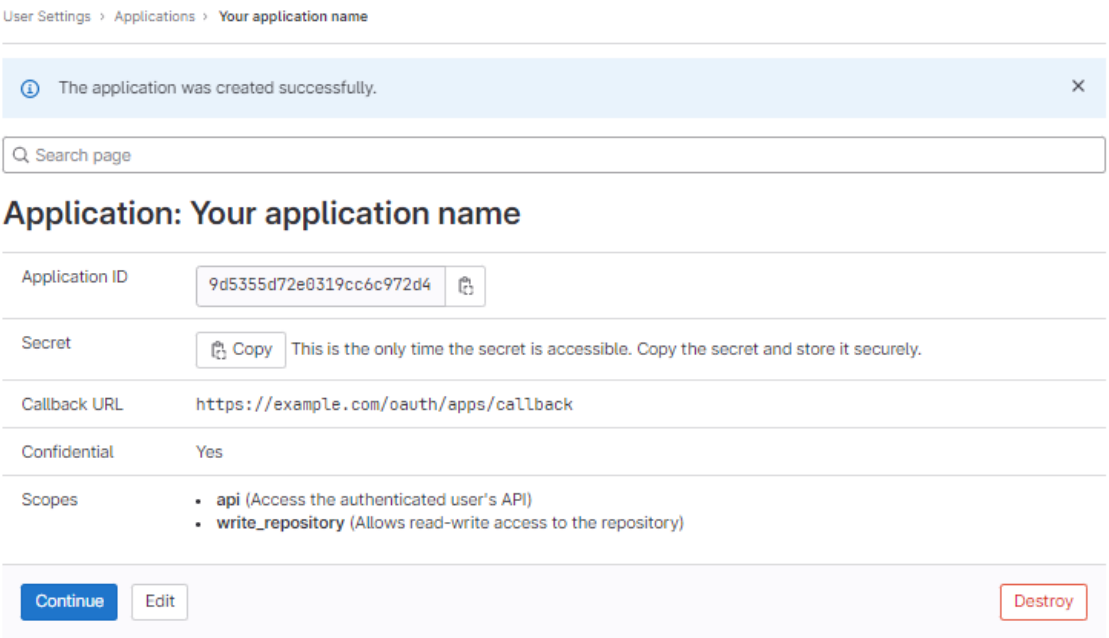
- **Navigate to Applications:**
- Using a GitLab account, go to user settings and then to "Applications".
- You can follow this https://gitlab.com/-/user_settings/applications to reach this menu directly.
- **Create a New OAuth App:**
- Click on "New application"and set the following fields:
  - **Name:** 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.




- **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.




# Configure Platform Login

October 2, 2025

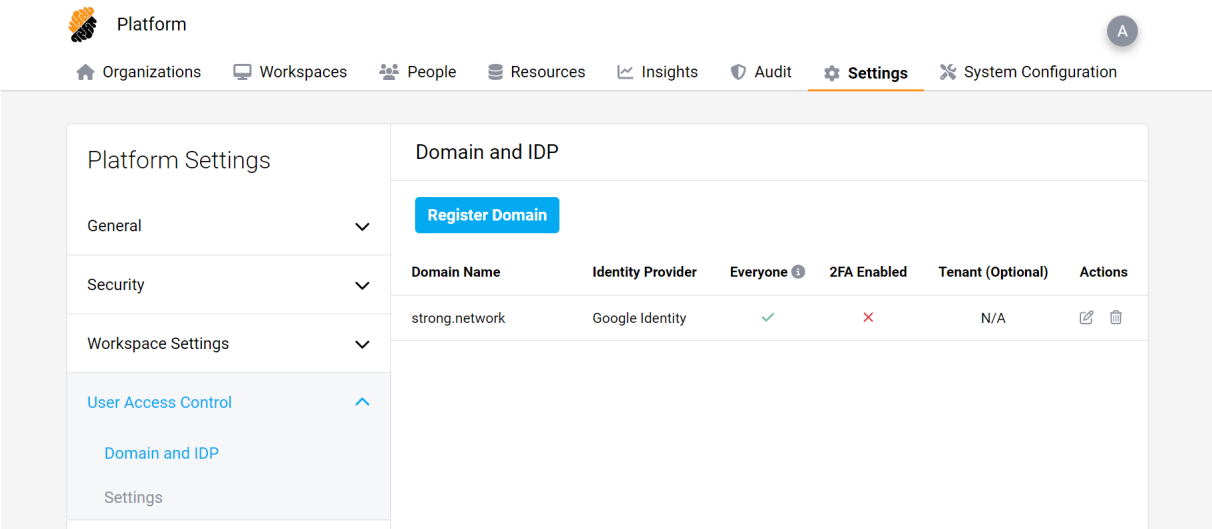
## Configure Login for Users

There are five ways users can log in to the platform:

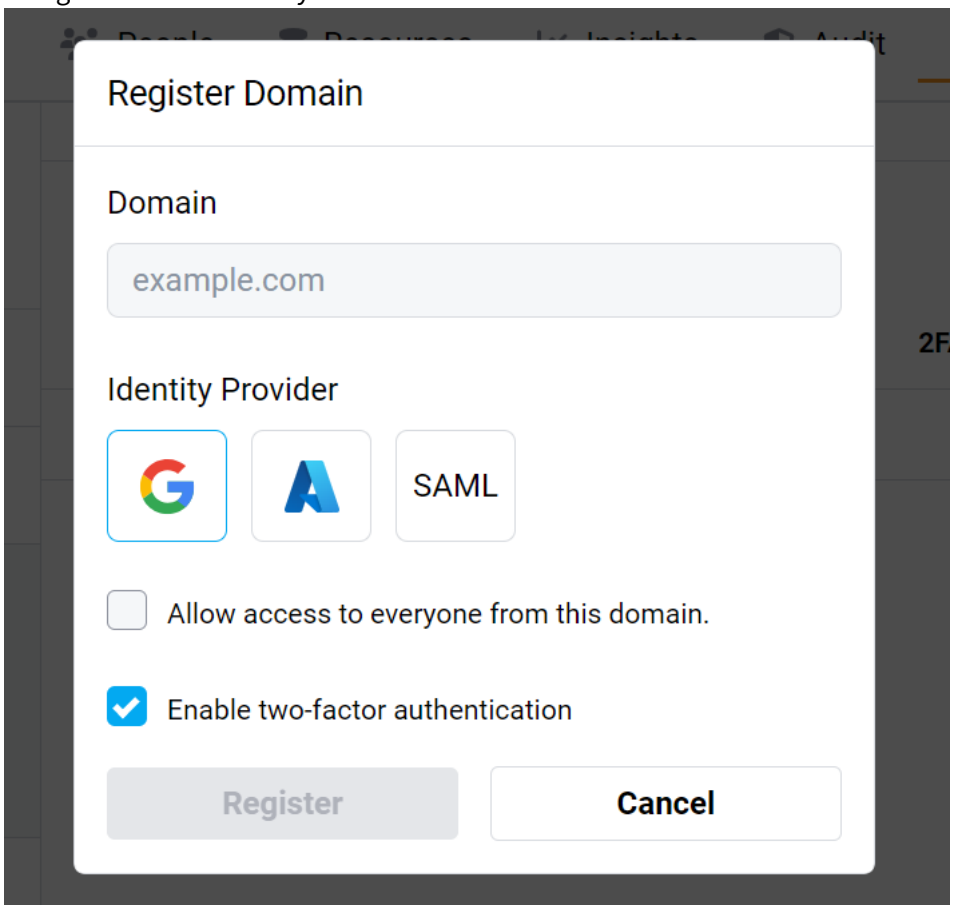
- Google OAuth provider
- Microsoft OAuth provider (Azure)
- SAML
- OpenID Connect
- Username and password

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".




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.

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.

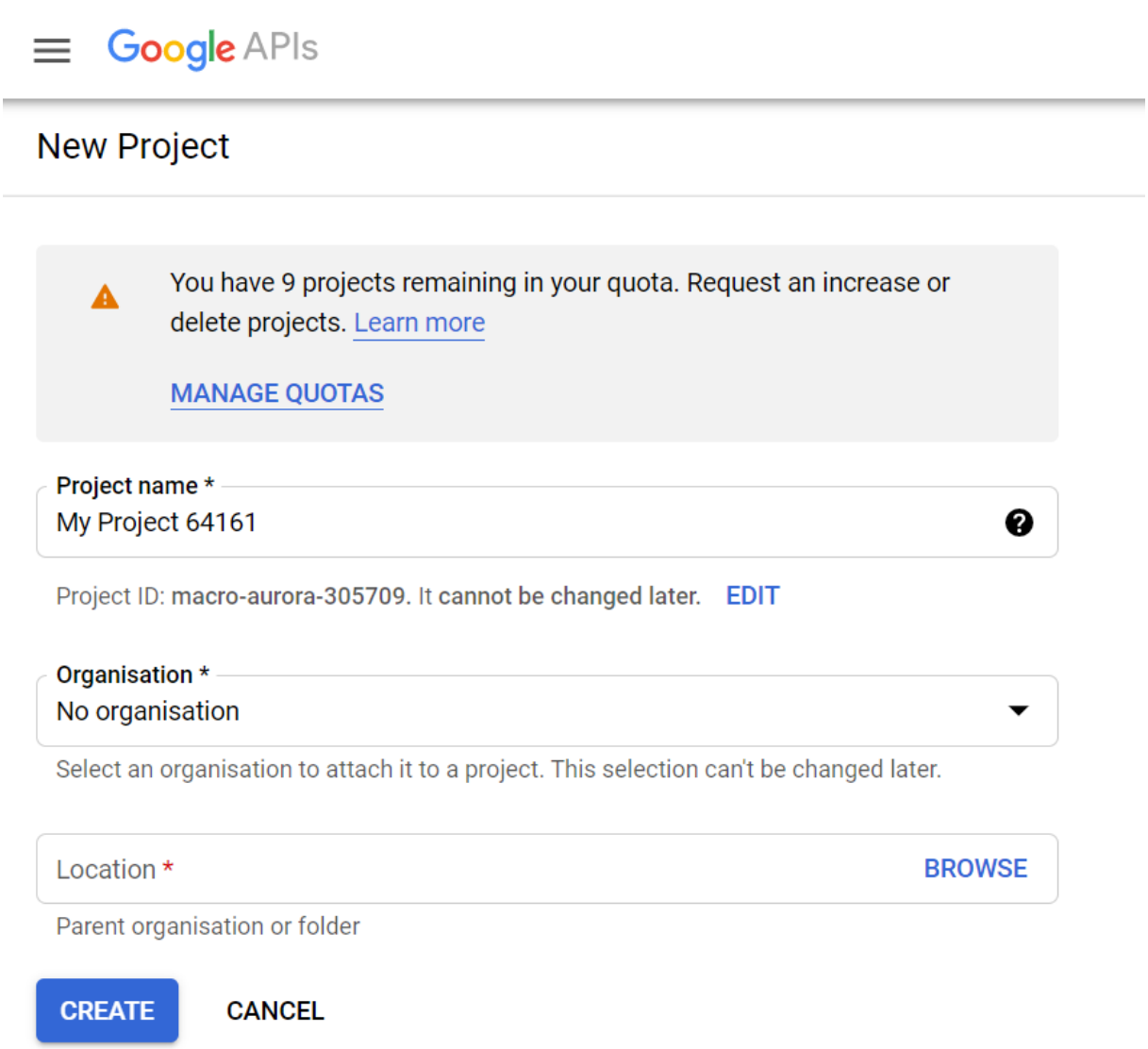
## Google Configuration as Identity Provider (OIDC)

October 2, 2025

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:

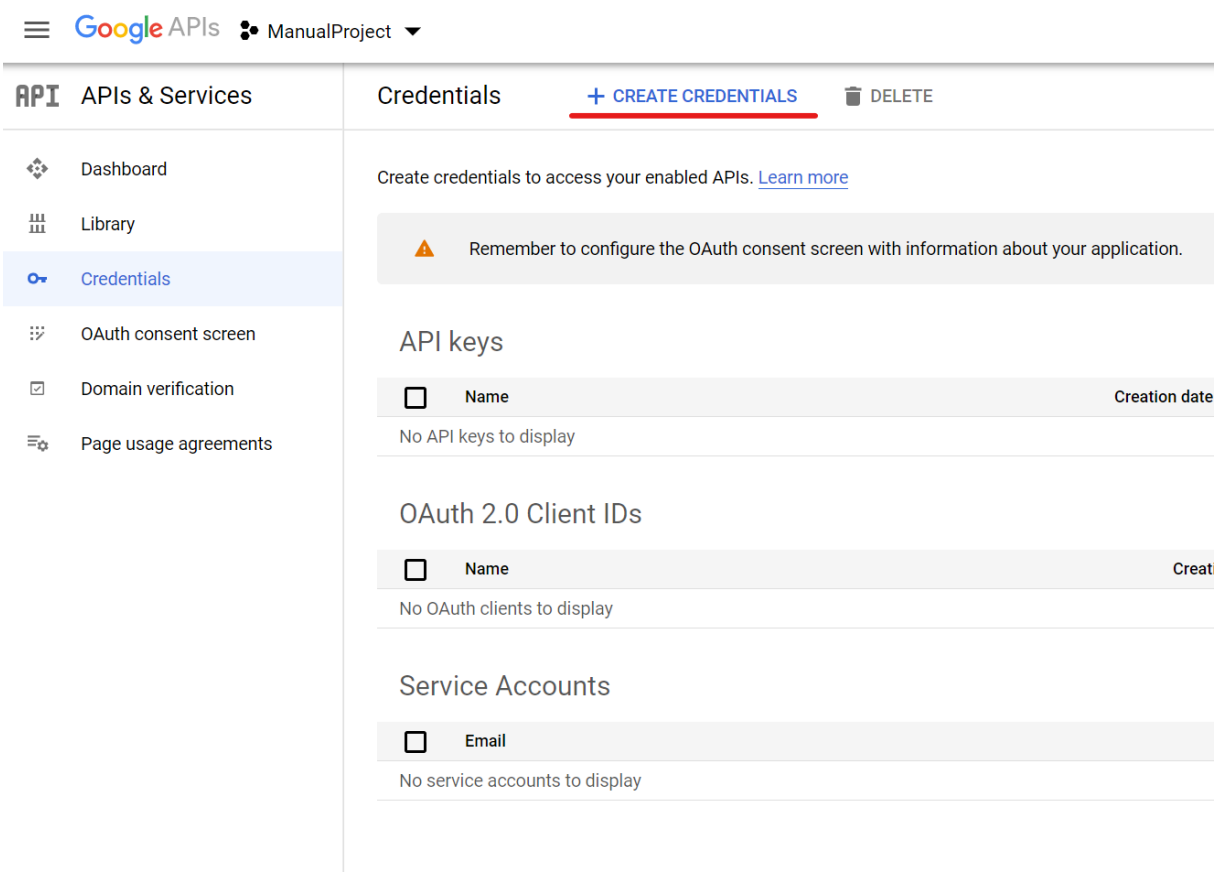
- Go to the Google API Console and create a new project (or use an existing one). The project name, organization, and location are left at your discretion.




- Inside the project, click on “+ Create Credentials”and select “OAuth client ID”from the submenu.




- 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.

## App domain

To protect you and your users, Google only allows apps using OAuth to use Authorised Domains. The following information will be shown to your users on the consent screen.

Application home page

Provide users a link to your home page

Application privacy policy link

Provide users a link to your public privacy policy

Application Terms of Service link

Provide users a link to your public Terms of Service

### Authorised domains

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the Google Search Console to check if your domains are authorised. Learn more about the authorised domain limit.

+ ADD DOMAIN

## Developer contact information

Email addresses *

These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE CANCEL

- 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”.

OAuth consent screen

eeee EDIT APP

Publishing status

Testing

PUBLISH APP

User type

External

MAKE INTERNAL

Test users

While publishing status is set to 'Testing,' only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. Learn more

+ ADD USERS

0 users (0 test, 0 other) / 100 user cap

- 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:

- `https://example.com/oauth/callback`
- Where “example.com”should be set to the proper domain name.

## Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information.

**Application type ***
Web application

Learn more about OAuth client types

**Name ***
Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorised domains.

### Authorised JavaScript origins

For use with requests from a browser

**URIs**

https://example.com

+ ADD URI

### Authorised redirect URIs

For use with requests from a web server

**URIs**

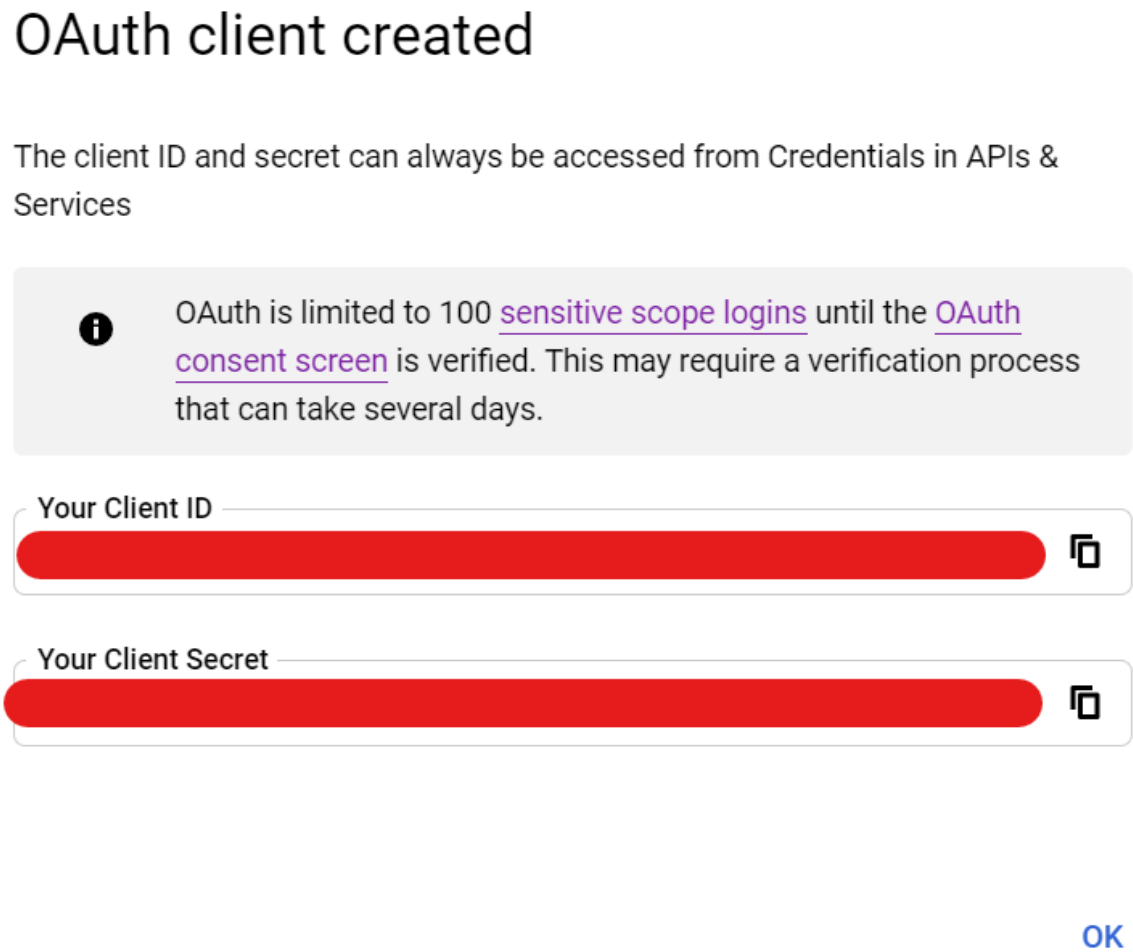
https://example.com/oauth/callback

+ ADD URI

CREATE CANCEL

- Click on “Create”and note the Client ID and Secret for the platform configuration.



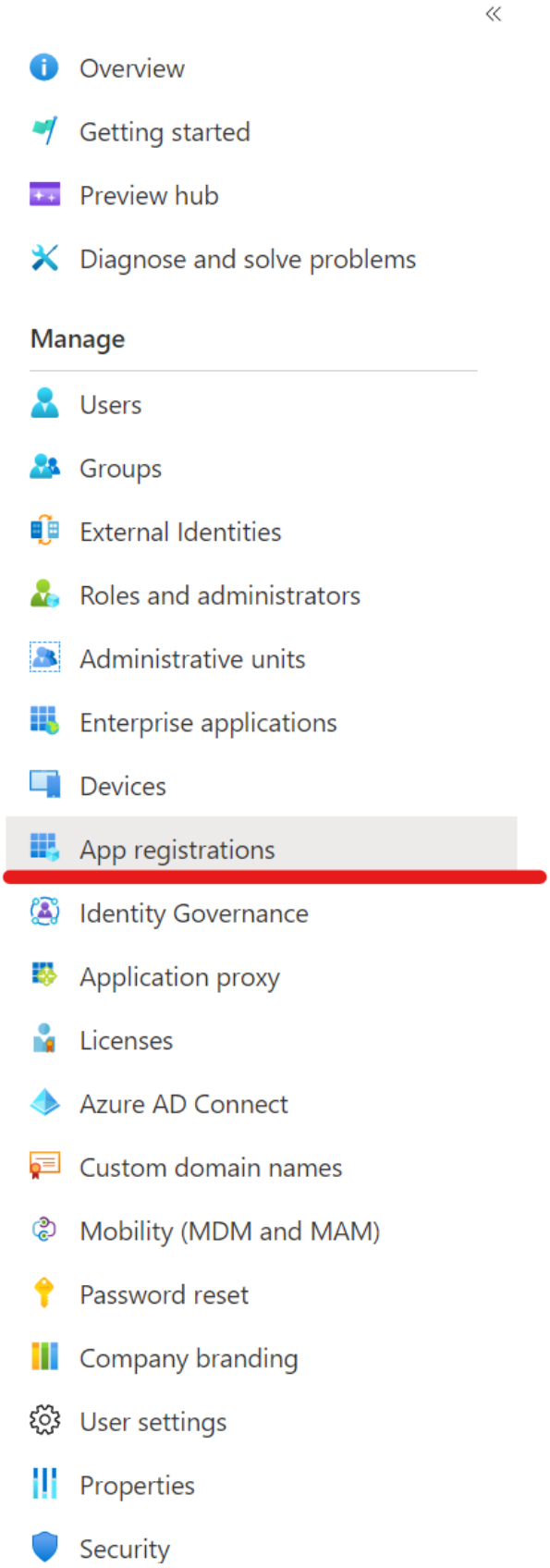

## Microsoft Azure Configuration as Identity Provider (OIDC)

October 2, 2025

The platform supports integration with Azure Active Directory for logging in with your Microsoft Azure account. To configure it:

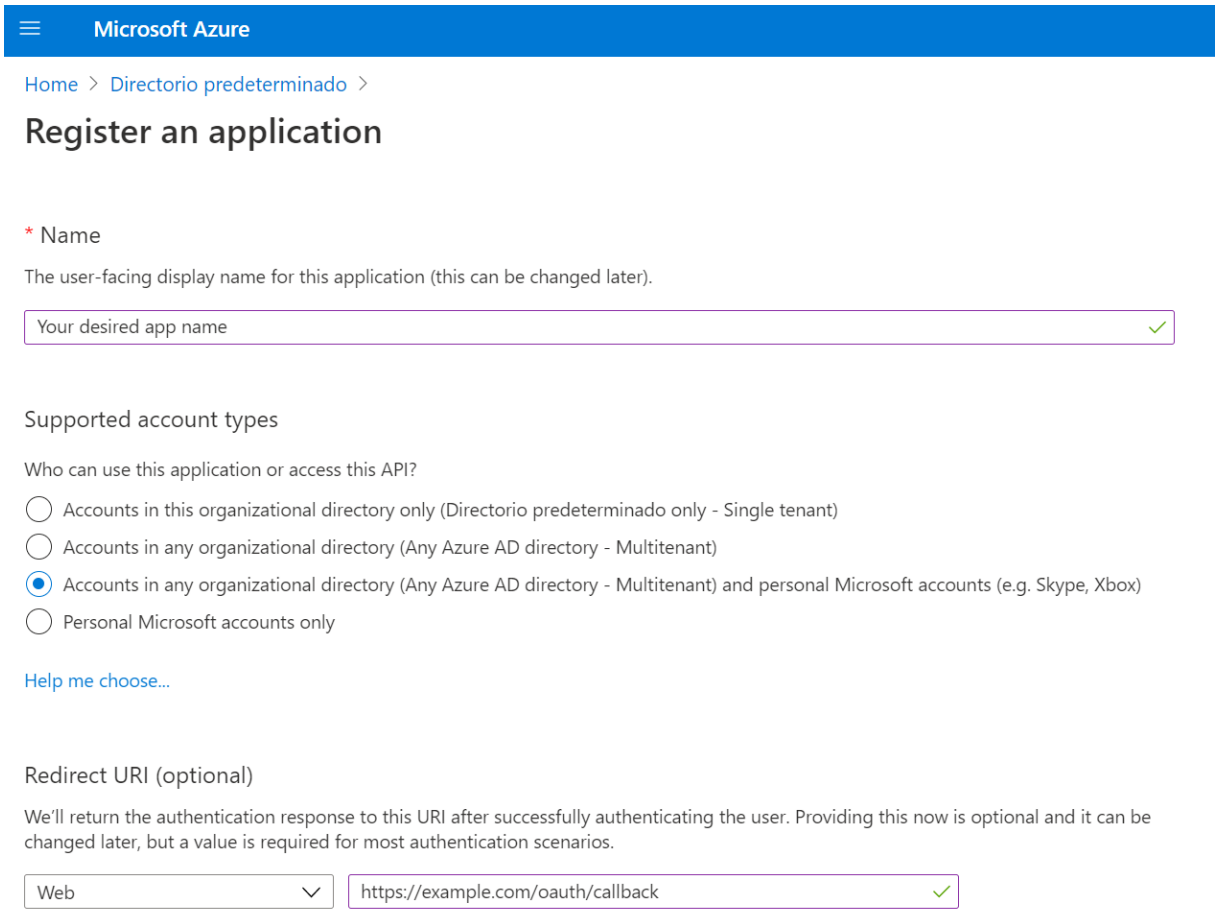
- Go to the Microsoft Azure portal.
- Navigate to the Azure Active Directory.



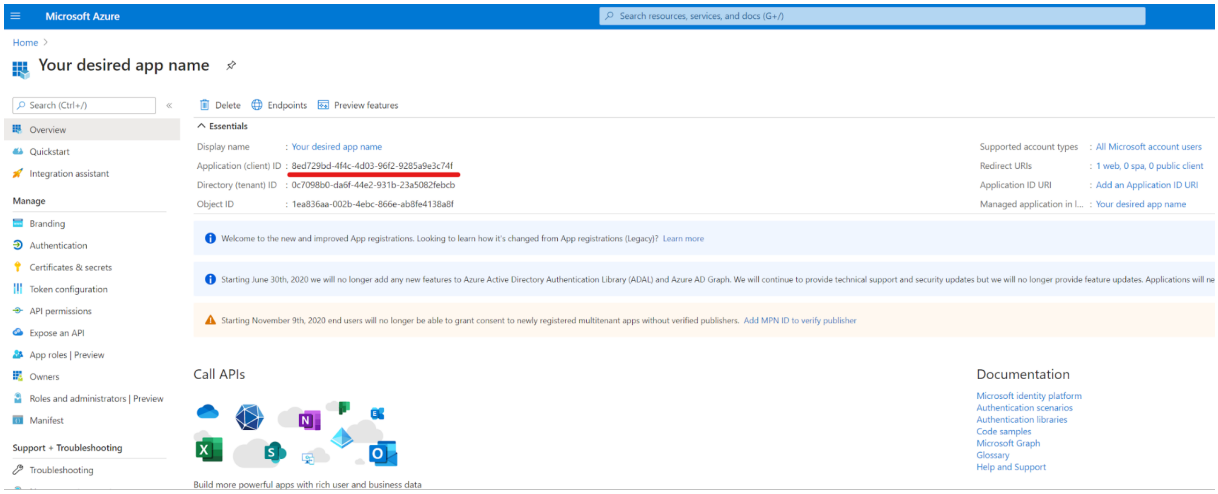

 

- Click on “App registrations”and then “New registration”. Set the following:
- **App name:** Choose a name that will be publicly visible to users logging into the platform.
- **Supported account types:** We recommend selecting “Accounts in any organizational directory (Any Azure AD directory - Multitenant) 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`.

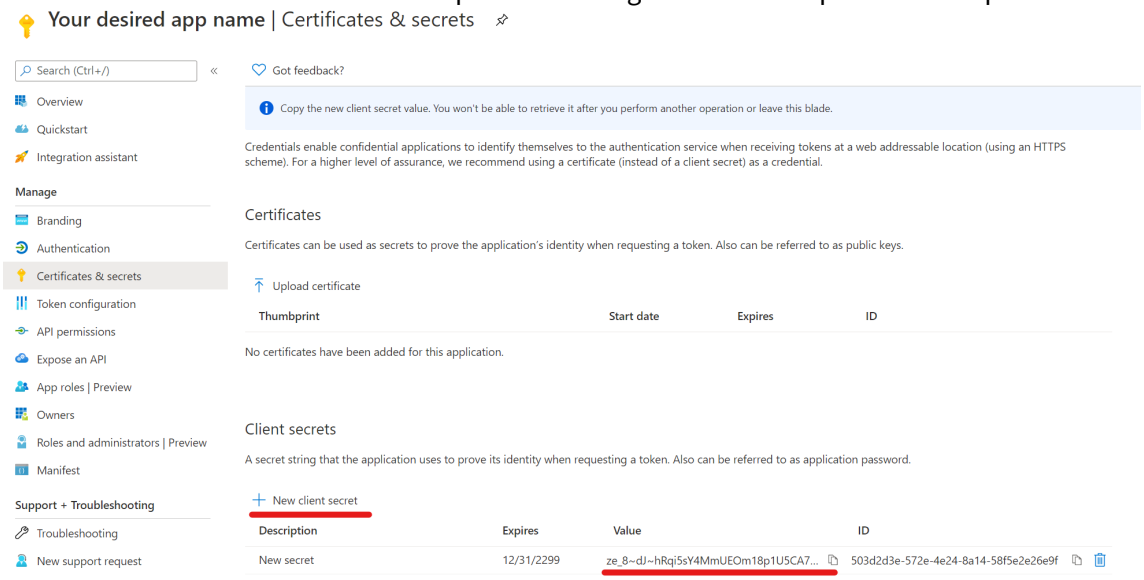



- Click on “Register”at the bottom.
- On the next page, note the OAuth Client ID for the platform configuration.




- 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 Strong Network™ platform, add another URL in the Redirect URI section with just the domain name used by the Strong Network 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”.

- To log the user out of the Strong Network 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 Strong Network platform. This endpoint will be called by Microsoft when a user logs out to also log out the user from the Strong Network platform.

## OpenID Connect Configuration as Identity Provider (OIDC)

September 29, 2025

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.

### 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.

## SAML Service Provider

October 2, 2025

To seamlessly onboard your users already registered in Okta to the Strong Network Platform using the SAML 2.0 protocol, follow these steps:

- **Configure Your SAML Identity Provider:**
  - **Single Sign-On URL:** Set to `http(s)://example.strong.network/saml/acs` where "example.strong.network"is the domain where the platform is deployed.
  - **Audience URI:** Set to `http(s)://example.strong.network/saml/metadata`
  - **Attribute Statements:**
  - **email:** This attribute is mandatory, and the configuration won't work without it.
  - **firstName:** Optional; if not set, the email will be used as the username.
  - **lastName:** Optional; if not set, the email will be used as the username.
- **Configure the Strong Network™ Platform:**
  - Log in to the platform as the administrator.

- Navigate to `http(s)://example.strong.network/platform/system_configuration/saml_sp` or click on System Configuration -> SAML Service Provider Configuration.
- Click on the "Configure"button to upload the metadata of your SAML Identity Provider. You can upload it either through a metadata URL or by uploading a .xml file.

Security Assertion Markup Language (SAML) Configuration

| | |
|---|---|
| Identity Provider Metadata URL | Not Set |
| Service Provider Metadata URL | Not Set |
| SSO URL (Assertion Consumer Service URL) | Not Set |

Configure

Below are the attributes that are needed to configured for the identity provider to be set up properly.

| Attribute | Type | Description |
|---|---|---|
| `email` | Required | The user email that is used for registration on the platform. |
| `displayName` | Recommended | The user's display name shown on the plaform. When empty, attributes firstName and lastName are used instead. |
| `uid` | Optional | A unique indentifier, usually provided for cross-platform traceability of user's operations. |
| `firstName` | Optional | First name of the user, only used if displayName is empty. |
| `lastName` | Optional | Last name of the user, only used if displayName is empty. |

The SAML configuration is now complete and ready to use.

## SCIM Configuration

November 7, 2025

The Citrix Secure Developer Spaces™ (SDS) platform adheres to the SCIM 2.0 specification. It is used for the automatic provisioning, synchronization, and deprovisioning of users. The SDS platform supports both the `/Users` and `/Groups` endpoints.

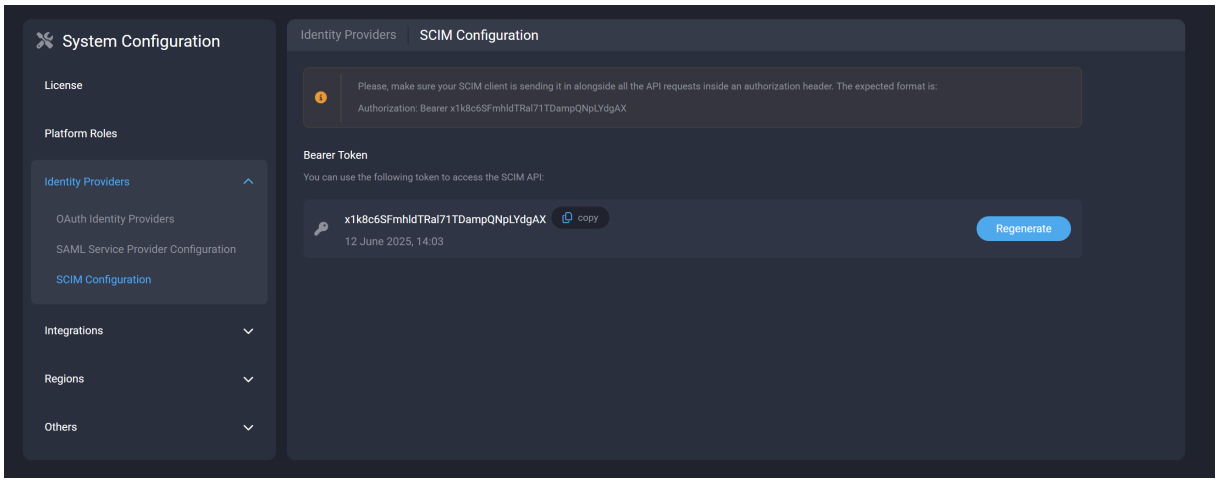
- The **Users** endpoint is used to create, update, and delete users in the SDS platform.
- The **Groups** endpoint is used to create, update, and delete groups in the SDS platform. You can then map these groups to organization(s) and/or project(s) within the SDS platform.

### Configure the SCIM Provider

A token is required to authorize requests between your SCIM provider of choice and the SDS platform. As an admin, you can obtain the token at: **System Configuration → Identity Providers → SCIM Configuration**

`https://example.strong.network/system_configuration/identity_providers/scim`




Please ensure that your SCIM provider of choice—such as Microsoft Entra, Okta, or any other SCIM 2.0-compliant provider—includes this token in all API requests, using the following authorization header format:

```
Authorization: Bearer <token>
```

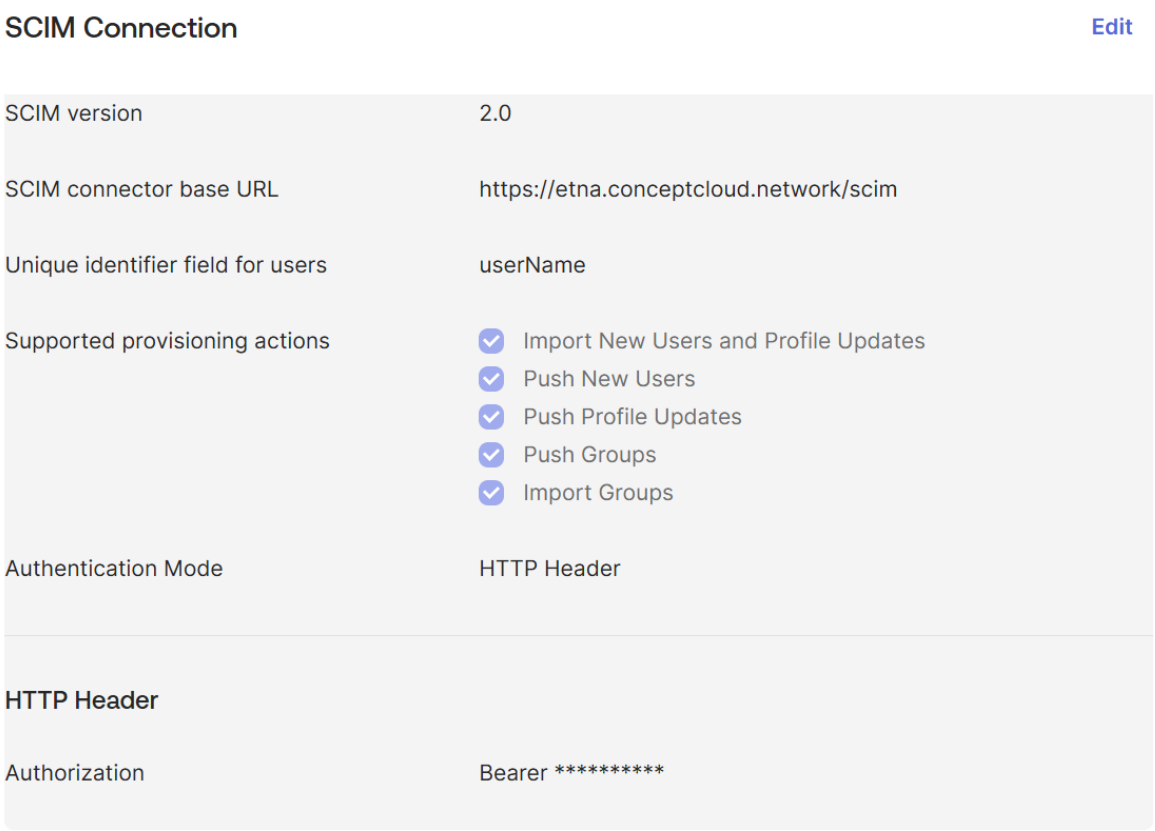
#### Using Okta

To use Okta, you will need to set these two fields:

- **SCIM connector base URL:** `https://example.strong.network/scim`
- **Unique identifier field for users:** `userName`

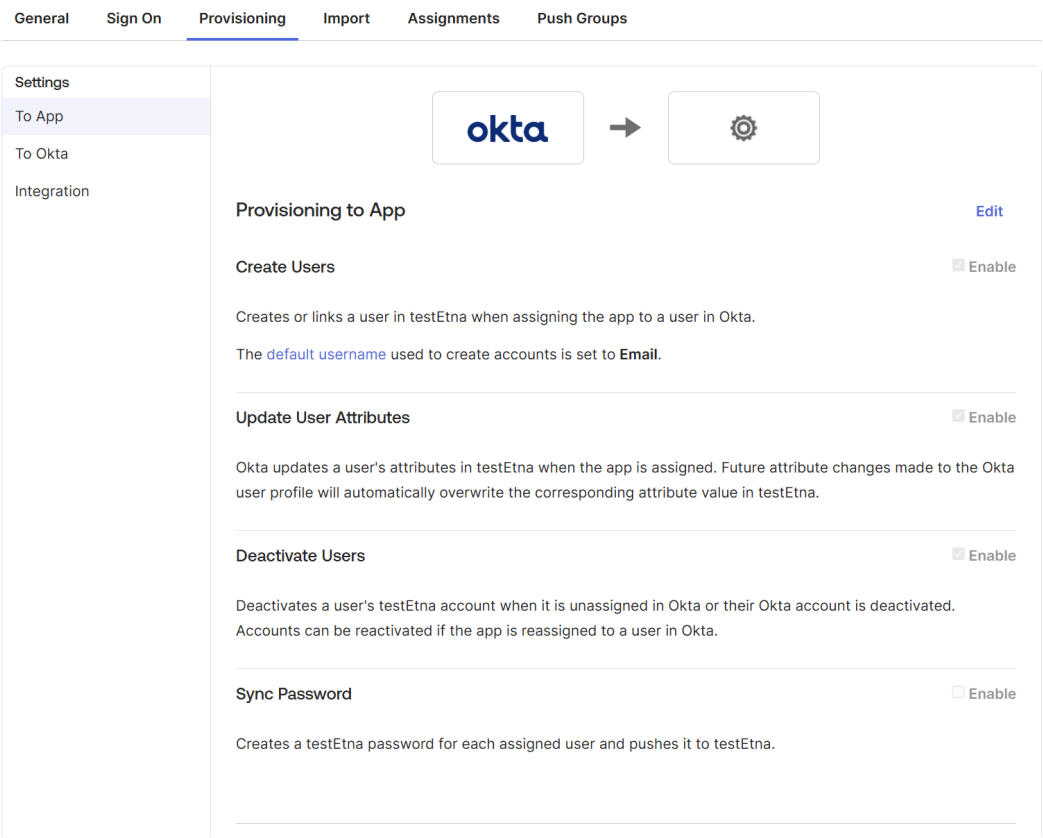
It will look similar to:




Under users you can enable the following options, as desired:



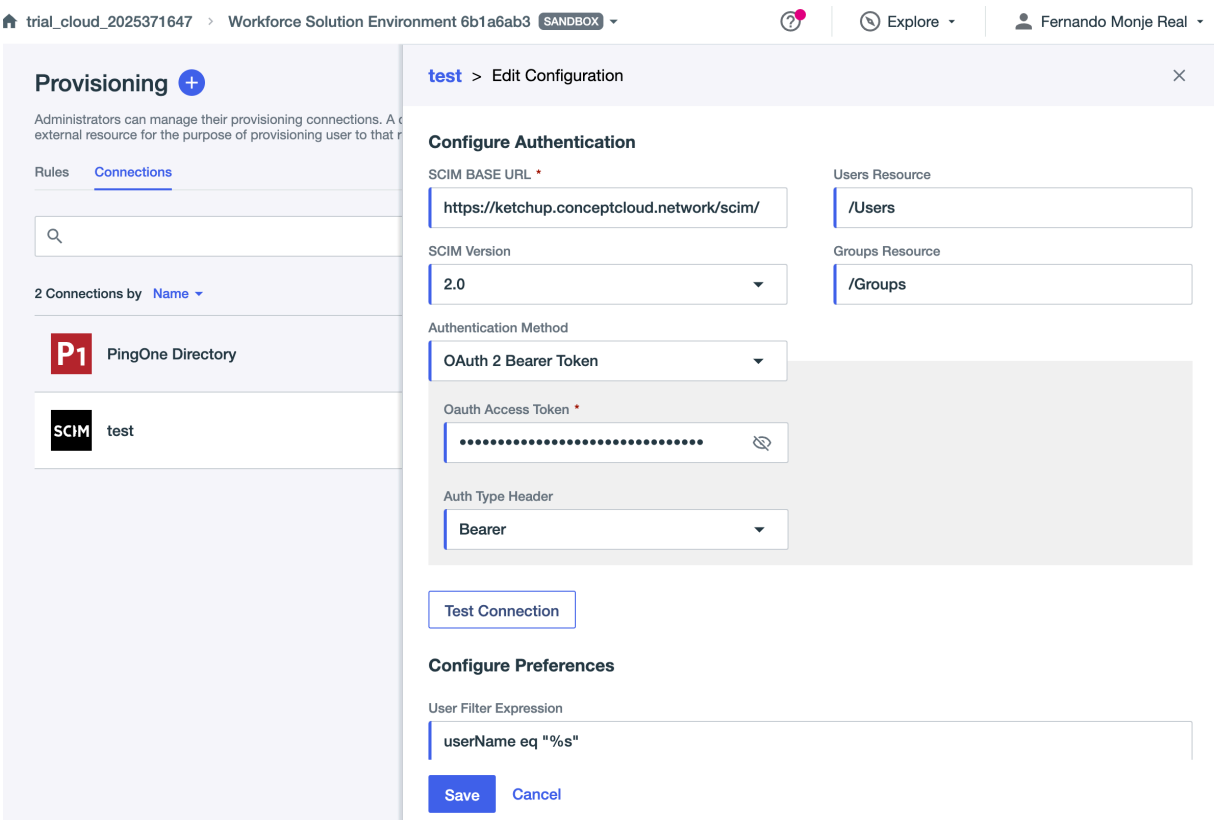

### Using PingOne

To use PingOne, you will need to set the following fields:

- **SCIM base URL:** `https://example.strong.network/scim`
- **User Filter Expression**: Modify “username”by “userName”as well as “Eq”to “eq”

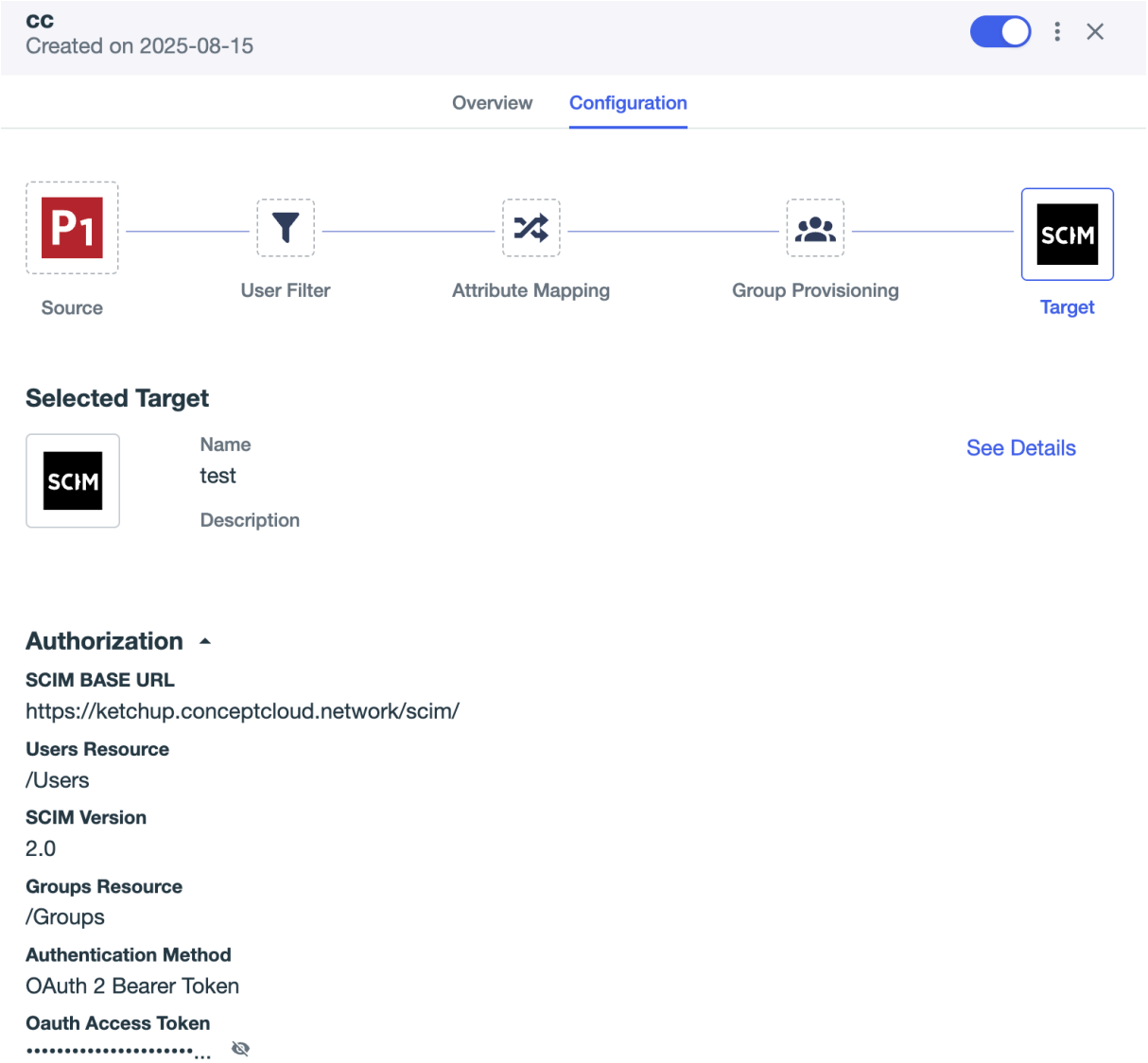
When configuring it should look like:

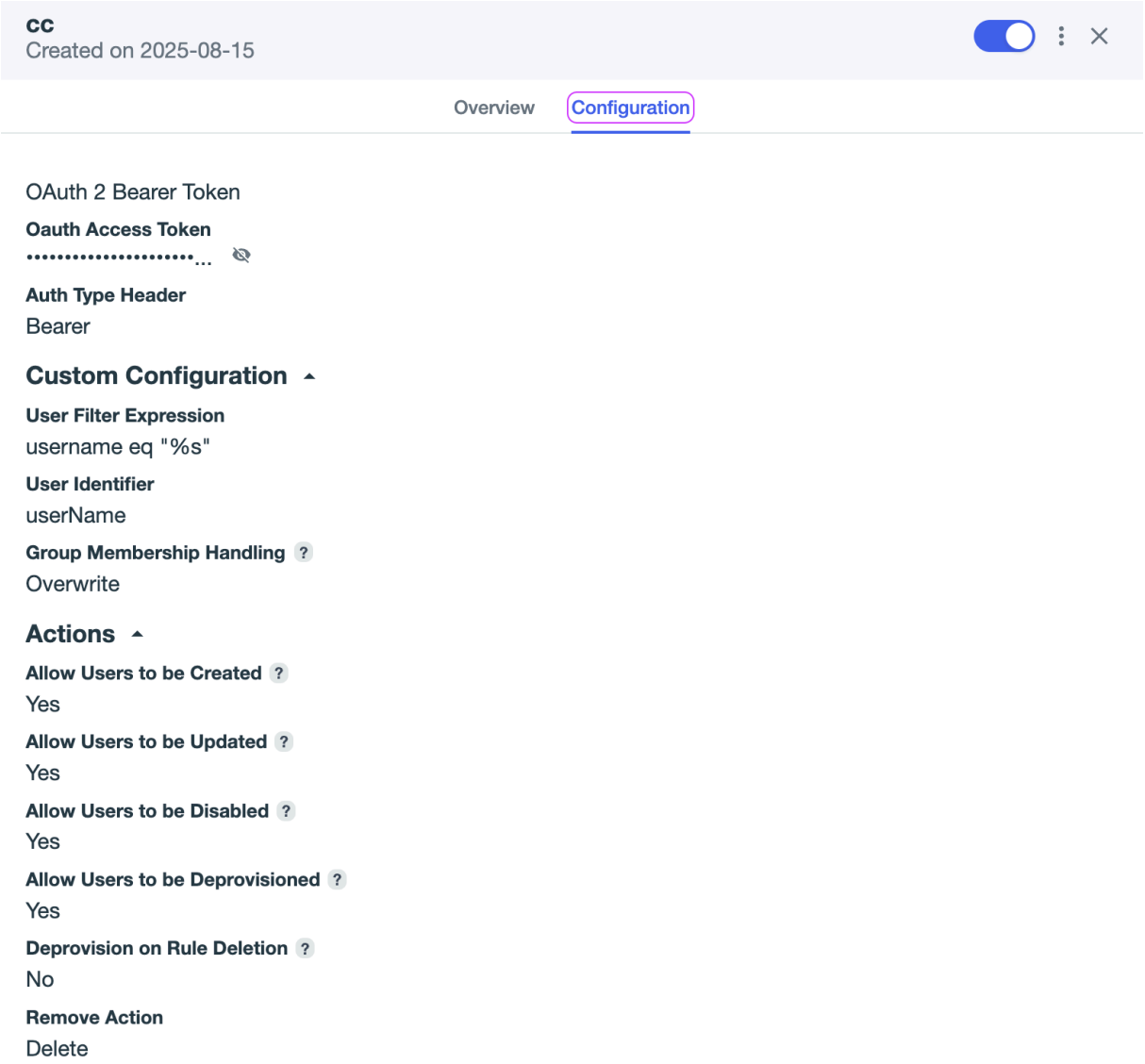



When configured the result should be like:





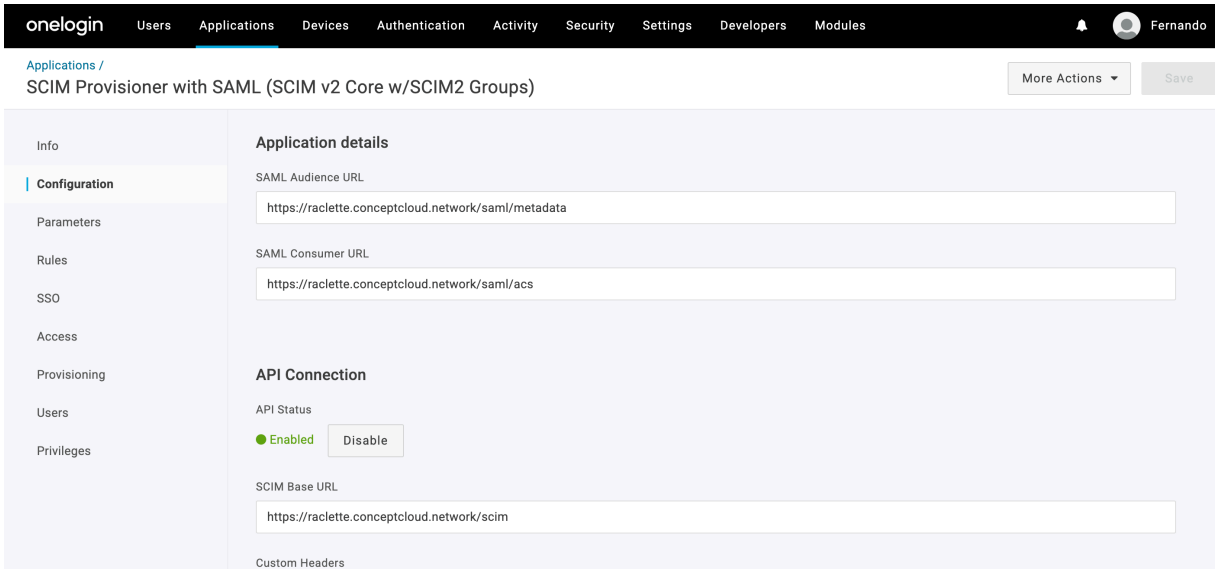

### Using OneLogin

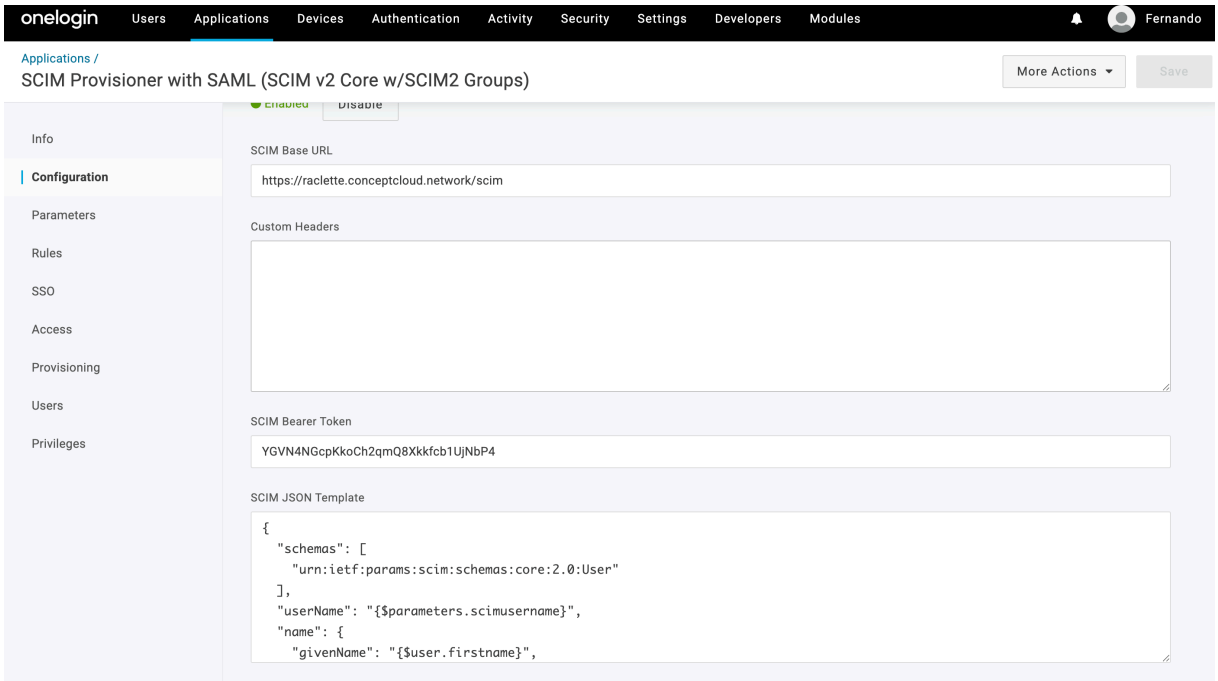
To use OneLogin, you will need to set the following fields:

- **SCIM base URL:** `https://example.strong.network/scim`
- **scimusername**: Set its value to Email

The configuration should look like:

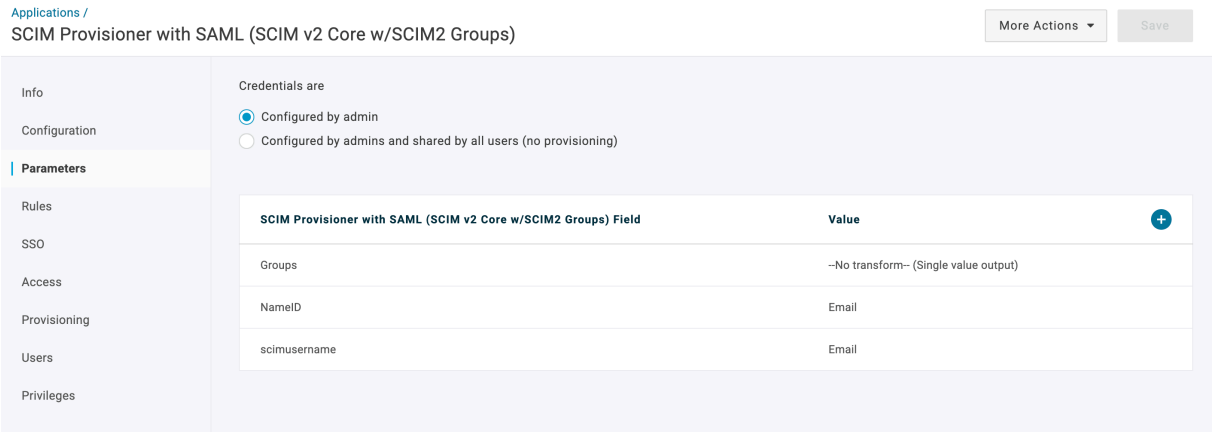
 




The parameters section should look like:

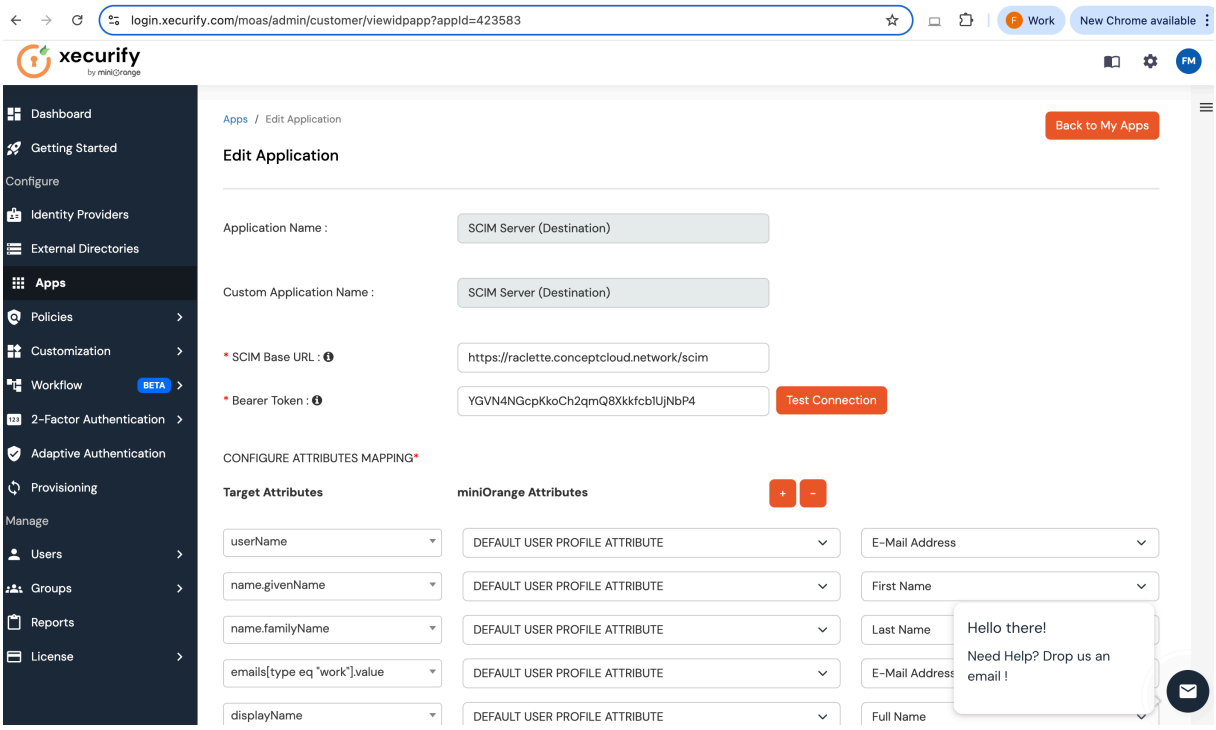



## Using Xecurify (miniOrange)

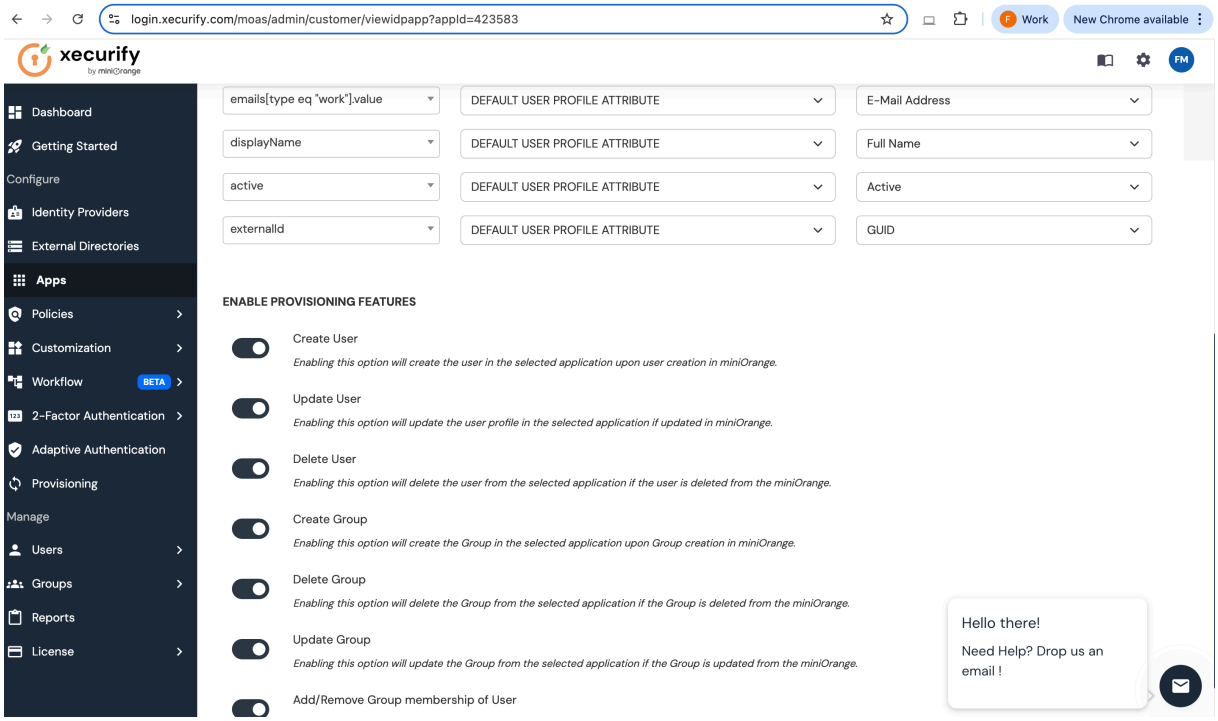
To use Xecurify, you will need to set the following fields:

- **SCIM Base URL:** `https://example.strong.network/scim`
- **userName**: Set its value to E-Mail Address

The configuration should look like:



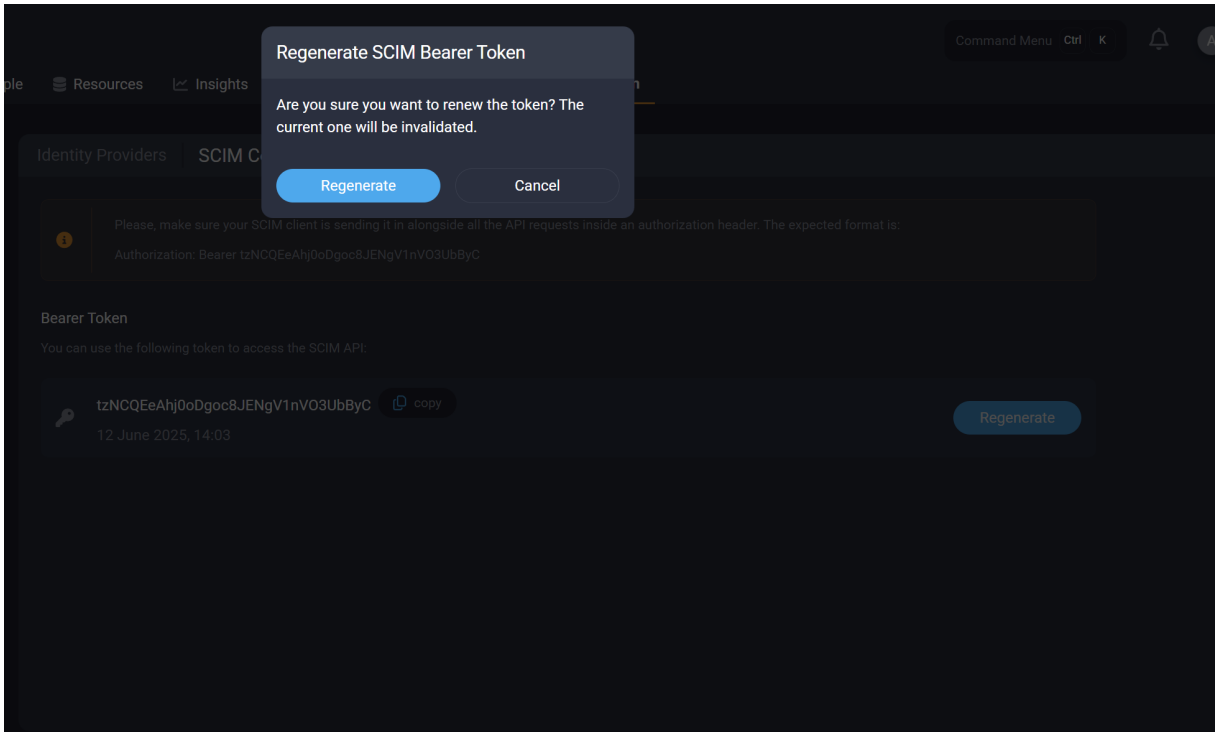



## Configure the SDS Platform

No additional configuration is needed on the SDS platform. SCIM is enabled by default.

You can renew the token at any time. Please note that renewing the token will invalidate any previously issued tokens.




## Nginx Ingress Recommended Settings

September 29, 2025

This is a recommended Nginx configuration to speed up the platform in customer deployments. A default configmap exists in the ingress-nginx namespace, typically named ingress-nginx-controller. The name may vary depending on how the ingress was installed on the cluster.

```
kubectl edit configmap ingress-nginx-controller
```

The configmap data:

```
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
```

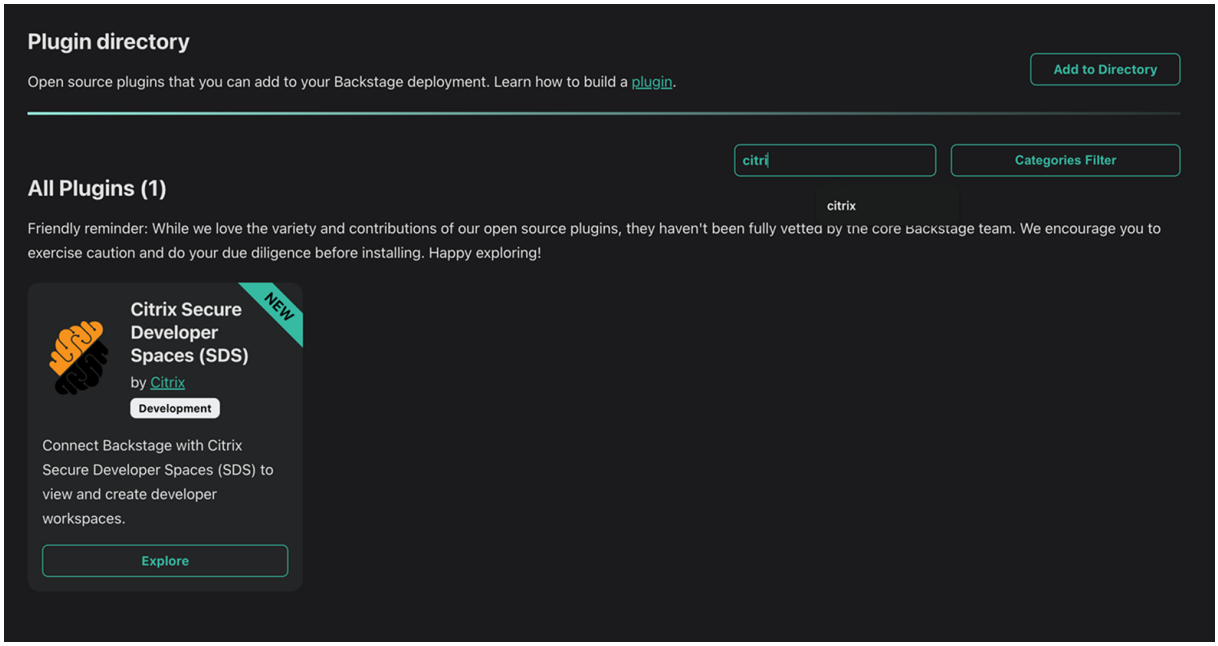
# Citrix SDS Workspaces Plugin for Backstage

November 5, 2025

Integrate Citrix Secure Developer Spaces™ (SDS) 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 Citrix SDS Workspaces Plugin allows developers to view and manage SDS Workspaces directly from Backstage entity pages. This integration enhances developer productivity by embedding workspace operations into the tools they already use.



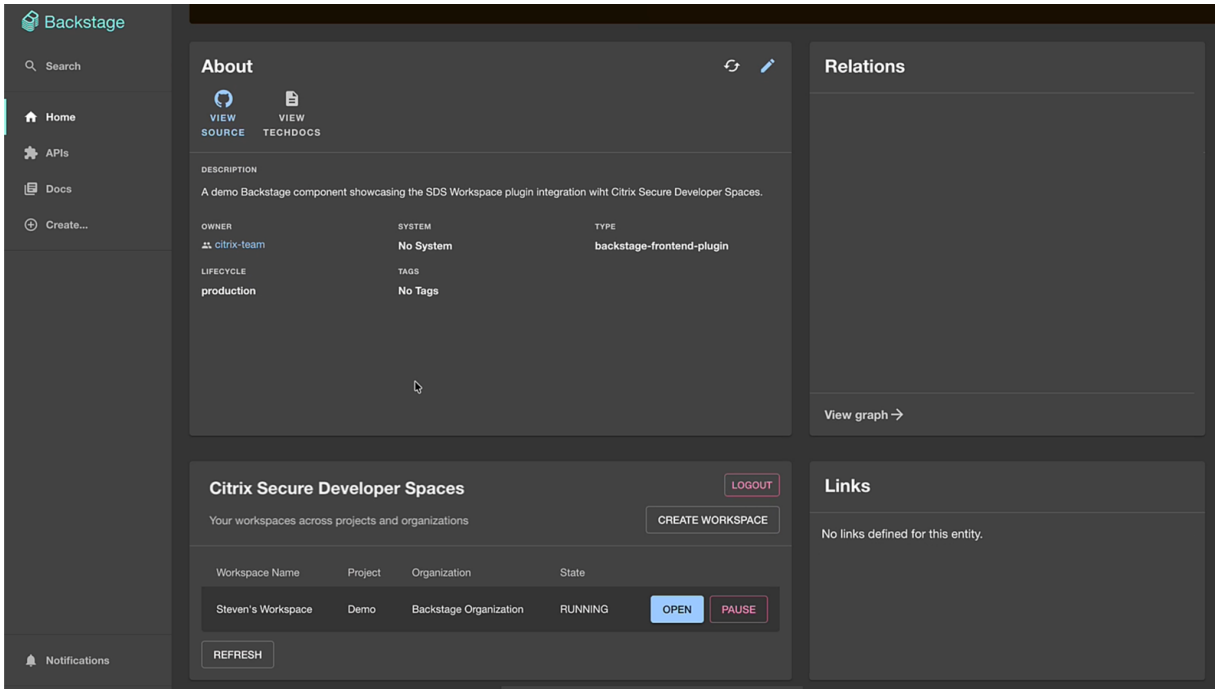

> Note:
>
> The SDS Workspaces integration requires both frontend and backend plugins to function correctly.

## Key Features

- View and manage SDS Workspaces from Backstage
- Custom SDS Workspace cards and tabs on entity pages

- Secure backend integration with SDS platform APIs
- No client-side exposure of credentials




## Prerequisites

- A running Backstage instance
- Access to Citrix Secure Developer Spaces
- SDS platform base URL

## Backstage Plugin Architecture

| Backstage Plugin Name | Description |
|---|---|
| @citrixcloud/backstage-sds-workspaces | Frontend plugin for displaying SDS Workspace cards and tabs |
| @citrixcloud/backstage-sds-workspaces-backend | Backend plugin for secure communication with the SDS Workspaces platform |

> Important:
>
> To use the SDS 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 SDS Workspaces platform, providing all required APIs for the frontend plugin.

## Enable SSH Access to Workspaces in Citrix Secure Developer Spaces™

November 10, 2025

This guide describes how to configure the Citrix Secure Developer Spaces™ (SDS) 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:**
>
> 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.
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:

```
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.

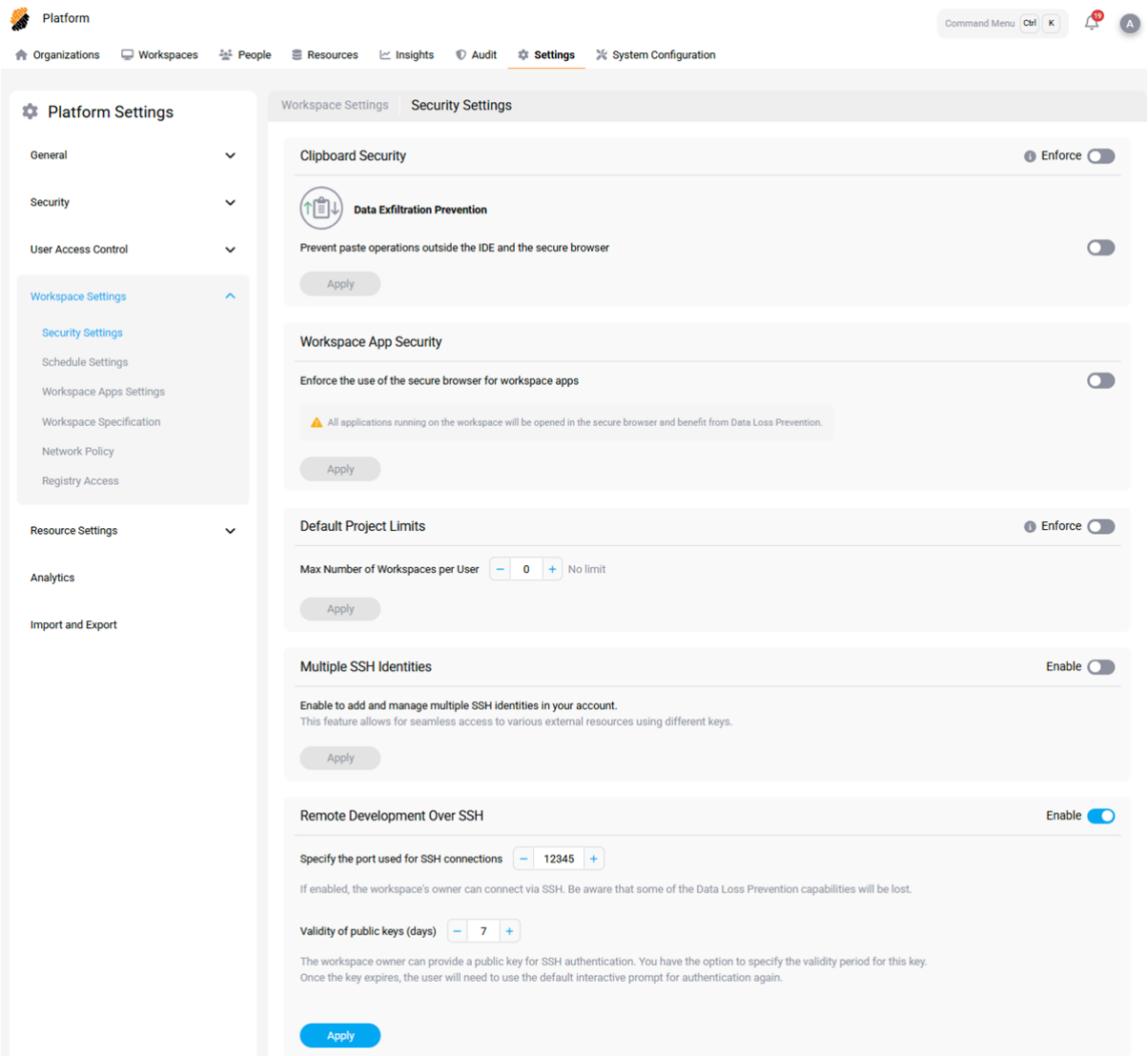
### Platform Level

> **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.
- 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.

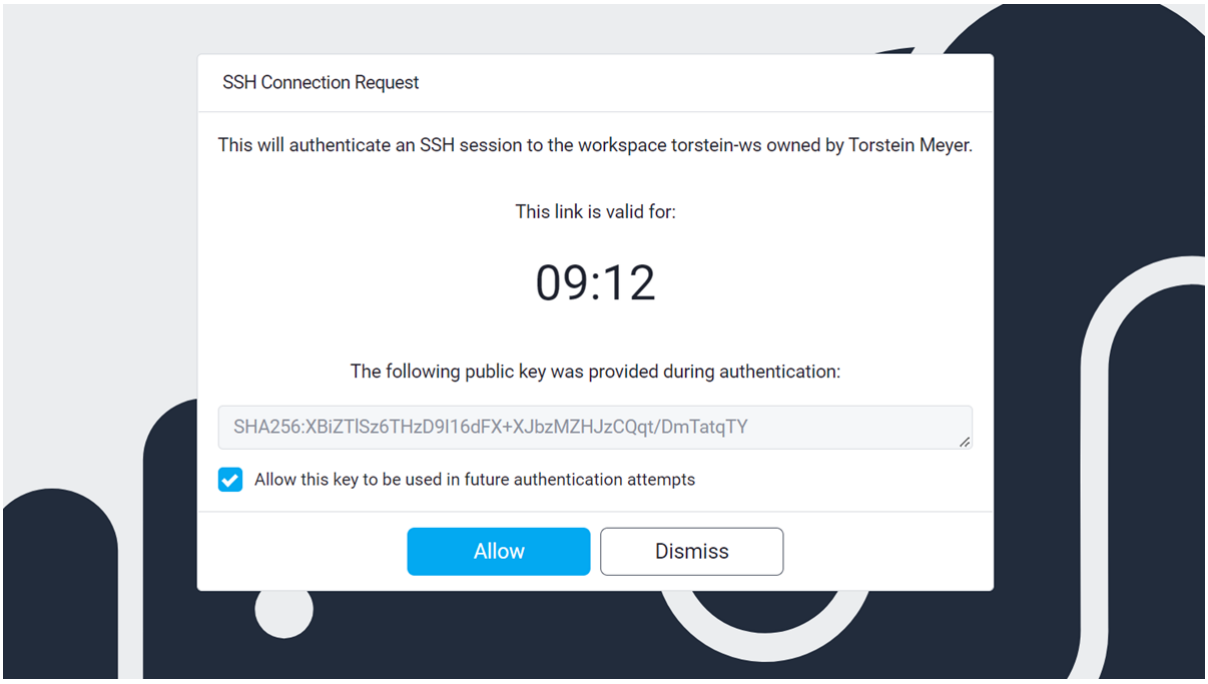
When connecting to a workspace with SSH, the user will be given the following prompt:

```
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 administrator through the validity of public keys setting.



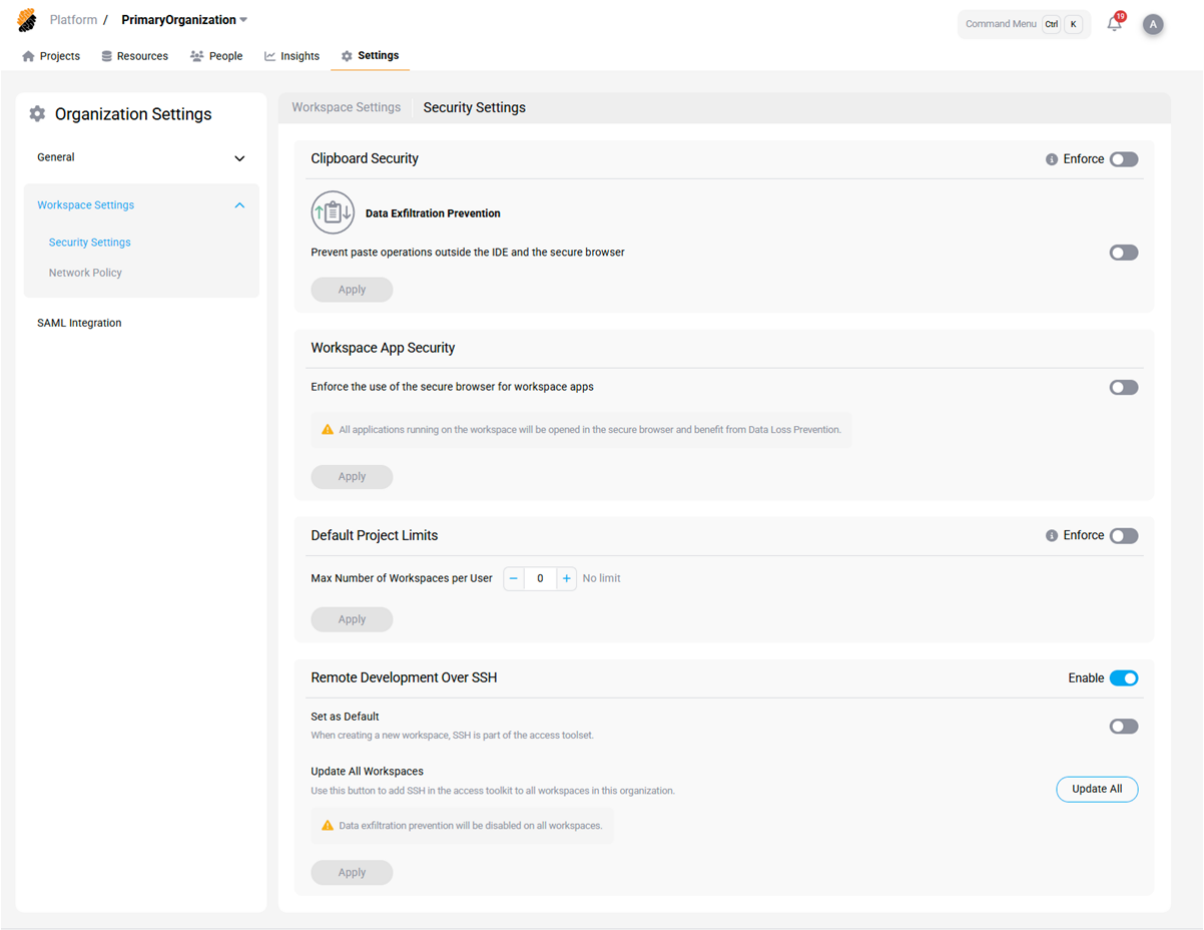

**Organization Level**

**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.



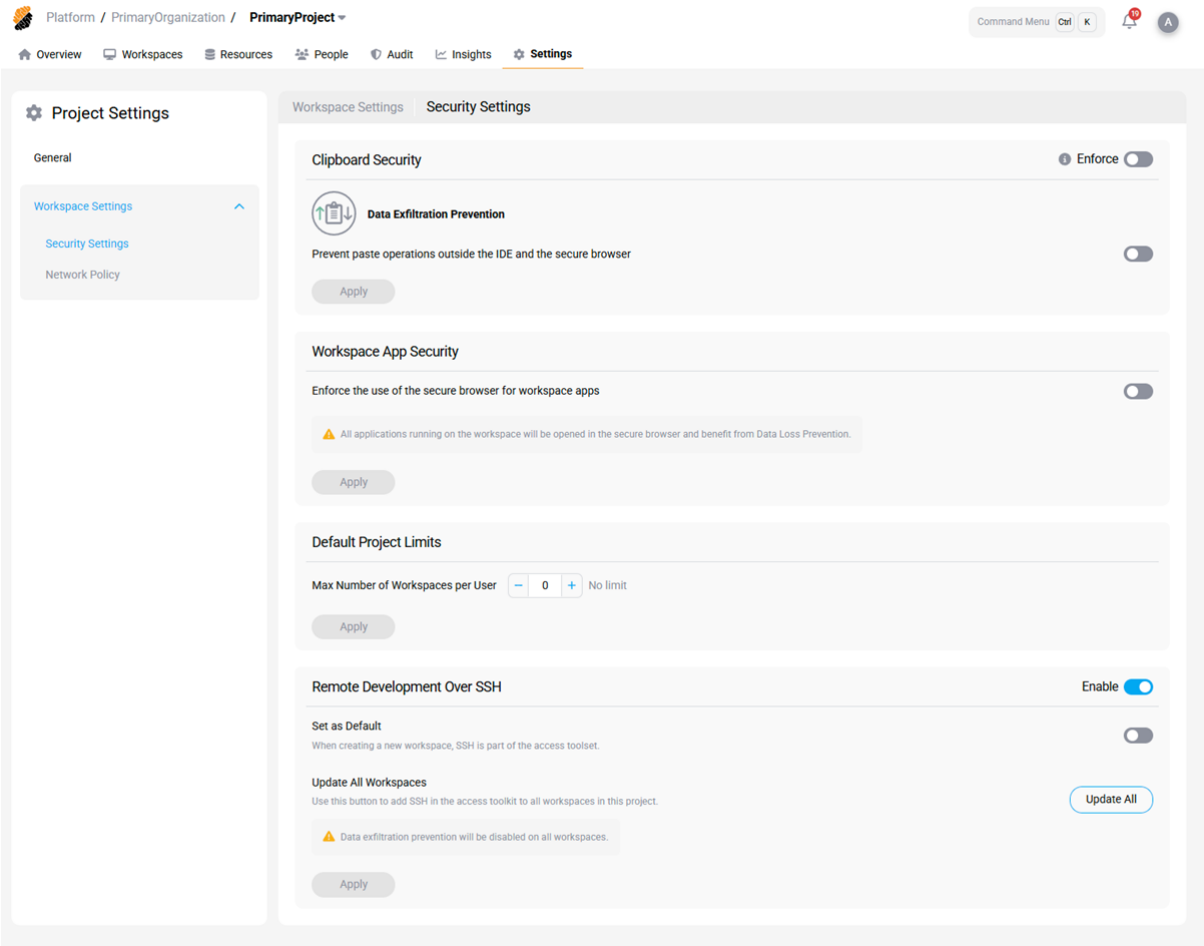

### Project Level

> **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.




## Use SSH to connect to Workspaces

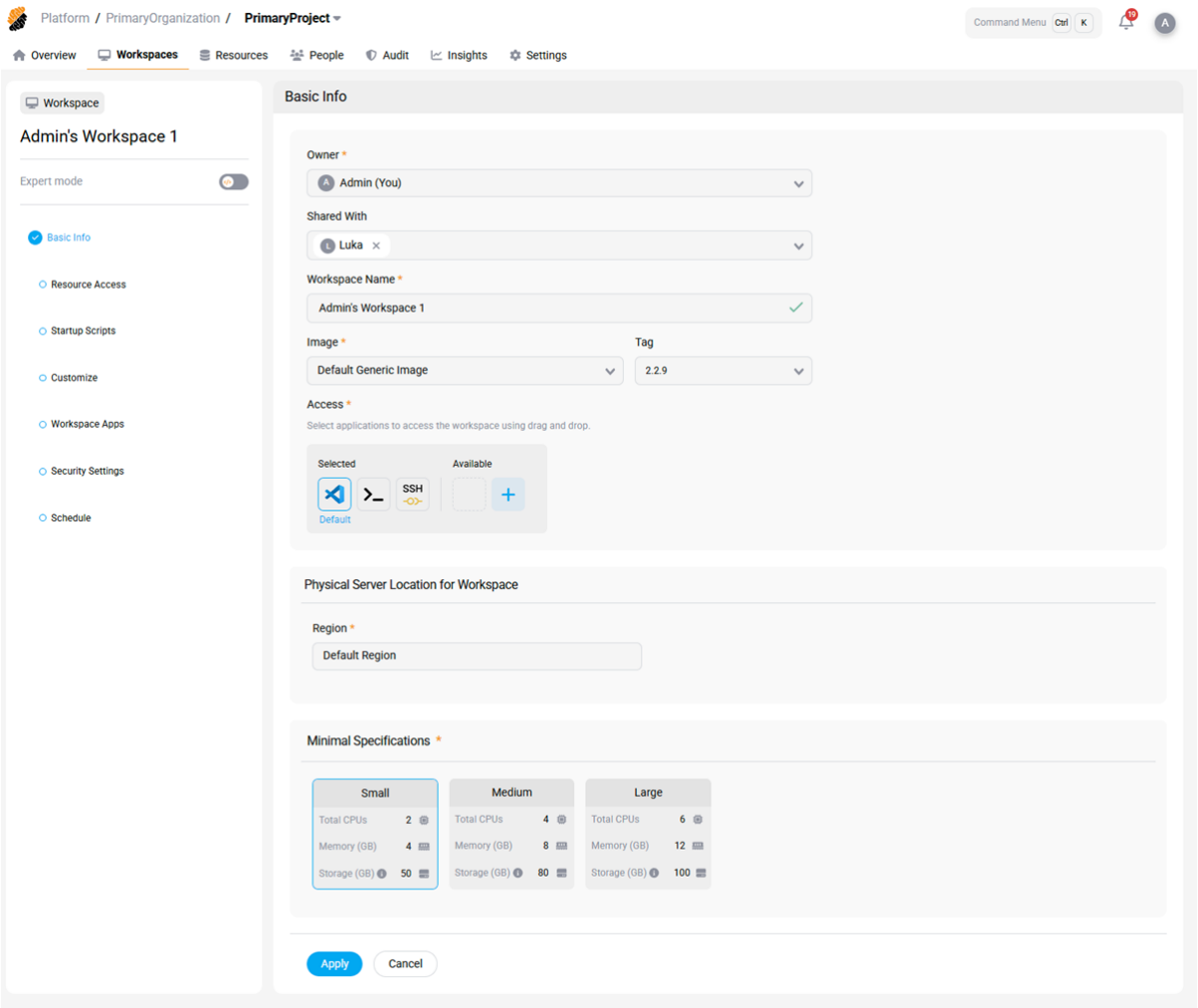
Once enabled, developers can connect to running Workspaces via SSH.

### Enable SSH on Individual Workspaces

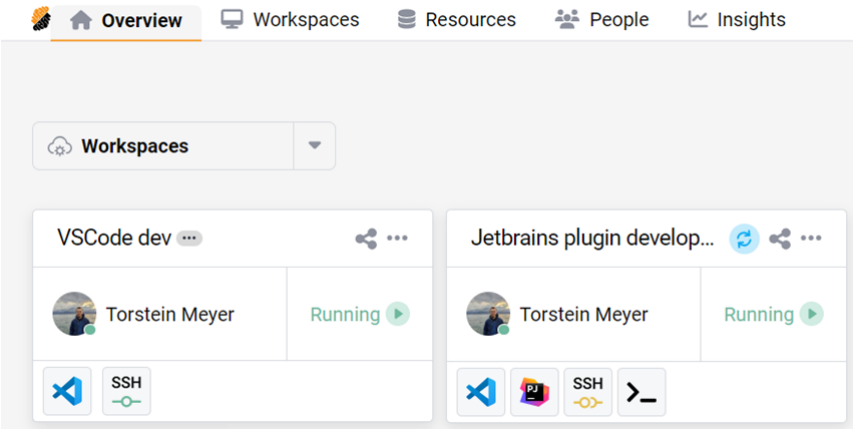
With the SSH feature enabled, developers on the platform can make use of the feature. As an additional safety measure, the feature can also be enabled or disabled on each specific Workspace. By default, SSH is disabled on individual Workspaces. To enable:

- Edit the Workspace.
- On the **Basic Info** page, under **Access**, drag the **SSH** icon from *Available* to *Selected*.
- Click **Apply**.

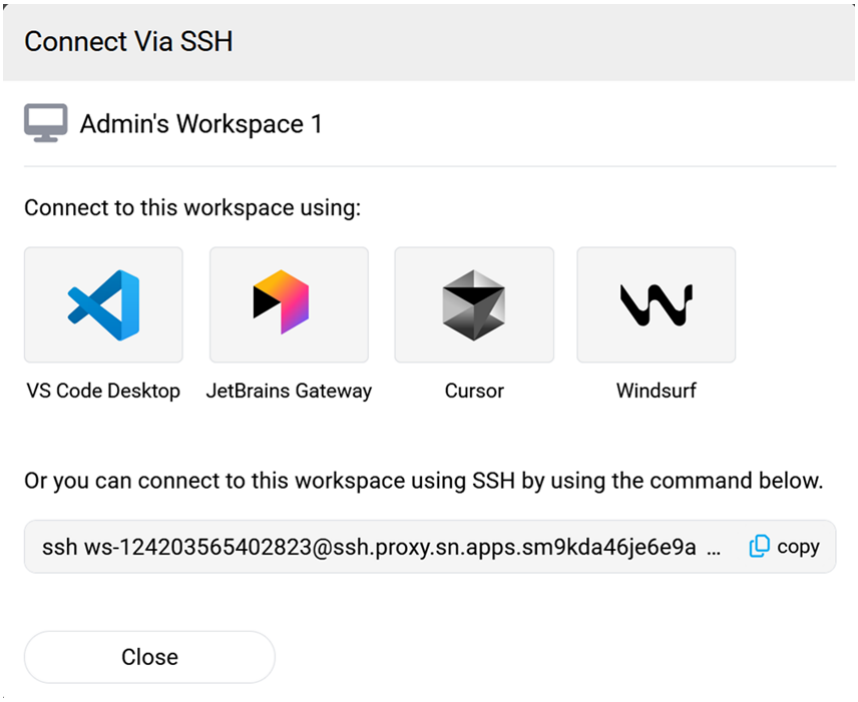
 




On workspaces with SSH enabled, the owner of the workspace will be able to access the workspace using SSH when the workspace is in a running state. To do so, first open the Connect Via SSH modal by clicking the SSH icon on the workspace card:





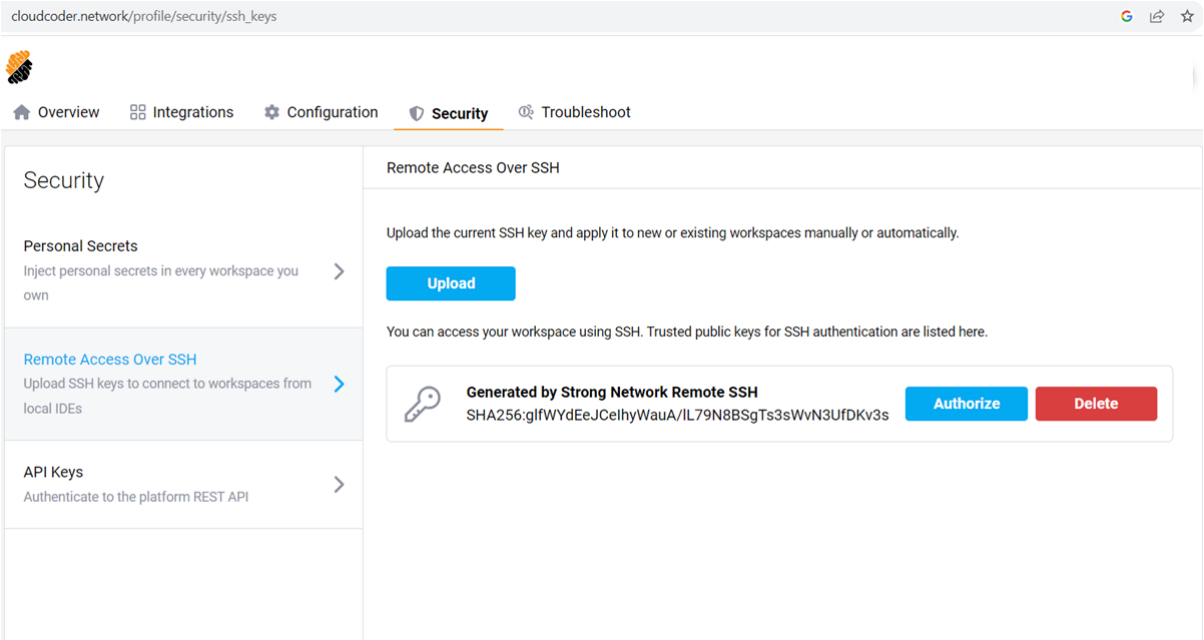

This will open the **Connect Via SSH** modal. Here the user can either connect directly to their local VS Code Desktop and/or JetBrains Gateway editors, or copy the SSH command in the format:

```
ssh ws-{
 id }
 .ssh.proxy.{
 domain }
  -p {
 port }
```

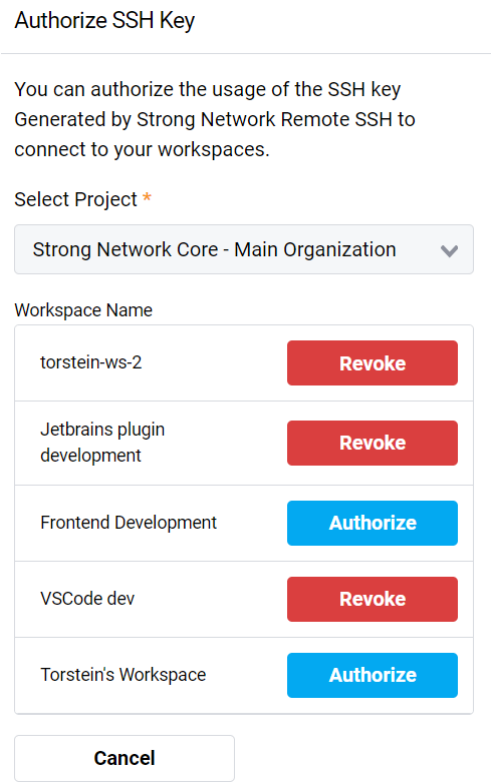
You can then use this command to access the workspace as you would any ordinary SSH server.

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:




Clicking **Authorize** will allow the user to specify the key's access to specific workspaces:



### 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.

## Third Party Application Setup

October 2, 2025

Jfrog Integration Setup

## Register JFrog as Third Party App

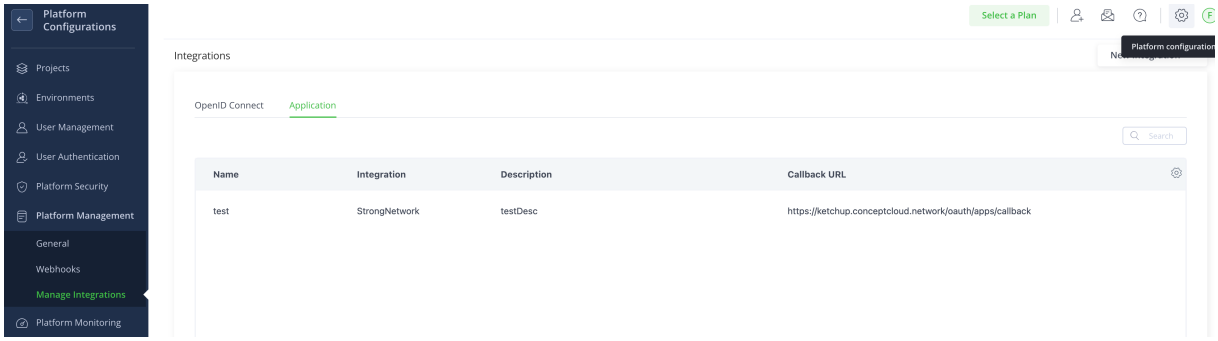
October 2, 2025

You can follow these steps to connect your JFrog instance and the Strong Network™ 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.

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
‘

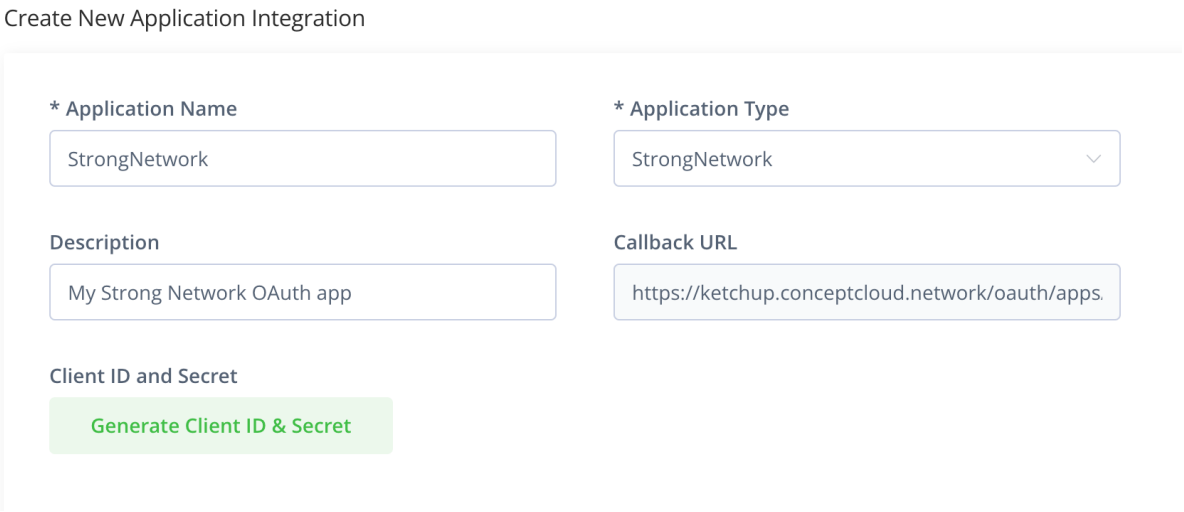



Click on “New Integration”of type “Application”and fill in the following fields:

- **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 admin menu in the Strong Network platform and has the format of `https://[your_strong_network_domain]/oauth/apps/callback`



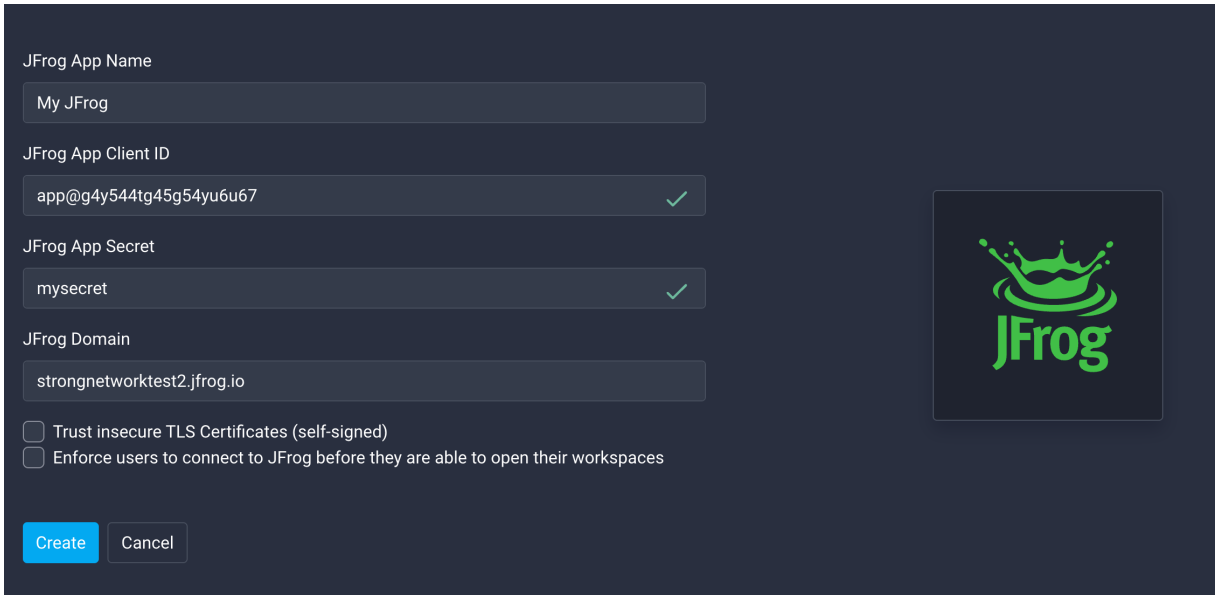

Click on **`Generate Client ID & Secret`** and copy the values.

Lastly, log in as admin in the Strong Network Platform, go to System Configuration → Third Party Applications, and select JFrog. You will need to introduce:

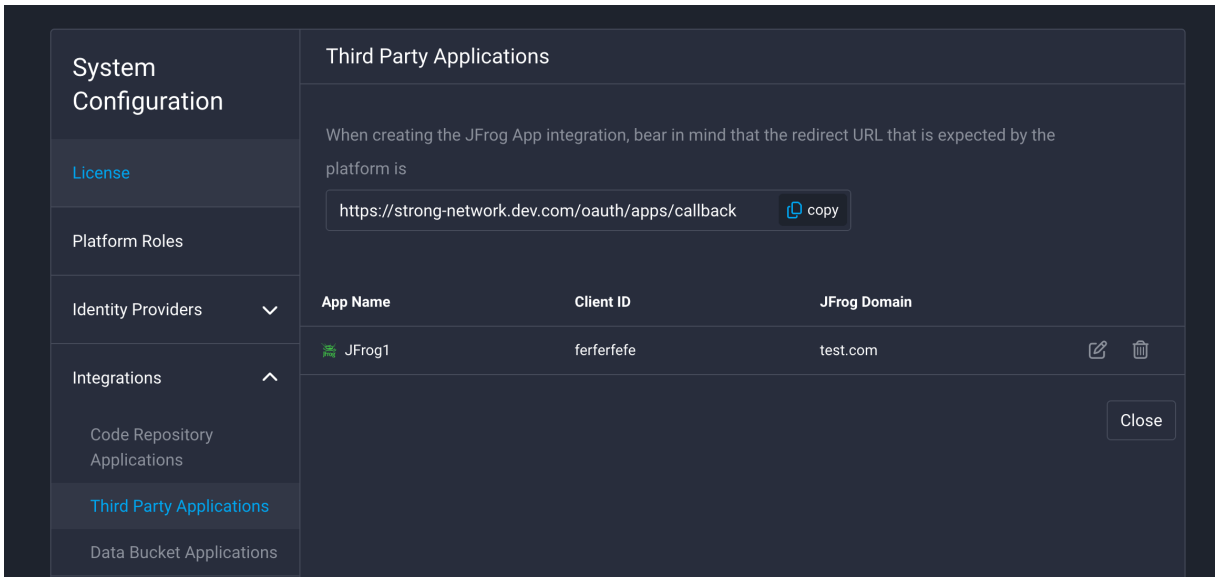
- **Name:** Up to you, it will be displayed to the platform users
- **Client ID and Secret:** Values copied from JFrog
- **Domain:** Your JFrog domain

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.



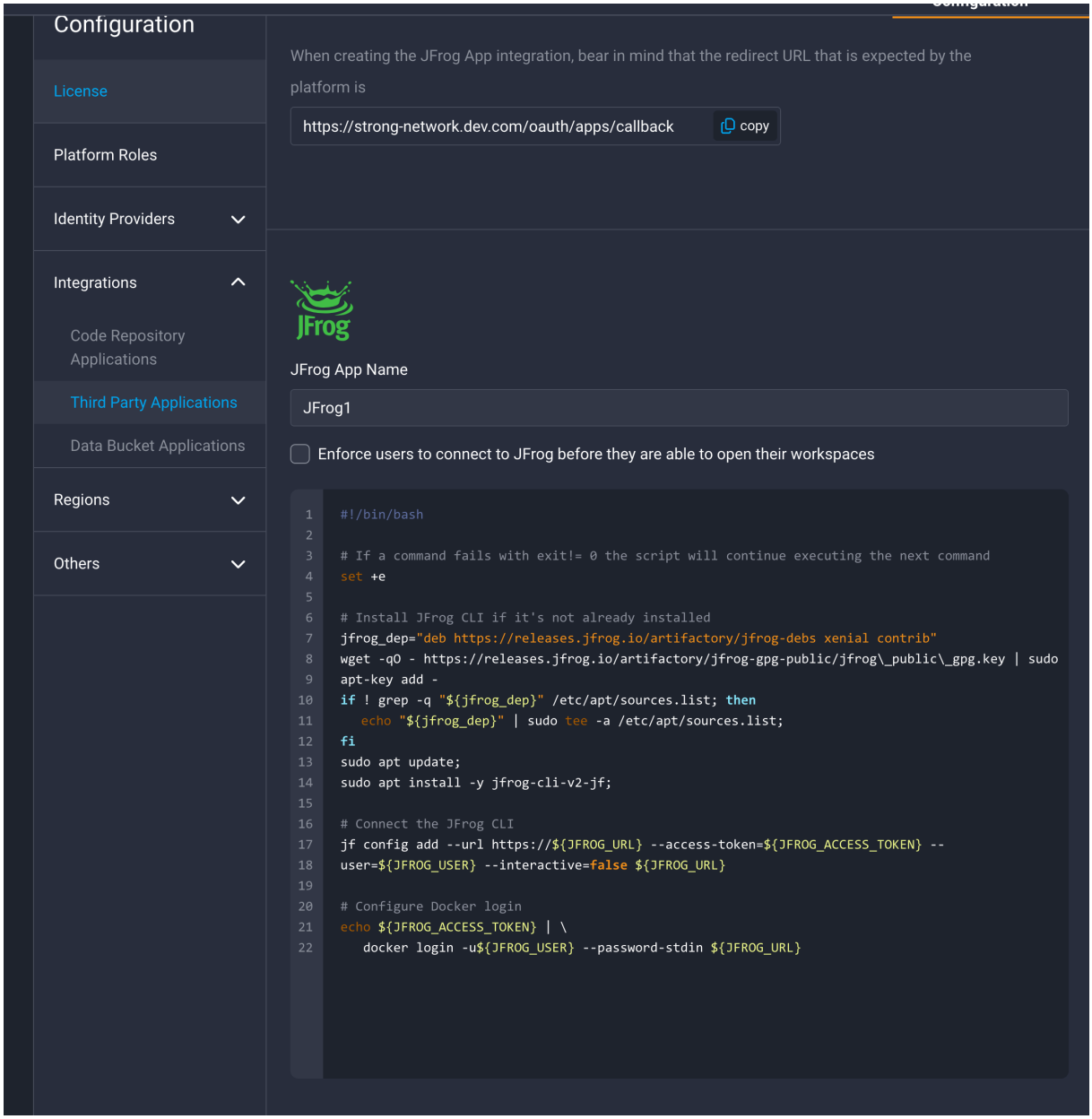

When the application is configured you can edit it by clicking on the edit icon on the right side.




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, 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 Strong Network Platform, ready to be used seamlessly by the users.

## Use HashiCorp Vault as a Secret Manager

October 9, 2025

You can use **HashiCorp Vault** to store all platform secrets instead of encrypting them in MongoDB.

Citrix Secure Developer Spaces™ (SDS) 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 SDS 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 SDS cluster is reachable by Vault.
  If it isn't, you must manually add the **signing public key(s)** of the SDS cluster.
  For details, see Use Kubernetes for OIDC authentication

## Configuration

You can configure Vault in the SDS 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"
#   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 SDS cluster. All platform services use this address to store and retrieve secrets. |
| **vaultRoleName** | Specifies the name of the Vault role configured for SDS. If different Kubernetes services use different service accounts, the `bound_subject` field may vary. You can omit this field when creating the role. |

| Parameter | Description |
|---|---|
| **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. |

## Upgrading the Citrix Secure Developer Spaces™ Platform

November 18, 2025

This article describes how to upgrade the Citrix Secure Developer Spaces™ (SDS) platform using the official installer. The upgrade process involves running a Docker-based installer, executing the upgrade command inside the container, and applying the resulting Helm upgrade to your Kubernetes cluster.

### Prerequisites

Before starting the upgrade, ensure the following:

- A recent backup of the SDS configuration database.
- Access to the terminal with Docker installed.
- 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.

### Run the Installer

Launch the installer using the following Docker command:

```
docker run -it --rm -v ${
 PWD }
 :/strong-network/shared strongnetwork/strong_installer:2025.10.2
```

> **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:

```
./strong-cli upgrade -c config_<your-current-version>.yaml
```

**Example:**

```
./strong-cli upgrade -c config_2025.5.0.yaml
```

The installer will guide you through the upgrade process. It validates your configuration, checks compatibility, and prepares the necessary resources.

### Apply the Helm Upgrade

After the upgrade process completes, the installer will output a Helm command tailored to your environment. This command applies the updated deployment to your Kubernetes cluster.

Run the provided Helm command in your terminal to finalize the upgrade.

### 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.

### Troubleshooting

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.

## How to Use this Guide

This guide is here to provide you with a description of the main functions provided by Citrix Secure Developer Spaces.

The guide covers the initial setup, configuration and general usage of workspaces, which are online Cloud Development Environments (CDEs) available for coding and data science. Workspaces can be accessed using a cloud IDE, include Microsoft Visual Studio Code, all Jetbrains'IDEs or through an SSH connection from a local installed IDE (see remote development for Microsoft Visual Studio Code)

This documentation is generally organized in a manner that follows the platform's UI pages. This provides a natural way to find information once on one of the platform's pages.

### Content



## Platform Level

October 8, 2025

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.

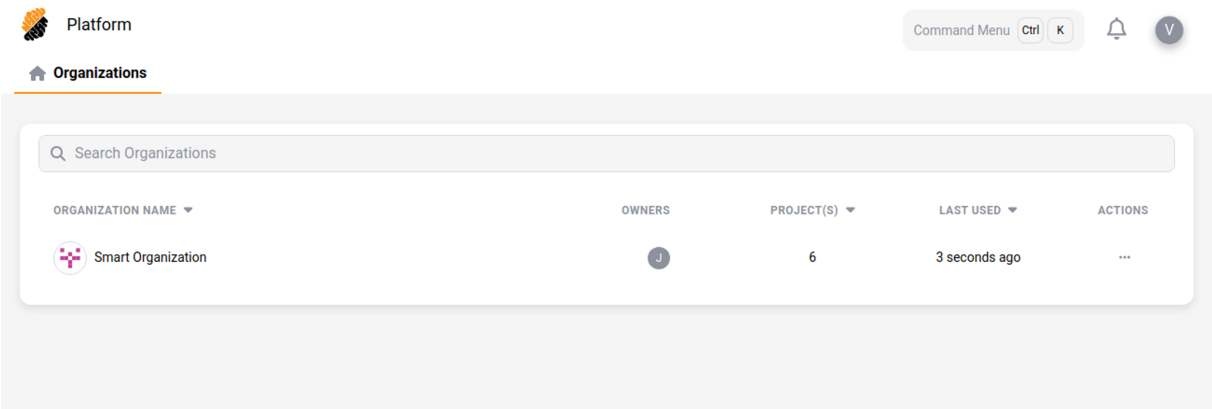
- View Organizations
- Platform Settings

### 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 Strong Network™ logo to **view your organizations** to which you belong.




*Organizations List*

### Platform Settings Admin

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

## Organizations

October 2, 2025

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

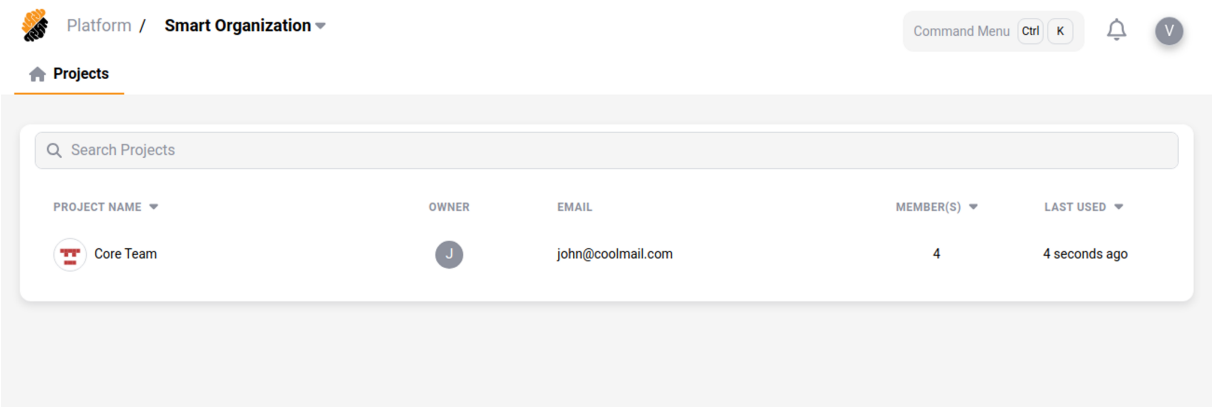
## Organization's Characteristics

An organization is defined by the following characteristics:

- **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.

## Projects

October 8, 2025

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.

### Project Settings `Project Owner`

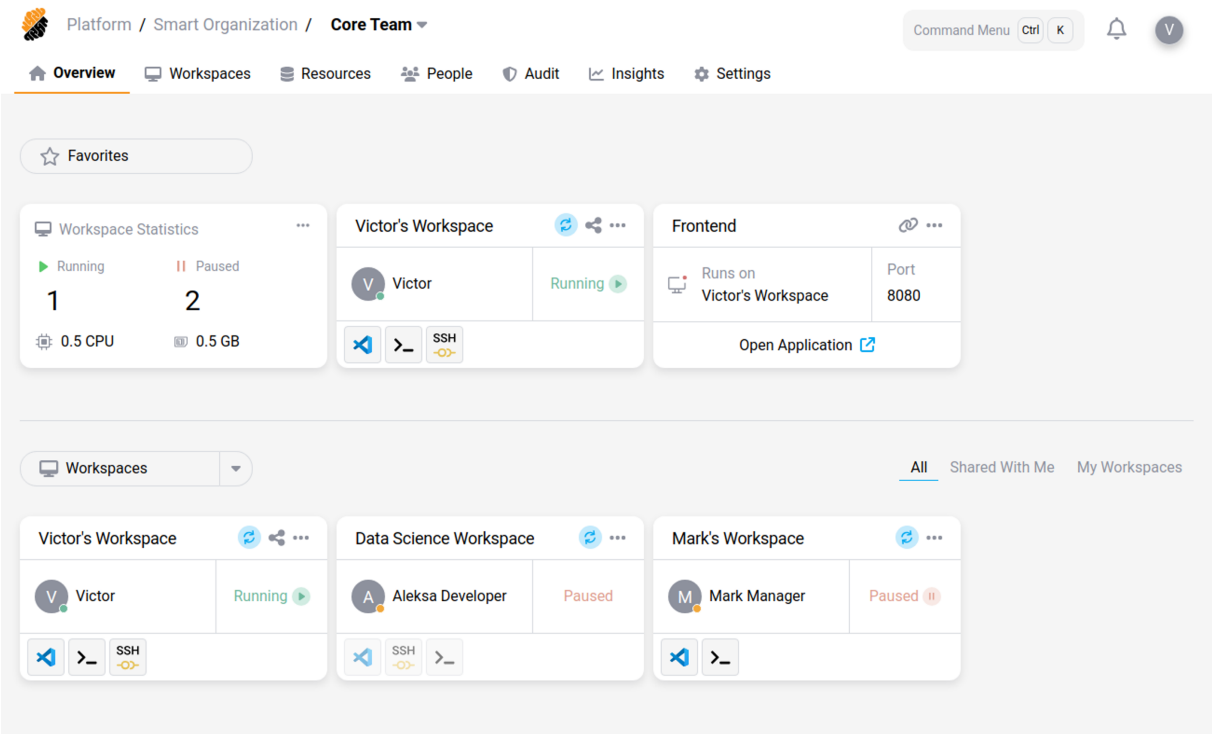
For comprehensive control over your Project's configurations, visit the dedicated Project Settings page.

## Overview Page

October 2, 2025

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.



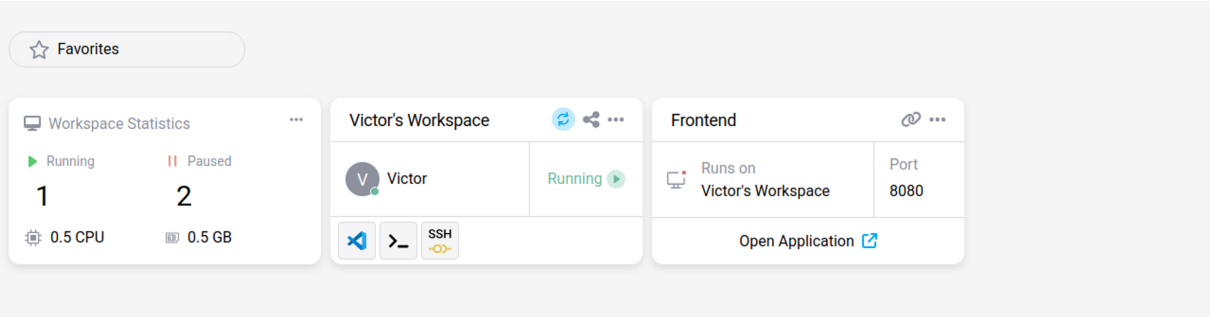

 

- Display Sections
  - Favorites
  - Workspaces
  - Workspace Apps
  - People & Other Metrics

## Display Sections

### Favorites

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

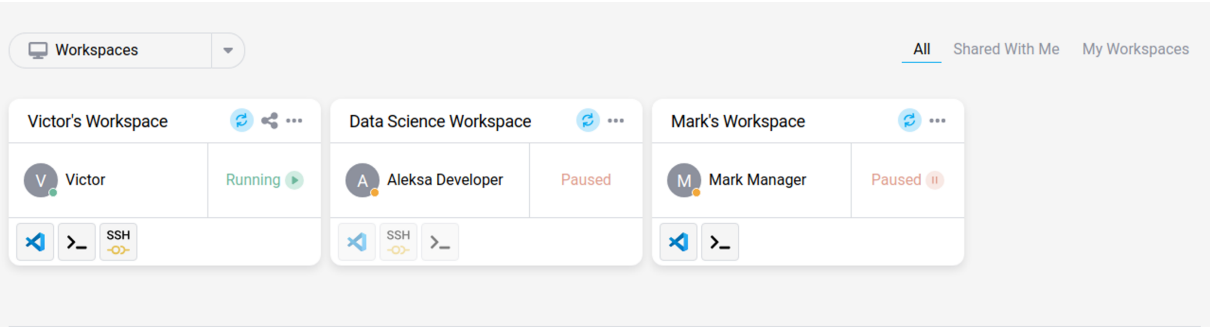



- To **add an element** to your list, click its "**...**"button and "**Add to Favorite**".
- To **remove an element** from your list, click its "**...**"button and "**Remove from Favorite**".

> 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.



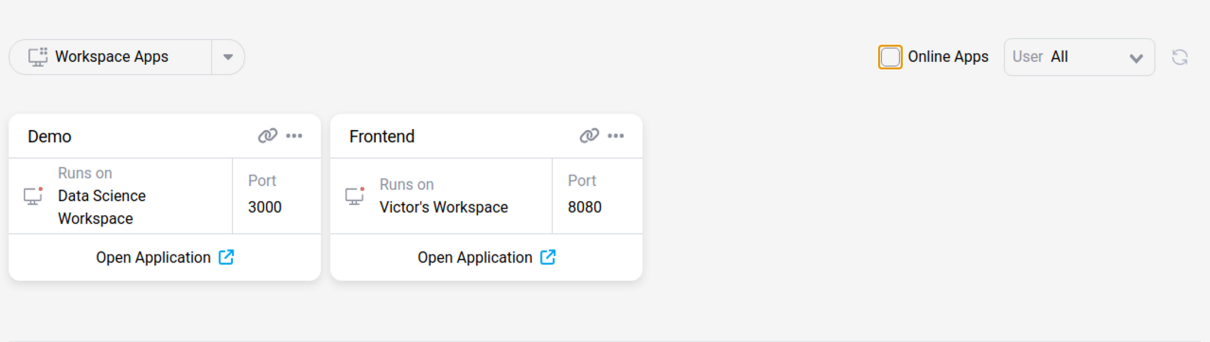

To only view your workspaces, select "**My Workspaces**".

- 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.

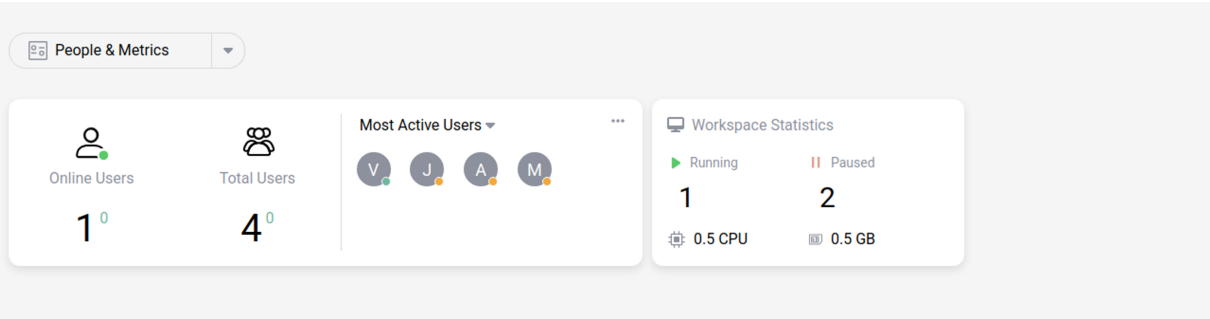



To only view your own, or any online workspace apps select "**My Apps**"or "**Online Apps**"respectively.

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

### People & Other Metrics

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




**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.

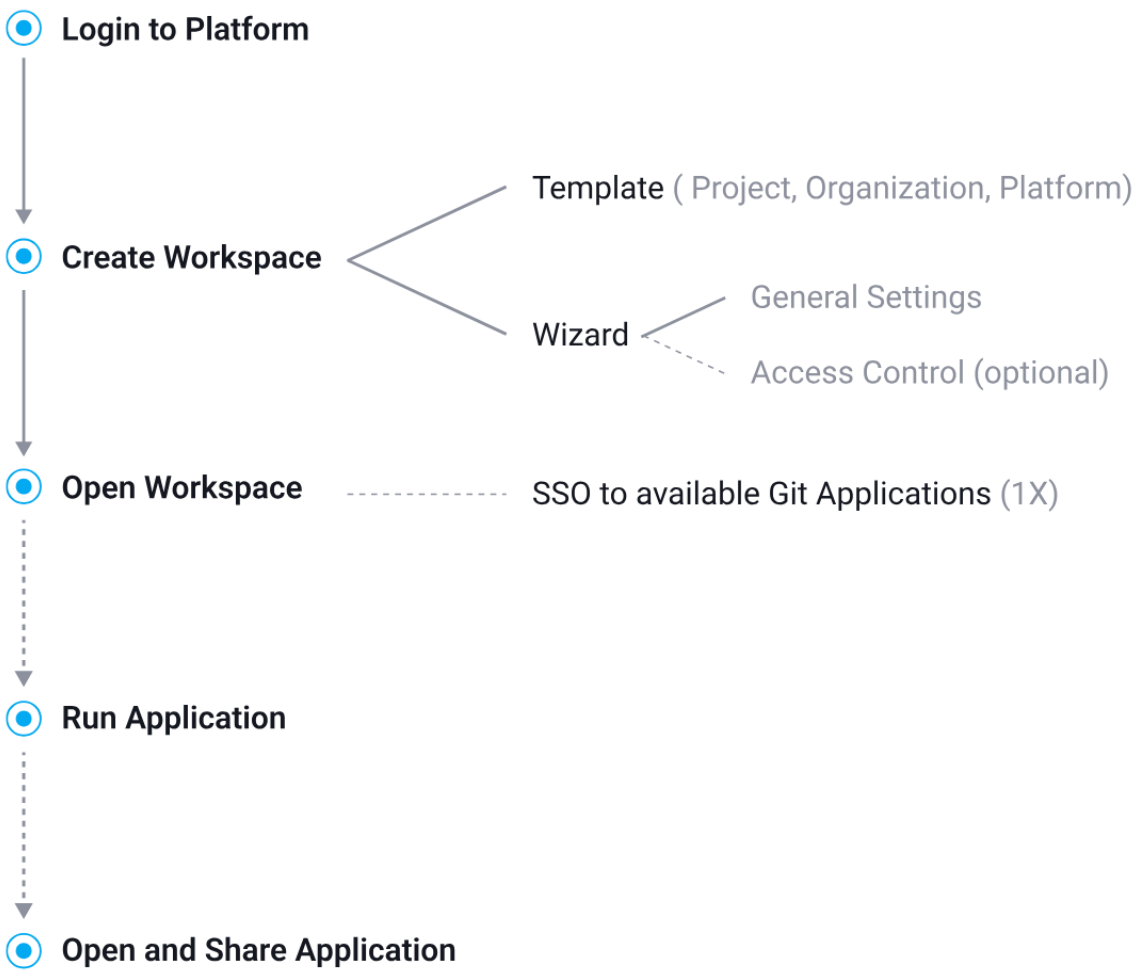
## Self-Served Developer

October 2, 2025

### Developer

This workflow exemplifies the most common onboarding case: a developer with the permission to create workspaces, i.e. a self-served onboarding process. This is typically an "internal" developer with permissions to access resources associated with the project, e.g. containers, services, secrets, etc. These resources are set up by the project owner and self-served developers are able to configure a workspace's access control setting.

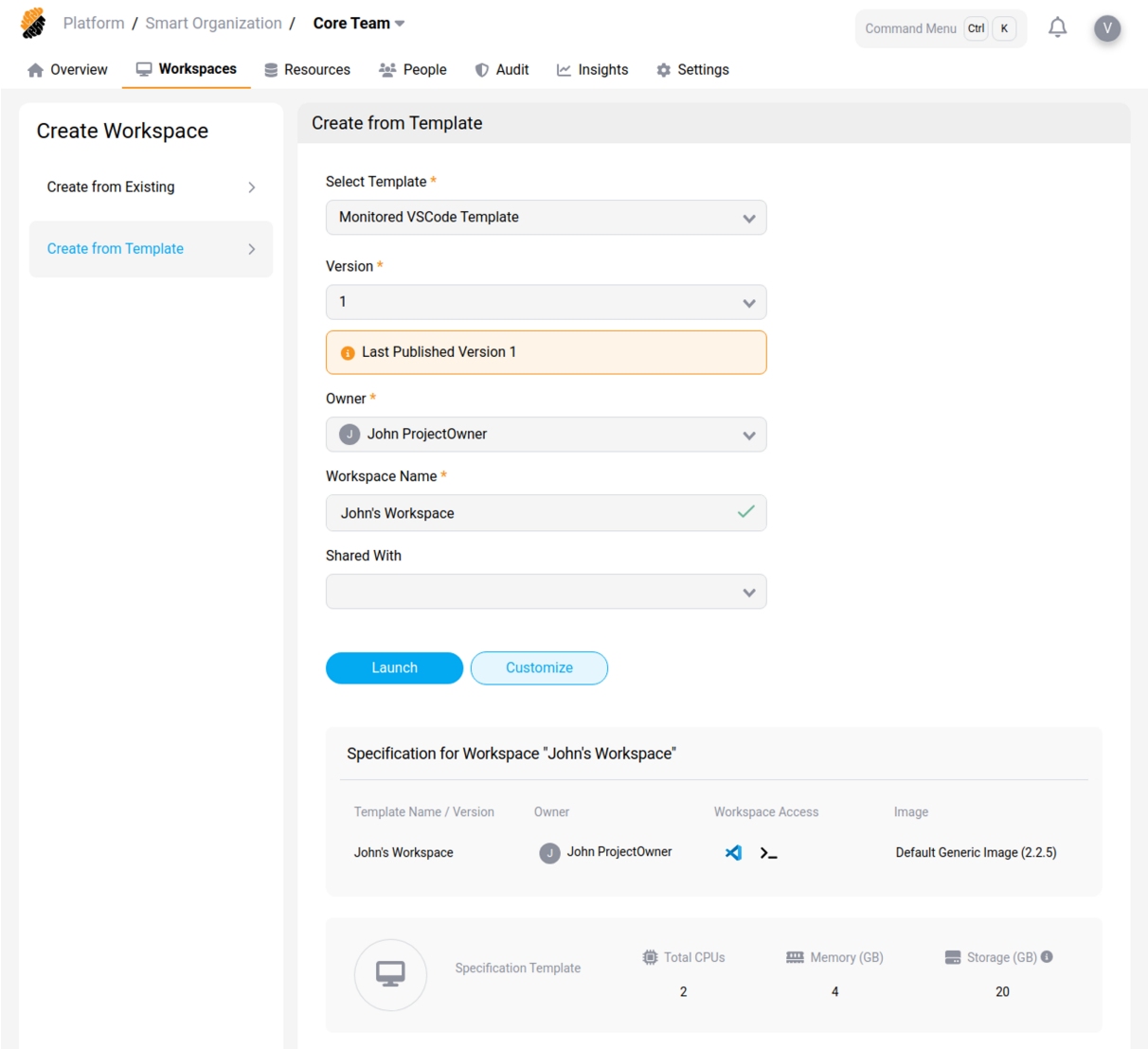



1. Log In & Create a Workspace
2. Configure Workspace Settings (Optional)
3. Access Workspace & Connect Platform Applications
4. Run, Open and Share Applications (Optional)

### 1. Log In & Create a Workspace

After logging in –having already been added to a project on the platform –the developer can independently create a workspace. This can be done using one of the pre-defined templates available on the platform or by following a guided setup process.

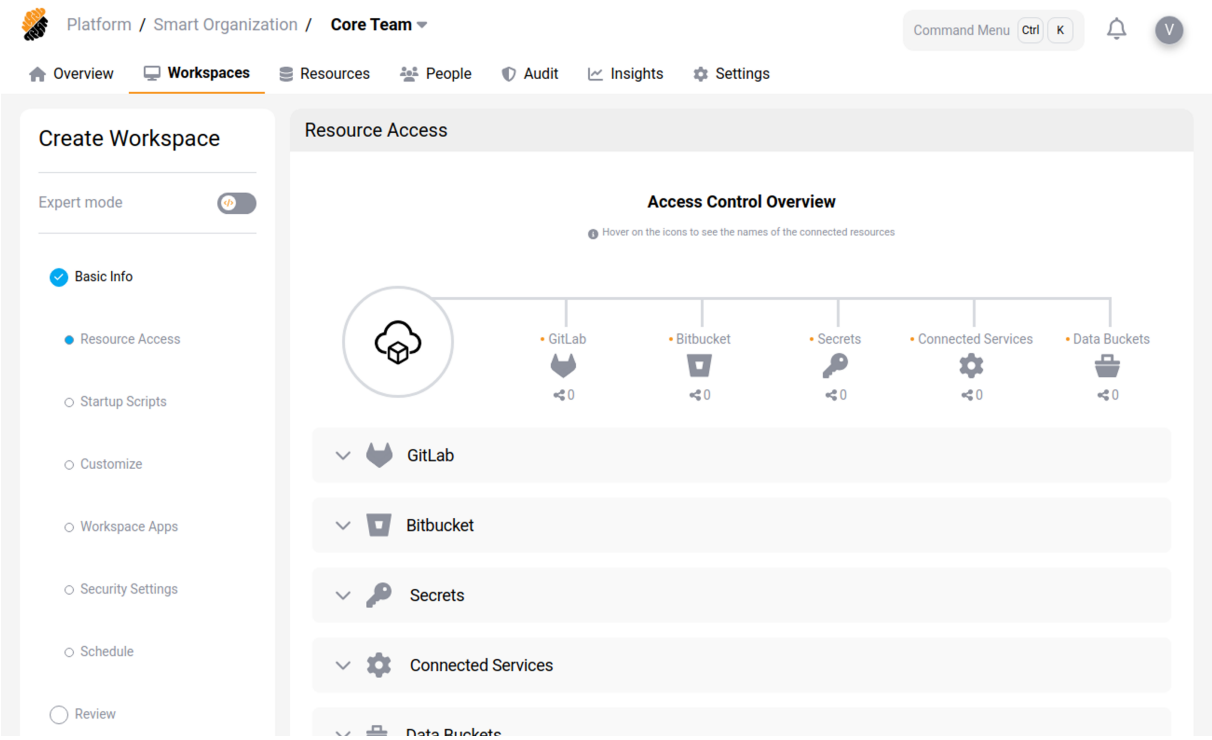



## 2. Configure Workspace Settings (Optional)

Through the guided setup (the wizard), the developer can configure the workspace's general settings, which include naming the workspace, selecting a specification template, and adjusting sharing preferences. Additionally, the developer can establish access controls to their entitled resources, covering options for git repositories, applications, services, and secrets.

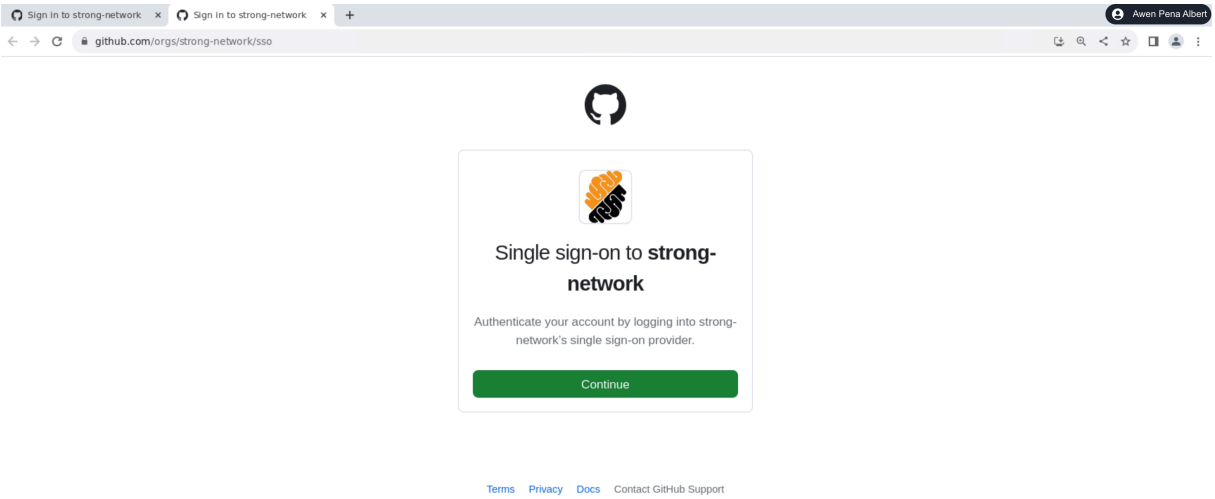
Implementing access control is not mandatory and can be addressed when the workspace is accessed for the first time.




## 3. Access Workspace & Connect Platform Applications

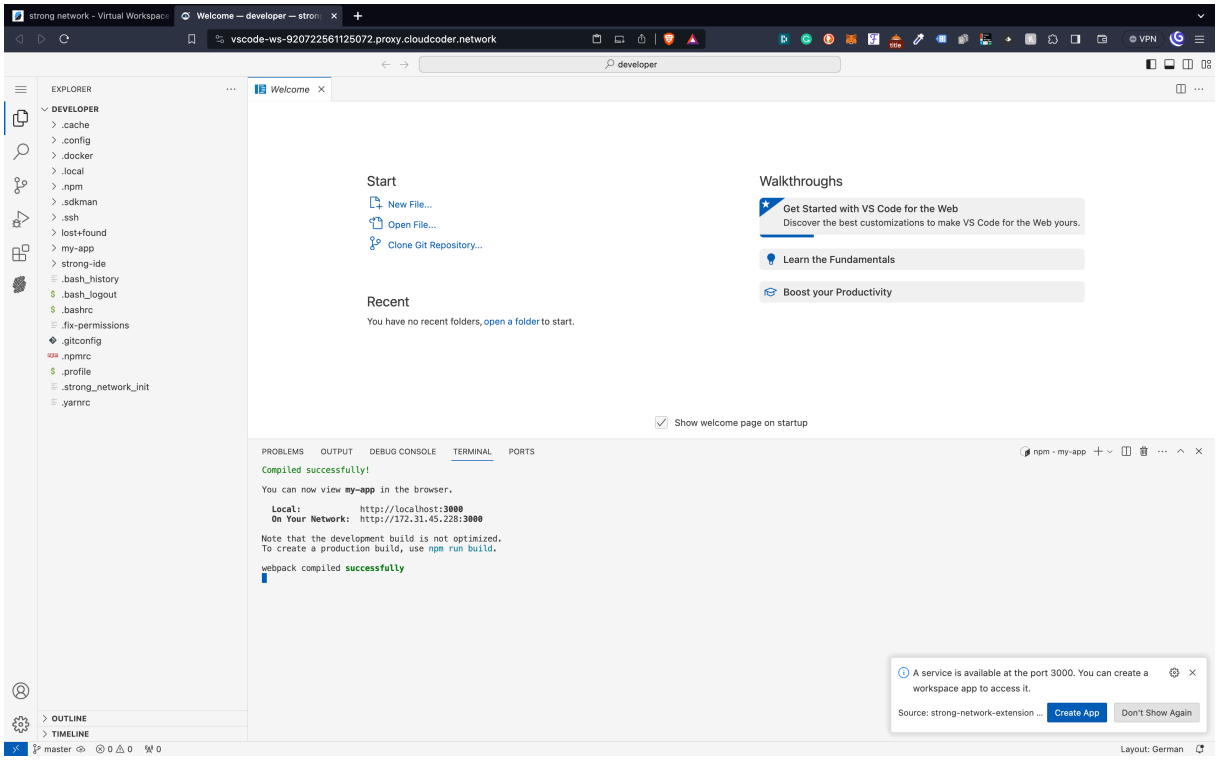
When first accessing a workspace, the developer may employ the single sign-on feature to gain entry to one or more gate applications linked to the platform, contingent upon the applications made available by the administrator.



### 4. Run, Open and Share Applications (Optional)

Once workspace access is secured, the developer is permitted to execute and, where authorized, access and share applications.



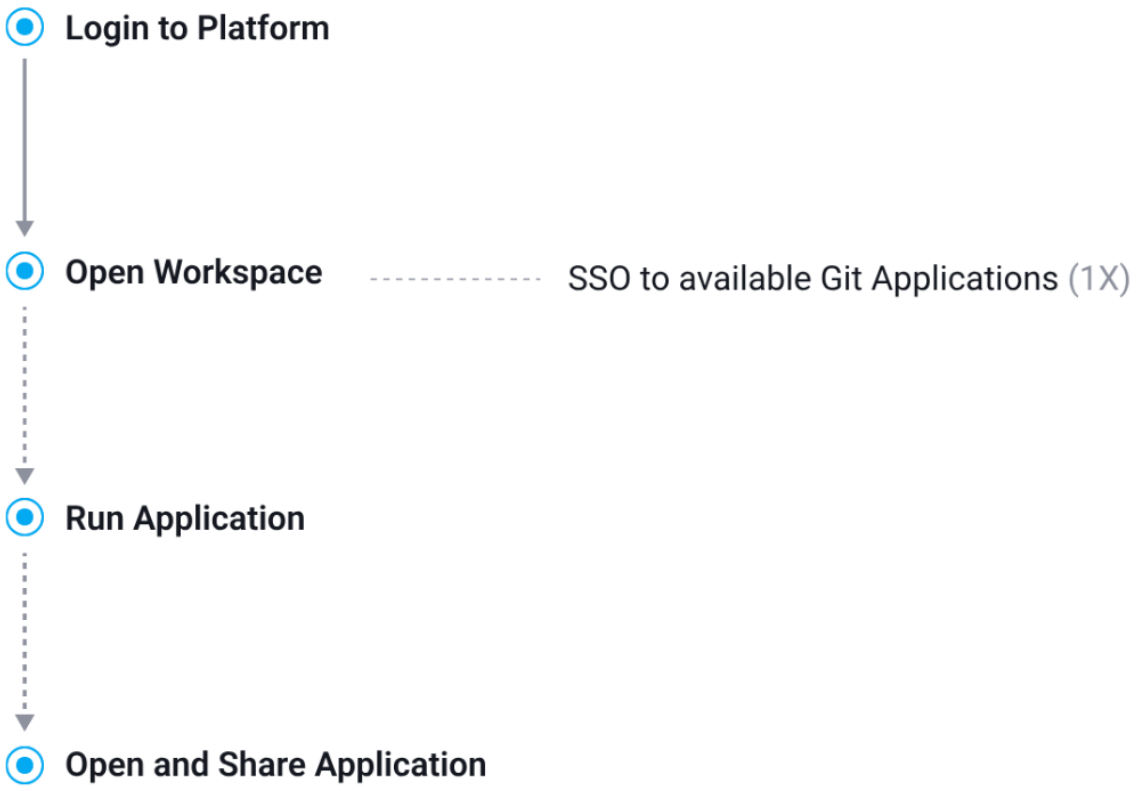

## Guest Developer

October 2, 2025

### Developer

This workflow exemplifies a particular onboarding case: a “guest”developer with permissions limited to access pre-configured workspaces, i.e. pre-set and immutable settings spanning resource access to security. This is typically a temporary developer, a contractor or an external collaborator. The entire workspace set-up is defined by the project owner and created in anticipation of onboarding the developer. Expectedly the developer cannot edit the workspace settings.

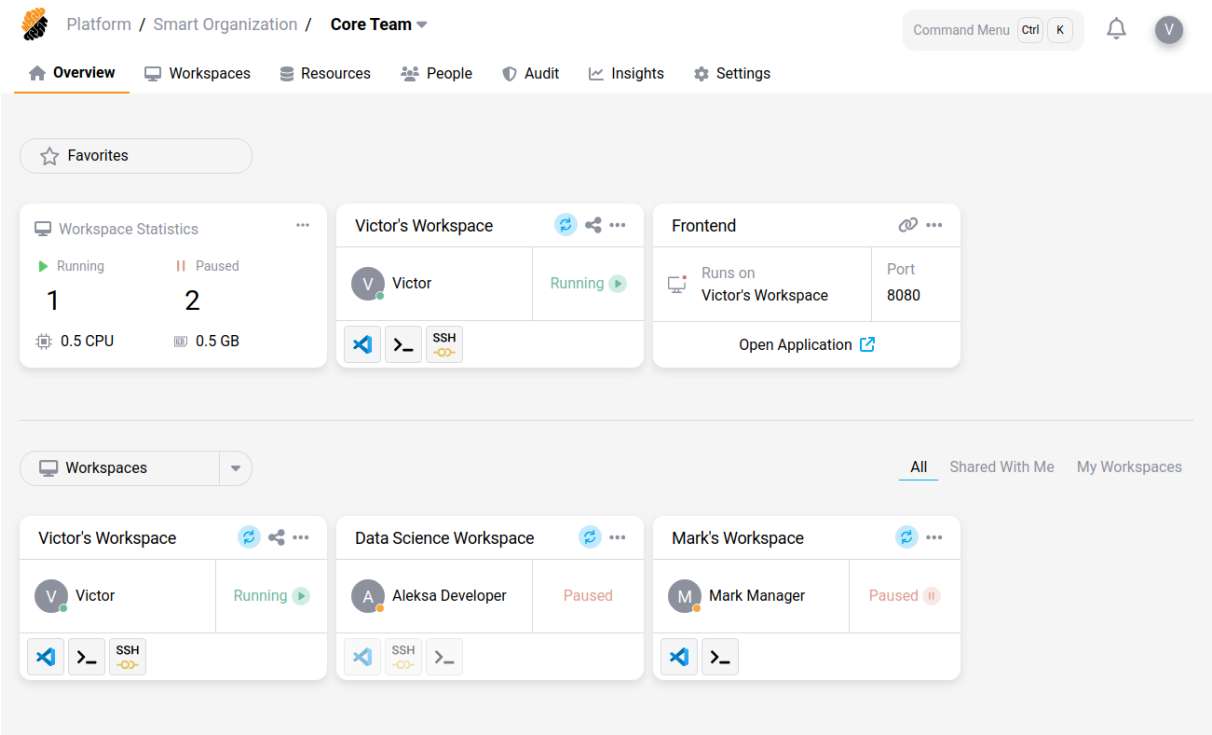



1. Log In & Access Workspace
2. Connect Platform Applications (Optional)
3. Run, Open and Share Applications (Optional)

### 1. Log In & Access Workspace

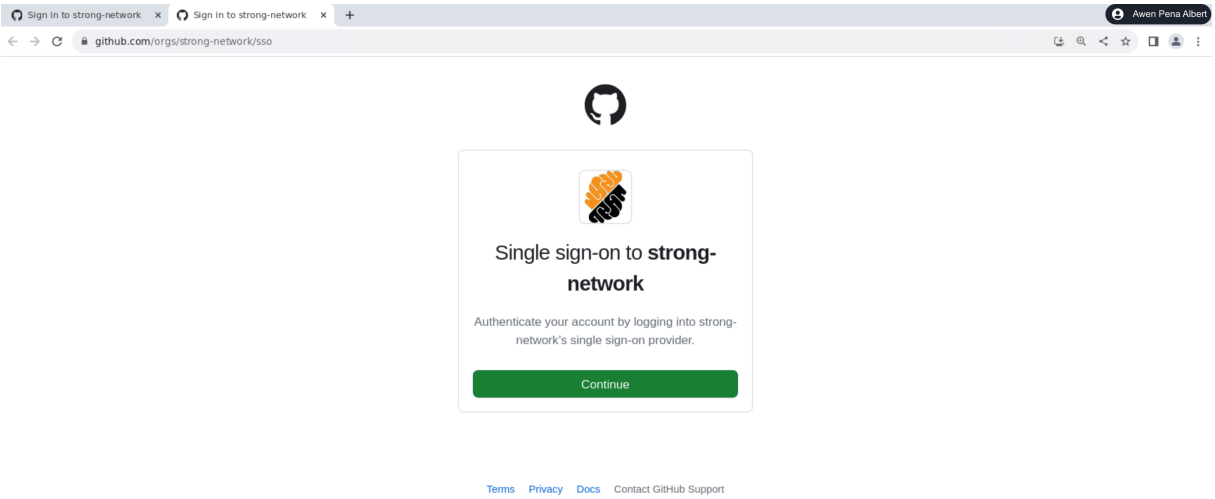
After logging in –having already been added to a project on the platform –the developer can access his assigned workspaces.




## 2. Connect Platform Applications (Optional)

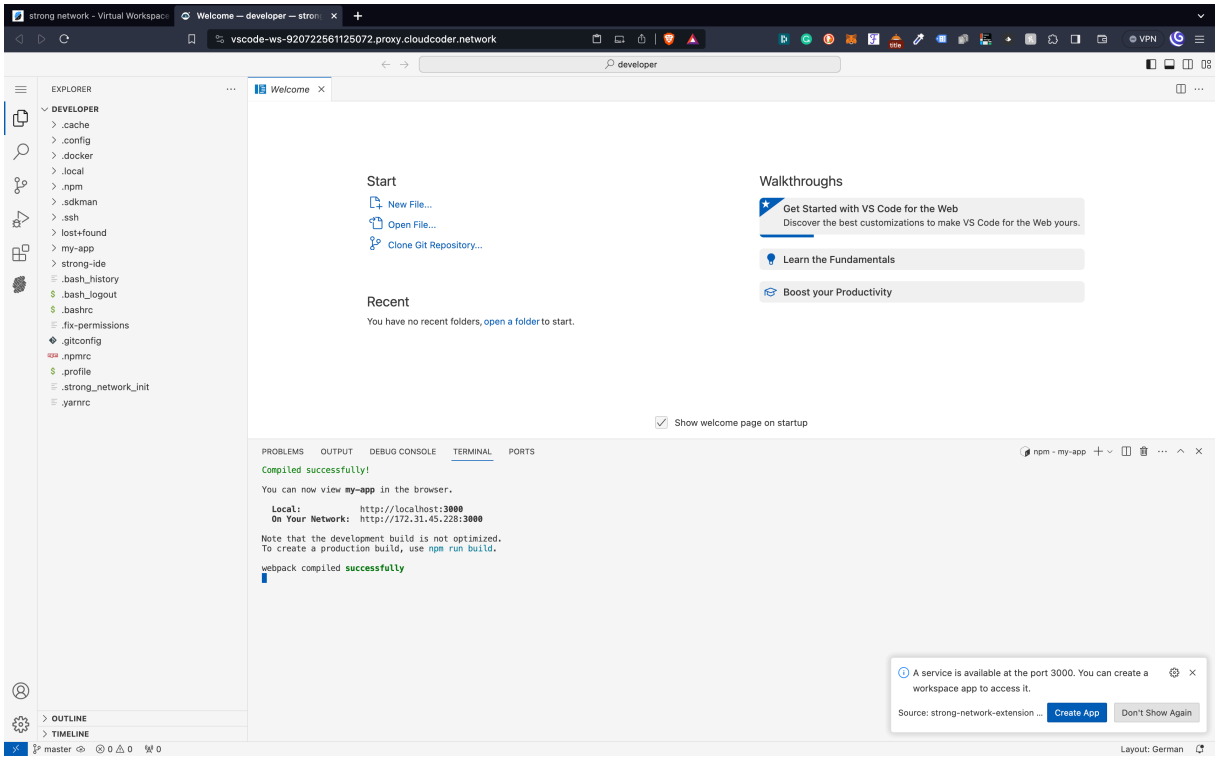
When first accessing a workspace, the developer may employ the single sign-on feature to gain entry to one or more gate applications linked to the platform, contingent upon the applications made available by the administrator.



### 3. Run, Open and Share Applications (Optional)

Once workspace access is secured, the developer is permitted to execute and, where authorized, access and share applications.



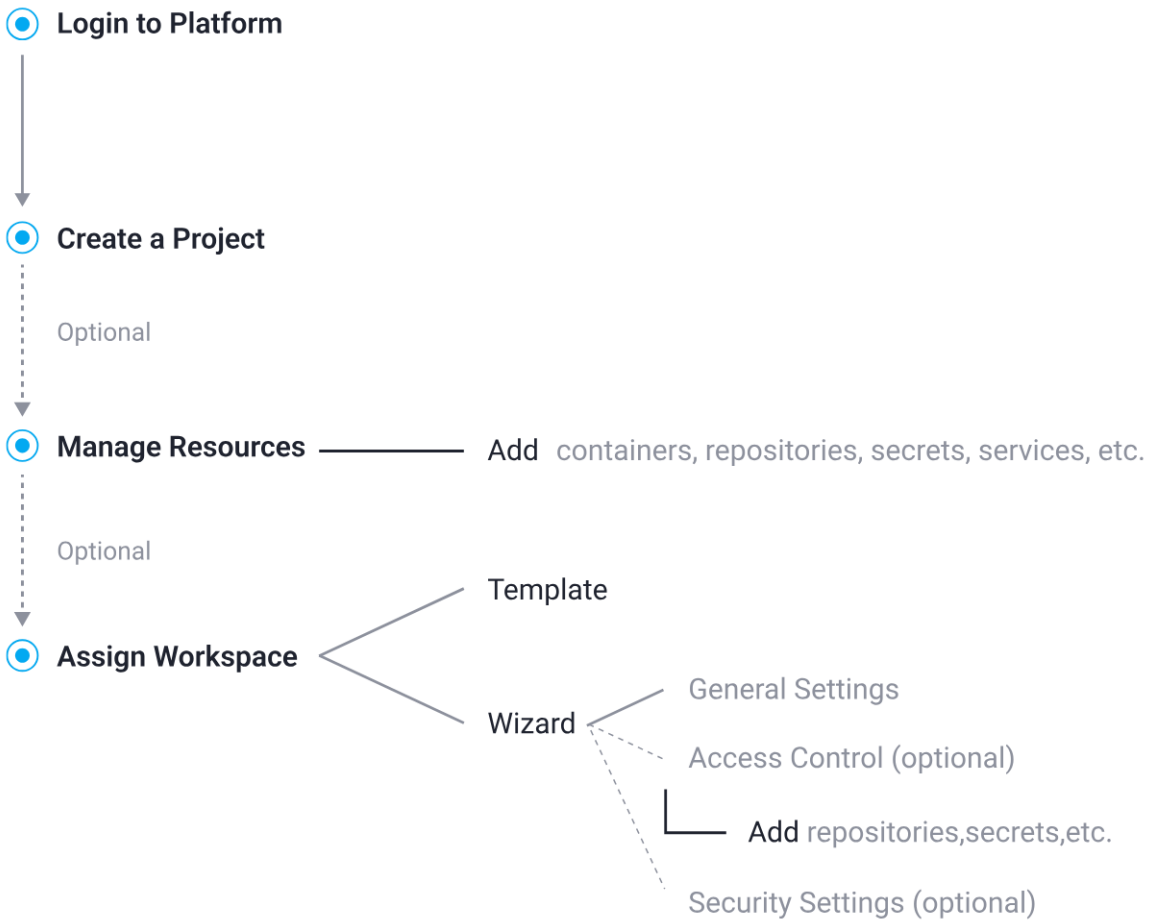

## Project Owner

October 2, 2025

### Project Owner

This workflow exemplifies the onboarding case of a project owner. Users with this role can create and edit settings of all the project's workspaces, including the workspace's access control and security settings. The project owner also creates workspaces for "guest"developers. In addition, he can manage resources for the project, such as importing containers, git repositories, secrets, etc.

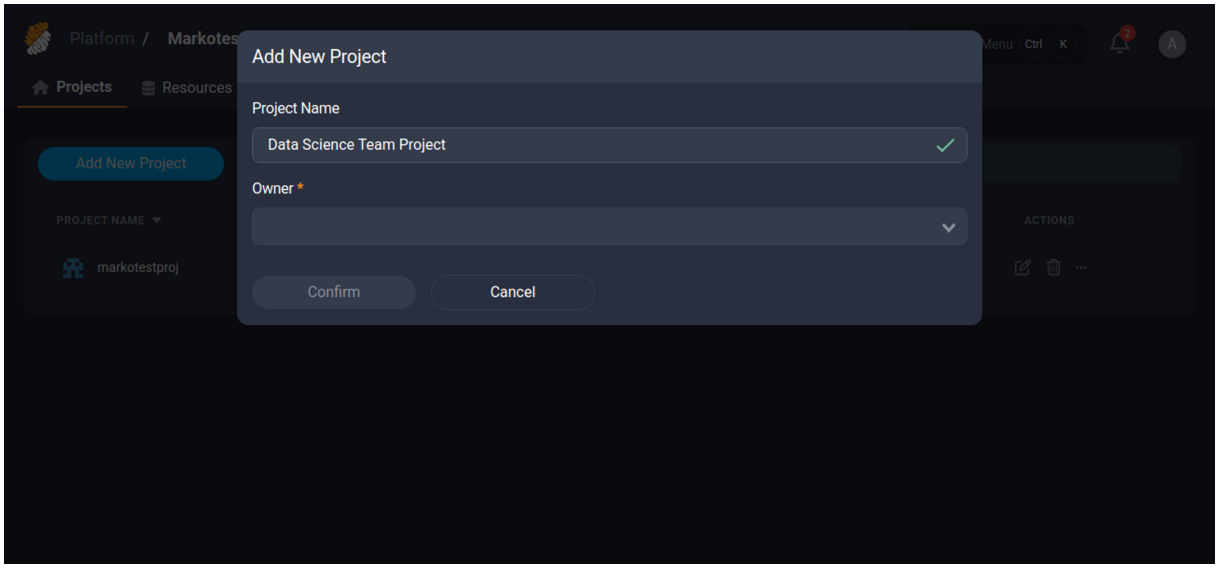



1. Log In & Create a Project
2. Manage Resources (Optional)
3. Assign a Workspace (Optional)
4. Configure Workspace Settings (Optional)

### 1. Log In & Create a Project

Upon logging in –having been affiliated with an organization on the platform –the project owner is equipped to establish a project for their team.

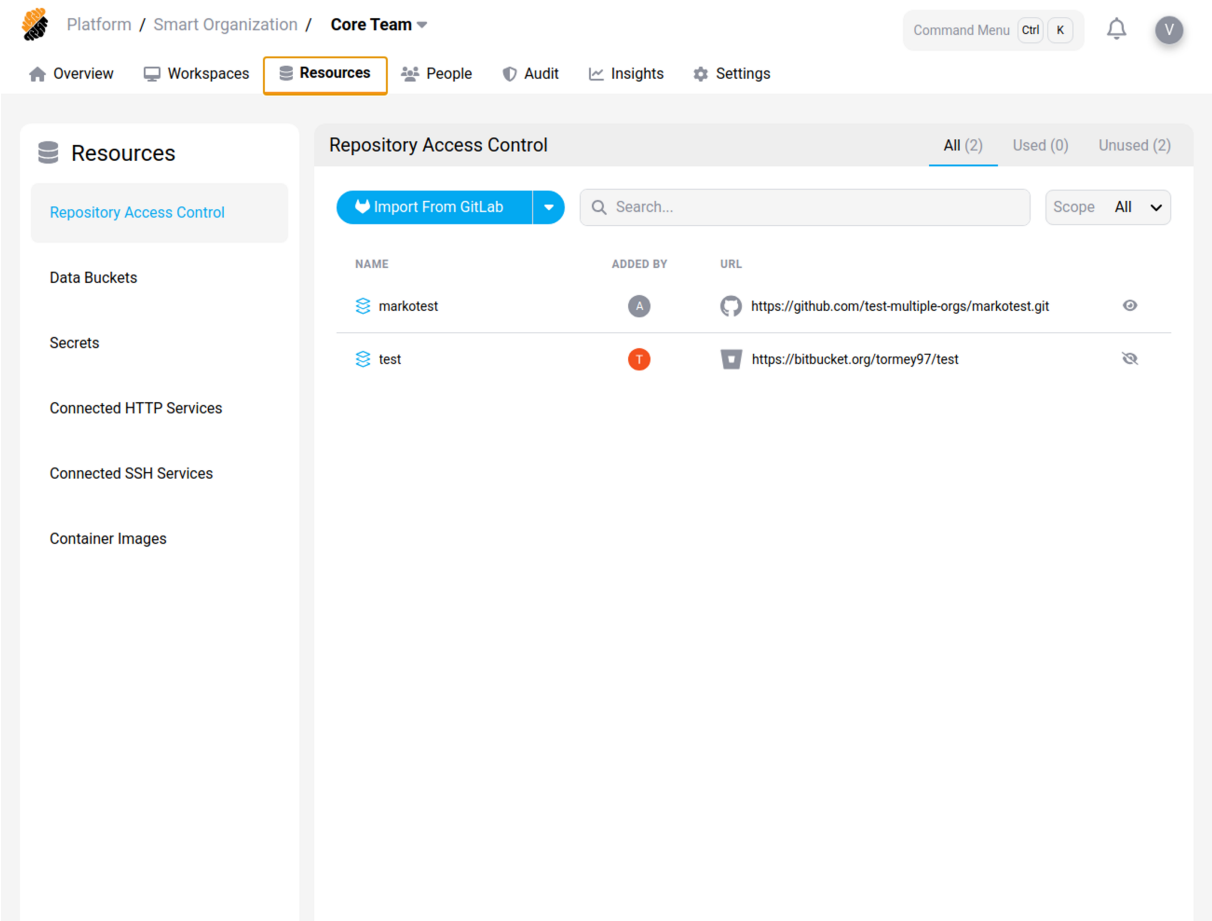



## 2. Manage Resources (Optional)

Additionally, a project owner can add and manage the resources leveraged by the development team.

Resources on the platform encompass code repositories, secrets, services, and data buckets. The project owner is responsible for determining user permissions, and stipulating who can view or alter resources to prevent unauthorized access.

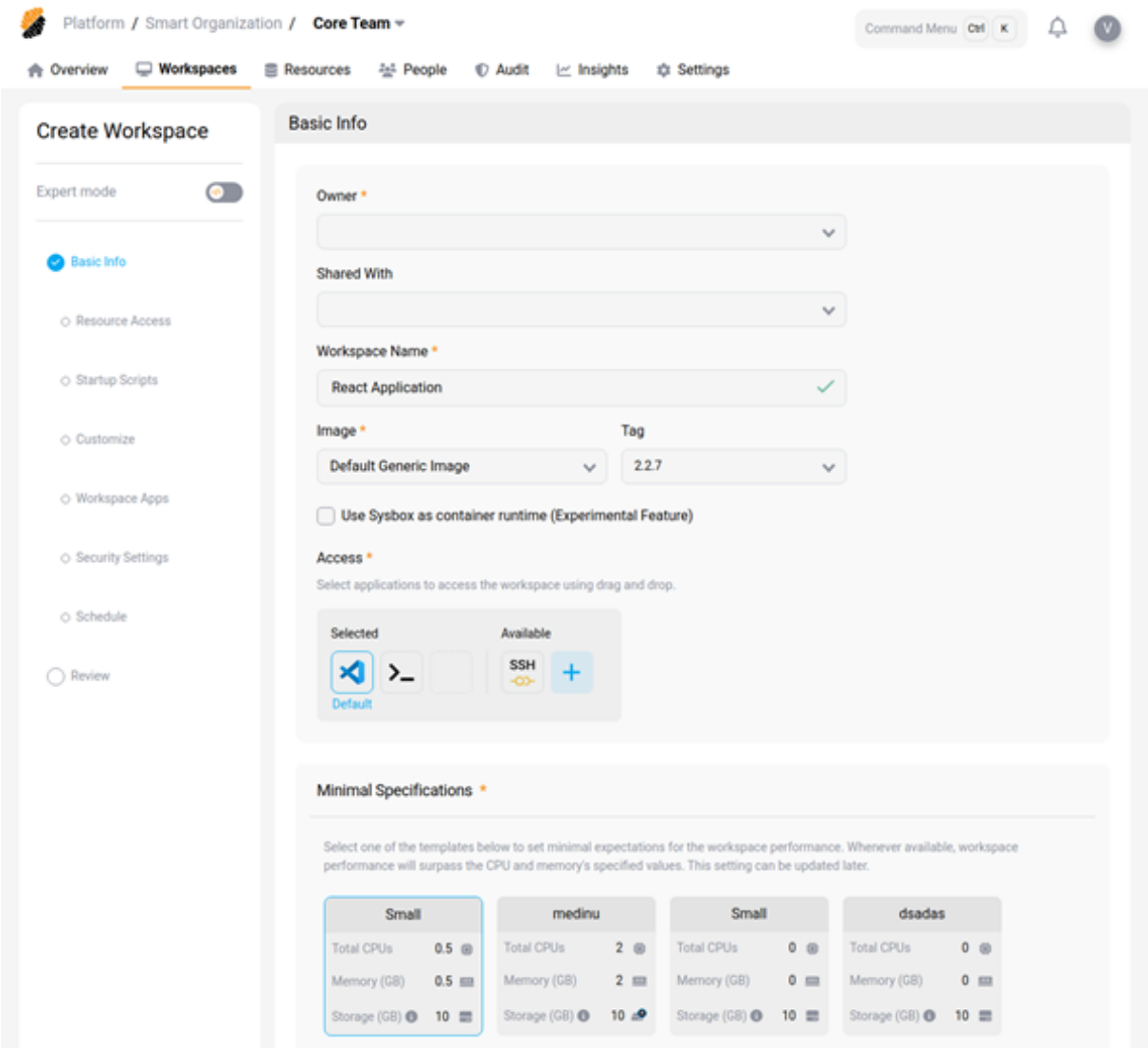



### 3. Assign a Workspace (Optional)

The project owner can create and assign a workspace to any user, however since developers with the permission Workspace:Manage Personal create their own workspaces (self-service), a project owner most commonly creates workspaces for developers without this permission, i.e. in order to onboard freelancers and contractors under a lesser permission model.

Therefore, project owners will create a workspace with a template or the workspace wizard and assign it to a user who is not entitled create it by himself.

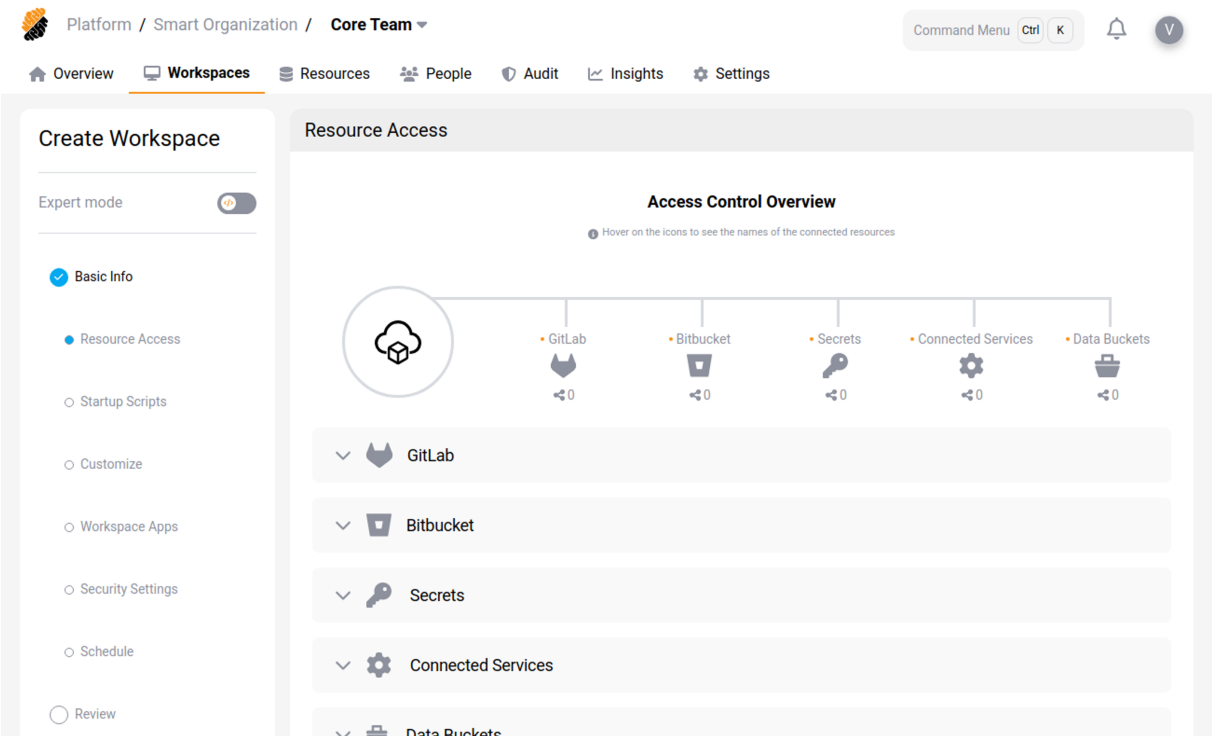
 




## 4. Configure Workspace Settings (Optional)

When the project owner creates a workspace on behalf of another user as explained in the previous section, he likely needs to set-up the access control and security settings. If the workspace is assigned to a user with the permission Workspace:Access (the user cannot create his/her own workspaces), the user won't be able to change the access control settings.




## What Is a Workspace?

October 2, 2025

A workspace is a Cloud Development Environments (CDEs) available for coding and data science. Workspaces can be accessed using a 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.

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

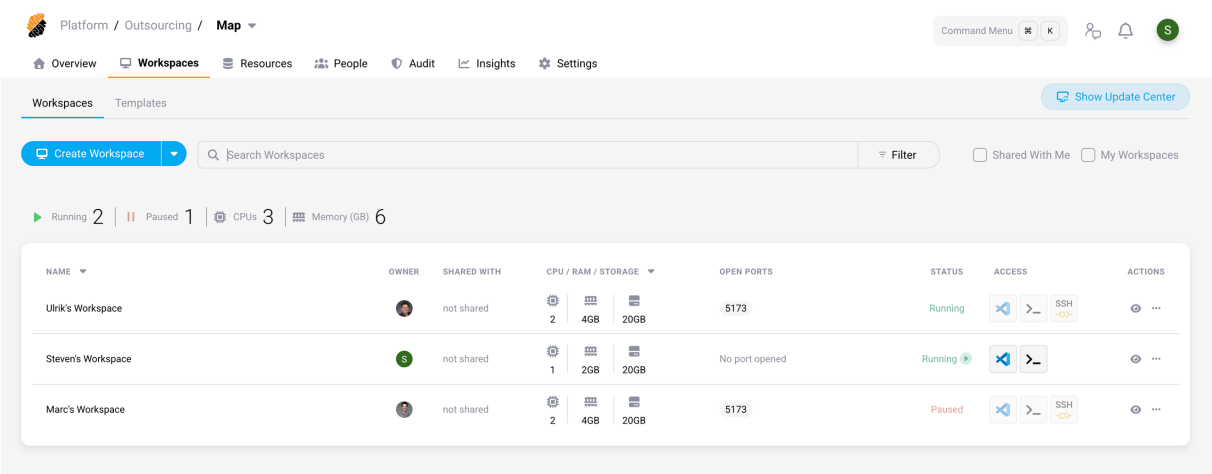
### Where to go next

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

## Workspaces Page

October 30, 2025

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.




### 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

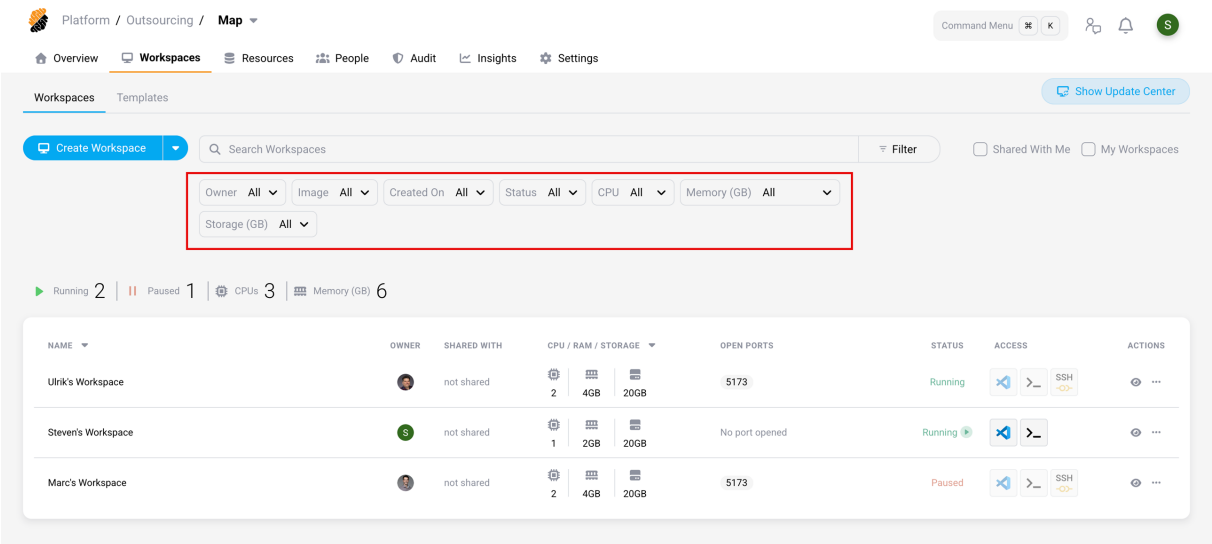
Below the search bar, you can view:

- 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



### Where to go next

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

## Create a Workspace

December 3, 2025

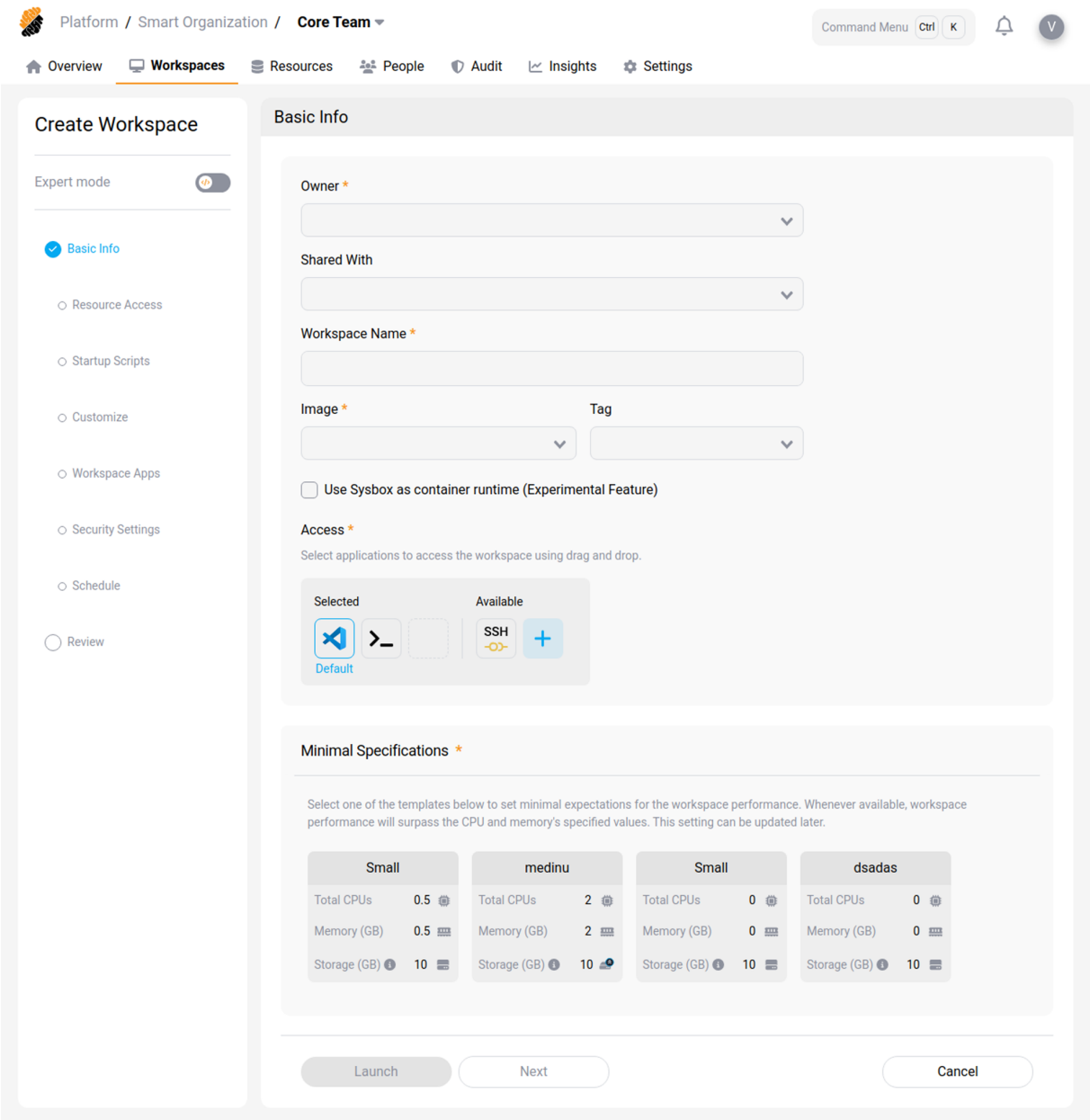
A workspace is created from the Workspaces Page. A workspace is, in essence, an online Cloud Development Environment (CDE) accessible via a Cloud IDE, a terminal or an SSH connection. Using an SSH connection is possible from a locally installed IDE supporting development from a remote container.

- Basic Set-Up
  - Basic info
  - Resource Access Control
  - Data Loss Prevention `Permission: Security::Manage`
  - Custom Work Schedule
  - Launch it
- From an existing Workspace
- From a template

### Basic Set-Up

You can create a workspace by pressing the "**Create Workspace**"button.



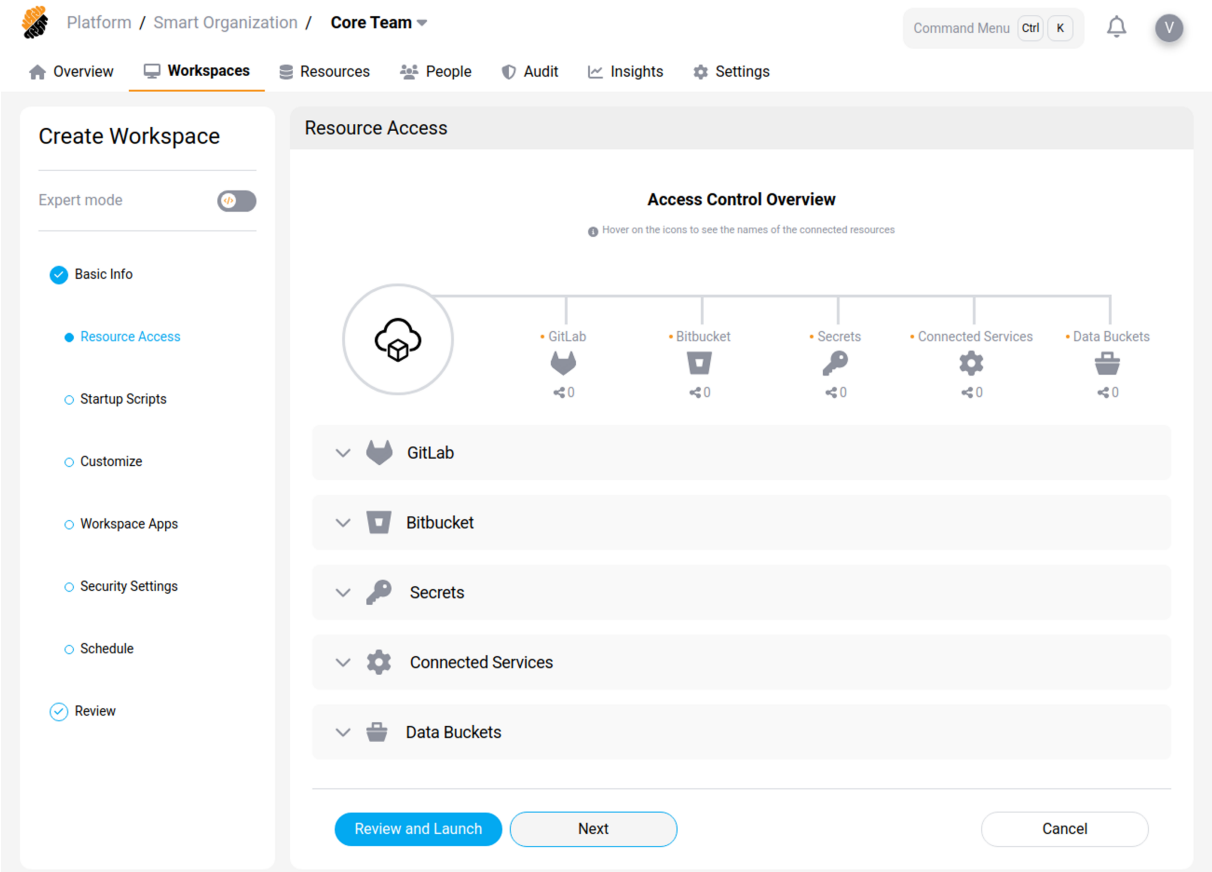

You will need to select the following information:

### Basic info

1. **Workspace Name**
2. **Embedded Cloud IDE**
3. **User Sharing Options**
4. **Docker Image**
5. **Image Version**
6. **Minimal Resource Specifications**

### Resource Access Control




You can attach various project resources to your workspace. Resources must have been previously added to the project. In addition, you might need the appropriate access rights to access them.

You can add the following resource:

- Git Applications And Repositories: You can connect the entire GIT applications available from your platform or single repositories that have been previously imported to the project's or organization's resources. Additionally, you can specify a default folder location within your workspace where the Git files will be cloned.
- Secrets: You can import secrets to the workspace as files or environment variables in the workspace. Choose from existing secrets or create a new one.
- Connected HTTP and SSH Services: You can connect services to appear as environment variables in the workspace. Supported and available services are part of the project's and organization's resources and depend on the platform's configuration.

### Startup Scripts

While the base container image (Dockerfile) provides core tools like languages and compilers, a startup script handles dynamic configurations. Because these configurations are often user-specific, they shouldn't be part of the shared image.
You can use a startup script to automate environment configuration every time the workspace launches and run it either pre-startup or post-startup, depending on your requirements. This ensures your workspace is ready for development immediately, without requiring manual setup.

Startup scripts are useful for tasks such as:

**Manage dynamic dependencies** Dependencies often change frequently or are specific to a branch, which makes them unsuitable for a static container image. You can use a script to:

- **Install dependencies:** Run commands like `npm install` or `apt update`. This ensures the environment has the latest libraries that match the code in your current branch.
- **Build binaries:** Compile the latest version of the application or helper tools so they are ready to run.

**Initialize services** A startup script can boot necessary background services that the container runtime doesn't automatically manage. Use the script to:

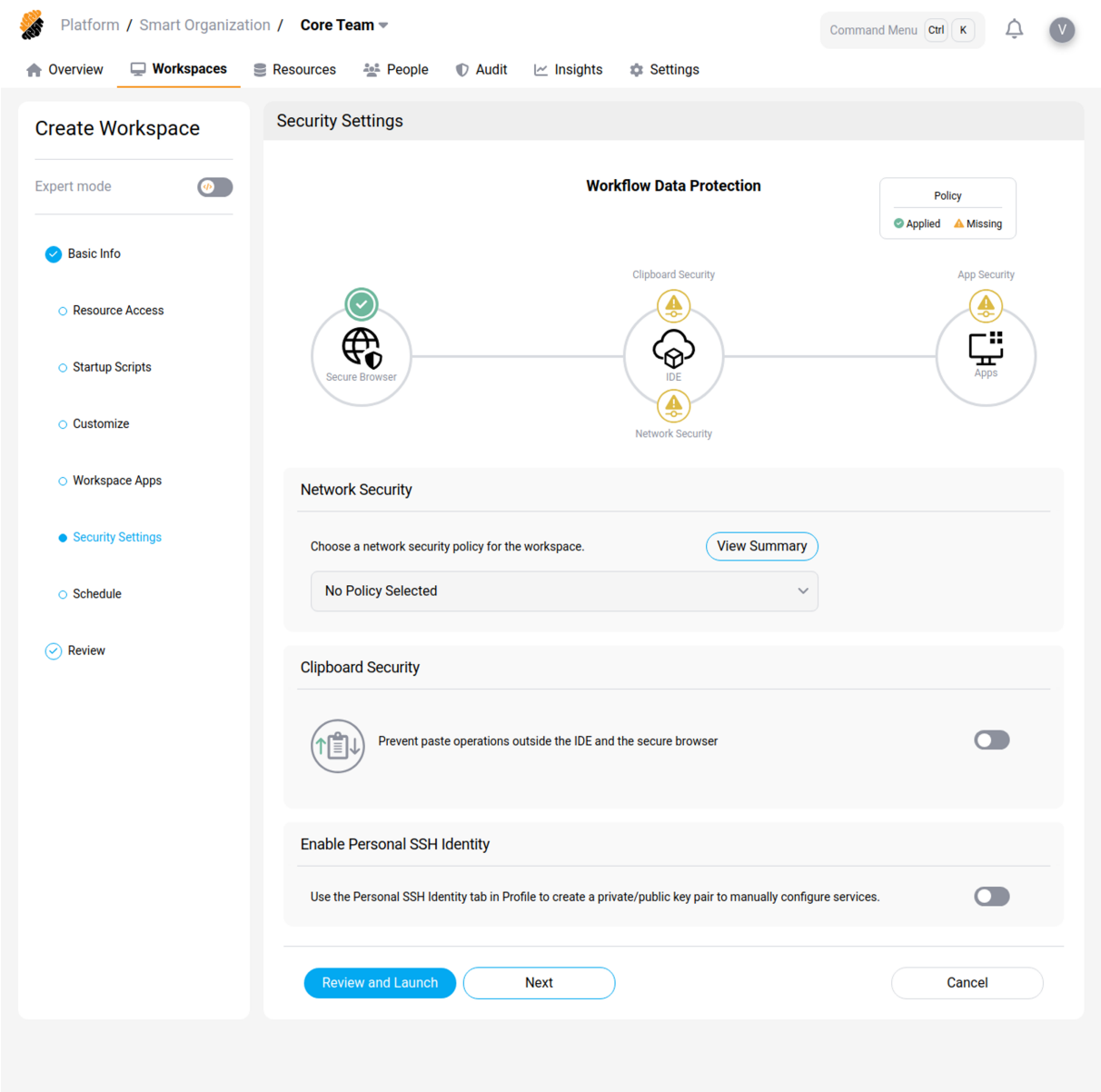
- **Start databases:** Launch local instances of services like PostgreSQL, Redis, or MongoDB if you need them for development.
- **Run daemons:** Start background processes, such as file watchers, test runners, or local servers.

**Run status checks** Scripts can provide feedback to let you know when the environment is fully ready. You can configure the script to:

- **Perform health checks:** Verify that all required services are running before giving you control of the terminal.
- **Print a welcome message:** Display a "Ready to code!"message or a list of available commands.

### Data Loss Prevention `Permission: _Security::Manage_`




In the Data Loss Prevention section you can configure the security of your workspace.

Under **Security Settings** you can configure:

- **Network Policy**: Select a network policy to enforce on the workspace. Network policies are part of the project's and organization's resources and are defined by the user with the Security::Manage permission. In particular, policies allow you to control outbound network traffic from the workspace.
- **Clipboard Security**: Prevent pasting outside of the IDE and the Secure Browser for this workspace.

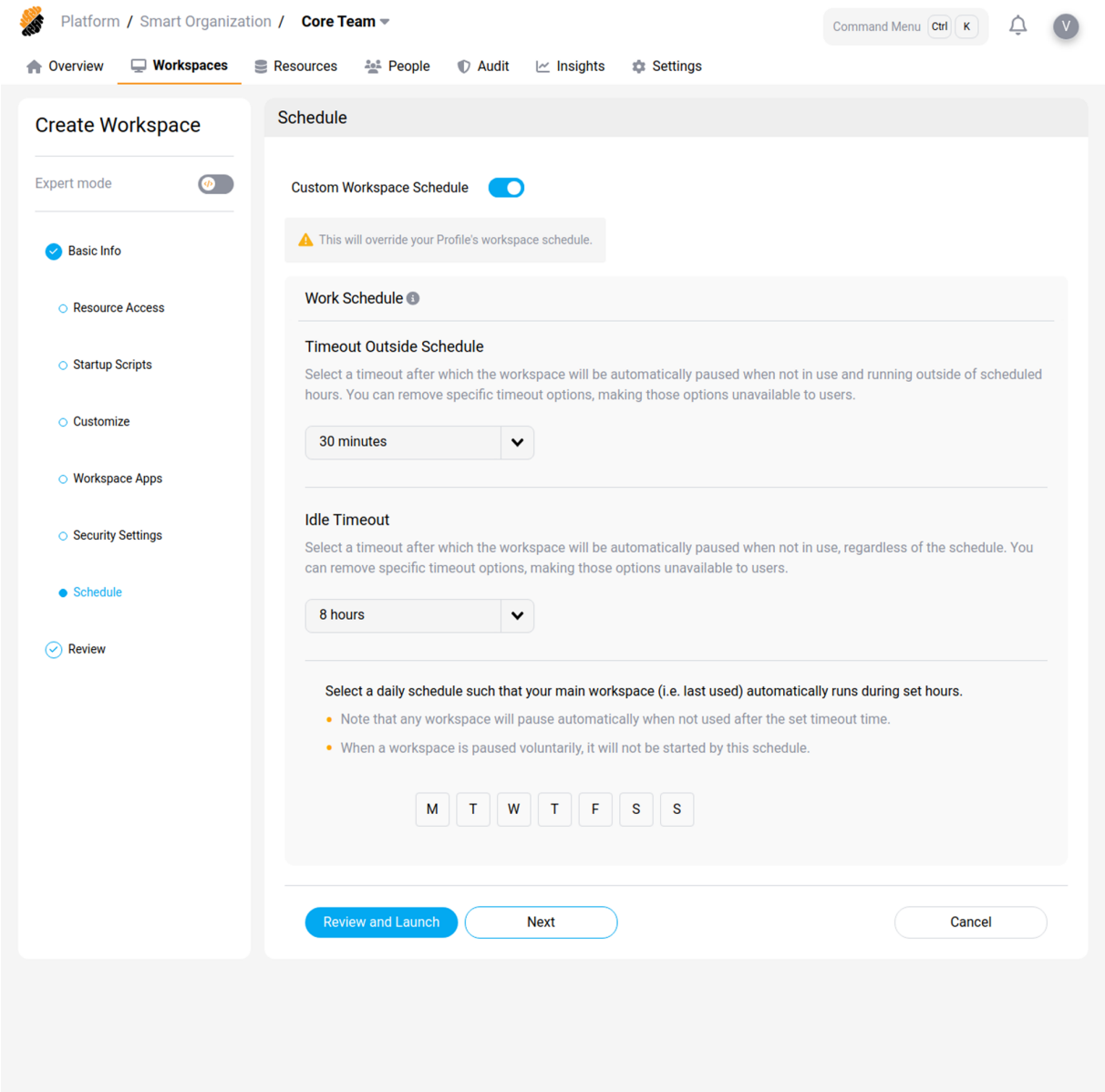
- **Apps Security**: Configure Workspace Apps to be accessed only through the Secure Browser.

Under **Secure Access Management** you can configure:

- **Enable Remote Development Over SSH**: Allow connection to the workspace via SSH.
- **Enable Personal SSH Identity**: Allow users to use their personal SSH identity from within the workspace.

### Custom Work Schedule

You can define a custom work schedule for your workspace.



#### Launch it

Finally, review your Workspace configuration, and launch it. Your workspace will be automatically deployed.
You can edit its configuration at any time from the Overview or Workspaces pages.

### From an existing Workspace

You can create a workspace from an existing one by pressing the "**Create from Existing**"button on the drop-down button of the "**Create Workspace**"button.

You will need to provide the following information:

1. **Workspace to Copy**
2. **Owner for the Workspace**

> **Tip**
>
> Click on "Customize"to edit the workspace as if you were creating it from scratch.
>
> Once done, press the "**Launch**"button.

### From a template

You can create a workspace from an existing one by pressing the "**Create from Template**"button on the drop-down button of the "**Create Workspace**"button.

You will need to provide the following information:

1. **Template Name**
2. **Owner for the Workspace**

> **Tip**
>
> Click on "Customize"to edit the workspace as if you were creating it from scratch.
>
> Once done, press the "**Launch**"button.

## Manage Workspaces

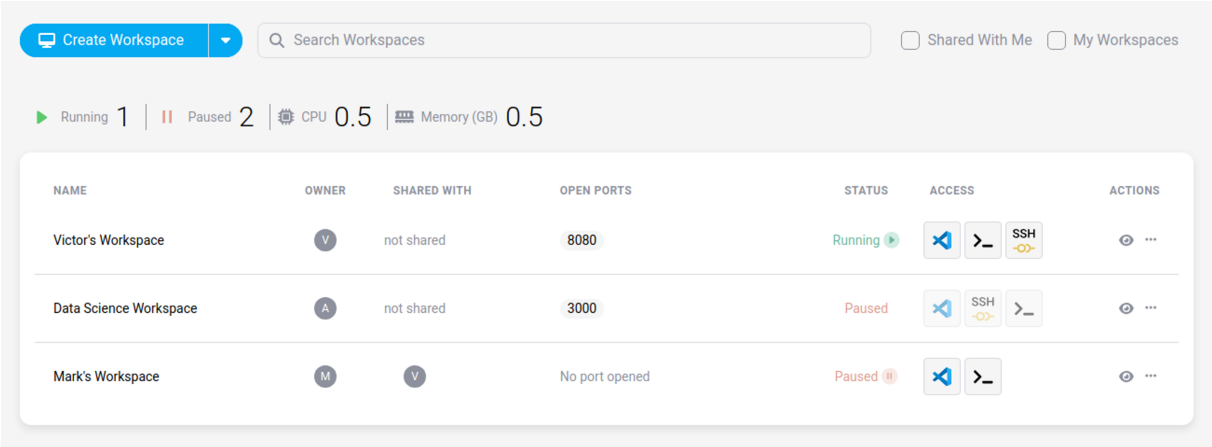
October 2, 2025

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.
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.

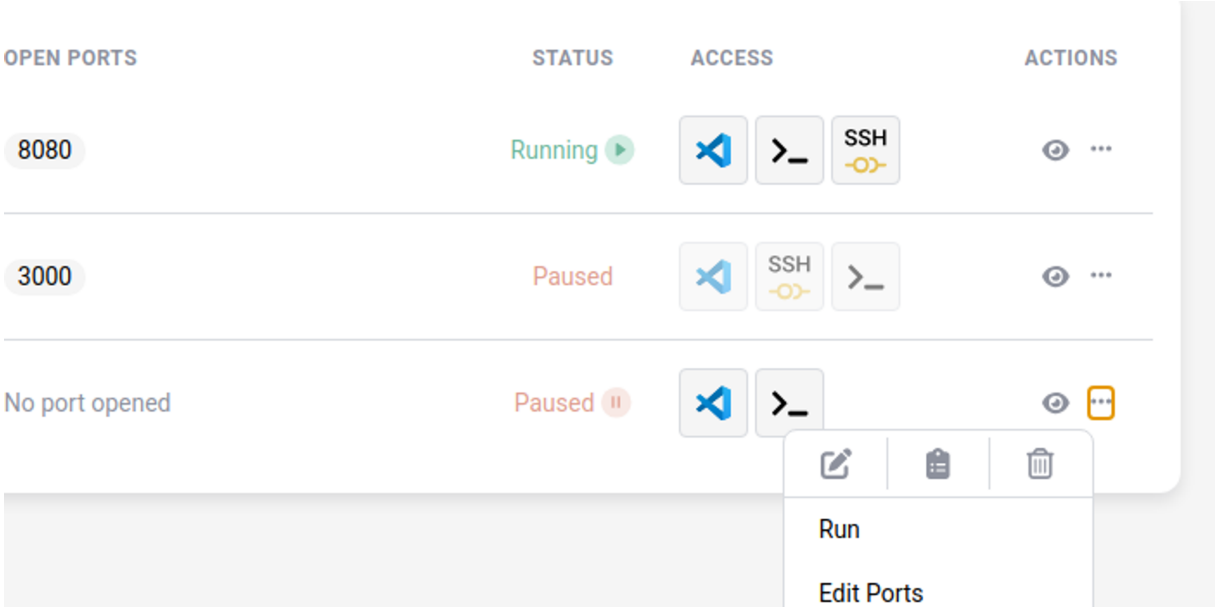
## Workspaces Actions

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::Manage Project* permission).




## Workspace Apps

October 2, 2025

A **workspace app** lets you access a local application running on a port of your Workspace. Depending on your workspace security's settings, your workspace app might open in a separate tab of your
web browser or in the Secure Browser. You can have multiple workspace apps attached to a single workspace, each accessing an application running on a different port.

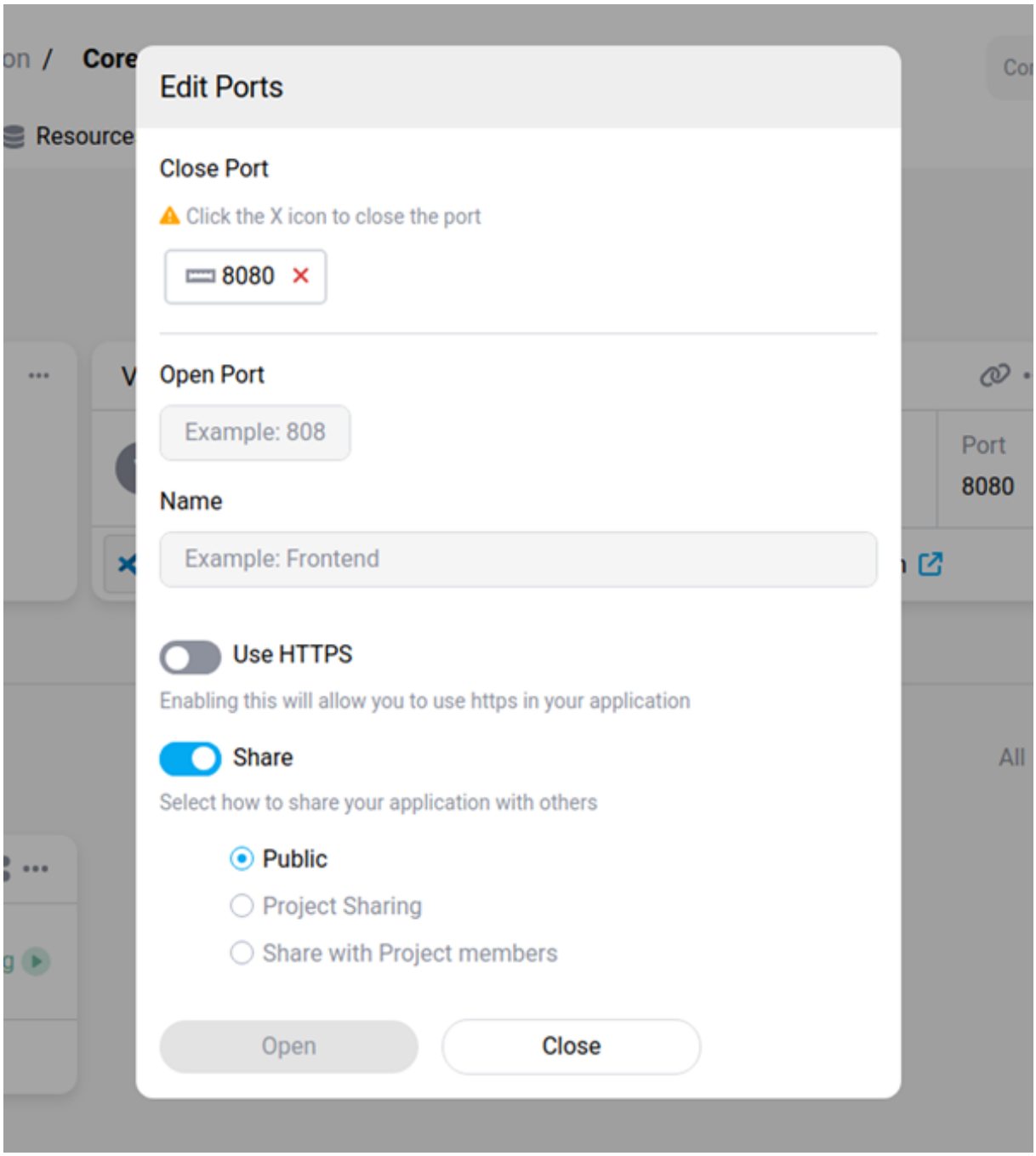
### Create a Workspace App

You can create a workspace app by pressing the "**Create Workspace App**"button from the Overview Page by selecting the drop-down menu next to "Workspace Apps".

> **Tip:**
>
> You can create a workspace app by clicking the "**...**"icon and select **Edit Ports** from a

> Workspace on the Overview or Workspaces Pages.




You will need to enter the following information:

1. **Port** where the app is running,
2. **Name** for the app,
3. **Use HTTPS** to allow to use https in the application,
4. **Share** to allow others to access the application (Public, Project Sharing or Share With Projectmate).

> **Tip:**
>
> When you create an app for a Node Js project, make sure the port number is the same as the one opened in the localhost of the workspace.

### 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, click the "**...**"icon and on the **Edit Ports** button from one of your workspaces.

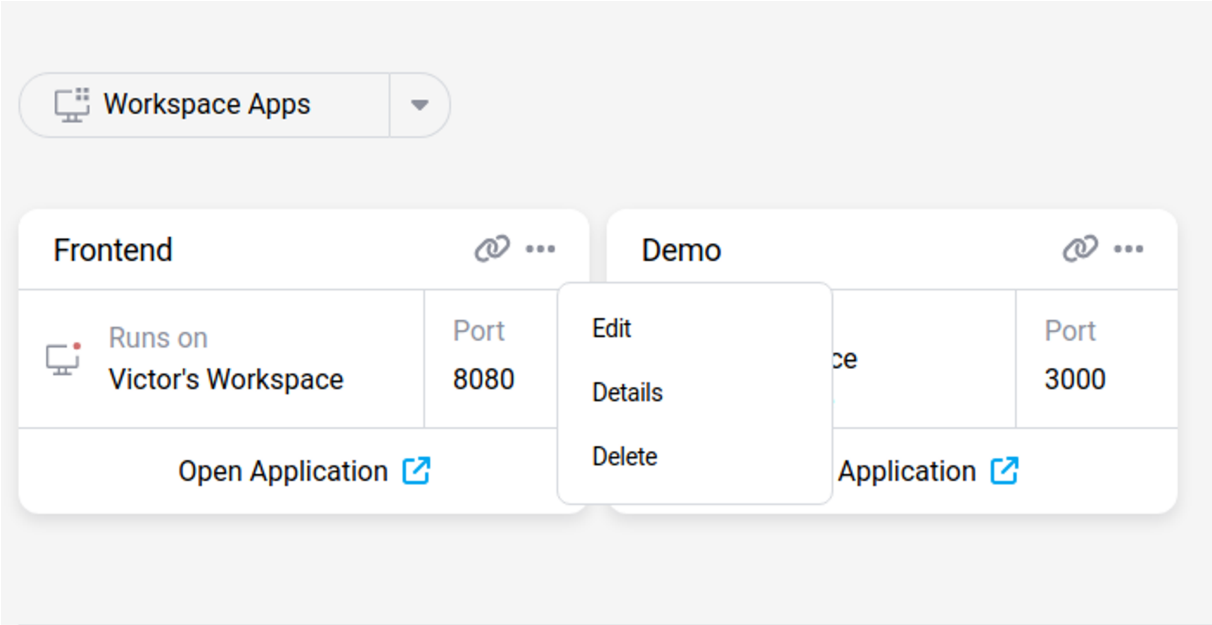
There are two sharing options:

- With a **group of users**: public (any external user) or project (all of your project-mates).
- With a **specific user**.

Granting access to one of your workspace apps does not provide access to the workspace running the app. To share a workspace with another user, check out the share a Workspace section.

### Delete a Workspace App

You can delete a workspace app from the Overview Page by pressing the "**...**"icon and clicking the **Delete** button.
You can also delete a workspace app from a Workspace by clicking the "**...**"icon and select **Edit Ports** from a Workspace on the Overview or Workspaces Pages.



# Templates

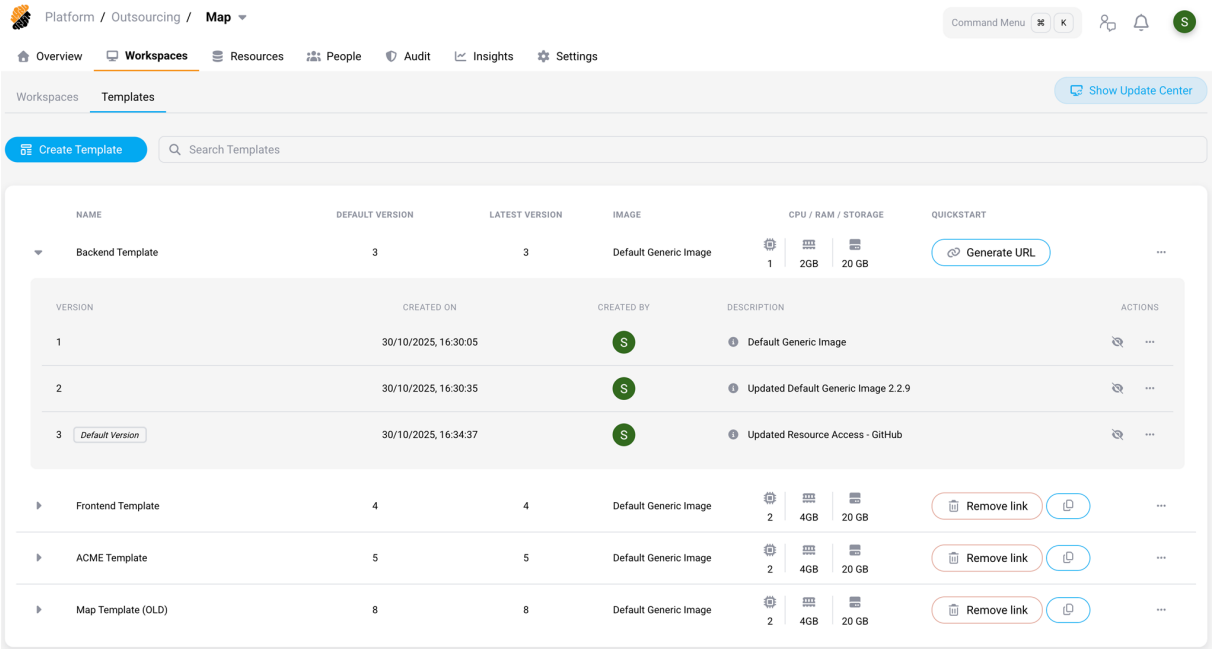
November 5, 2025

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 Templates
- Create a Template `Permission: Workspaces::Manage Project`
- Create a new version of a Template
- Quickstart
- Duplicate a template
- Create a Workspace from 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.




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 Templates

There are a few example templates provided in a standard project: Monitored VSCode, Restricted VSCode and Inspected VSCode. They are provided as examples with the characteristics below:

| Name | Image | CPU / RAM / Storage | Description |
|---|---|---|---|
| **Monitored VSCode Template** | Default Generic Image | 2 CPU / 4 GB / 20 GB | This is a standard template to create an instance of a fully-updated Ubuntu container with monitored traffic and clipboard. |
| **Restricted VSCode Template** | Default Generic Image | 2 CPU / 4 GB / 20 GB | This is a standard template to create an instance of a fully-updated Ubuntu container with restricted traffic with a series of exceptions (apt, npm, pip) and monitored clipboard. |

> **Warning**
>
> For the **Inspected VSCode Template**, applications using certificates in custom locations (folders) in the container will likely fail. Contact your administrator for more details.

### Create a Template `Permission: _Workspaces::Manage Project_`

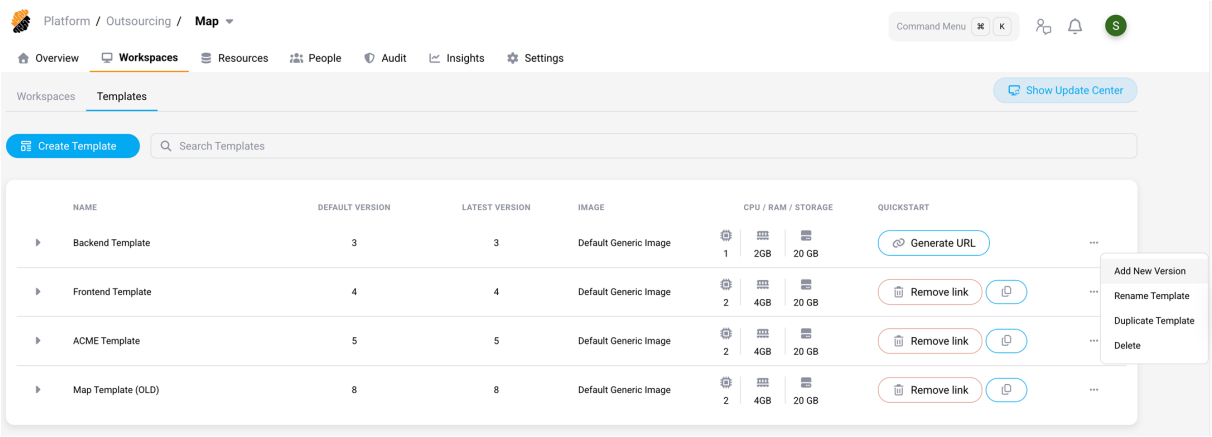
On the Workspaces Page, in the **Templates** section, you can create a template by clicking on the **Create Template** button.
You would follow the same steps as during the initial setup of a Workspace.

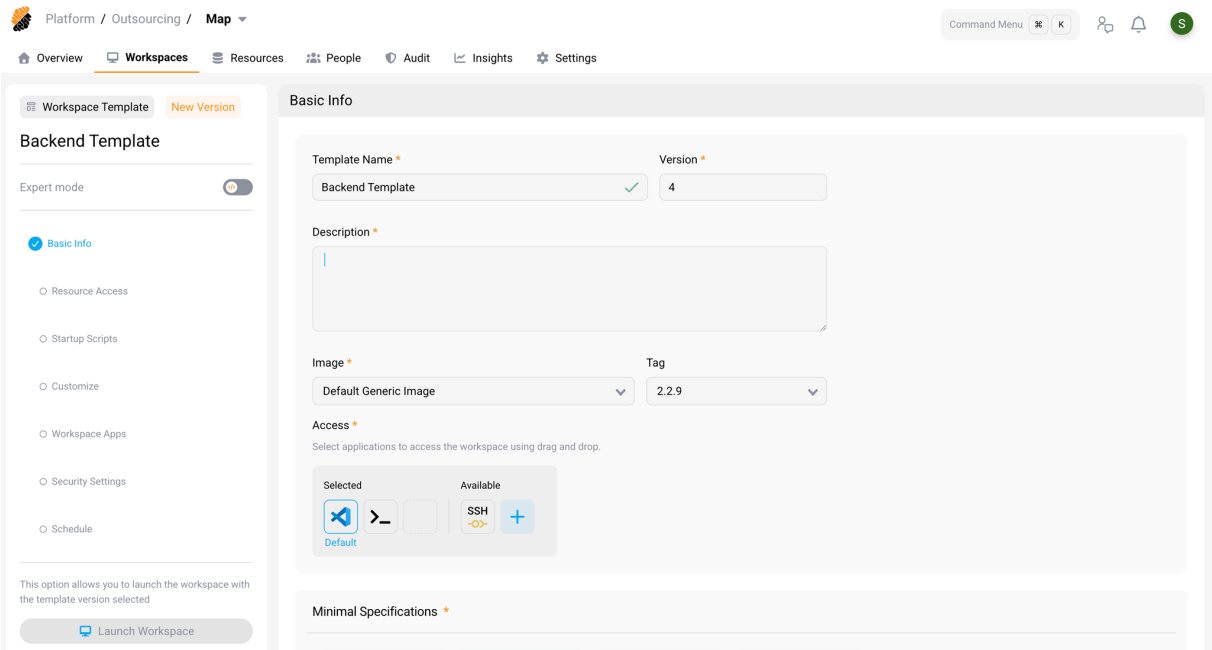
> **Tip**
>
> You can save a Workspace as a Template by clicking on the "…"button and on **Save As Template**.

## 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**.




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

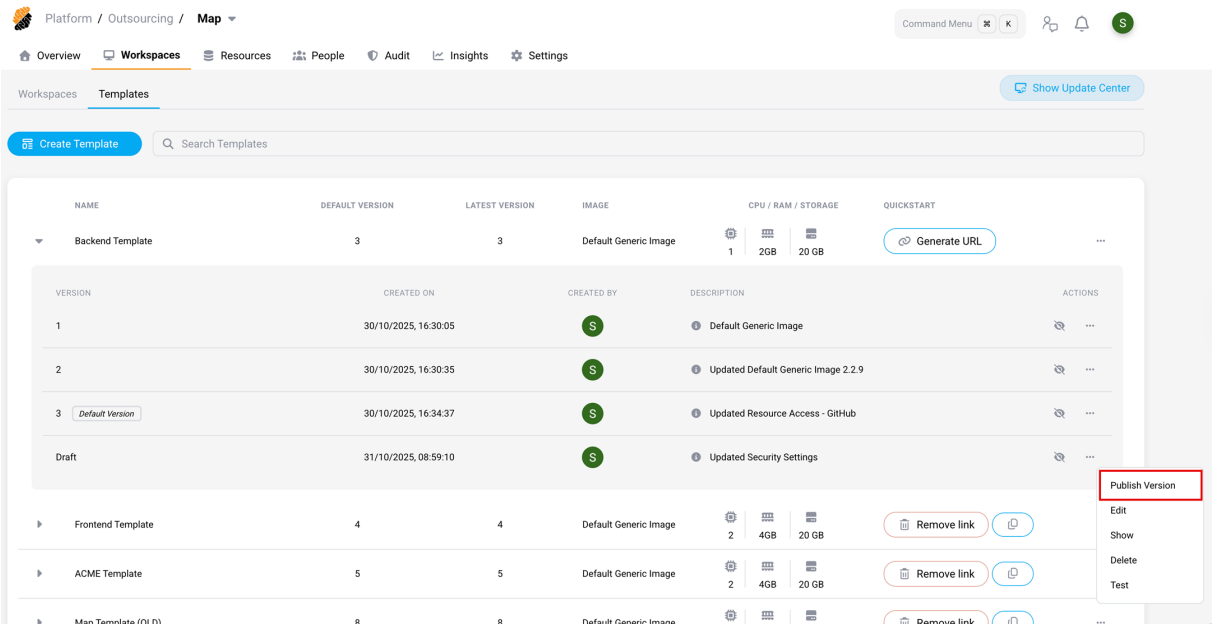
 

further modifications, or save it as a final template version right away, which cannot be changed afterwards.

A draft or new template version can be tested by either:

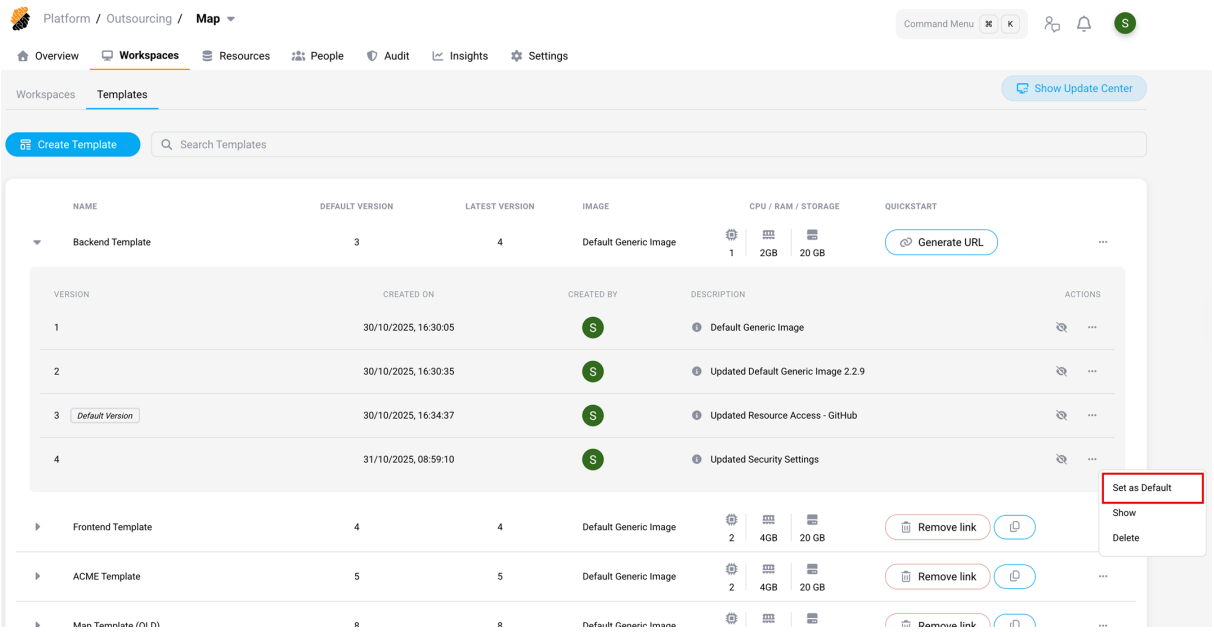
- Selecting **Launch Workspace** right within the template editor.
- Clicking on the "…"button on the right of a template and select **Test**.
- Manually selecting it from the list of version in the **Create Workspace from Template** wizard.

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**.




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**.

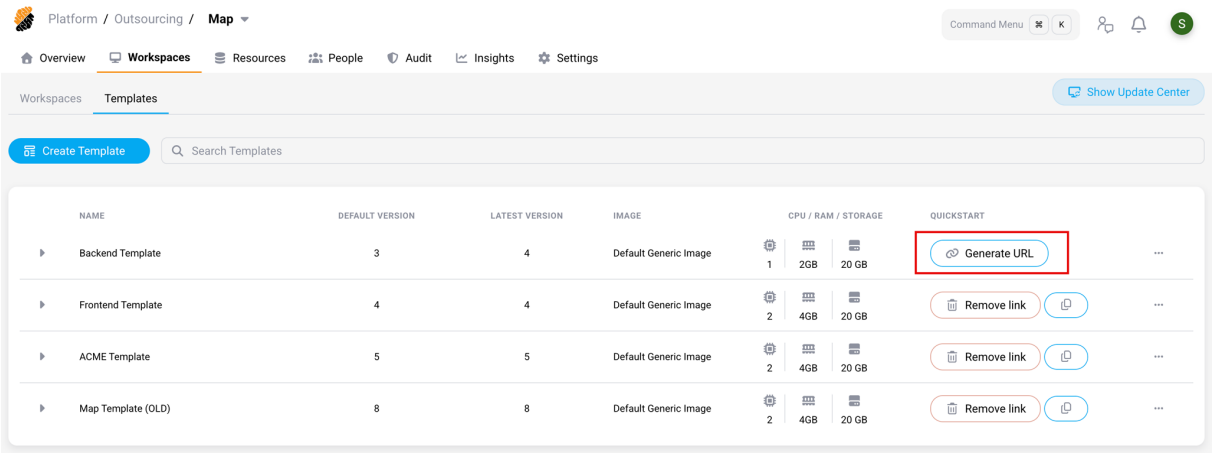
 




## Quickstart

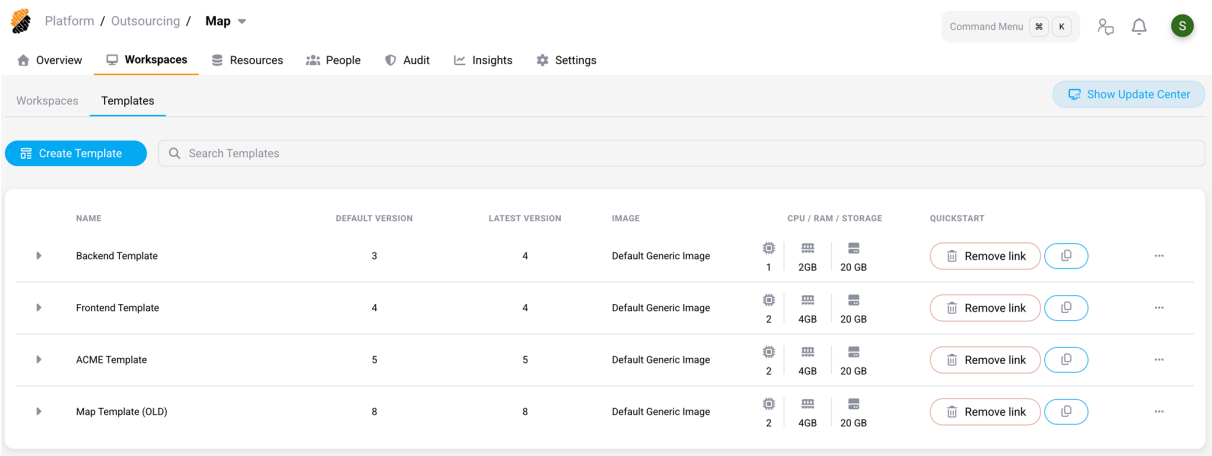
The Quickstart functionality allows developers to create a new workspace with a single click from a code repo, engineering portal or any other location outside of SDS.

Create a Quickstart link by clicking the **Generate URL** button on the right of a template.

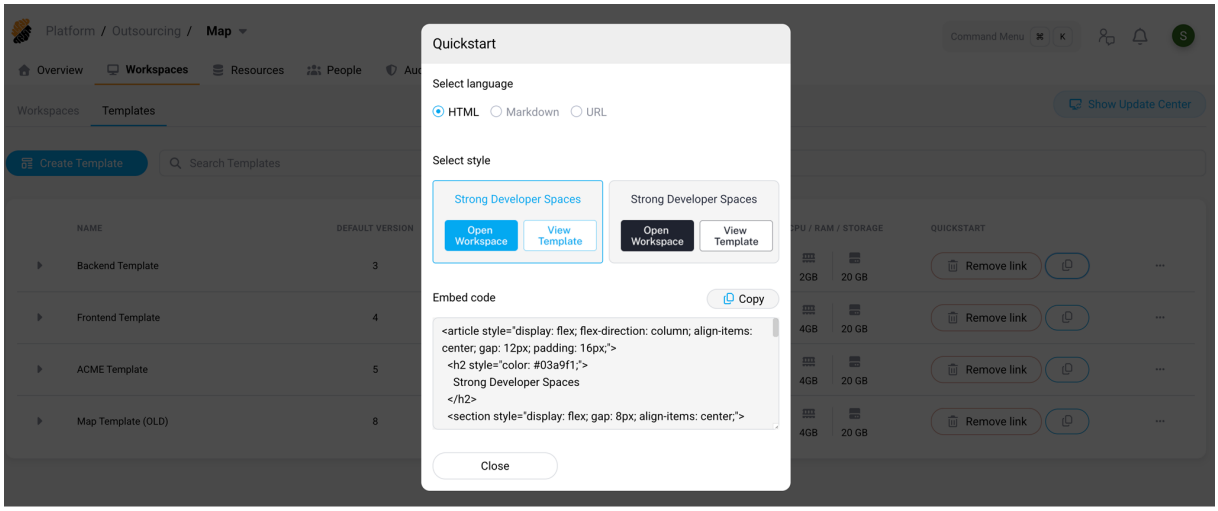



Then click the **Copy** icon.



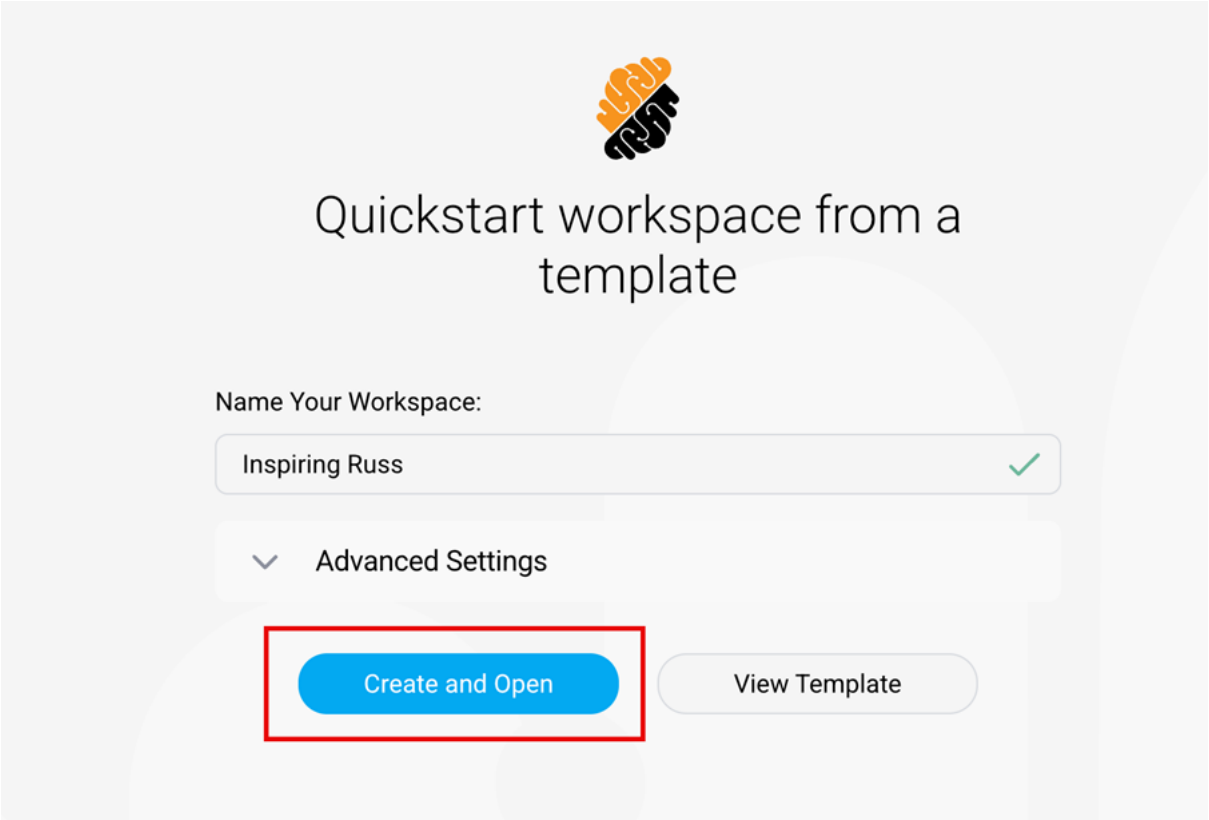

Select any of the available options.




When a user accesses the Quickstart URL SDS 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 forwarded to the respective Workspace automatically.

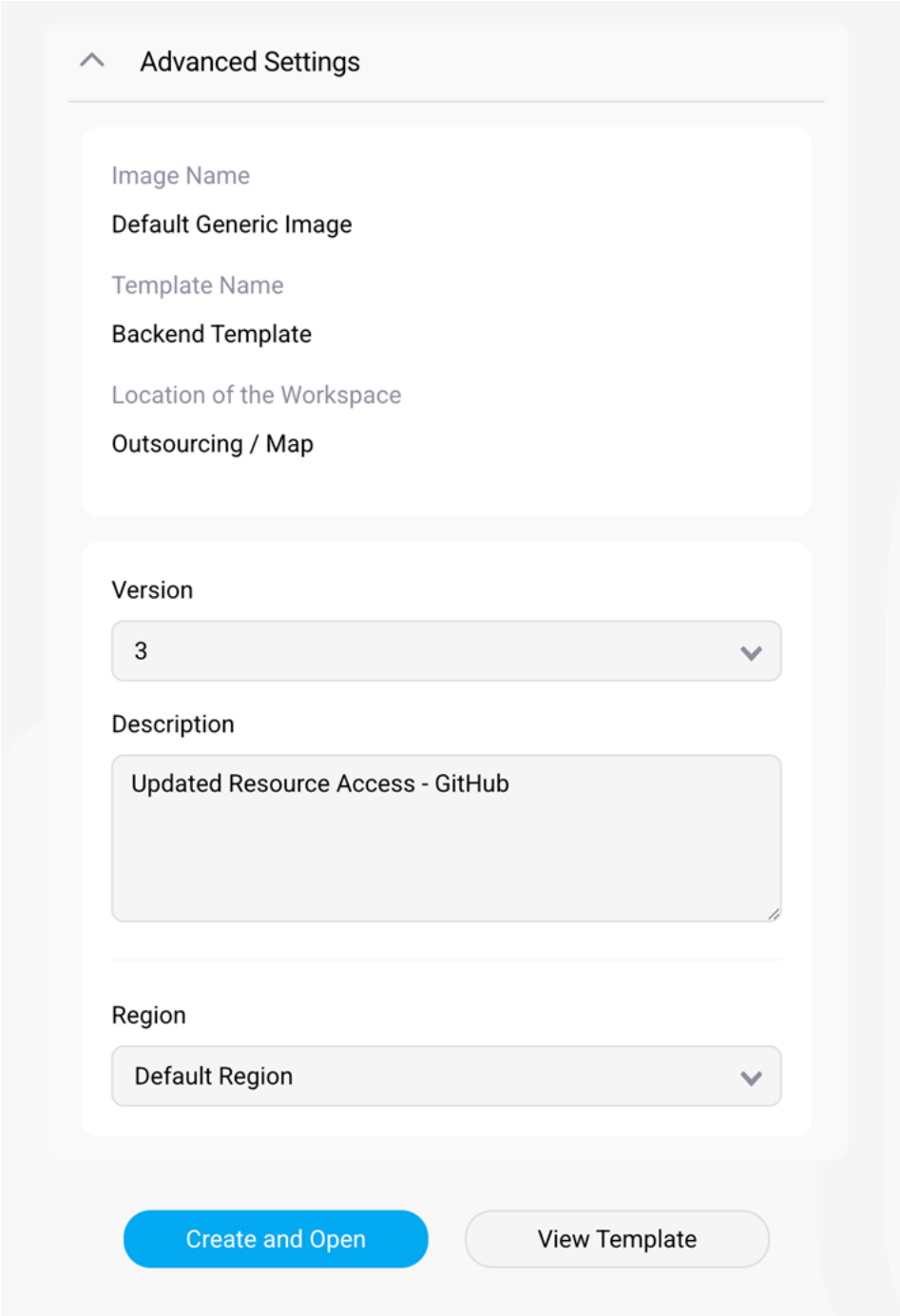
When a new Workspace needs to be created the user can configure the name of the Workspace and finalize the creation flow, by selecting **Create and Open**.



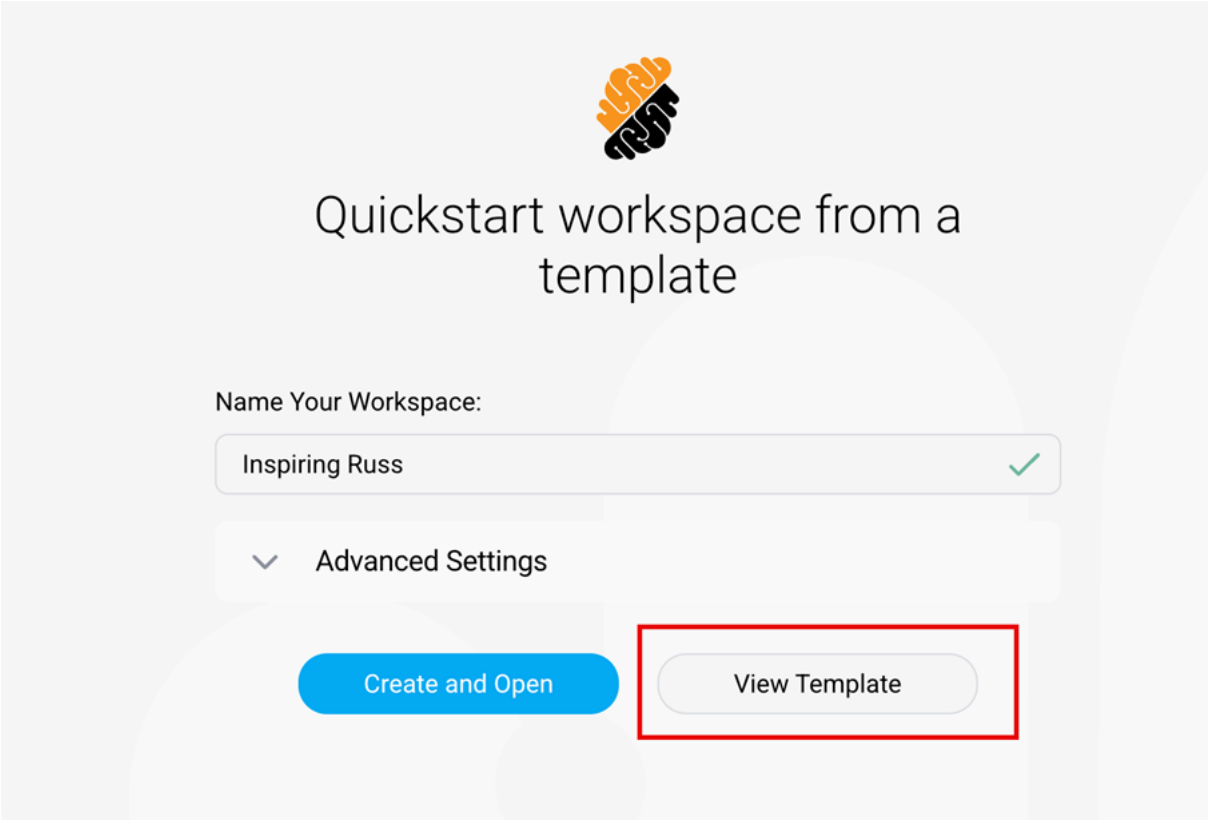

Via the **Advanced Settings** menu, configuration details, such as base template or related SDS project, can be verified and template version as well as deployment region can be configured.




The **View Template** button opens the Workspace Template editor for the selected version, to verify further configuration details.



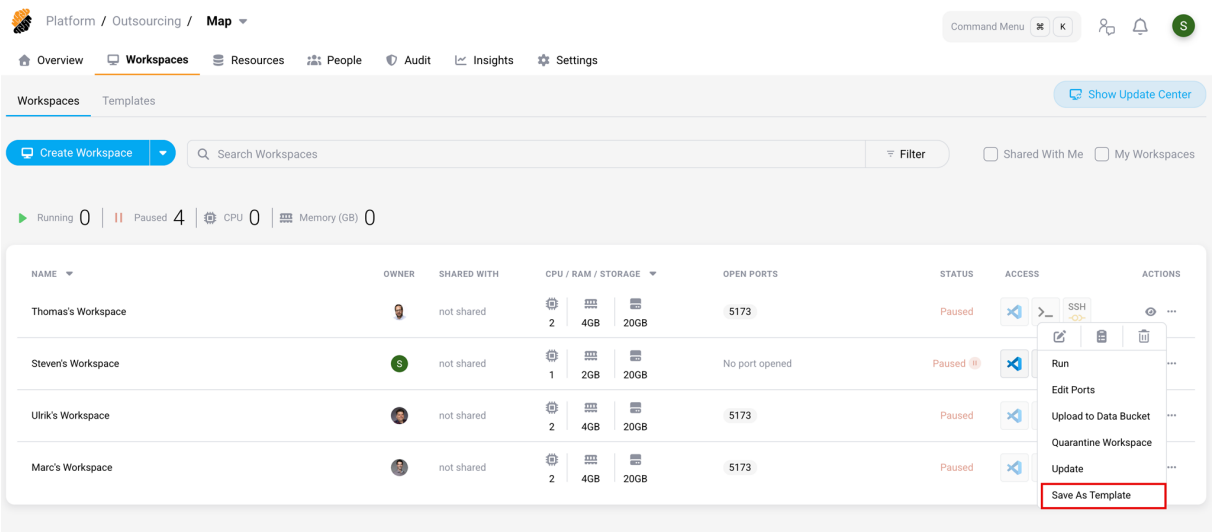

### Duplicate a template

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.

### Create a Template from a Workspace

You can create a Template using an existing Workspace by clicking on the "**...**"button on the right of a Workdpace and select **Save As Template**.




## Coding in a Workspace

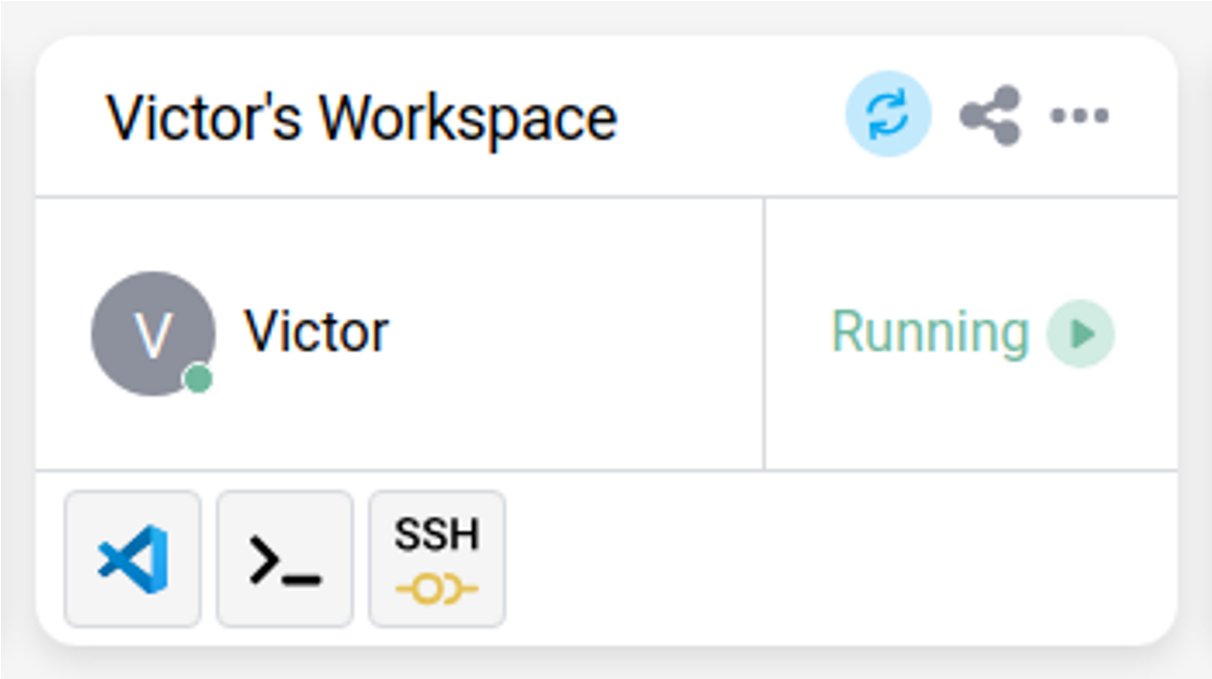
October 2, 2025

The easiest way to code in a workspace is through a Cloud IDE. A Cloud IDE runs directly in the web browser and does not require other software installation on the endpoint, i.e. your development machine. Alternatively, a workspace can be accessed via an SSH connection from a locally installed IDE that allows "remote development". See how it works in Microsoft vscode.

### Cloud-Based Integrated Development Environments (Cloud IDEs)

The platform supports a series of Cloud IDEs that might differ based on your particular deployment. Typically supported IDEs are Microsoft Visual Studio Code and Jetbrains' IDEs. Note that the version of The vscode running in the web browser is the same as the one available for local installation (including the marketplace). Hence, you can refer to any available documentation online to understand its functioning and options.




> **Tip:**
>
> To access a workspace using the Cloud IDE attached to it, just click the button indicating the workspace execution status. This is only possible if you own or have shared access to the workspace.

### Import Local Files in a Cloud IDE

The ability to import local files in the Cloud IDE depends on the setting of your platform. The most common way to do so is to simply drag a file from a user interface such as a browser, to the IDE interface. Please contact the platform administrator to inquire about potential security restrictions imposed on such an operation.

### Workspace Access Using SSH With a Local IDE

You can access your workspace using SSH via a locally installed IDE such as Microsoft VSCode or using JetBrains Gateway. For this, you must register a SSH authentication key to your account in your Profile Page.

Once the key has been registered, you can access the workspace via a two-factor authentication process. This process ensures that you are indeed accessing your workspace remotely and at preventing an authorized user to do so.

You can find a full guide on how to SSH into your Workspace here.

### Work With a Shared Workspace

After sharing a workspace, you may work with other users in the same workspace.
Working in a shared workspace is similar in a way to use work simultaneously in the same document.
The benefit of doing so is that it provides a way to co-edit content, also known as **peer editing**.

> **Tip:**
>
> When modifying files on the same workspace, **changes are displayed in real-time**.
> You may see who is accessing the workspace live from the "(*show component*)".

### Recover a deleted Workspace

After deleting a Workspace, you may recover it for 7 days from the Project Settings.

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

## SSH Into Your Workspace

October 2, 2025

This guide provides instructions for accessing your workspace via SSH, enabling you to edit code directly using a local command-line editor. This process requires the generation of an SSH Key pair.

- 1. Generate an SSH Key Pair on UNIX and UNIX-like Systems
- 2. Upload Your Public Key to the Platform
- 3. Authorize Your Workspace to Use Your SSH Key
- 4a. Connect to Your Workspace Using a Shell
- 4b. Connect to Your Workspace via SSH Using VSCode
  - 4b.1. Install the VSCode SSH Extension
  - 4b.2. Initiate a New SSH Connection from the VSCode SSH Extension
  - 4b.3. Input the SSH Command into the Extension Prompt
  - 4b.4. Select the Default SSH Configuration
  - 4b.5. Click the "Connect"Button after the Host is Added
- 4c. Connect to Your Workspace via SSH Using JetBrains Gateway
  - 4c.1. Install JetBrains Gateway

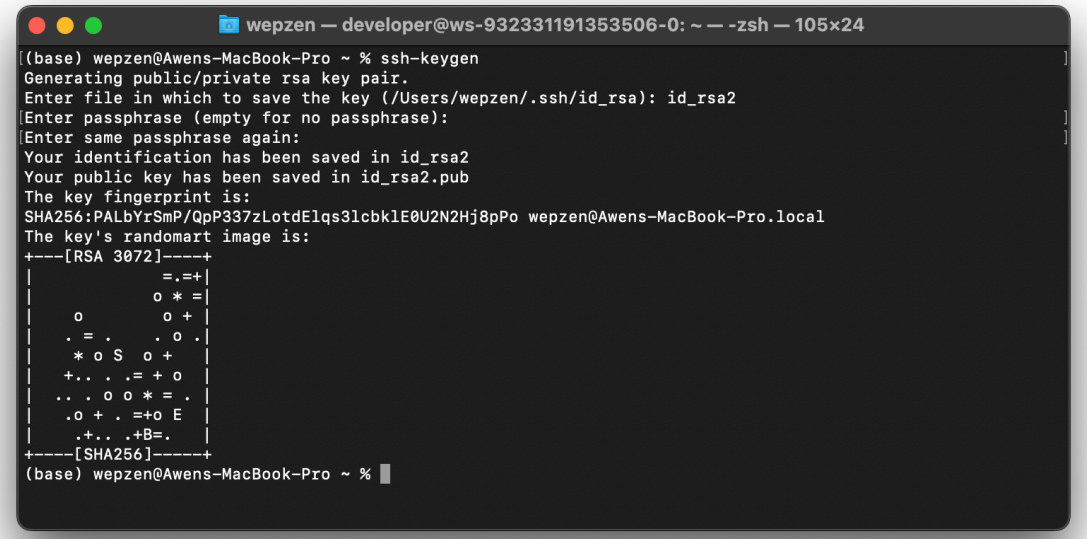
### 1. Generate an SSH Key Pair on UNIX and UNIX-like Systems

- To generate an SSH key pair on UNIX and UNIX-like systems, run the `ssh-keygen` command in your terminal:

```
ssh-keygen
```

- The terminal will suggest a default path and file name (for example, /home/user_name/.ssh/id_rsa). To accept the default path and file name, press Enter. If you want to specify a different path and file name, enter those details and then press Enter.
- The command prompts you to enter a passphrase. Although optional, it's recommended to set a passphrase for additional security against unauthorized use of your private key.
- If you set a passphrase, you will be prompted to enter it again for confirmation. If you didn't set a passphrase, simply press Enter.
- The command generates an SSH key pair - a public key and a private key - and saves them in the specified path. The public key file name is automatically created by appending .pub to the private key file name. For instance, if the private key file is named id_rsa, the public key file will be named id_rsa.pub.

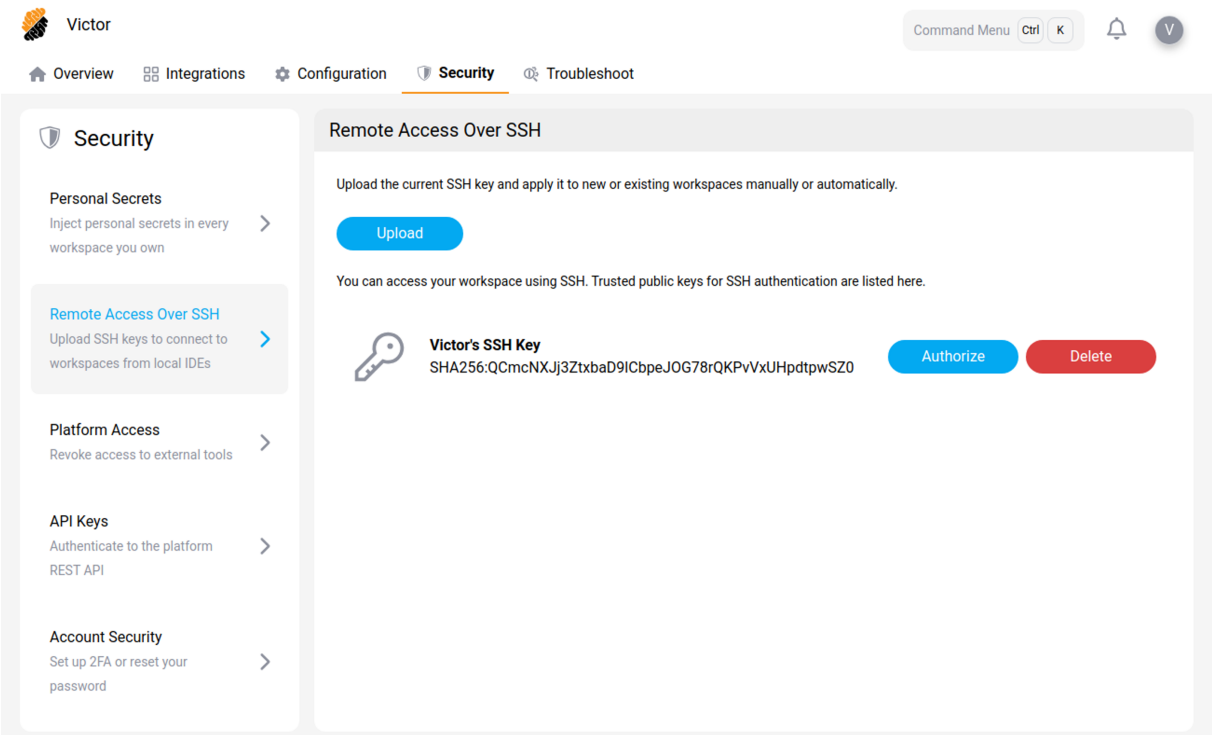



## 2. Upload Your Public Key to the Platform

Once your SSH Key pair is generated, you need to upload it to the SSH Keys Section in your Profile.

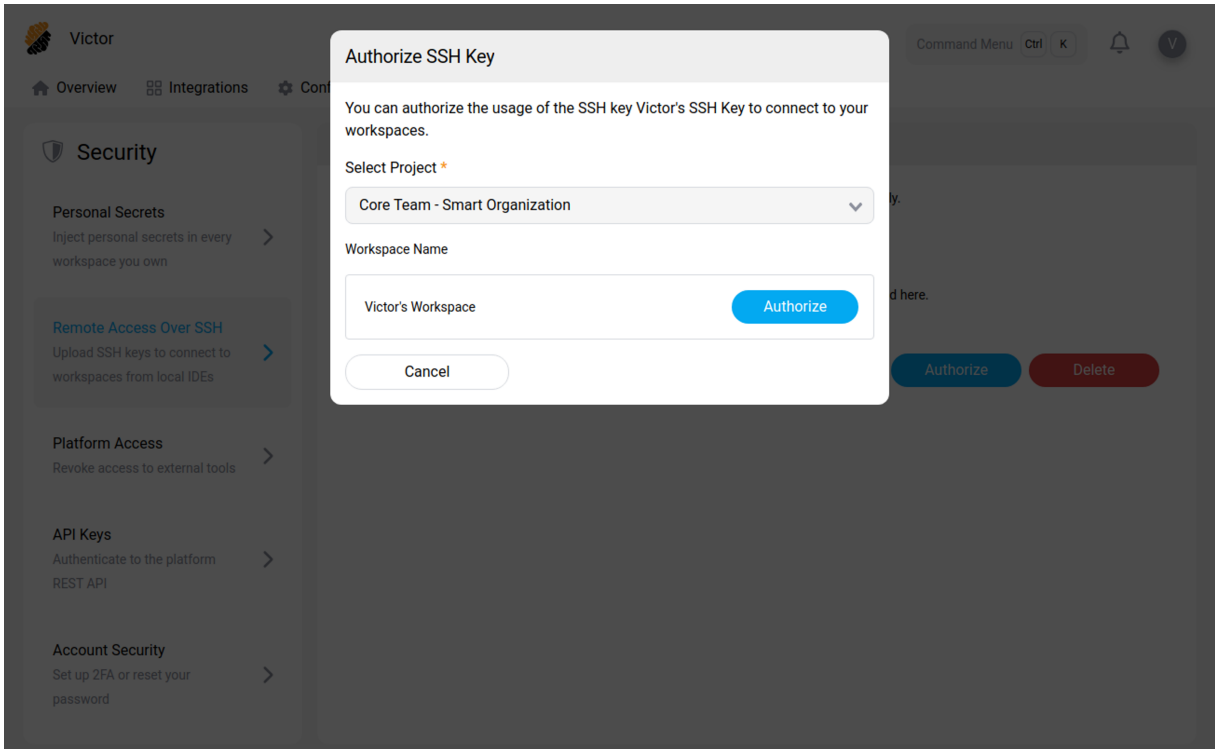
The key begins with ‘ssh-rsa’, ‘ecdsa-sha2-nistp256’, ‘ecdsa-sha2-nistp384’, ‘ecdsa-sha2-nistp521’, ‘ssh-ed25519’, ‘sk-ecdsa-sha2 nistp255@openssh.com’or ‘sk-ssh-ed25519@openssh.com’.




## 3. Authorize Your Workspace to Use Your SSH Key

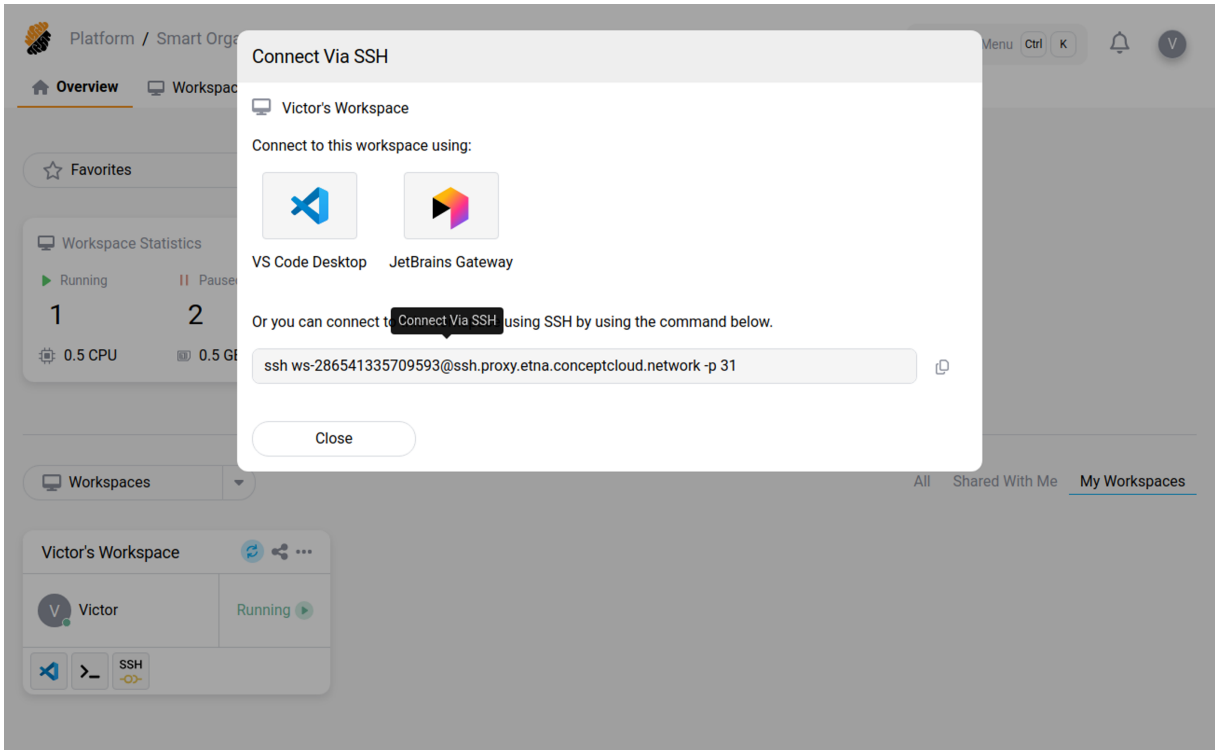
After uploading your SSH key to your profile, you need to authorize your workspace(s) to access it.



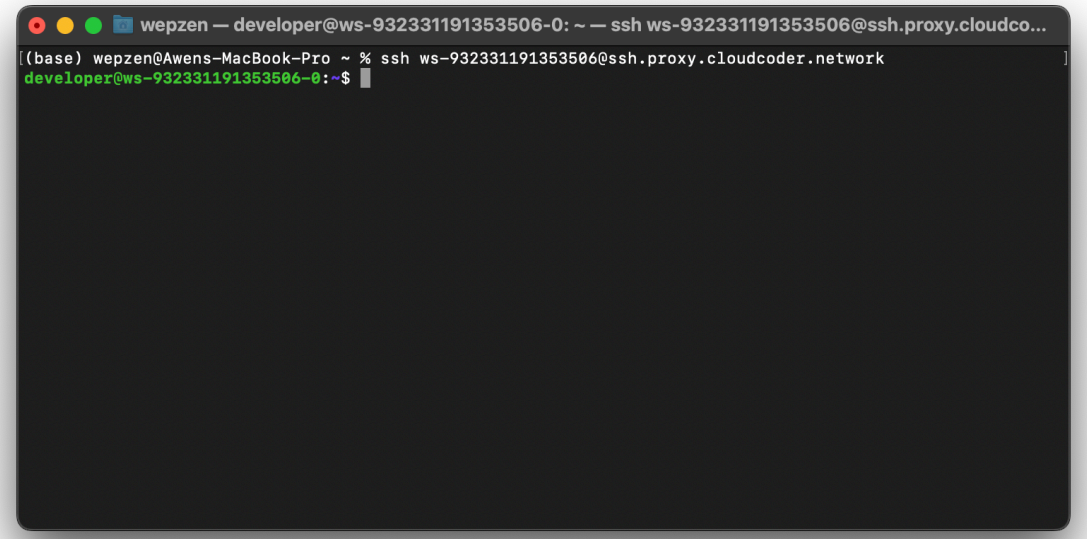
## 4a. Connect to Your Workspace Using a Shell

Navigate to the Running Actions List of Your Workspace and select the "Connect With SSH"option. This action will display the `ssh` command that you need to establish an SSH connection to your Workspace.




Input this command in your terminal.




Once this is done, you will have successfully established an SSH connection to your Workspace!

## 4b. Connect to Your Workspace via SSH Using VSCode

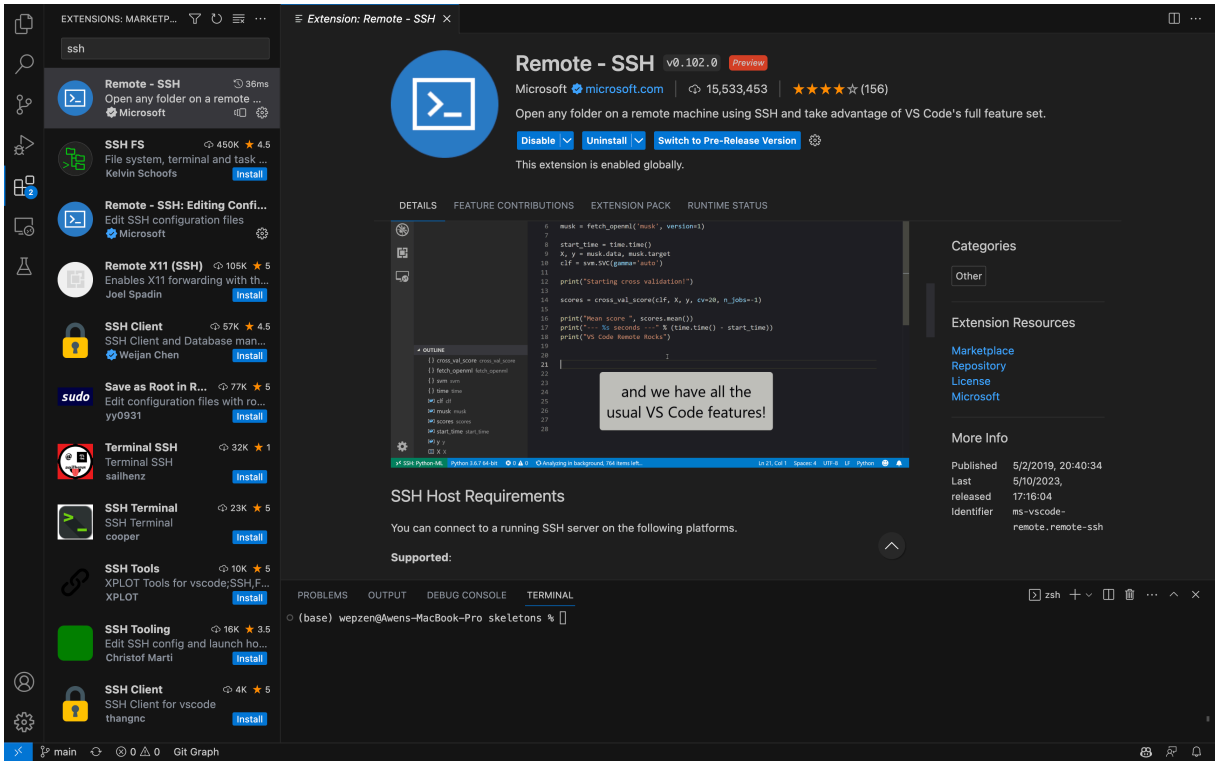
This section provides a detailed walkthrough on setting up an SSH connection to your workspace using the VSCode SSH extension.

> **Tip**
>
> Note that you can execute the same steps directly from your terminal, beginning with step 5b.3.

### 4b.1. Install the VSCode SSH Extension

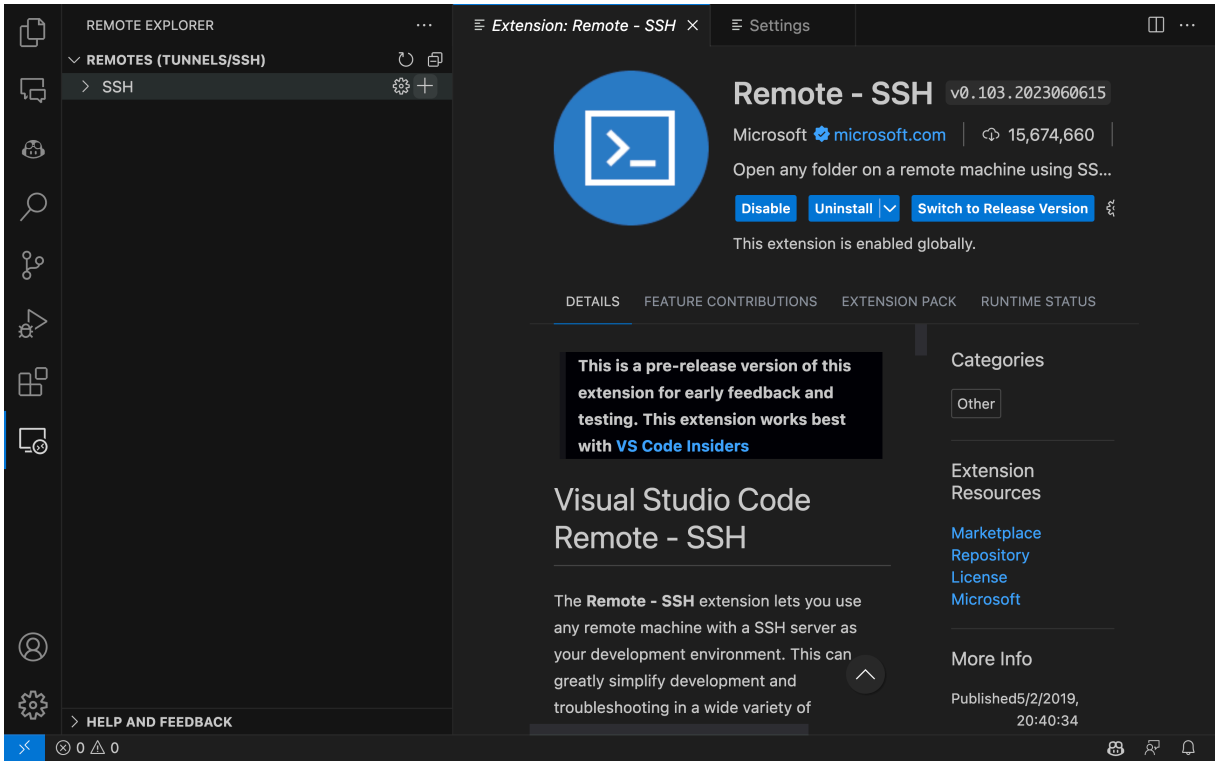
To SSH into your workspace directly from your local VSCode IDE, you can download the Microsoft SSH Extension. This extension replicates the usual SSH command you would perform from your terminal, but allows you to work directly within your local VSCode.




## 4b.2. Initiate a New SSH Connection from the VSCode SSH Extension

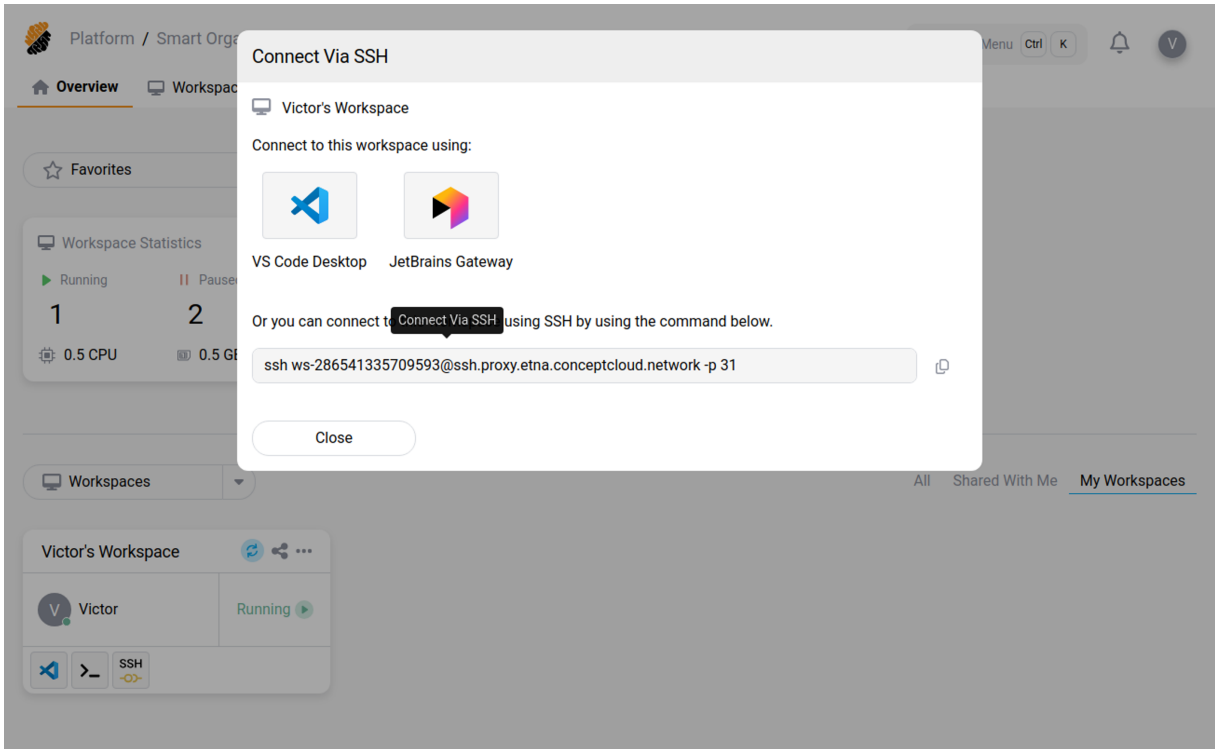
By clicking the “+”button next to the “SSH”panel in the VSCode Extension section.



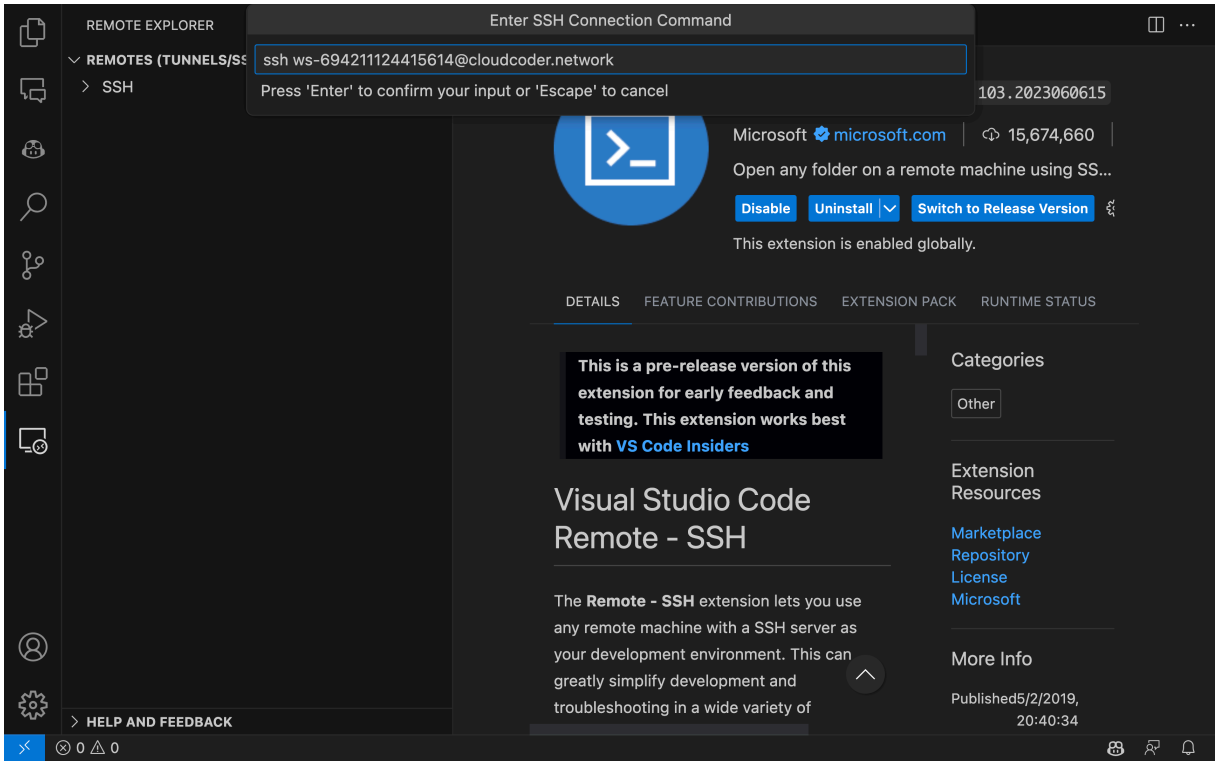
### 4b.3. Input the SSH Command into the Extension Prompt

From your Workspace's Running Actions List select the "Connect With SSH"option to display the `ssh` command you need to connect to your Workspace via SSH.

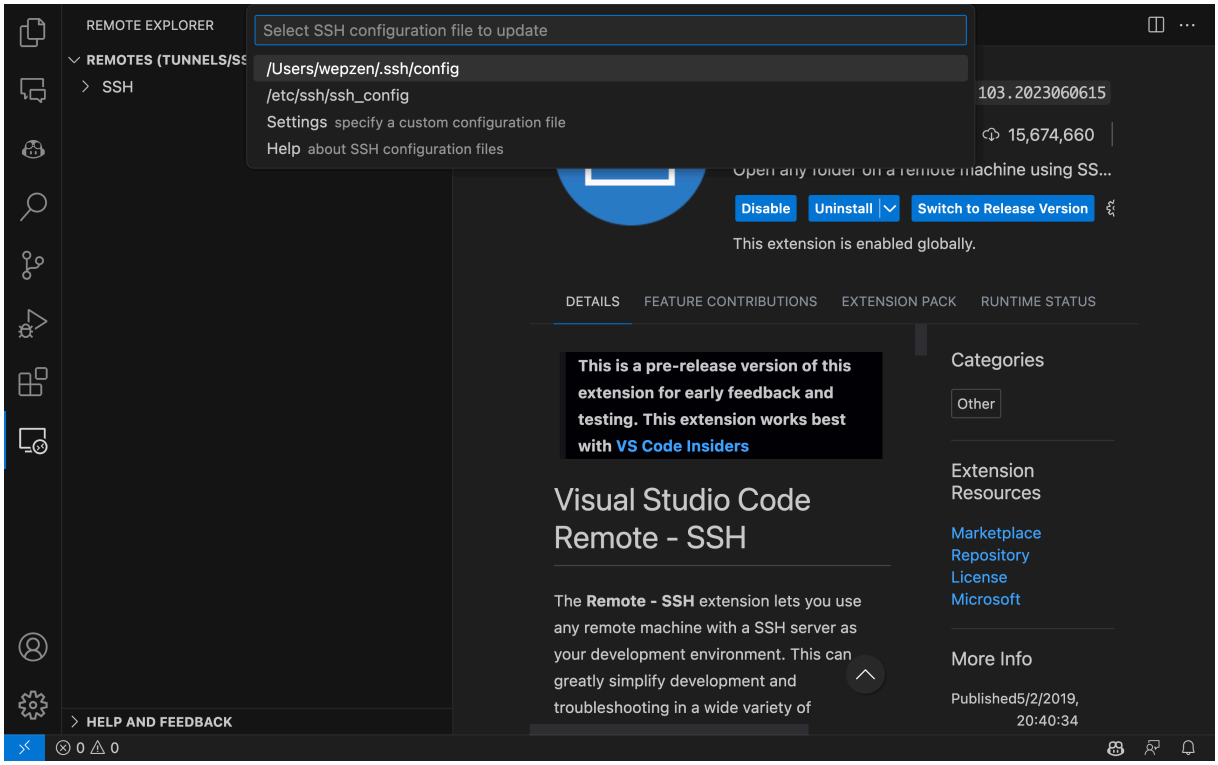



Enter this command in the VSCode extension prompt.

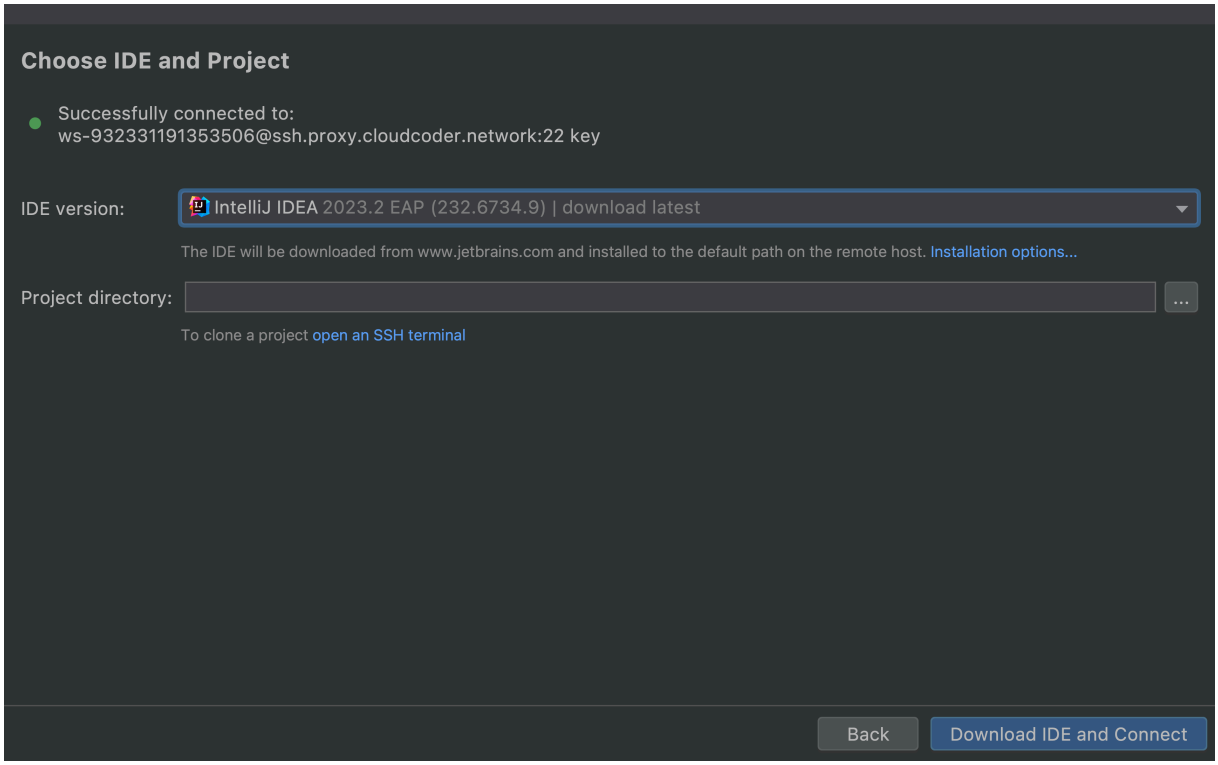



## 4b.4. Select the Default SSH Configuration



#### 4b.5. Click the “Connect”Button after the Host is Added




You are successfully connected to your Workspace with SSH!

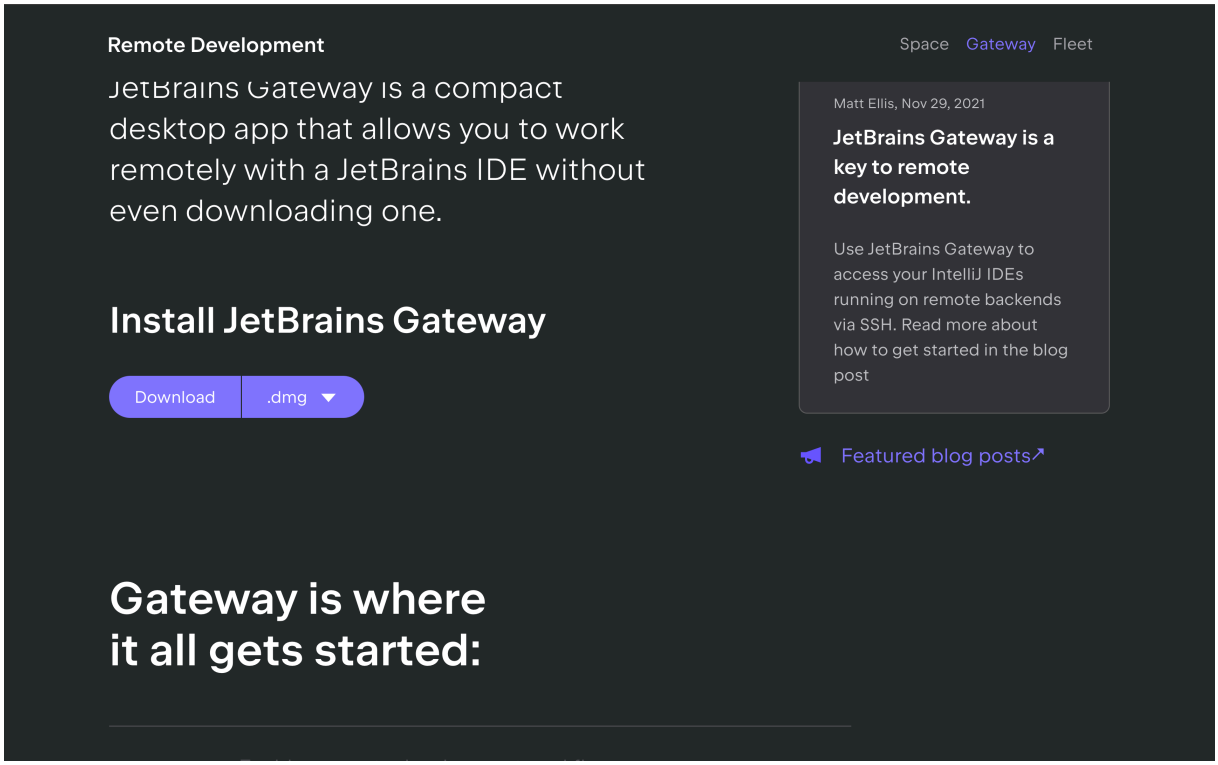
### 4c. Connect to Your Workspace via SSH Using JetBrains Gateway

This section offers a comprehensive guide on establishing an SSH connection to your workspace using JetBrains Gateway.

#### 4c.1. Install JetBrains Gateway

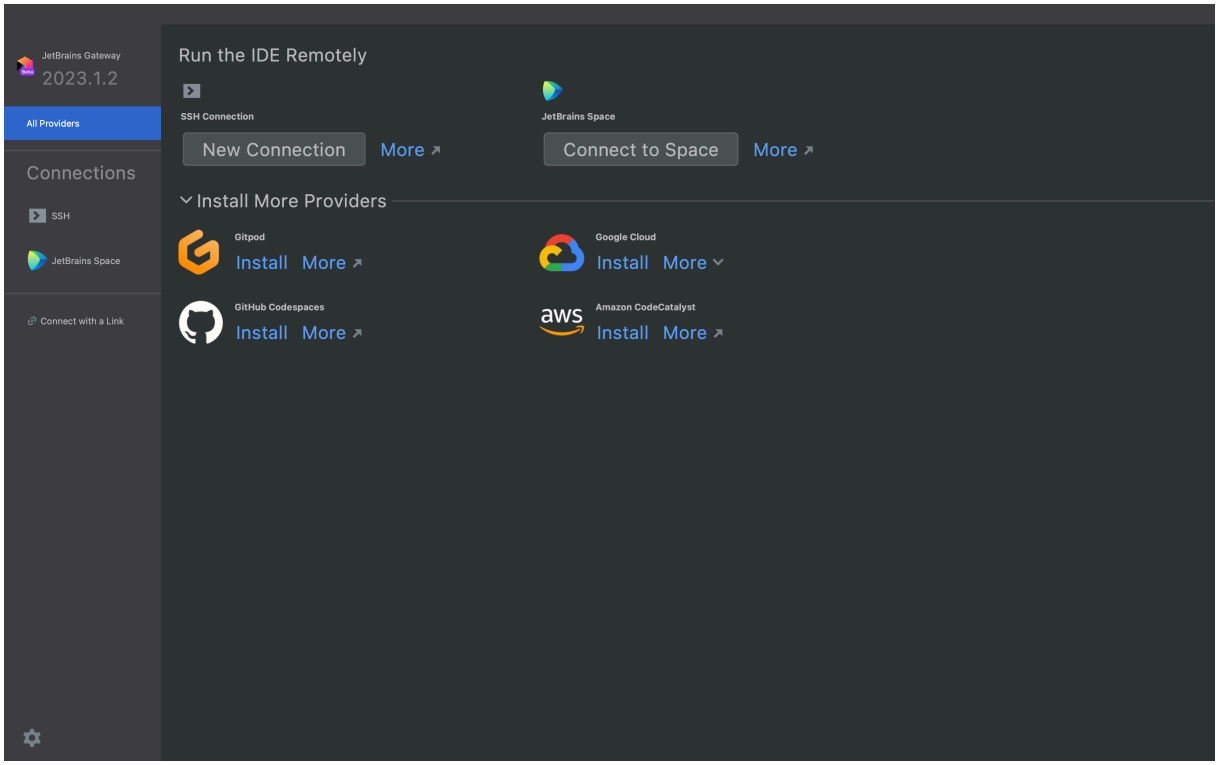
To access your workspace directly from your local JetBrains IDE, download JetBrains Gateway. This software enables SSH connection to your workspace using JetBrains.




### 4c.2. Begin a New SSH Connection

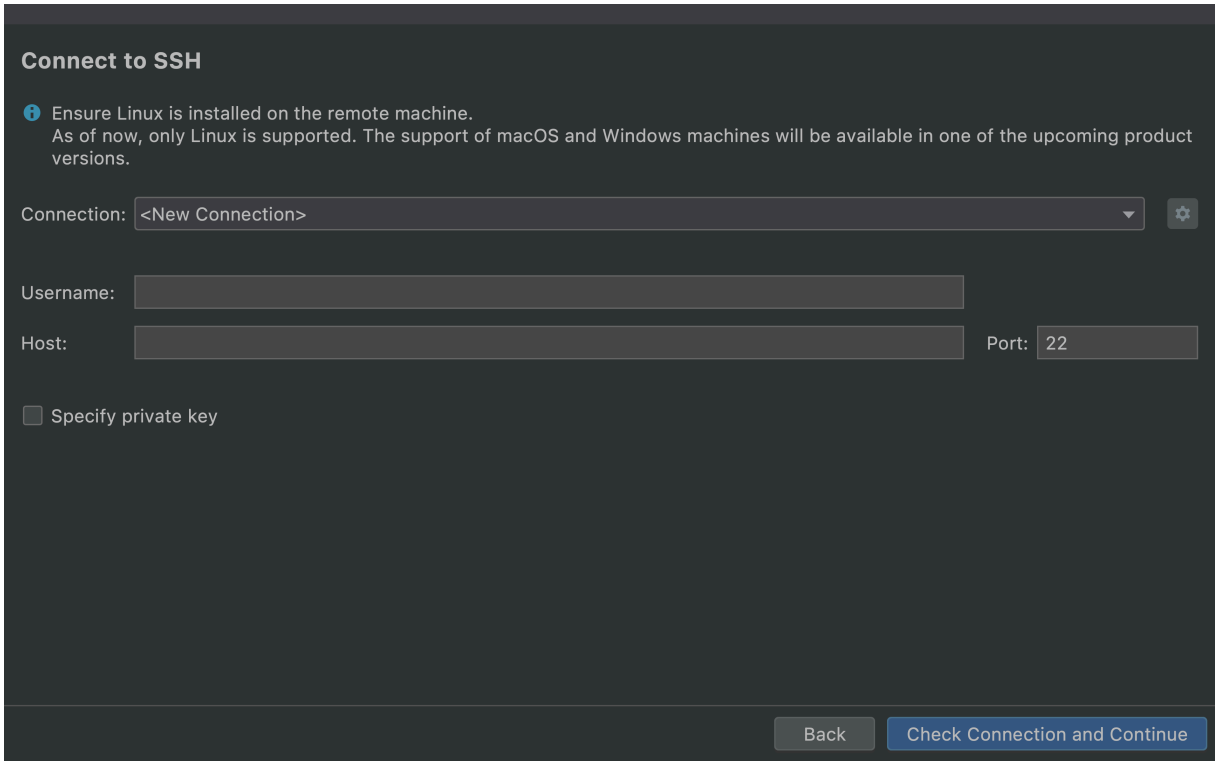
Start by clicking the “New Connection”button found below the “SSH Connection”title.



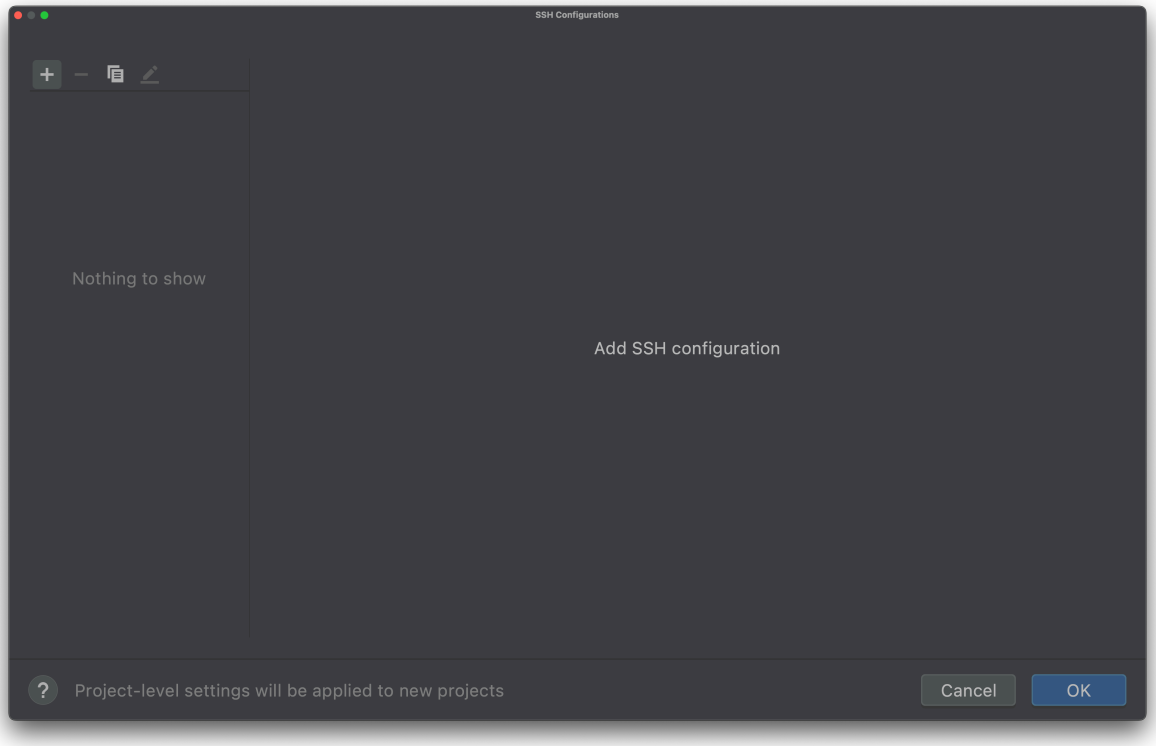
### 4c.3. Create an SSH Configuration

Click the “settings icon”next to the “New Connection”option.




Then click the “+”icon to add a new SSH configuration.

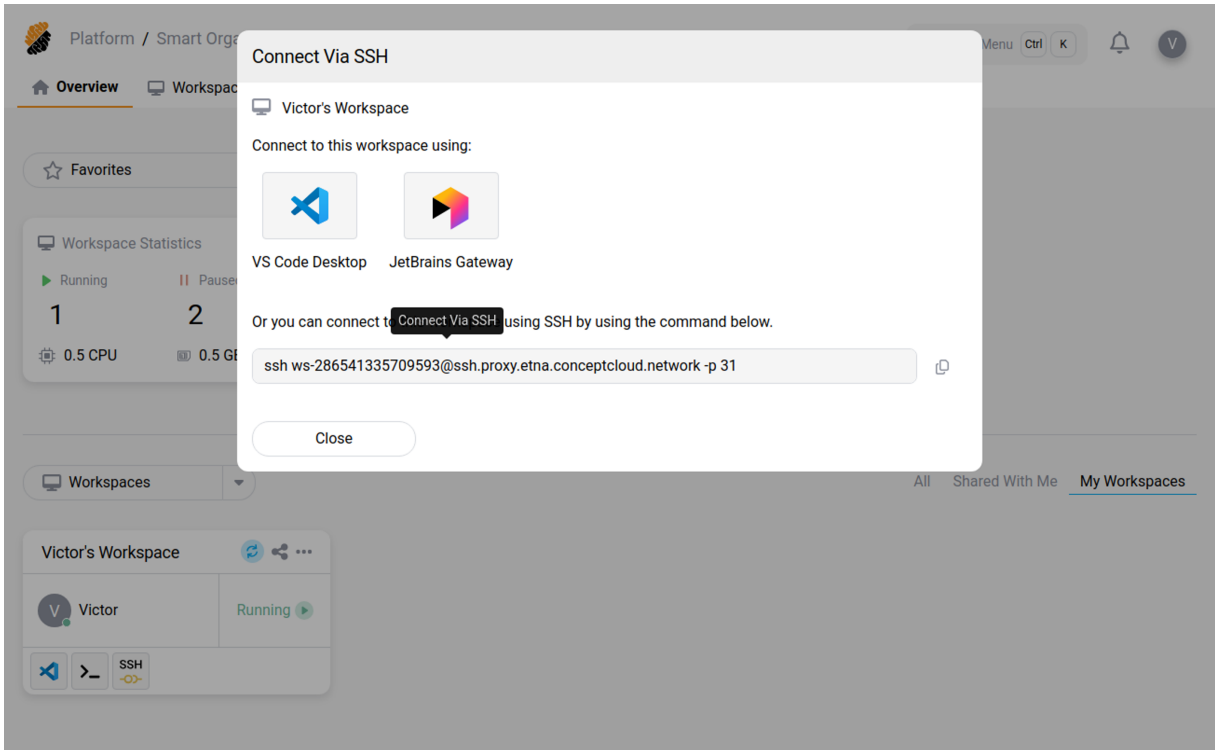
 




### 4c.4. Enter the Host and Username Information

Select the “Connect With SSH”option from your Workspace’s Running Actions List to view the shh command necessary for the SSH connection to your workspace.

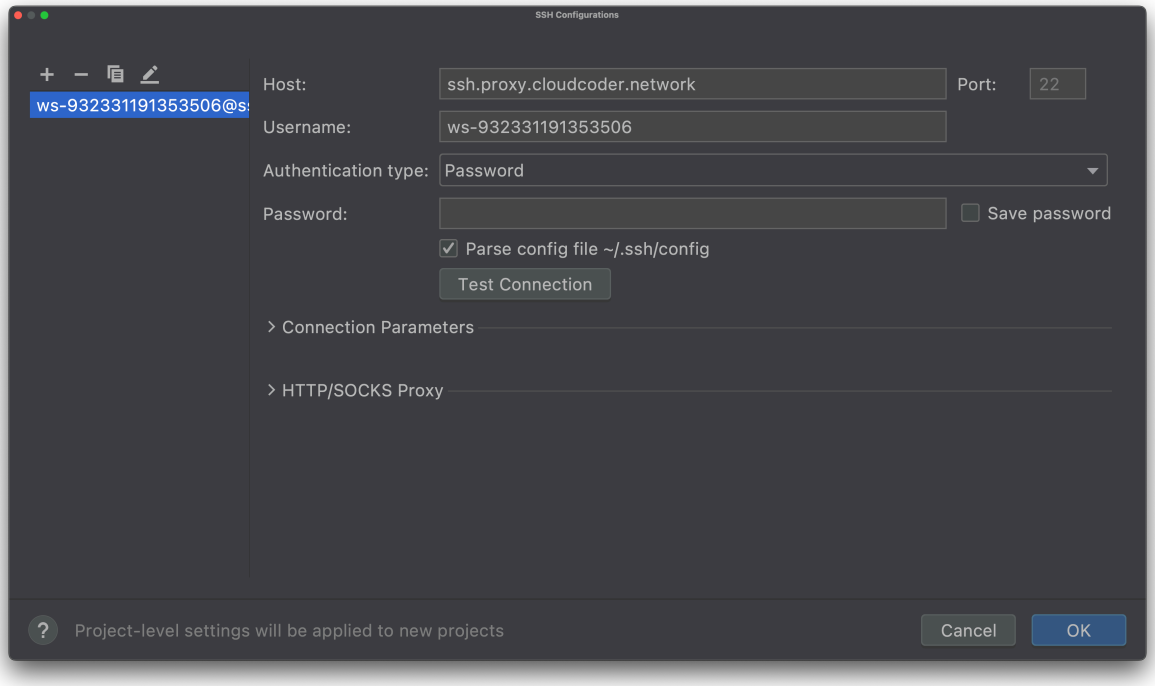
 




Enter the command details into the SSH configuration settings.

> **Tip**
>
> - Host = second part of the command (example: ssh.proxy.cloudcoder.network)
> - Username = first part of the command (example: ws-694211124415614)
>
> Disregard the `ssh` and `@` characters.

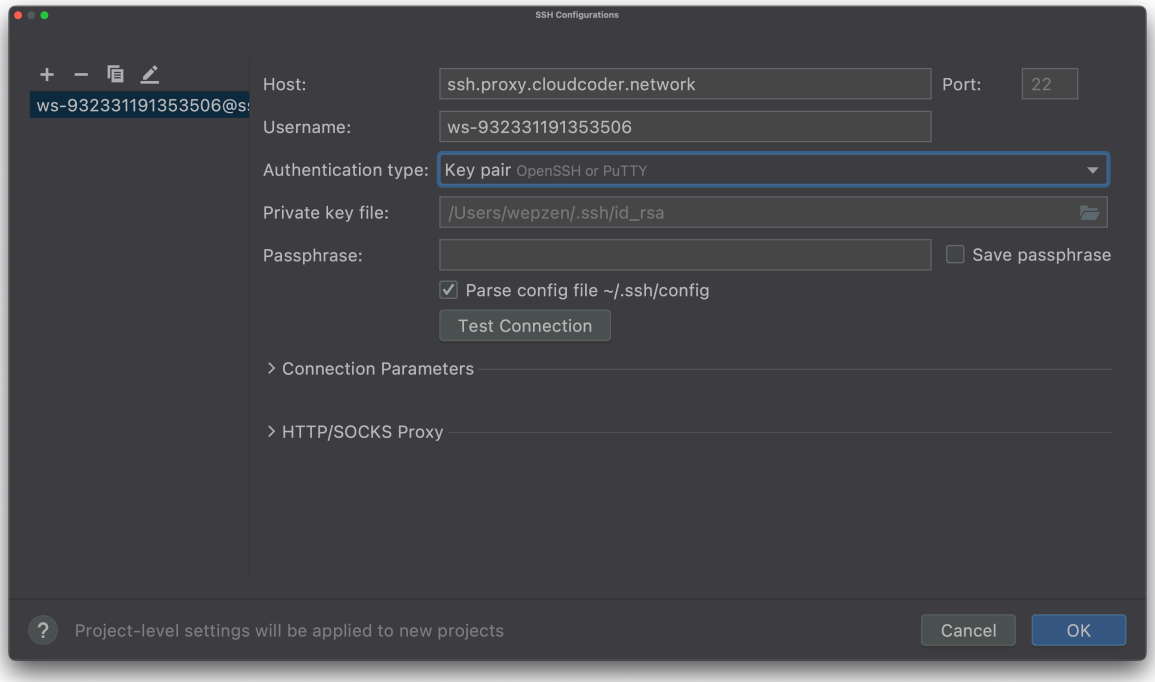



### 4c.5. Choose Authentication Method and Test Your SSH Configuration

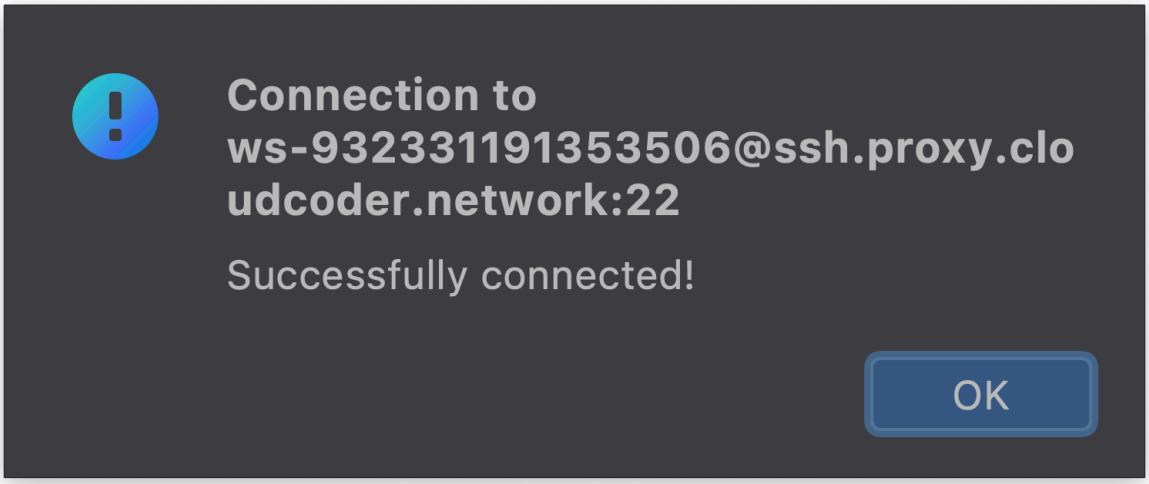
Select “Key Pair”as the “Authentication type”and provide the path for your key (the default field can be left as is).

> **Warning**
>
> By default, the “Password”option is selected as the authentication method.



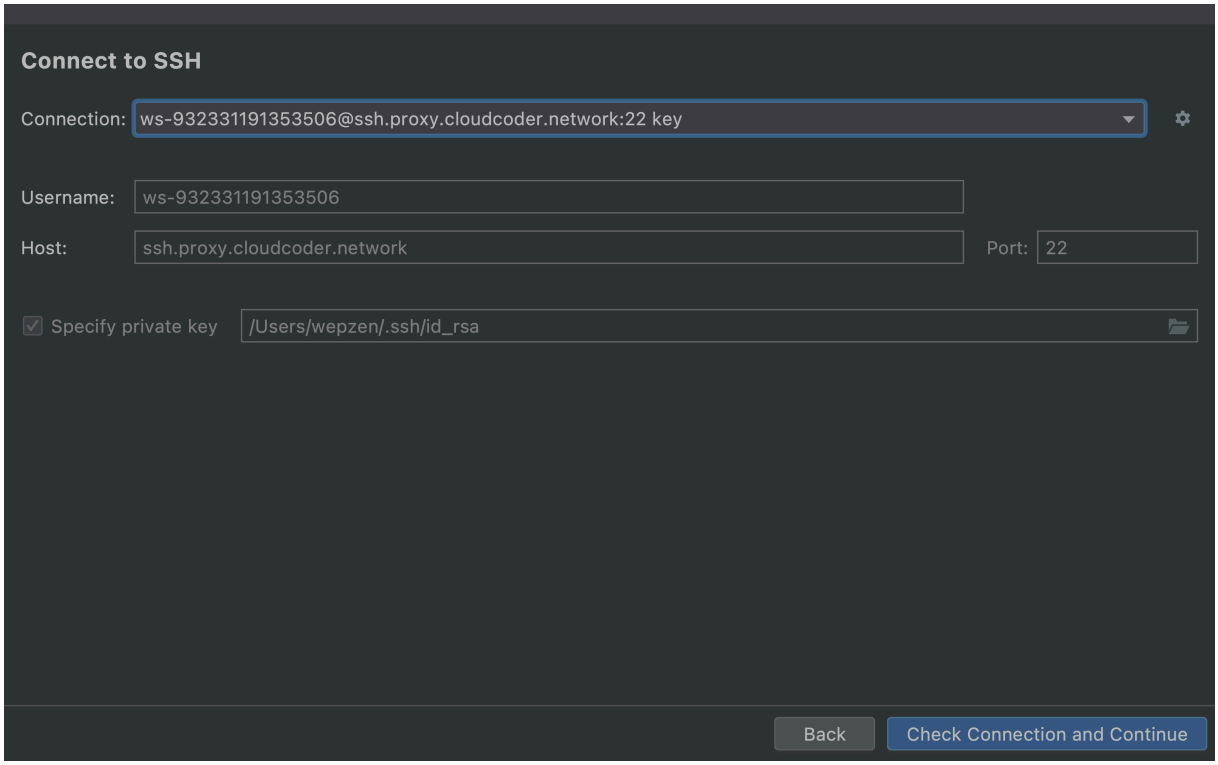

After filling in the “Host”, “Username”, and “Authentication method”fields, test your SSH configuration by clicking the “Test Connection”button. You should see the following:




### 4c.6. Select an SSH Configuration

Upon validating your SSH configuration by clicking “Ok”, select your new configuration as the “Connection”in the “Connect to SSH”menu.

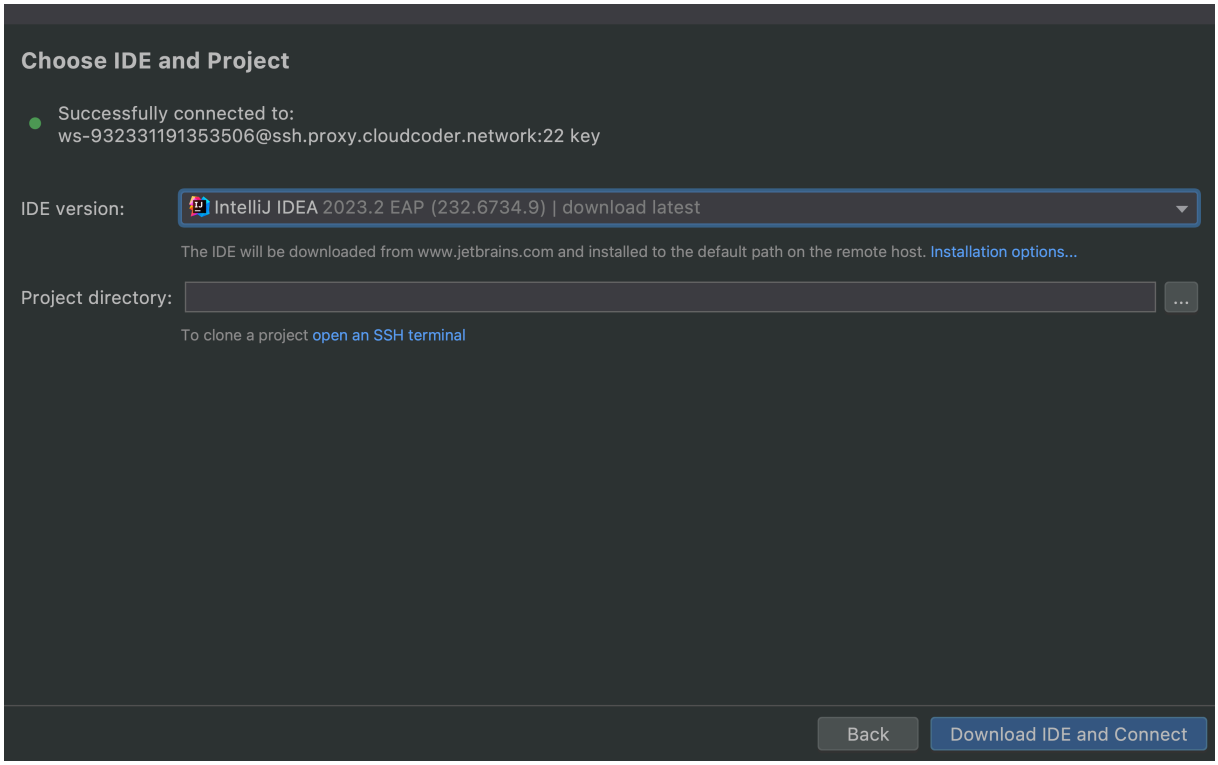
 




### 4c.7. Verify Your SSH Configuration and Connect to Your Workspace

Validate your connection by clicking the “Check Connection and Continue”button. If the connection is successful, you will be directed to the following screen:

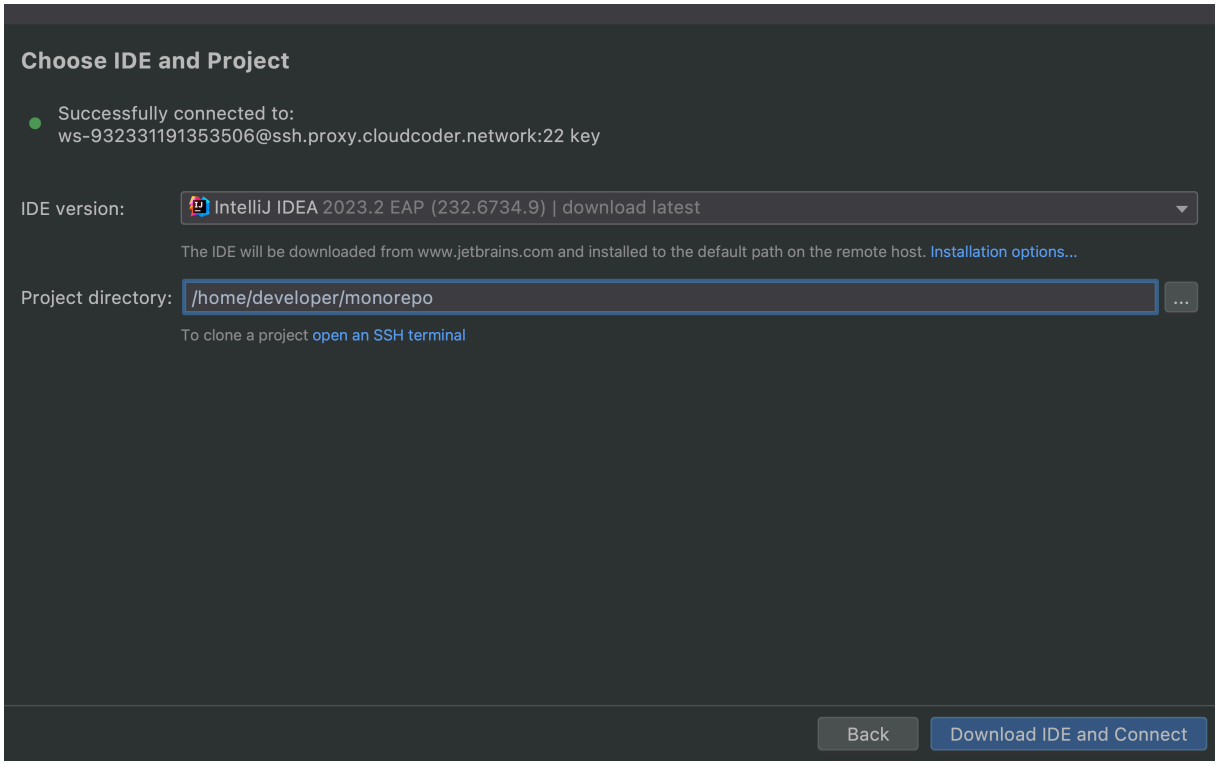
 




### 4c.8. Choose and Download the JetBrains IDE

On the successful connection screen, select the JetBrains IDE you wish to use and the folder you intend to open.

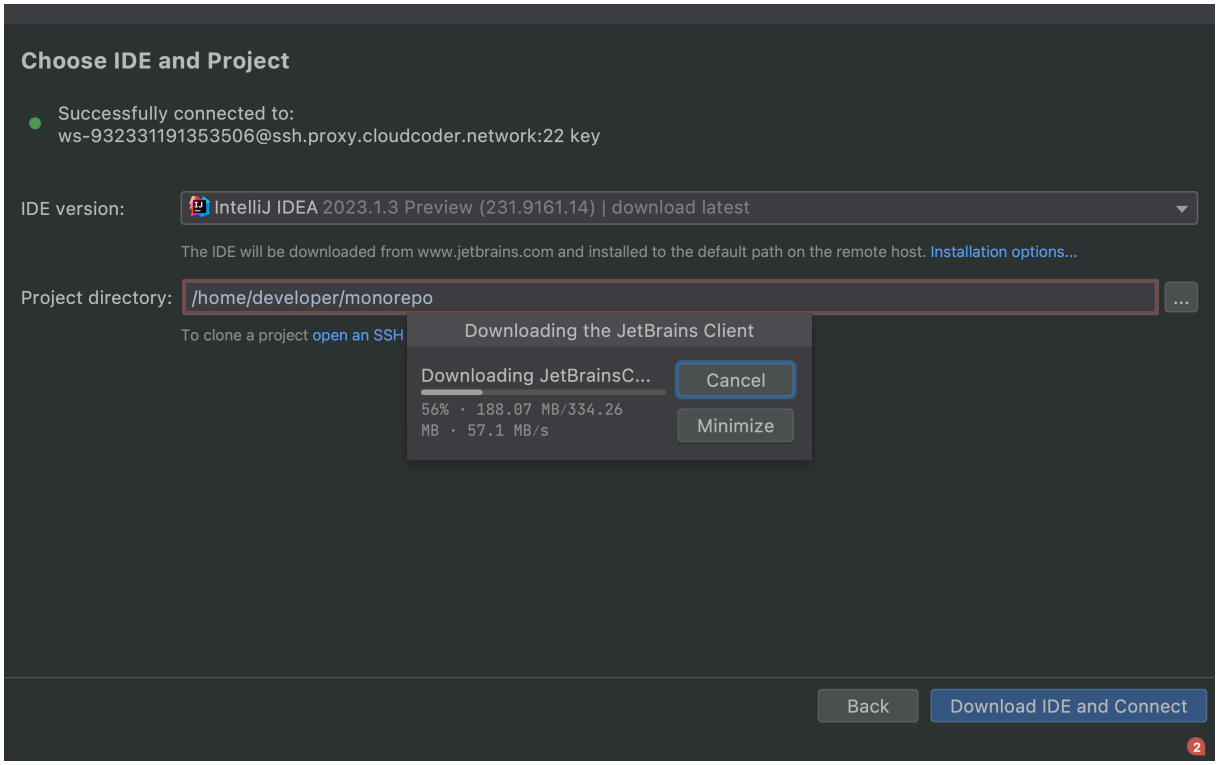
 




Confirm your selections by clicking "Download IDE and Connect". The following screen indicates that the IDE is being downloaded to your workspace.

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

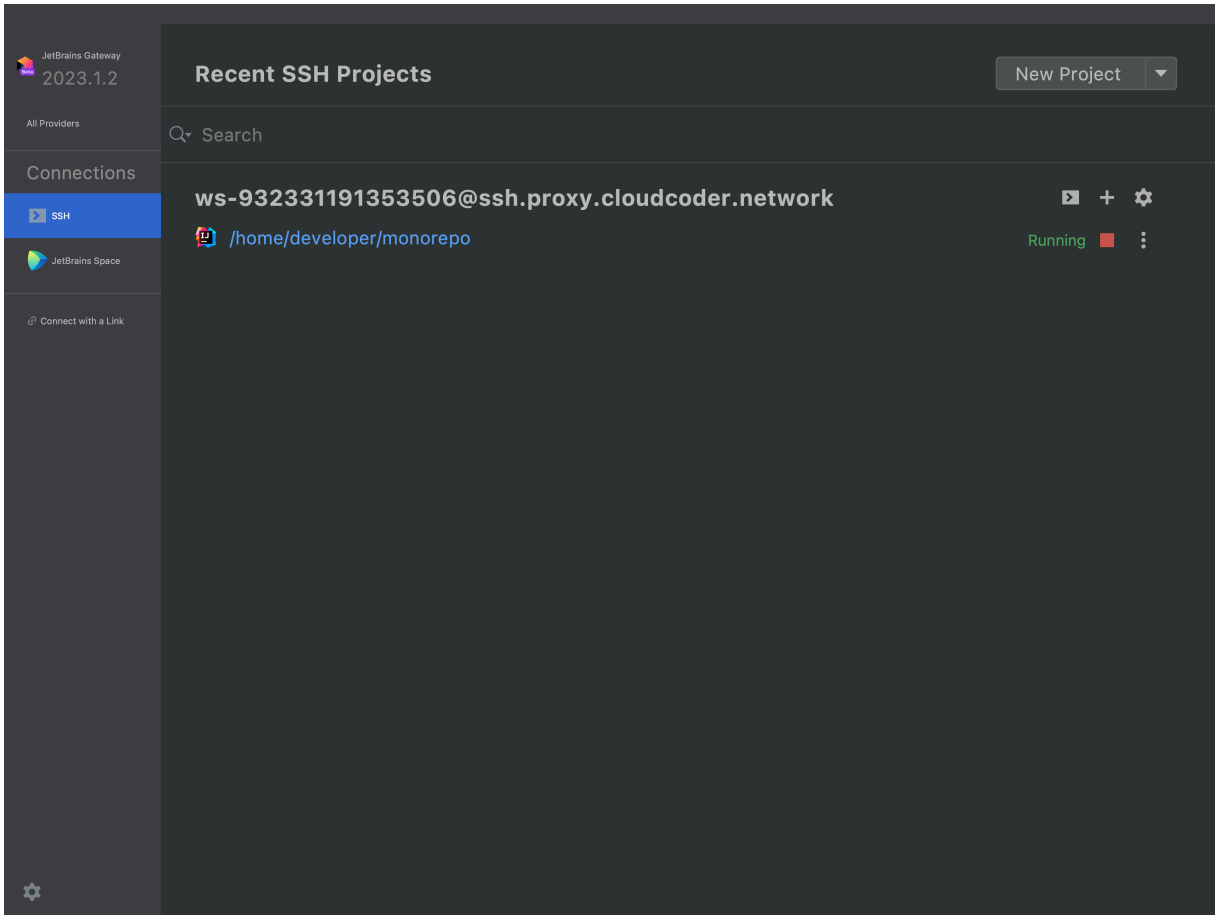
 




### 4c.9. Access Your Workspace

After the completion of the IDE installation, you can now access your workspace via JetBrains Gateway!




## Workspace resource usage insights

November 5, 2025

Citrix Secure Developer Spaces™ (SDS) provides historical insights into workspace CPU and memory usage. This data is automatically collected and stored in the SDS database and is accessible via API 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.

### Requirements

To enable workspace metrics collection, 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, storage, and access

- **Data Consolidation**: SDS 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**: 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, please leverage the **workspace-measurements-samples** API (e.g. `/v1/metrics/workspace-measurements-samples`)
  - For aggregated data, please leverage the **workspace-measurements** API (e.g. `/v1/projects/{ projectId } /metrics/workspace-measurements`)

For API documentation and usage examples, see the Secure Developer Spaces API documentation

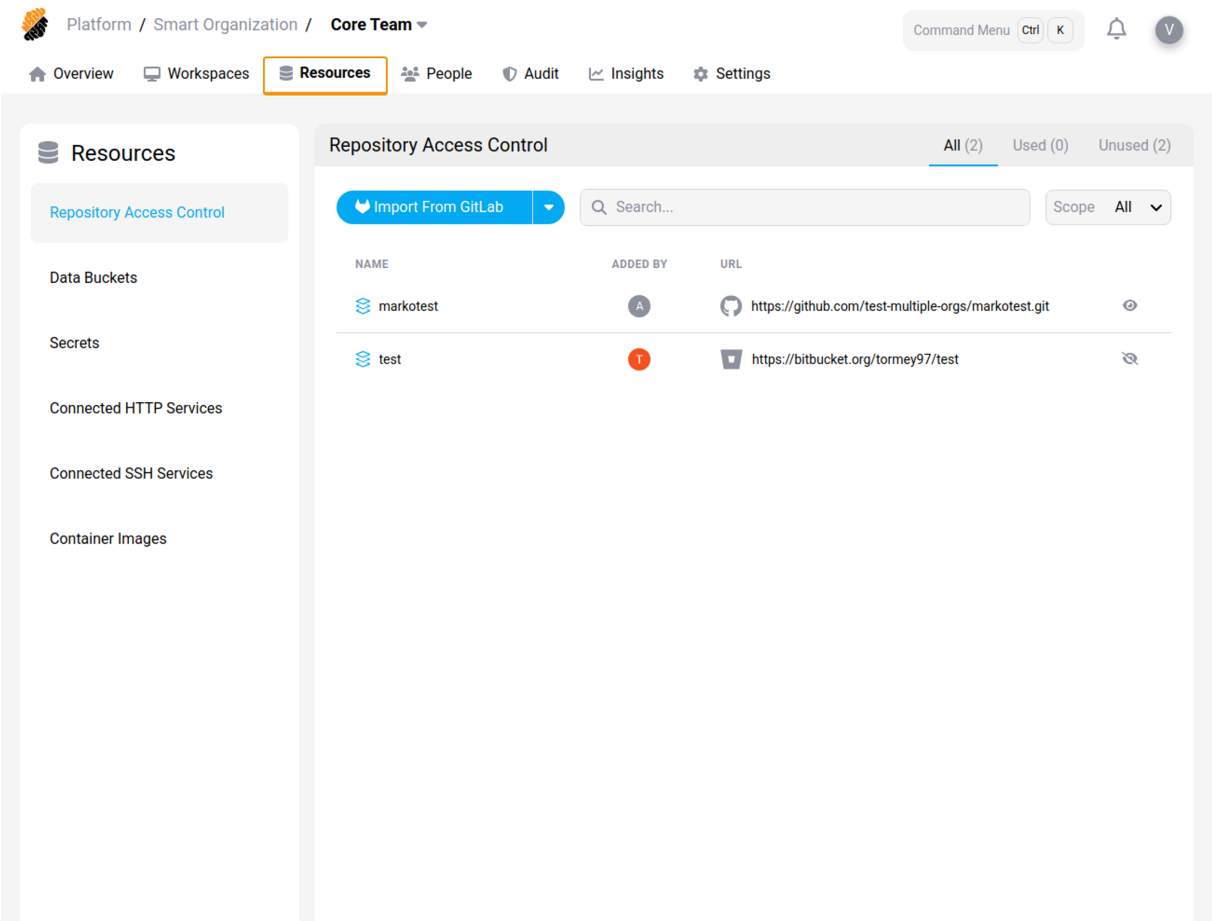
## Resources Page

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.




### Content

- Repository access control
- Data buckets
- Secrets
- Connected HTTP services
- Connected SSH services
- Container images

## Code Repositories

October 2, 2025

Code repositories are used for storing, tracking, and collaborating on source code developed using software development projects. The format supported by the

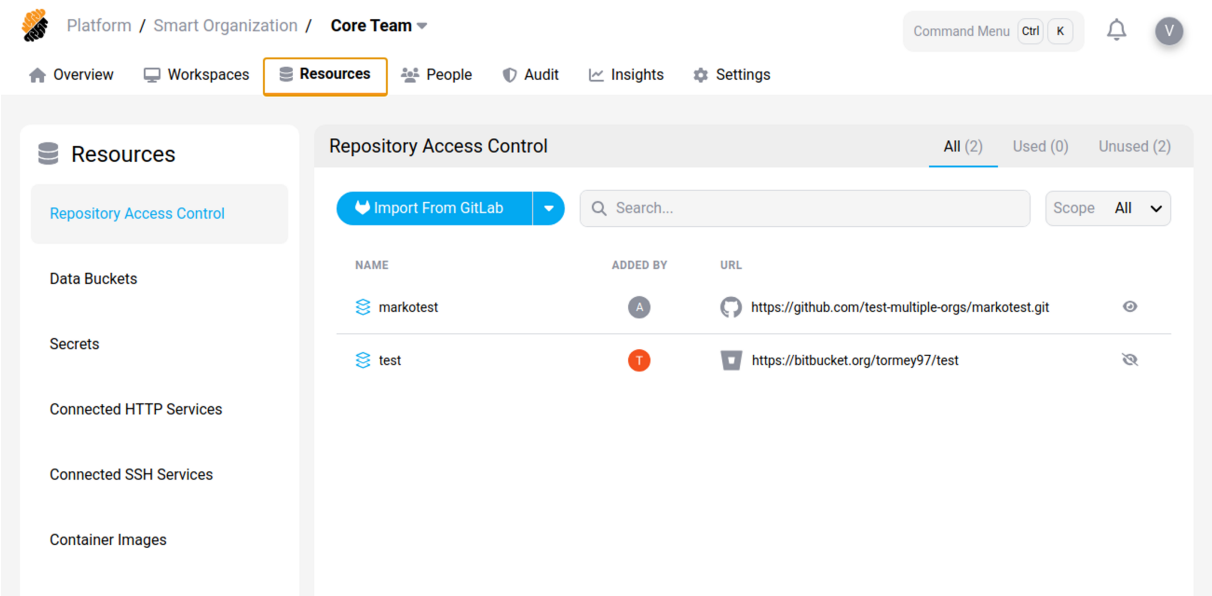
 

platform to manage source code repositories is GIT. Therefore assets from providers using this format can be imported to the platform and attached to
workspaces. Currently, providers such as GitHub, GitLab and BitBuckets are supported. In addition, you can import GIT repositories manually by providing
the necessary information.

- View Repositories
- Import a Repository `Permission: Resources::Import`

## View Repositories

Code Repositories whose information has been imported in the project are displayed
in the table. You may search for one or filter those used in workspaces.



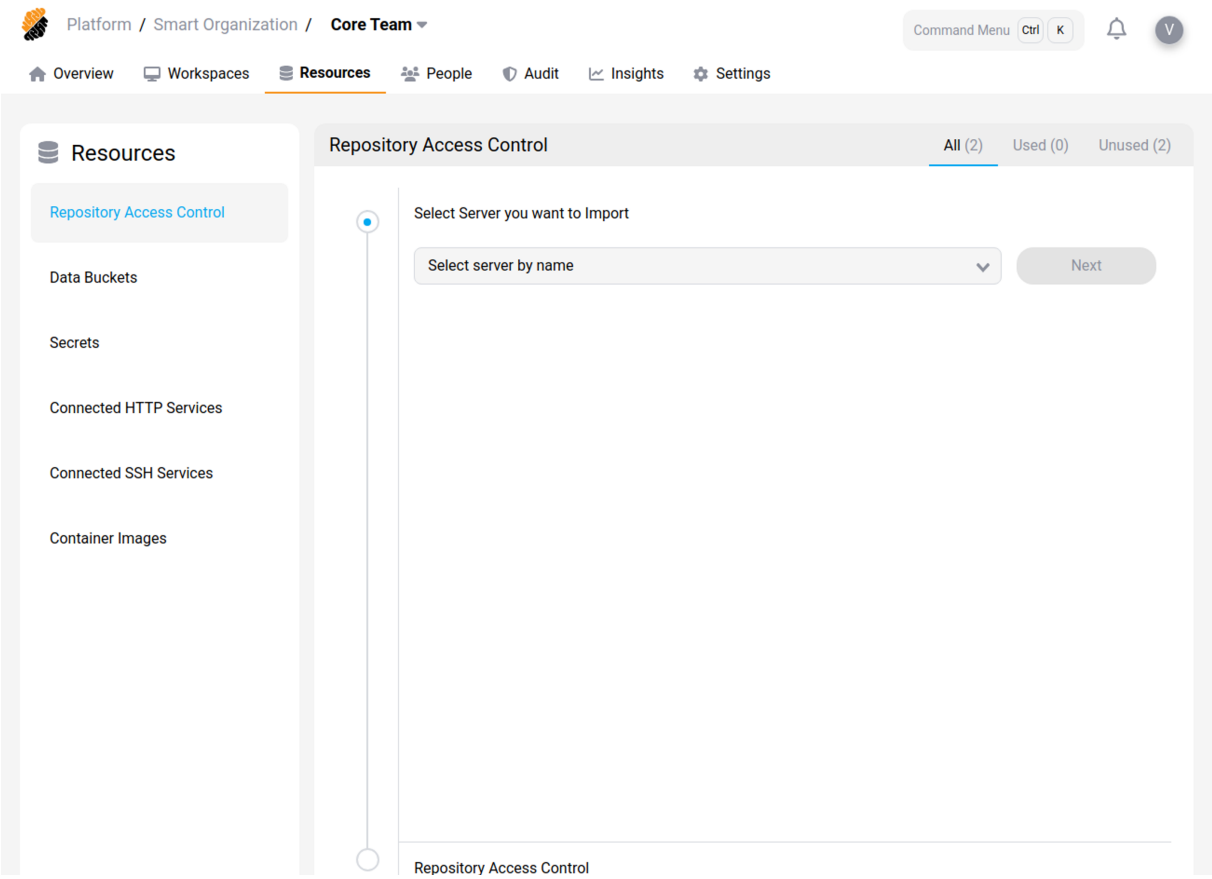

A code repository is defined by the following characteristics:

- **Basic information**: Information such as name, scope of use (platform, organization or project), the user who added it, GIT service provider e.g. GitHub, GitLab, BitBucket, URL.
- **Class Level**: This option defines the visibility for the repository based on the user's permissions.
- **Asset Information**: This option allows for providing a description of the repository.

## Import a Repository `Permission: _Resources::Import_`

You can import a code repository by pressing the "**Import Repository**"button. Make sure to select the actual provider, i.e. GitHub, GitLab or Bitbucket. The remote GIT application is scanned for code repositories and you can import the repo information by clicking the button next to the name.




## Data Buckets

October 2, 2025

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.

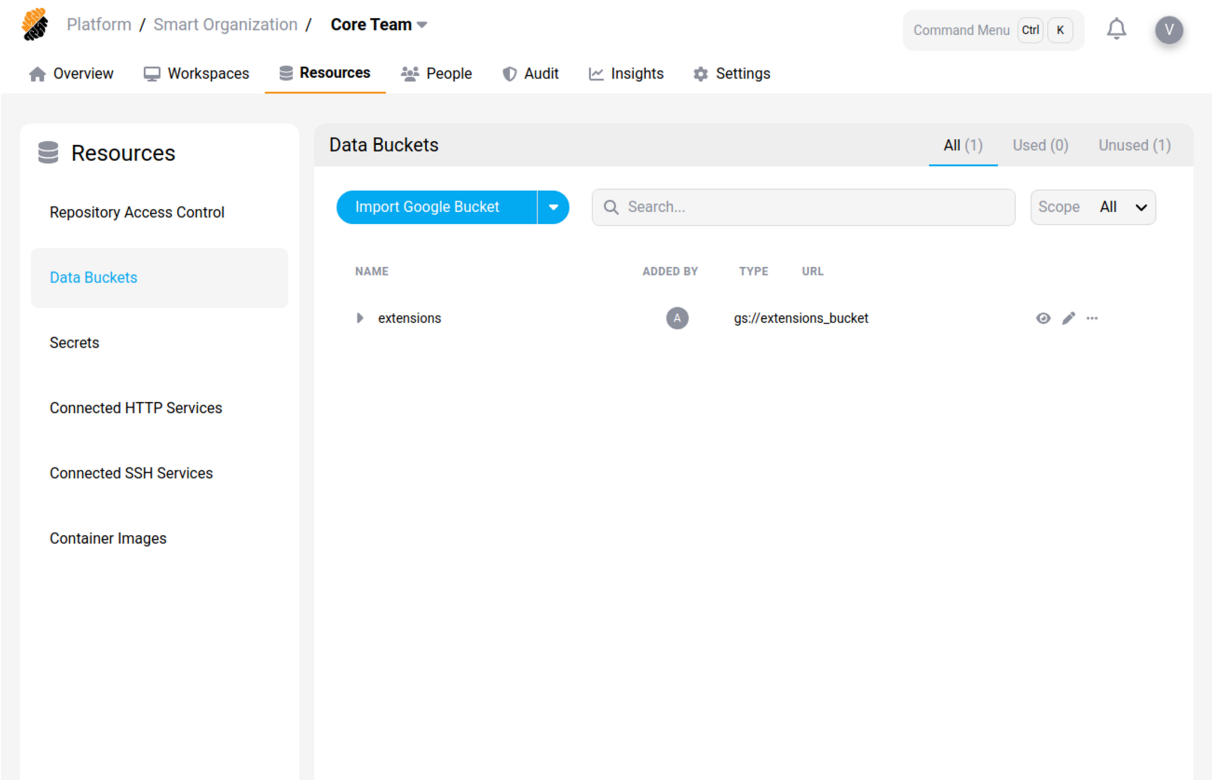
As for the other types of resources, data buckets are first imported to the platform such that they become available when creating or updating the configuration of a workspace.

- View Data Buckets
- Import a Data Bucket `Permission: Resources::Manage`

## View Data Buckets

Data Buckets used in the project are being displayed. You may filter those in use.




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*.

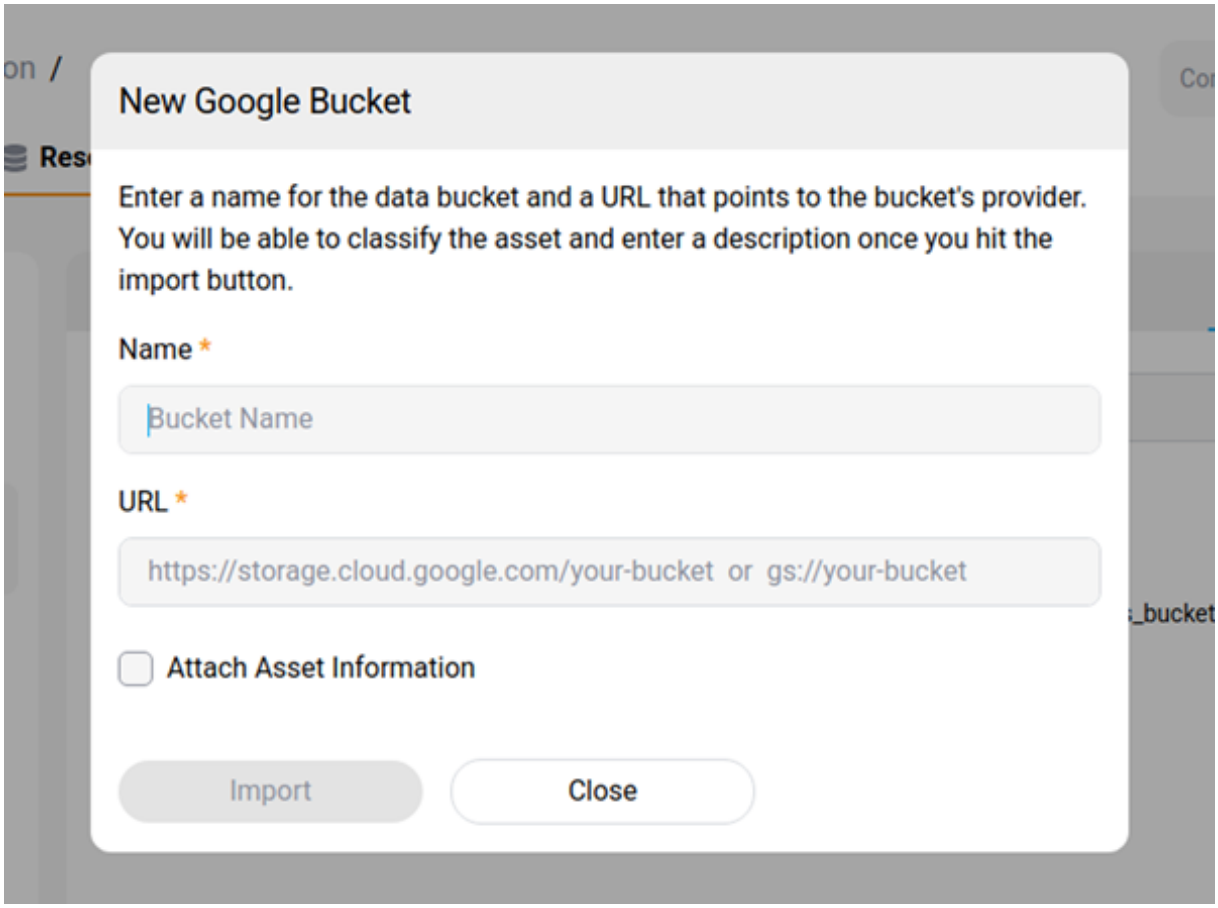
## Import a Data Bucket `Permission: _Resources::Manage_`

You can import a bucket by pressing the "**Import Bucket**"button. Make sure to select the correct provider of your bucket (Google, Amazon or Microsoft).

You will need to enter the following information:

1. **Name**, a name to identify the data bucket, and a
2. **Bucket URL** that points to the Cloud provider's storage location.




> **Info**
>
> When importing Amazon buckets, you need to specify its region to optimize the data access performance.

## Secrets

October 2, 2025

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 secrets is a service that can be provided by the platform or

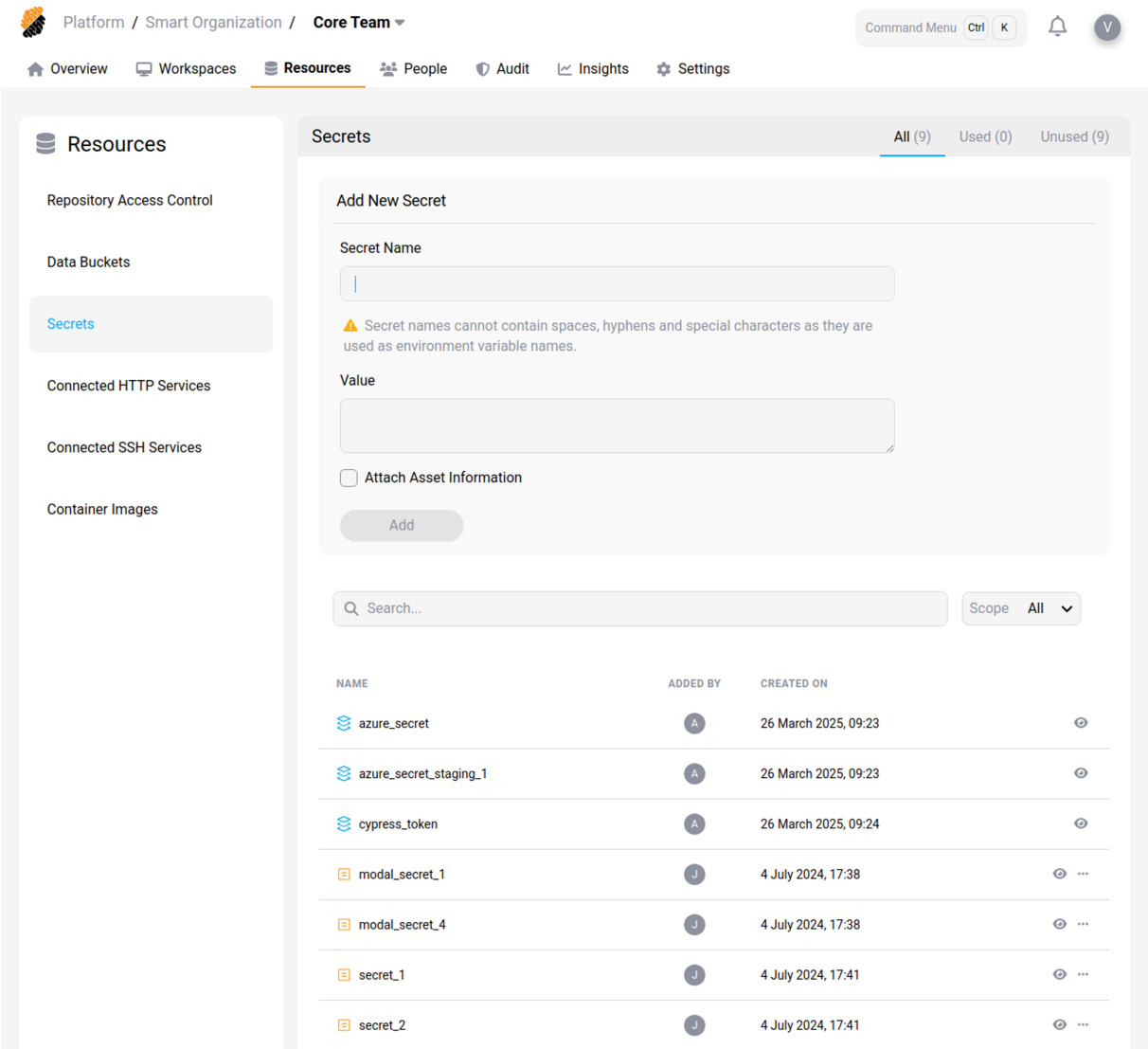
 

by an external mechanism. Once registered on the platform, secrets attached to workspaces are available in the container's filesystem as environment variables or files. This section explains how secrets are managed by the platform, but note that your platform might use an external service for that purpose.

- View Secrets
- Add a New Secret `Permission: Resources::Manage`

## 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.



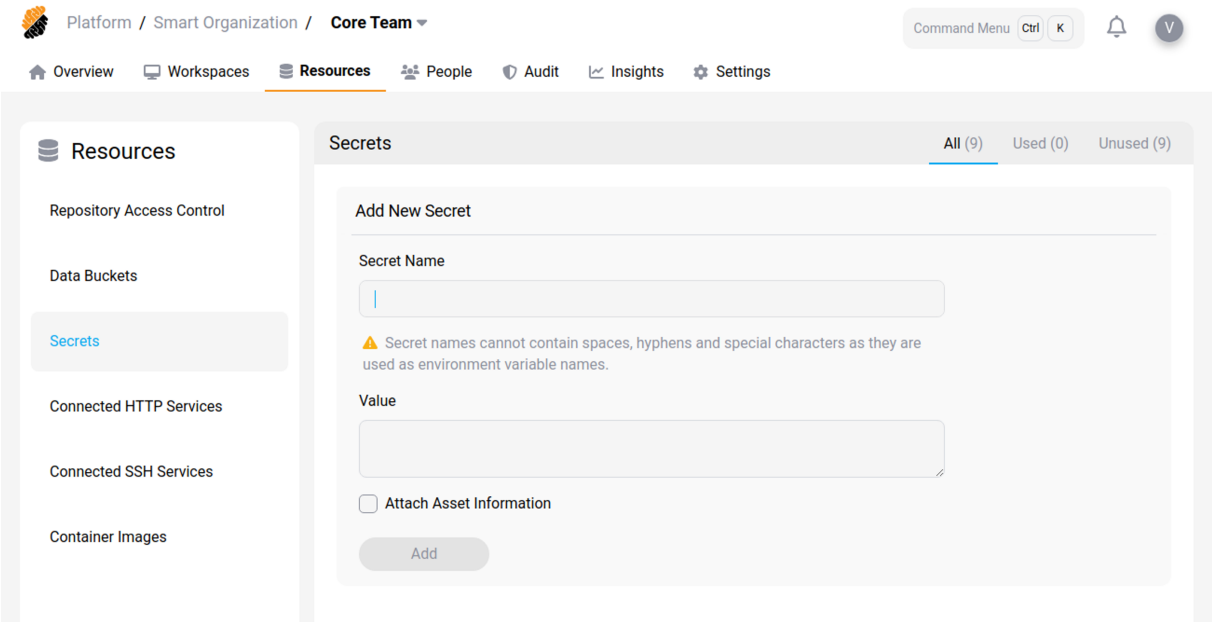

A Secret is defined by the following characteristics:

- **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 `Permission: _Resources::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.

## Connected HTTP Services

October 2, 2025

**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.

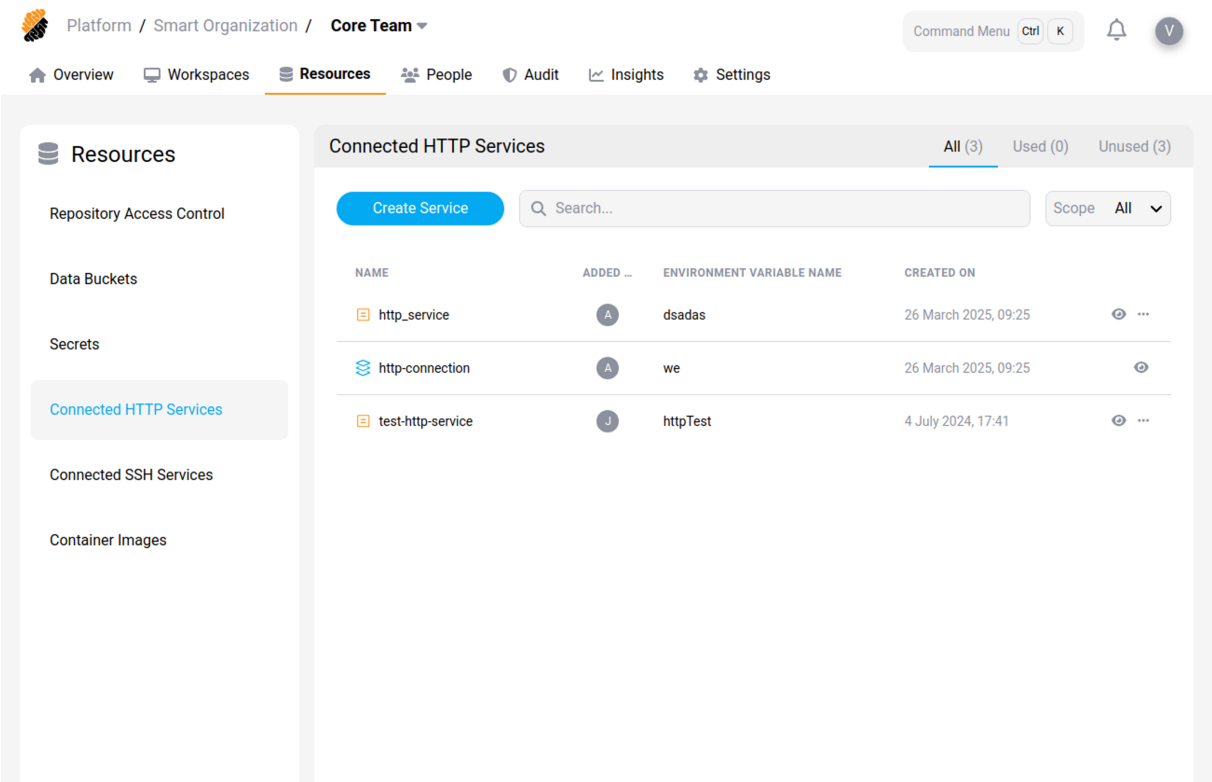
 

> **Tip**
>
> The nature and protocol of services that can be attached to workspaces depend on your platform's implementation.

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 `Permission: Resources::Manage`

## View Connected HTTP Services




Connected HTTP services are defined by the following characteristics:

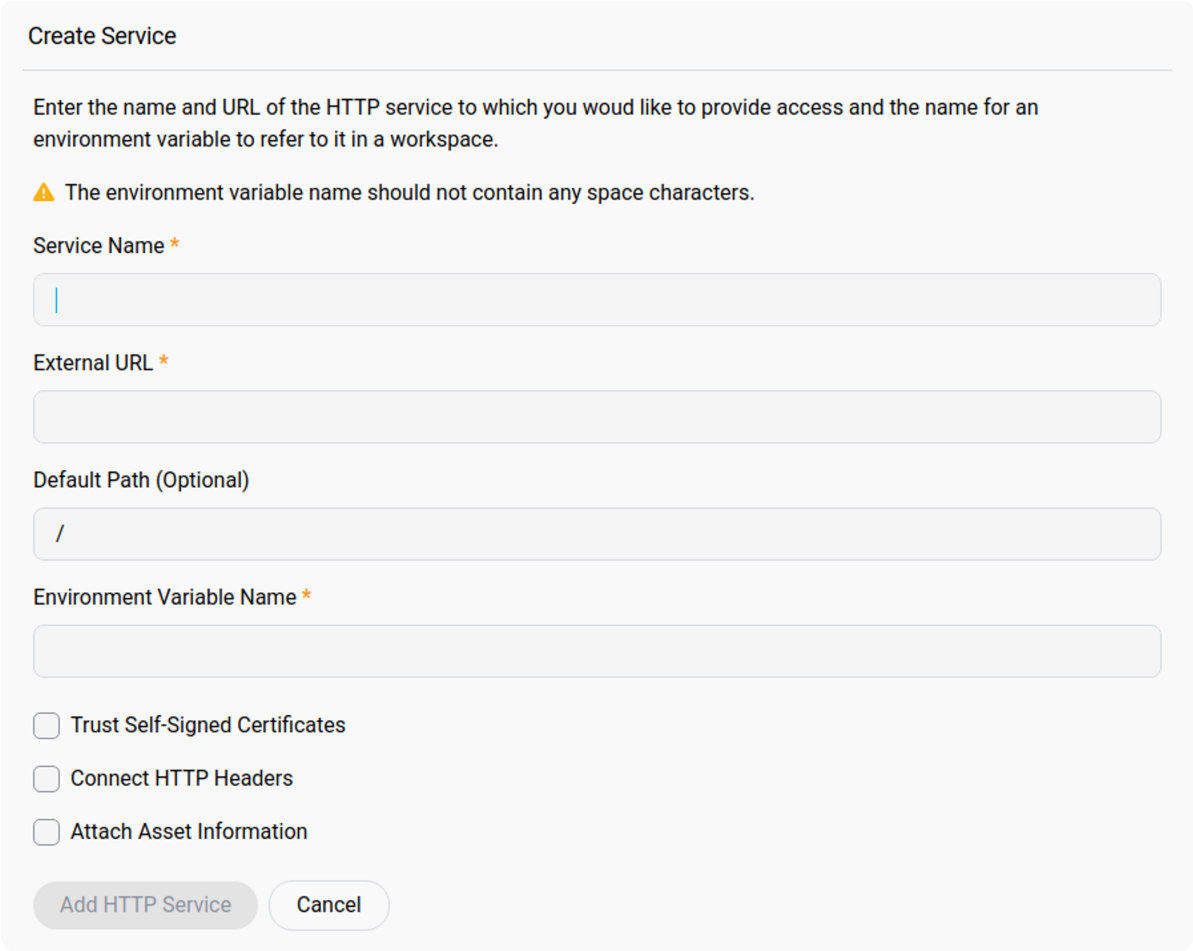
- **Basic information**: Name, scope of use (platform, organization or project), URL and tag.
- **Class Level**: This option defines the visibility for the service based on the user's permissions.
- **Asset Information**: This option allows for providing a description of the container.
- **Environmental Variable Name**: This allows access to the service simply by naming an environment variable.

### Add an HTTP Service `Permission: _Resources::Manage_`

You can register a service by selecting "**New HTTP Service**"and provide the following information:

1. **Name**, a name to identify the service,
2. **Service URL** that points to the service location,
3. **Environment Variable Name**, to name the service in the context of the container's environment,
4. **HTTP headers (optional)**, used to pass authentication data when necessary to access the service,
5. **Asset Information**, used to provide a description of the service.



# Connected SSH Services

October 2, 2025

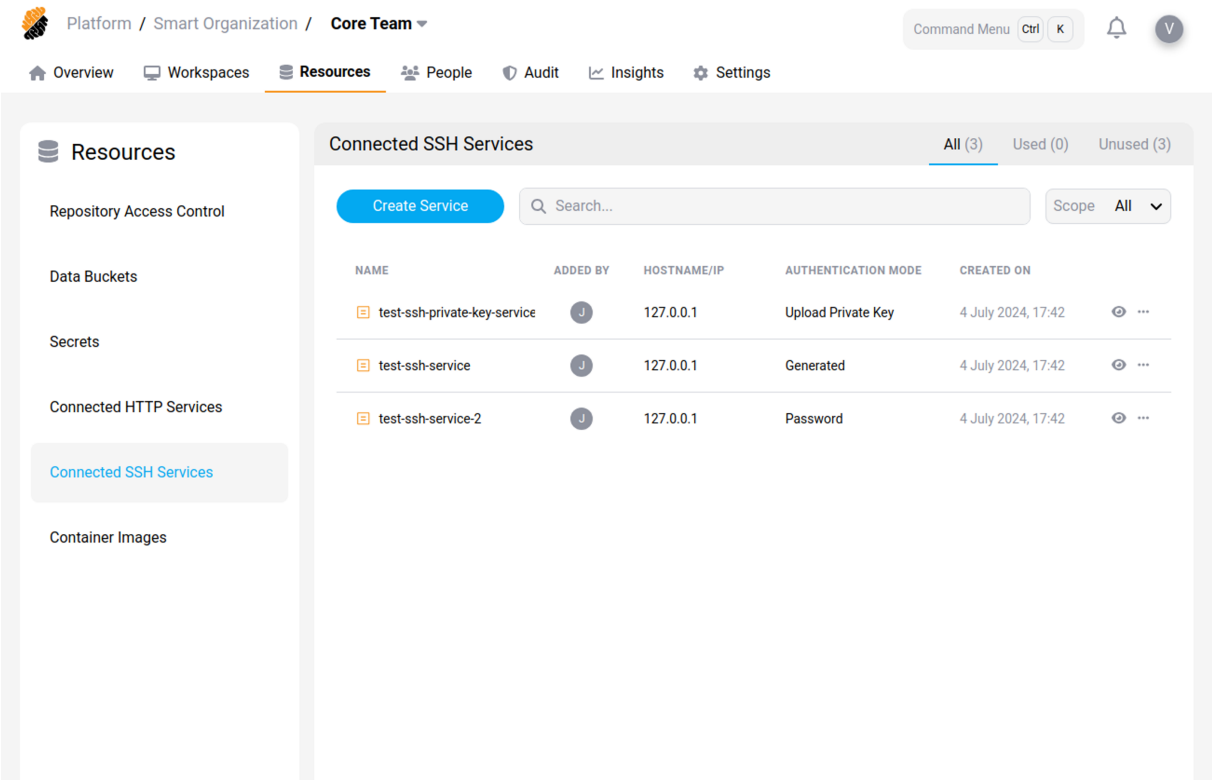
**Connected SSH Services** consist of services used for the implementation of software applications. These services are typically providing functions, data or host access via APIs over the SSH network protocol.

> **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, SSH services are attached to workspaces during the creation of the update of the workspace's settings.

- View Connected SSH Services
- Add an SSH Service `Permission: Resources::Manage`

## View Connected SSH Services




Connected SSH 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.
- **Environmental Variable Name**: This allows access to the service simply by naming an environment variable.
- **Hostname/IP**: The IP address or hostname of the SSH host,
- **Authentication Mode**: the mechanism to authenticate with the service.

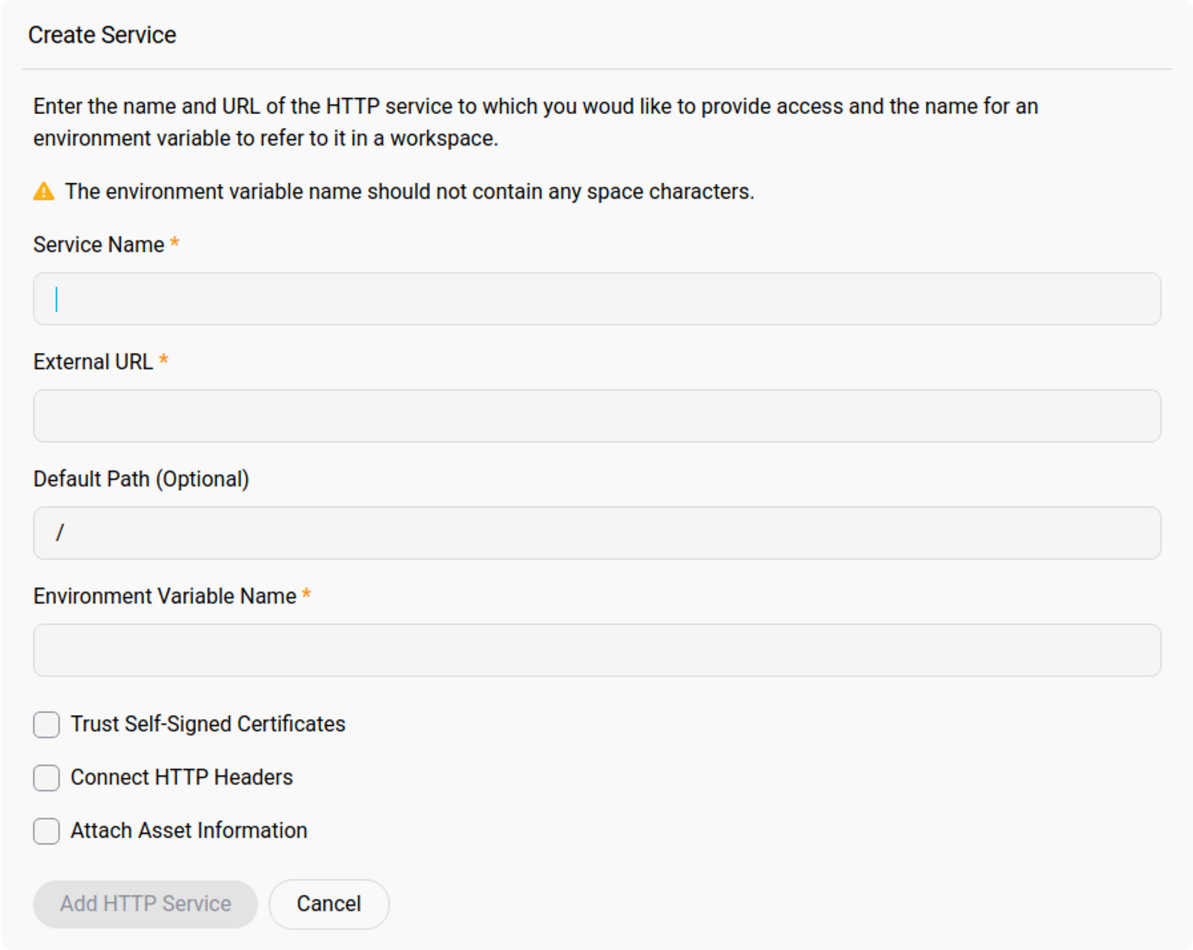
### Add an SSH Service `Permission: _Resources::Manage_`

You can register a connected service by selecting "**New SSH Service**".

You will need to enter the following information:

1. **Name**, a name to identify the host,
2. **SSH Username**, a username to access the host,
3. **Hostname or IP address of the SSH service**, that points to the host location,
4. **Port number the SSH service is running on**, a port number for the service,
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.




## Container Images

October 31, 2025

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.

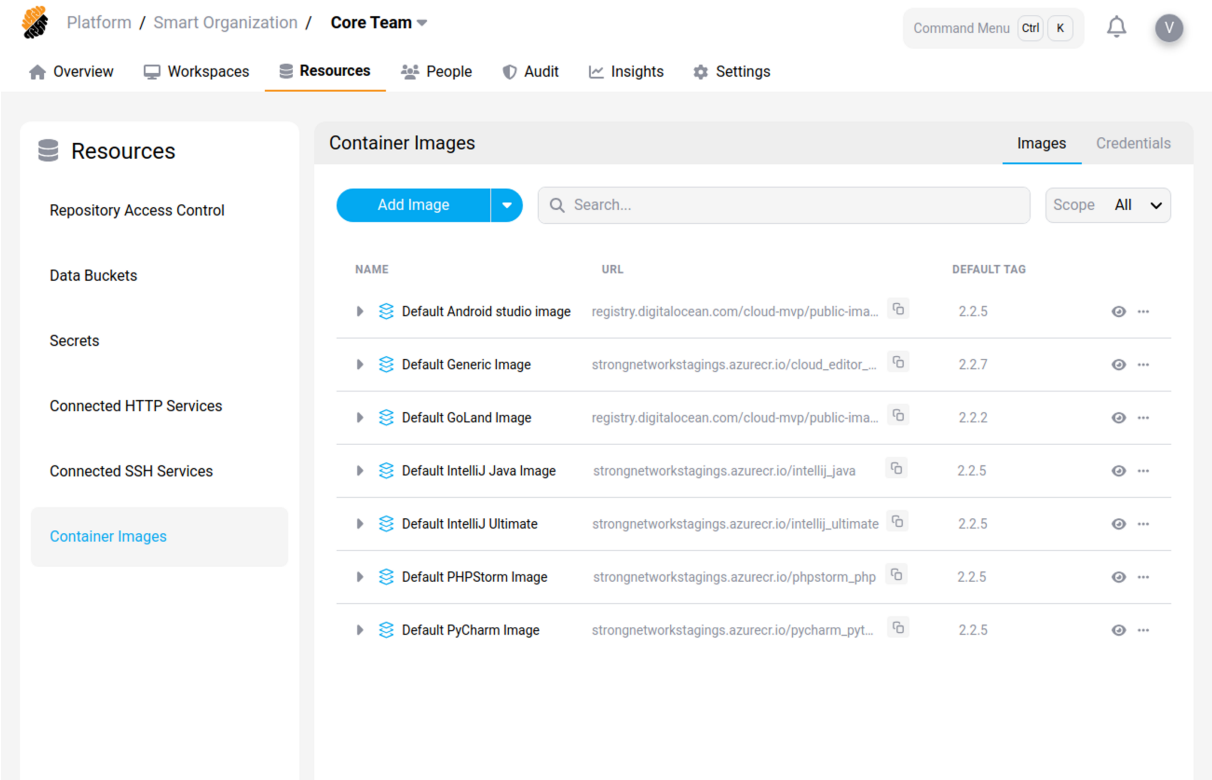
CDE images are imported from a registry as part of the resources available to users on the platform. Registries are either public or private.
For private registries, you need to provide credentials to authenticate properly before importing the image. Public registries, by definition, do not need credentials.

- View CDE Images
- Add a CDE Image `Permission: Security::Manage`
- View Registry Credentials
- Add a Registry Credential `Permission: Security::Manage`
- 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.




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.

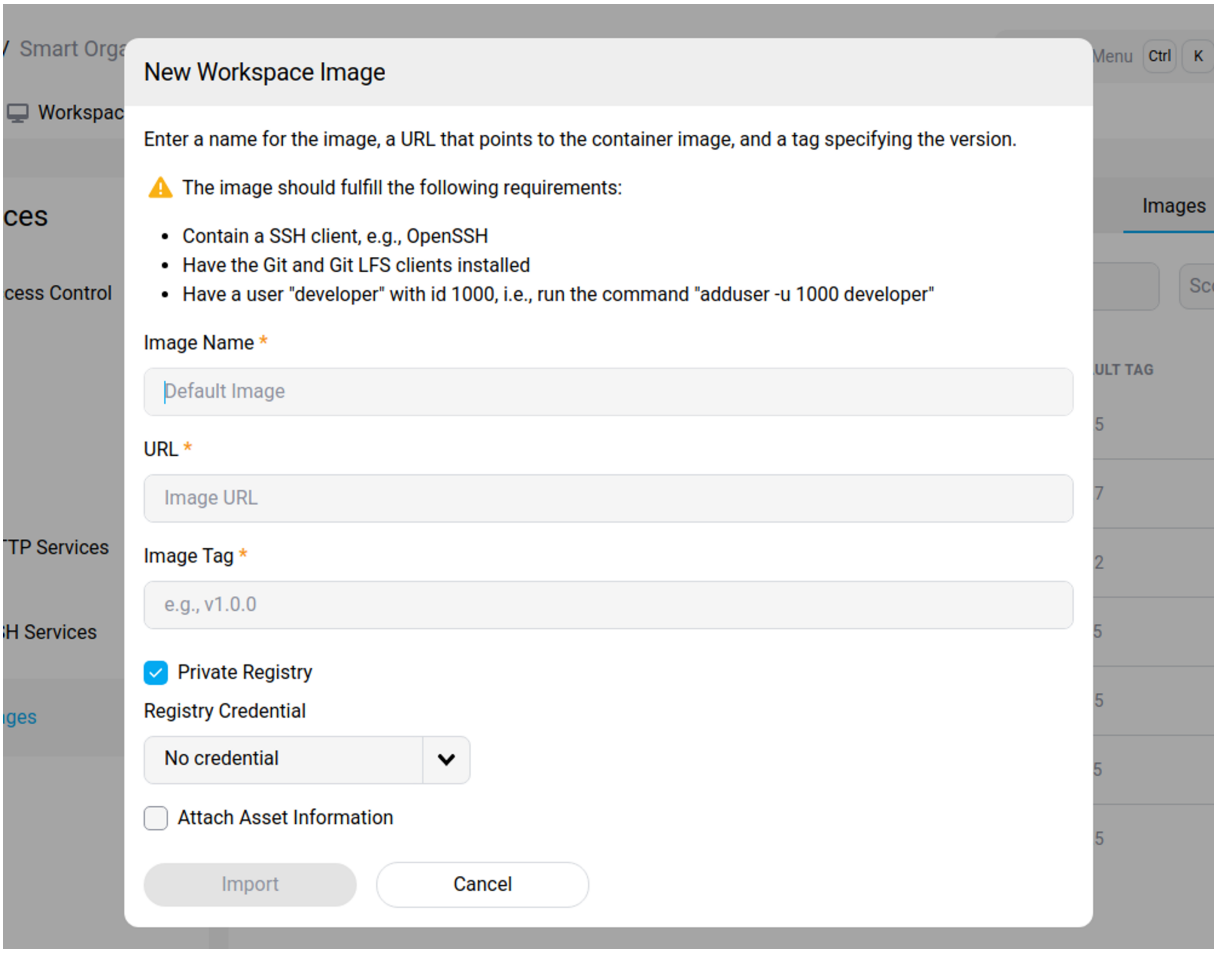
By clicking on a CDE image, you can see a list of the CDE's versions followed by basic details such as imported date, status.

## Add a CDE Image `Permission: _Security::Manage_`

You can add a CDE image by pressing the "**Add New Image**"button.
You will need to provide the following information:

1. **Name**, a name to identify the CDE,
2. **Images URL**, that points to the CDE's location,
3. **Image's latest tag**,
4. **Private registry** (optional),
5. **Asset Information** (optional).




> **Warning**
>
> The CDE image should fulfill the following requirements:

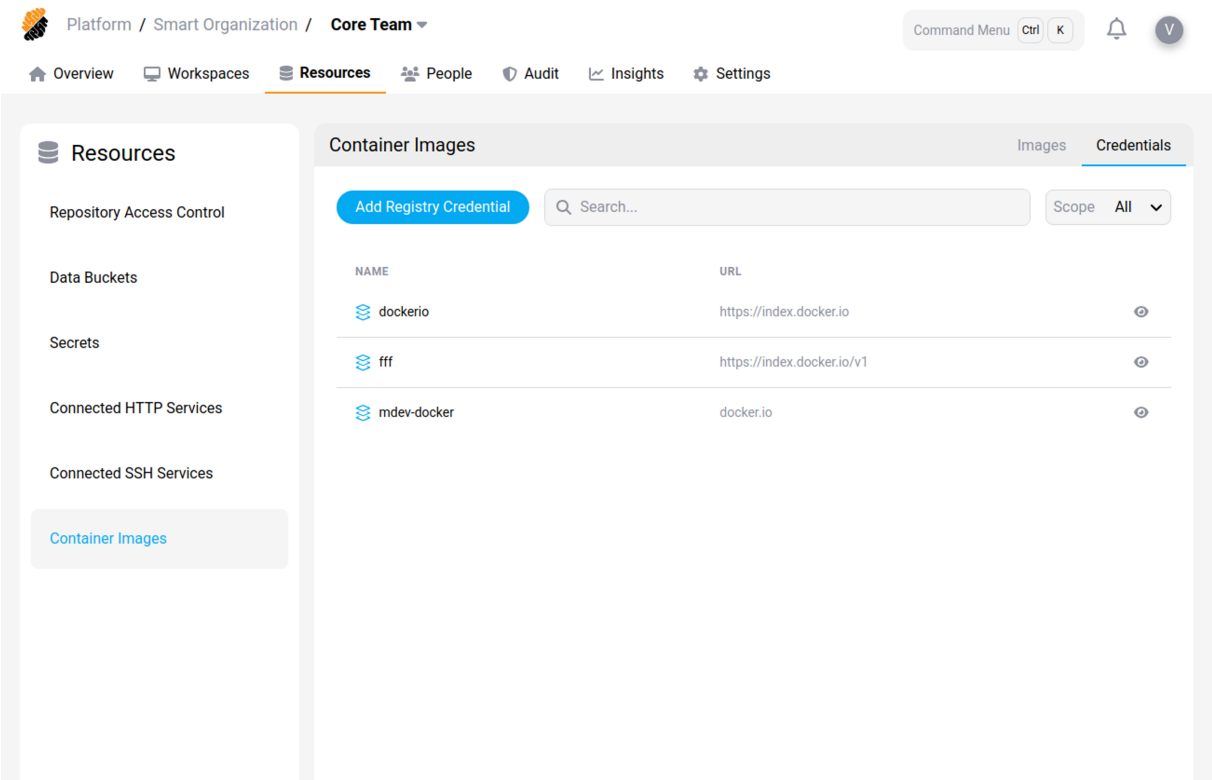
1. It should contain an SSH client.
2. It should have both GIT and GIT LFS clients installed.
3. It should have a user named "developer"with ID 1000 (this is obtained by running the command "adduser -u 1000 developer).

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.




A Registry Credential is defined by the following characteristics:

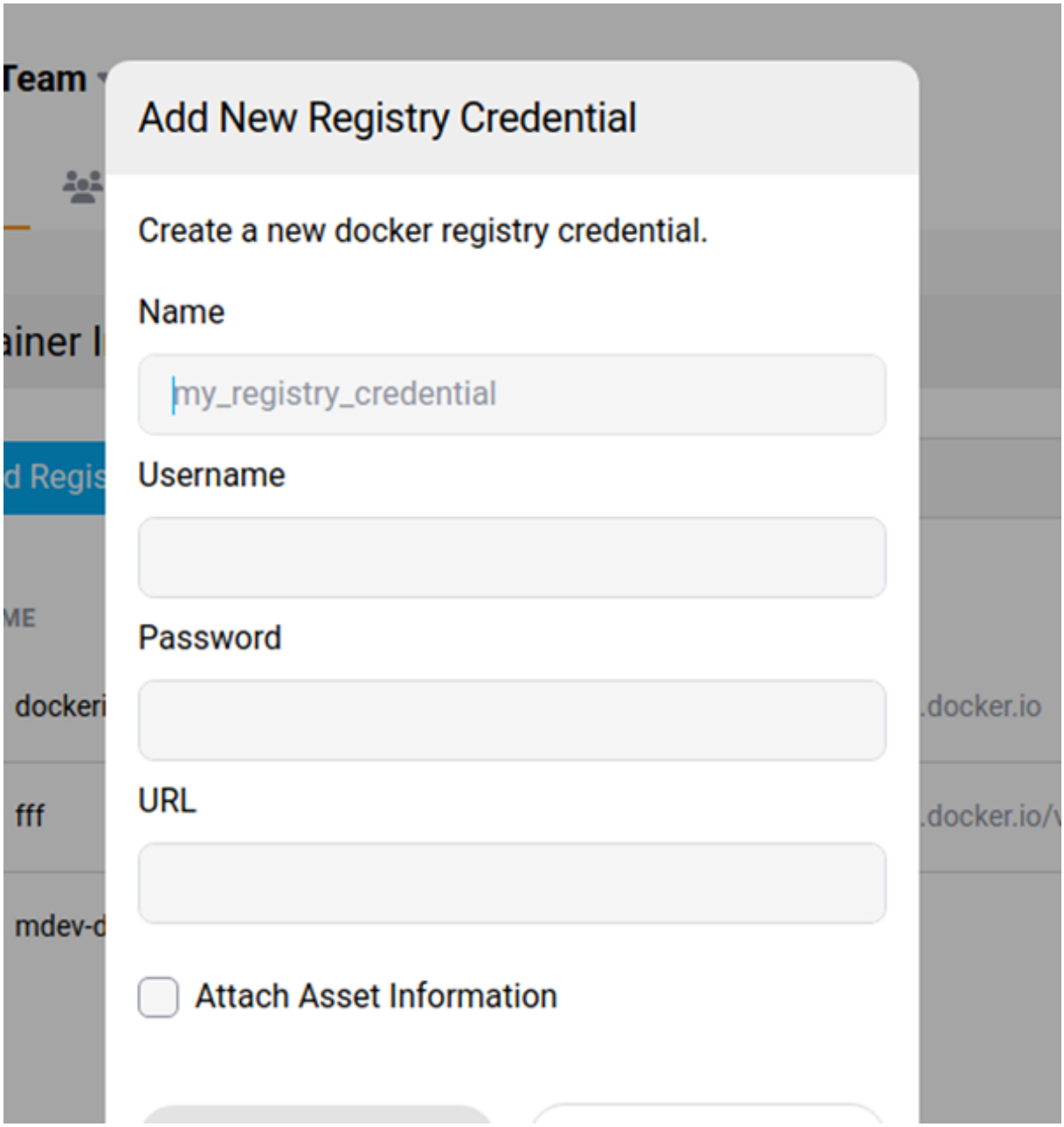
1. **Name**,
2. **Scope** and
3. a **URL**.

For security purpose, no credentials are directly exposed or available for consultation.

### Add a Registry Credential `Permission: _Security::Manage_`

You can add a Registry Credential by pressing the "**Add Registry Credential**"button.



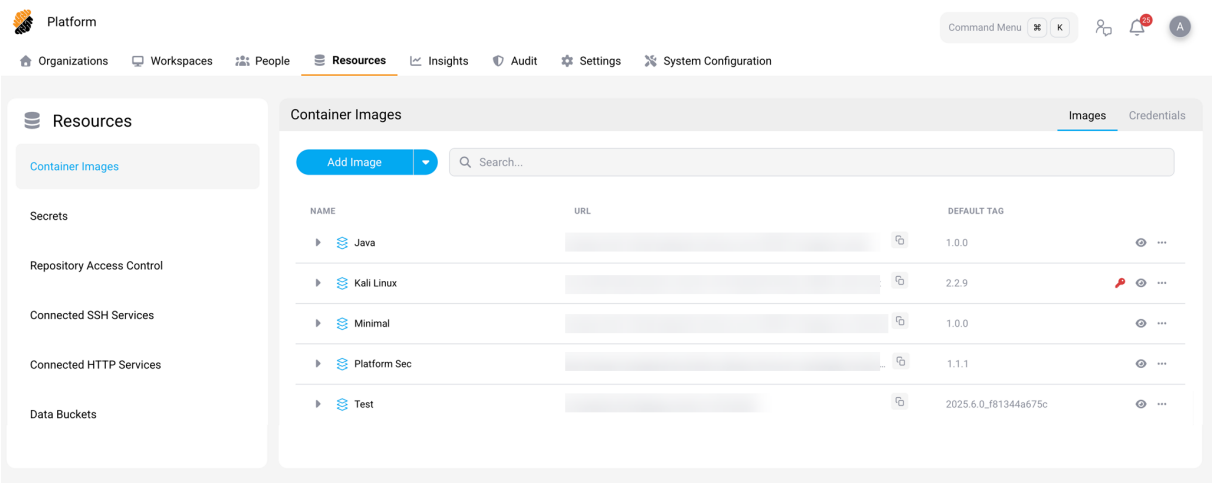

You will need to enter the following information:

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.

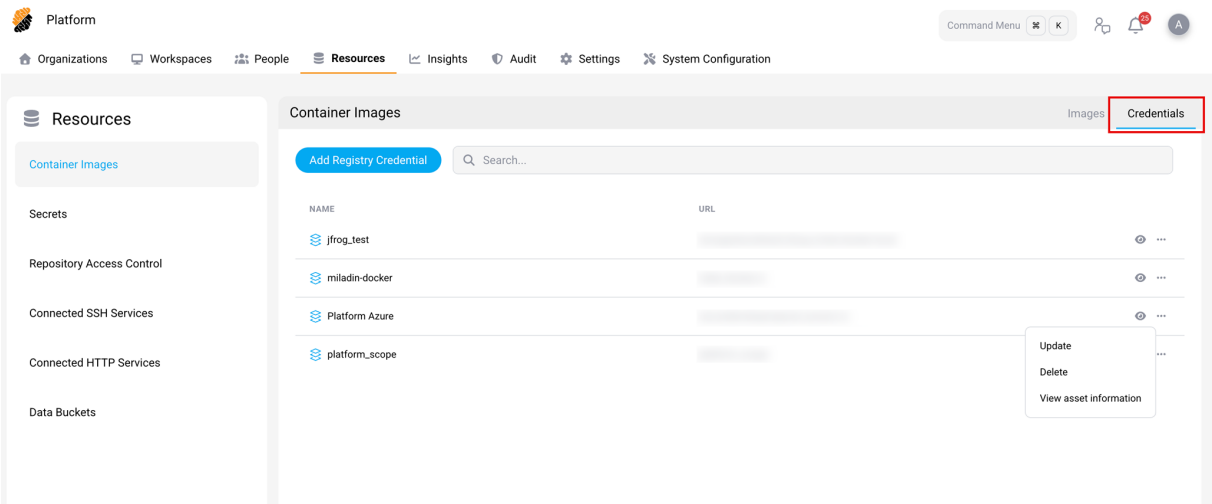
 

### Update a Registry Credential

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.




## People Page

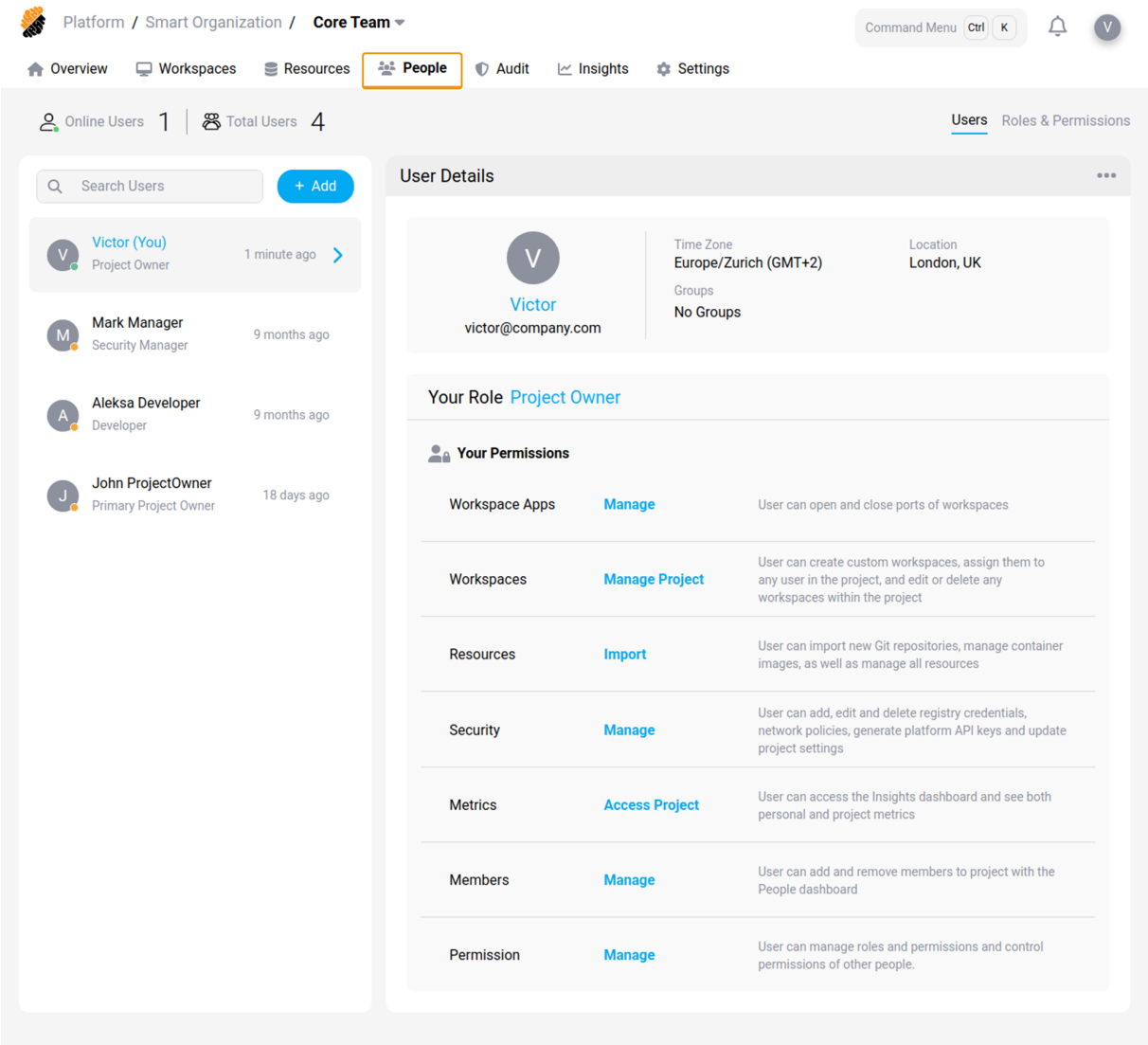
The People page contains information about users onboarded to a project, an organization or the entire platform. Switching projects or organizations therefore updates the membership in the table.

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.




> **Info:**
>
> A regular user can view all of the roles in the project and the associated permissions even if he does not have the ***Members::Manage*** permission.

## Content

- **View Users** panel.
- **Access Control** panel.

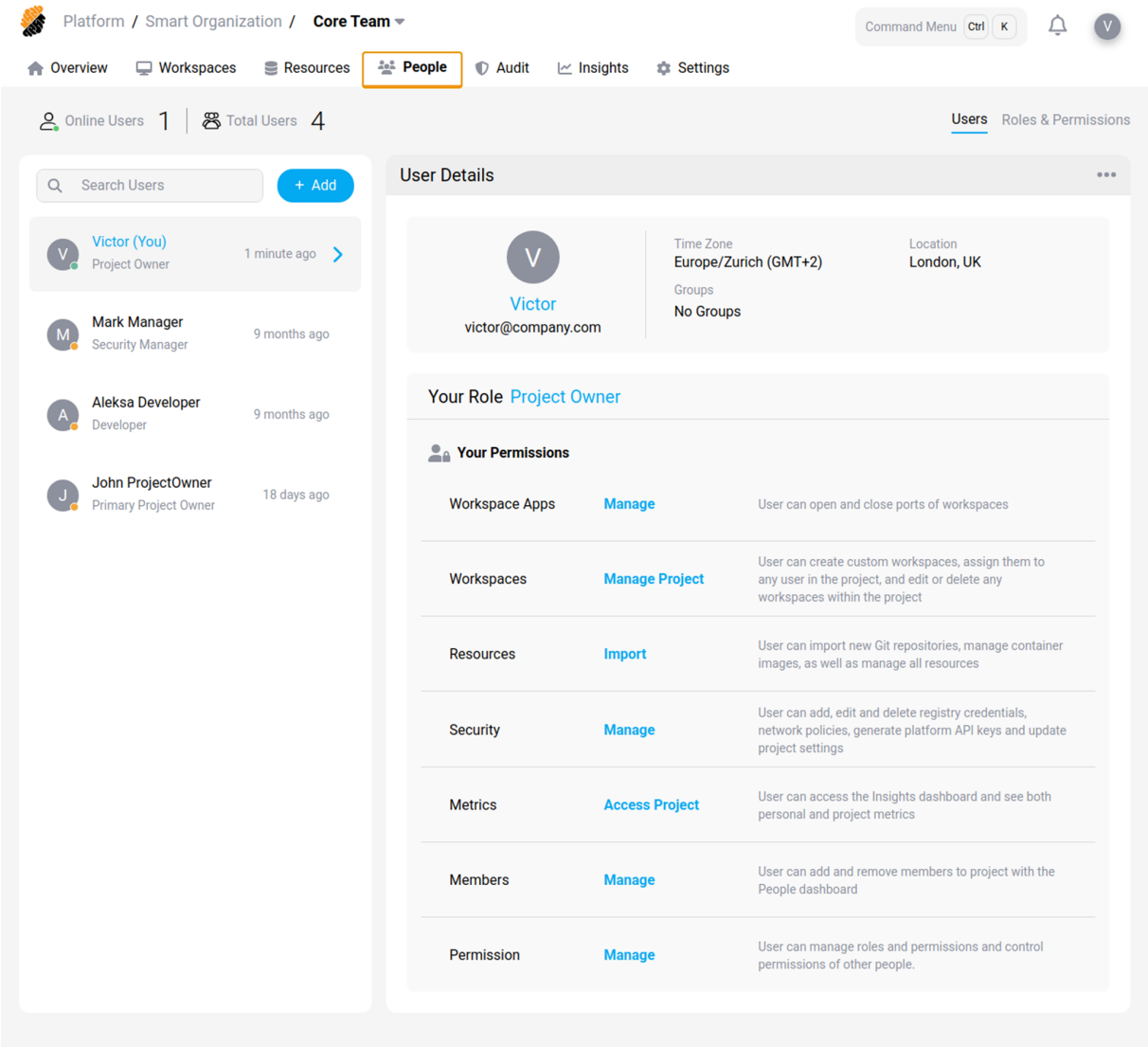
 

# Users

November 6, 2025

Users participating in the project or organization are displayed in the table at the top of the People page.

You can see the role, permissions and public details for each user.




- Search for Users
- Onboard a User in a Project `Permission:Members::Manage`
- Remove a User `Permission:Members::Manage`
- User Details Page
- Public Details

- Roles and Permissions

### Search for Users

You can look for a specific user in the project using the *search bar* or by *browsing the tabs*.

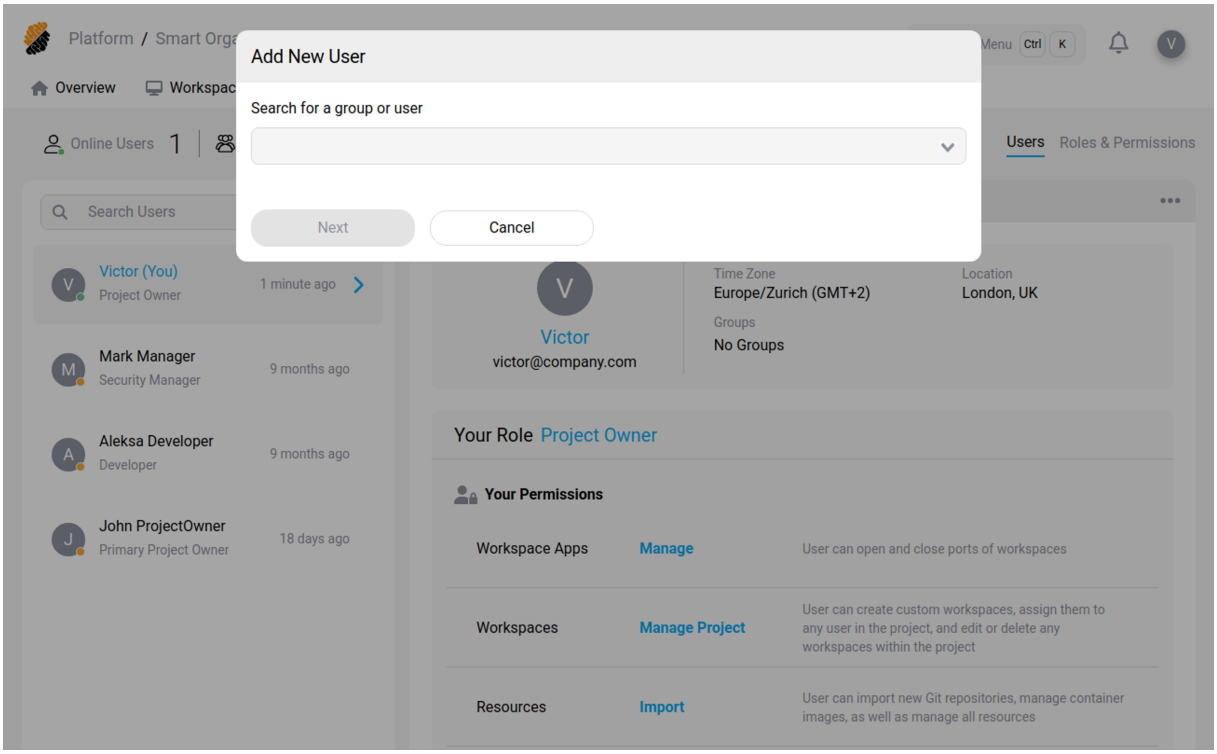
Recent activity and roles are displayed next to the username. Counts of connected users and total users are visible above the search bar.

### Onboard a User in a Project `Permission: _Members::Manage_`

By clicking on the "Add New User"button, you will be prompted to enter the email address of the user to be added. Based on the email's domain name, an appropriate identity provider (IdP) is selected. Domain names have to be registered with the Settings menu at platform-level to attach it to the correct IdP.

When the domain is not detected, a temporary password can be generated for the user. This password will have to be communicated to the user, unless a mechanism to do so is available with your instance.

Each user must be assigned a role in the scope of a project during the onboarding process. Once a user has been onboarded in the project, a workspace can be assigned to her or She can create a workspace on her own granted she has the appropriate permission, at least **Workspaces::Manage Personal**.

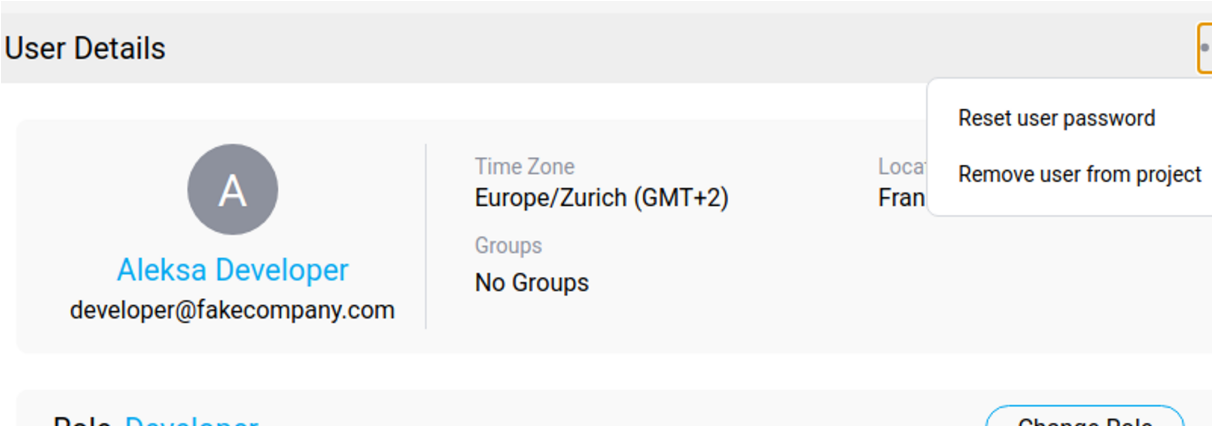


> **Tip**
>
> You can set an expiration date to the participation of the user in the project. Once the date is passed, the user won't have access to the project, the workspaces or to any resource associated with it.

### Remove a User `Permission: _Members::Manage_`

By clicking on the "**...**"icon on the top right of the user detail you can remove him from the project. The user won't have access to the project or to any resource associated with it. The user is however still in the platform database. To fully remove a user from the platform, the user has to be removed from the list of users, i.e. People Dashboard, when accessed at the platform level. This can be done with a user with a platform-level role such as *admin* or *security officer*.




### User Details Page

The user details page can be accessed in different ways:

- At the Platform or Organization hierarchy level, select a user from the list
- At the Project level, select the "**...**"icon on the right and choose **More Details**

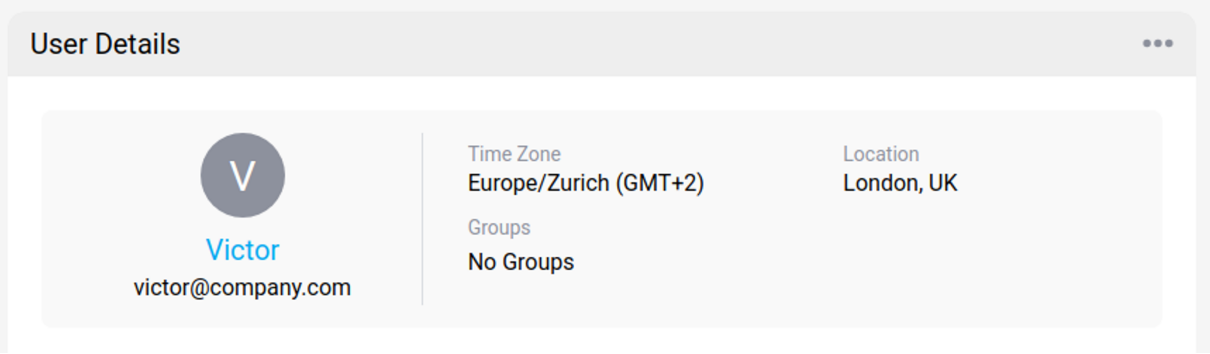
This page provides an overview of the user's access and activity, including:

- Organizations the user belongs to
- Projects the user can access
- All workspaces owned by the user
- Workspaces that have a custom schedule
- The user's personal work schedules
- Location history

### Public Details

On the user profile you can see his email address, time-zone and location.
These details are visible by everyone with the *Members::Access* permission in the project.

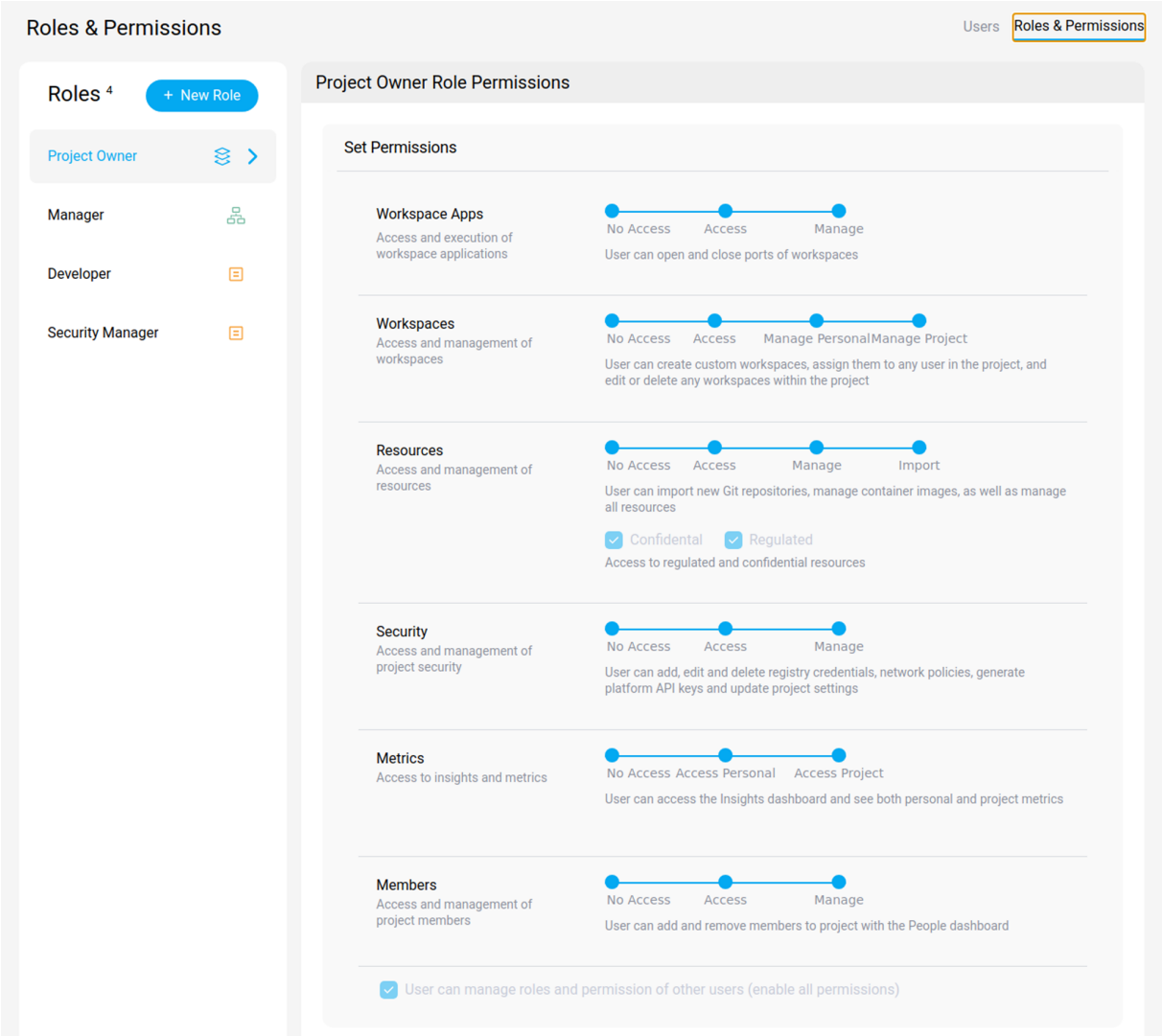



> **Tip**
>
> Your public details can be modified in the profile page

### Roles and Permissions

On the user profile you can view your current role in context of the currently selected project.




> **Tip**
>
> Tip for privileged users with permission *Members::Manage*
> The user role can be modified using the user table found on the people page.

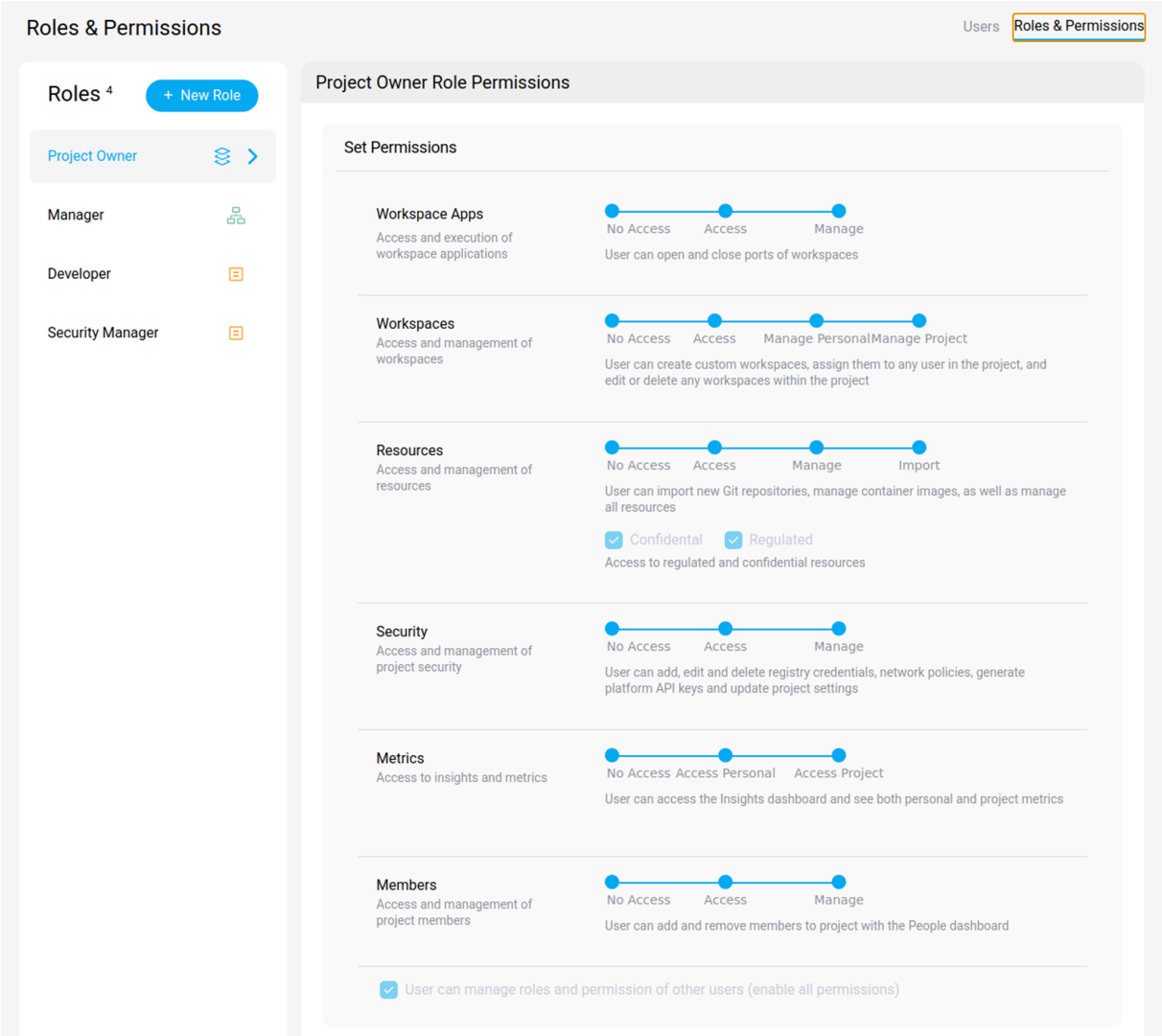
Refer to the Access Control page for more details around the access control policies on the platform.

## Access Control

October 2, 2025

Roles and permissions in the organization are displayed on the People page.
If you are a *project owner*, you can create new roles or update existing ones from the access control panel.

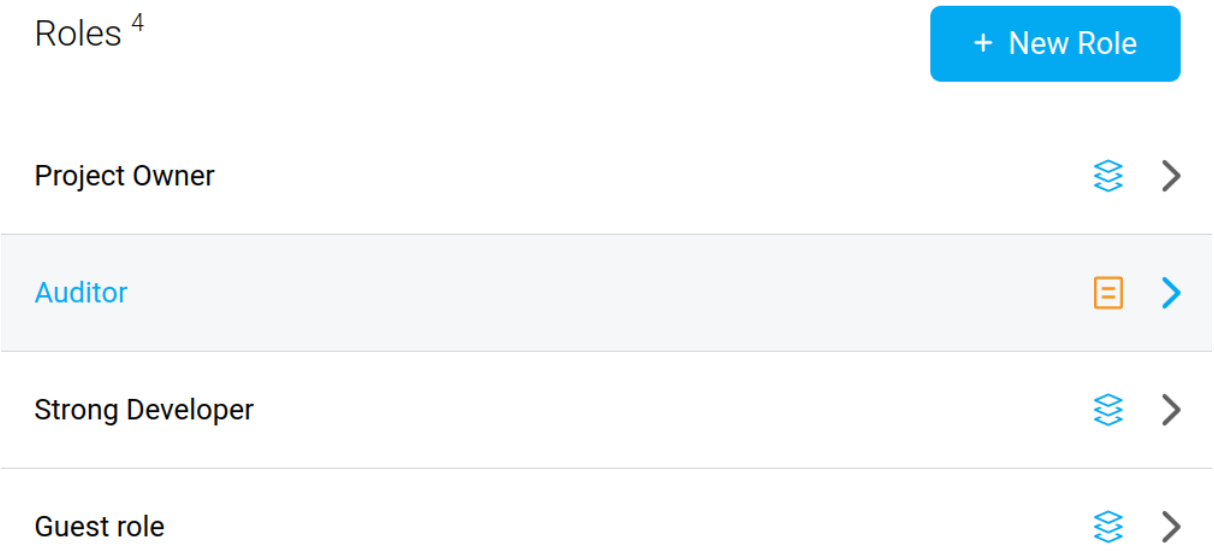



- Roles
- Default roles
- Create a new role `Project Owner`
- Permissions

## Roles

Roles define a set of permissions given to a user or a group of user.
They allow to determine the rights given to each user. Roles are project bound. This means that the same user may have a different role depending on the project. Roles defined on the project level are only available within that project.




> **Warning**
>
> Roles are a crucial element to consider when securing your resources. Roles must be attributed following a **least privilege** policy to avoid any unwarranted access.

### Default roles

There are 4 default roles in a standard project: **Guess**, **Developer**, **Manager** and **Project Owner**. They are meant for the following use:

- **Guest**: The guest role allows a user to view the platform without having access to sensitive data or the ability to make any modifications.
- **Developer**: The "default"developer will be able to create workspaces based on admin-defined project rules.
- **Manager**: The manager has all the tech lead's permissions.
- **Project Owner**: The project owner has all the manager's permissions, in addition to accessing the project's audit and manage the user's security feature, such privilege elevation.

To each role is attached the set of permissions described below.

Refer to the permissions section for an explanation about each permission.

| Permission | Guest | Developer | Manager | Project Owner |
|---|---|---|---|---|
| **Workspace Apps::Access** | **Yes** | **Yes** | **Yes** | **Yes** |

| Permission | Guest | Developer | Manager | Project Owner |
|---|---|---|---|---|
| **Workspace Apps::Manage** | **Yes** | **Yes** | **Yes** | **Yes** |
| **Workspaces::Access** | **Yes** | **Yes** | **Yes** | **Yes** |
| **Workspaces::Manage Personal** | No | **Yes** | **Yes** | **Yes** |
| **Workspaces::Manage Project** | No | **Yes** | **Yes** | **Yes** |
| **Resources::Access** | **Yes** | **Yes** | **Yes** | **Yes** |
| **Resources::Manage** | No | **Yes** | **Yes** | **Yes** |
| **Resources::Import** | No | No | No | **Yes** |
| **Resources::Regulated** | No | **Yes** | **Yes** | **Yes** |
| **Resources::Confidential** | No | **Yes** | **Yes** | **Yes** |
| **Security::Access** | No | **Yes** | **Yes** | **Yes** |
| **Security::Manage** | No | No | No | **Yes** |
| **Metrics::Access Personal** | No | **Yes** | **Yes** | **Yes** |
| **Metrics::Access Project** | No | No | **Yes** | **Yes** |
| **Members::Access** | No | **Yes** | **Yes** | **Yes** |
| **Members::Manage** | No | No | **Yes** | **Yes** |

### Create a new role `Project Owner`

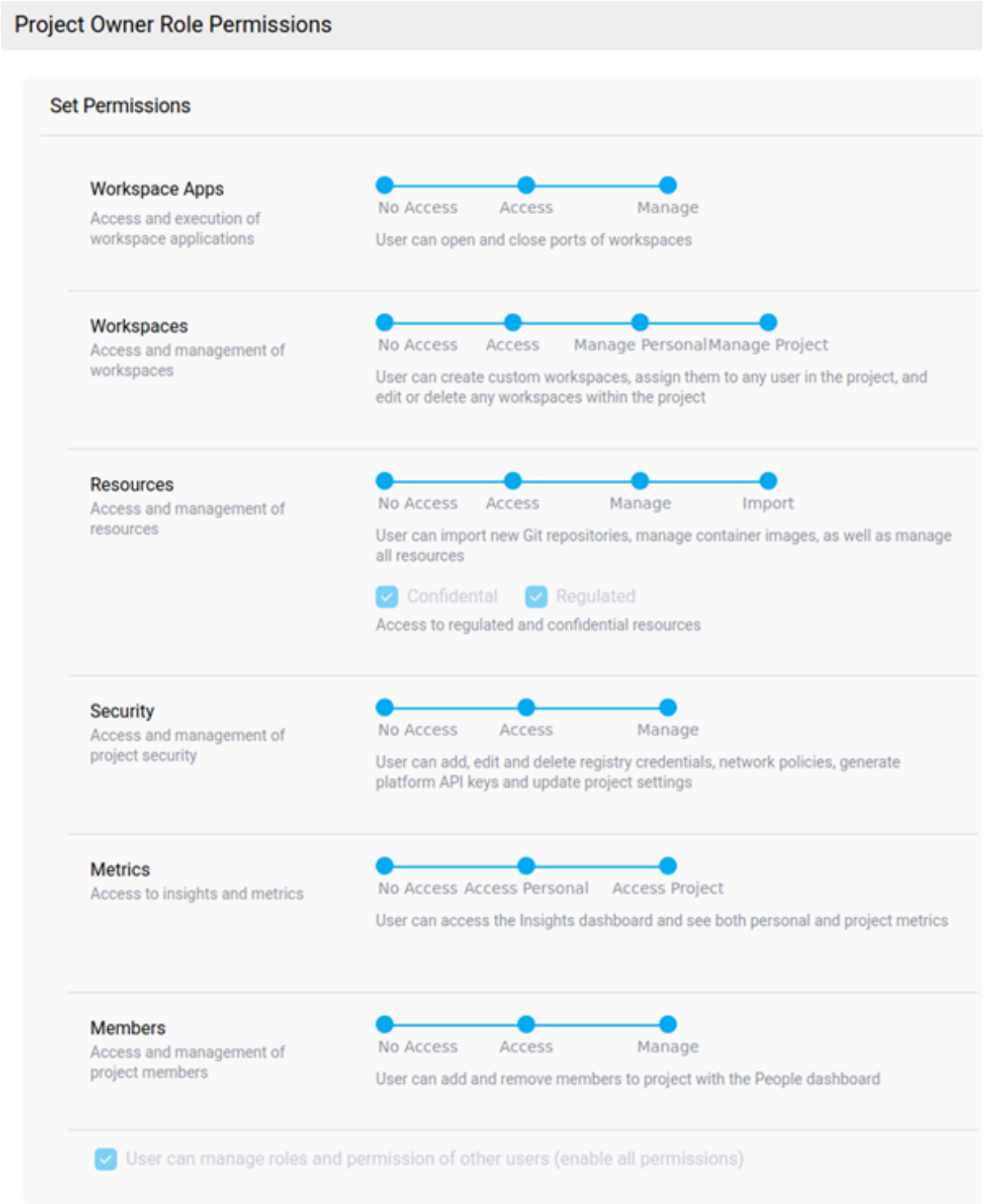
By clicking on the button at the top left of the **access control** panel, you can create a new role. Select a name and the set of permissions that characterize the new role.

> **Warning**
>
> Granted permissions must follow a **least privilege** policy.
> Be careful when naming a role, a poorly chosen name can be misused and end up giving too much privilege to a user.

## Permissions

Permissions describe the rights given to a user for a specific access.




Please find below the detail of each access mentioned above.

| Permissions | Description |
|---|---|
| **Workspace Apps::No Access** | The user cannot access apps running on the workspace. |
| **Workspace Apps::Access** | The user can access and view apps shared with the user by other users. |
| **Workspace Apps::Manage** | The user can open and close ports of workspaces. |
| **Workspaces::No Access** | User cannot access workspaces |
| **Workspaces::Access** | User can access workspaces assigned to her, but cannot edit properties or modify access control to resources, or delete her workspace. |
| **Workspaces::Manage Personal** | User can create personal workspaces (i.e. with admin pre-defined characteristics), manage access control to the project resources, and delete personal workspaces. |
| **Workspaces::Manage Project** | User can create custom workspaces and assign it to any user in the project. The user can edit or delete any workspaces in the project. |
| **Resources::No Access** | The user cannot access the Resources dashboard and see registered resources. |
| **Resources::Access** | The user can access the Resources dashboard and see registered resources, but cannot edit or delete them. |
| **Resources::Manage** | The user can access the Resources dashboard and see, edit and delete project repositories, secrets, external services and data buckets. |
| **Resources::Import** | The user can import new git repositories, container images and SAML connected apps, as well as manage all resources. |
| **Resources::Regulated** | The user can access resources registered as regulated, i.e. falling under some regulations |
| **Resources::Confidential** | The user can access resources registered as confidential such as intellectual property, etc. |
| **Security::No Access** | The user does not have access to security metrics. |
| **Security::Access** | The user has access to the Audit dashboard, define network policies (Resource Dashboard), but cannot add, edit or delete them. |

| Permissions | Description |
|---|---|
| **Security::Manage** | The user can add, edit and delete workspace images, registry credentials, network policies, generate platform API keys and update project settings. |
| **Metrics::No Access** | The user has no access to the Insights dashboard. |
| **Metrics::Access Personal** | The user has access to the Insights dashboard and see only personal metrics. |
| **Metrics::Access Project** | The user has access to the Insights dashboard and see both personal and project-level metrics. |
| **Members::No Access** | The user cannot see the project's members (no People dashboard). |
| **Members::Access** | The user can see the project's members in the People dashboard. |
| **Members::Manage** | The user can add and remove members to the project with the People dashboard. |

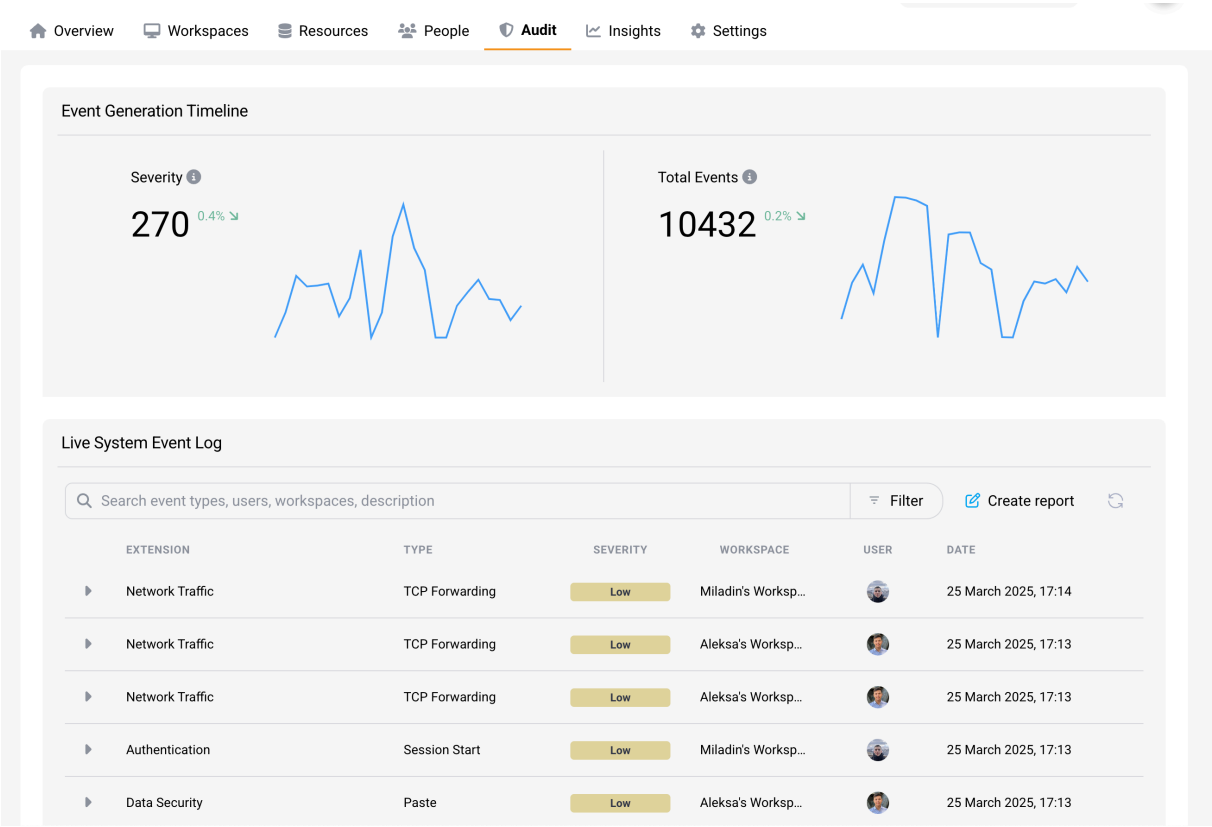
## Audit Page

October 2, 2025

`Permission: _Security::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.




- Event Generation Timeline
- Event Logs
- Filtering Logs
- Log Display

## Event Generation Timeline

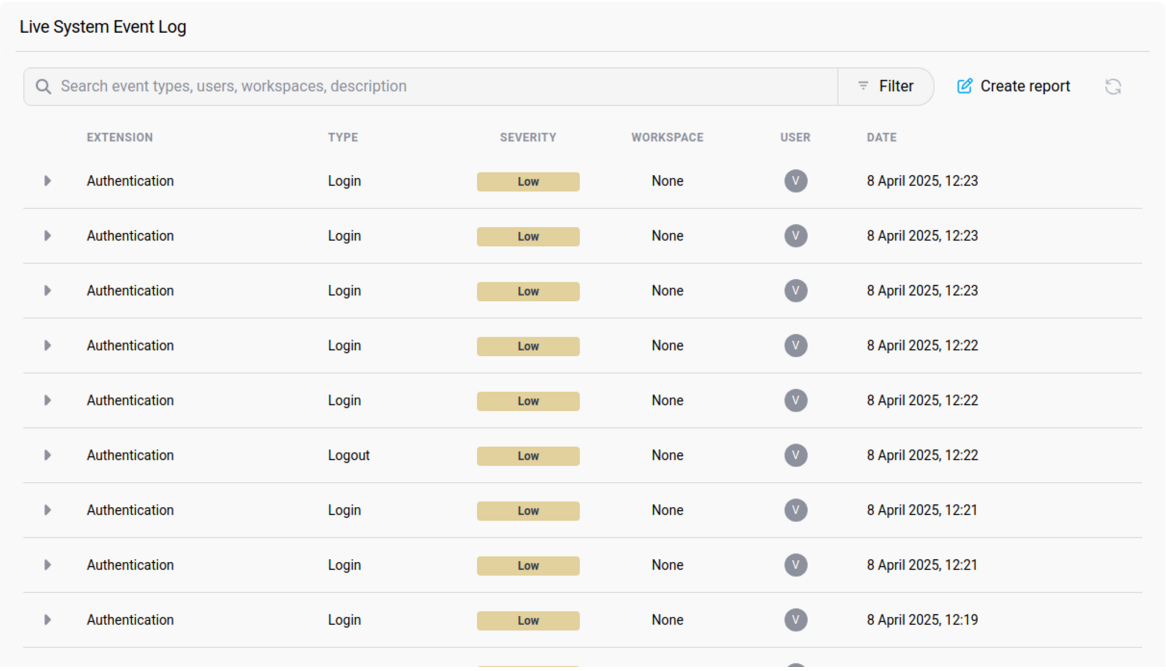
The event generation timeline are two graphs, where the first one shows the severity level of triggered events over time, and the other displays the total amount of triggered events over time.

## 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,

- **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.

## Real-time Auditing Section: Event Log Catalog Reference

September 29, 2025

The tables below offers a quick reference to events monitored in real time on the Citrix Secure Developer Spaces (SDS) 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

| 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

| 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 |

| 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**

| ID | Category | Event Type | Event Description | Attributes |
|---|---|---|---|---|
| 21 | System | WorkspaceSpecsUpdated | Modifications or updates made to the specifications of a workspace | |

## 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 |
| 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 |
| 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 |
| 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 |

| ID | Category | Event Type | Event Description | Attributes |
|---|---|---|---|---|
| 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 |
| 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 |

| ID | Category | Event Type | Event Description | Attributes |
|---|---|---|---|---|
| 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 |
| 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 |

| ID | Category | Event Type | Event Description | Attributes |
|---|---|---|---|---|
| 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 |
| 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 |

| ID | Category | Event Type | Event Description | Attributes |
|---|---|---|---|---|
| 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

| 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 |

| Attributes | Attribute Description |
|---|---|
| git_branch | The git branch name, if applied |
| 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 |
| o_auth_app | Third party app name, if applied |
| project_id | Project ID |
| project_name | Project name |
| 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 |
| --- | --- |

## 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** `Permission: _Metrics::Access Project_`
- **Container Process Metrics**

## Resource Allocation

October 2, 2025

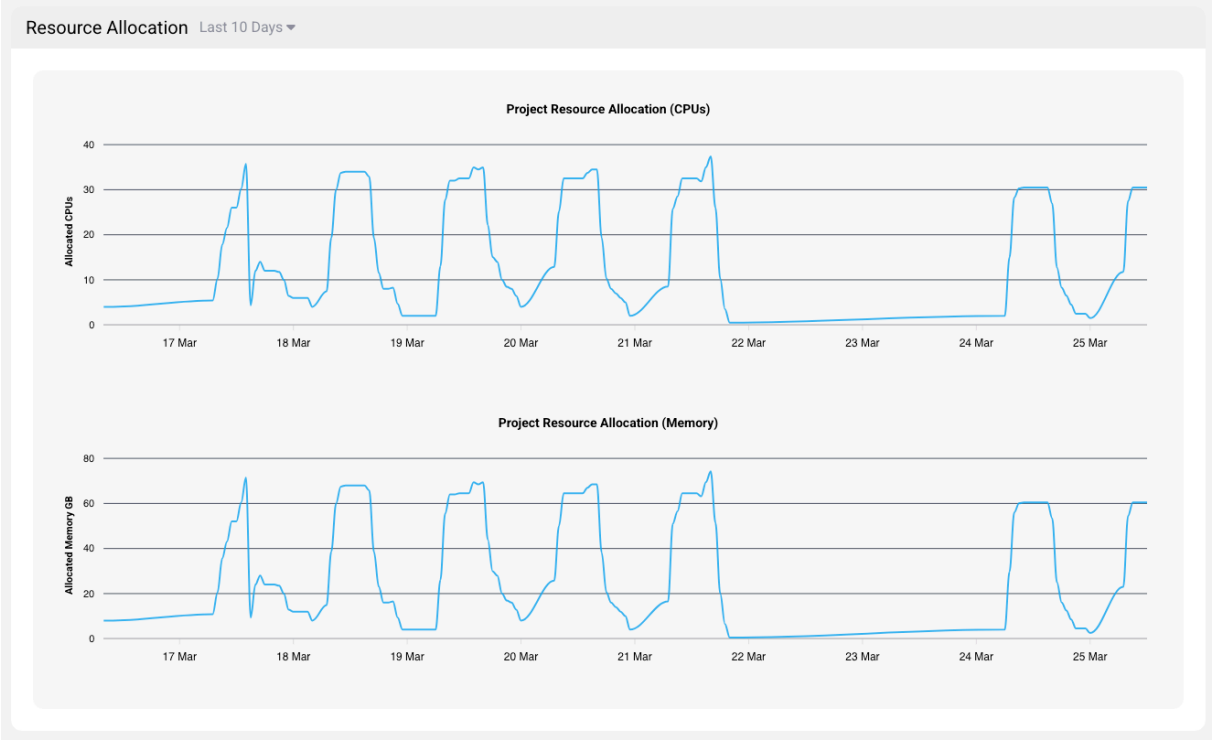
`Permission:_Metrics::AccessProject_`

Within the **Resource Allocation** tab, you can view the current usage of resources by your workspace.

- Resource Allocation Graph

### Resource Allocation Graph




You can also view a sortable list of the total consumption based on activities for each workspace in the project.

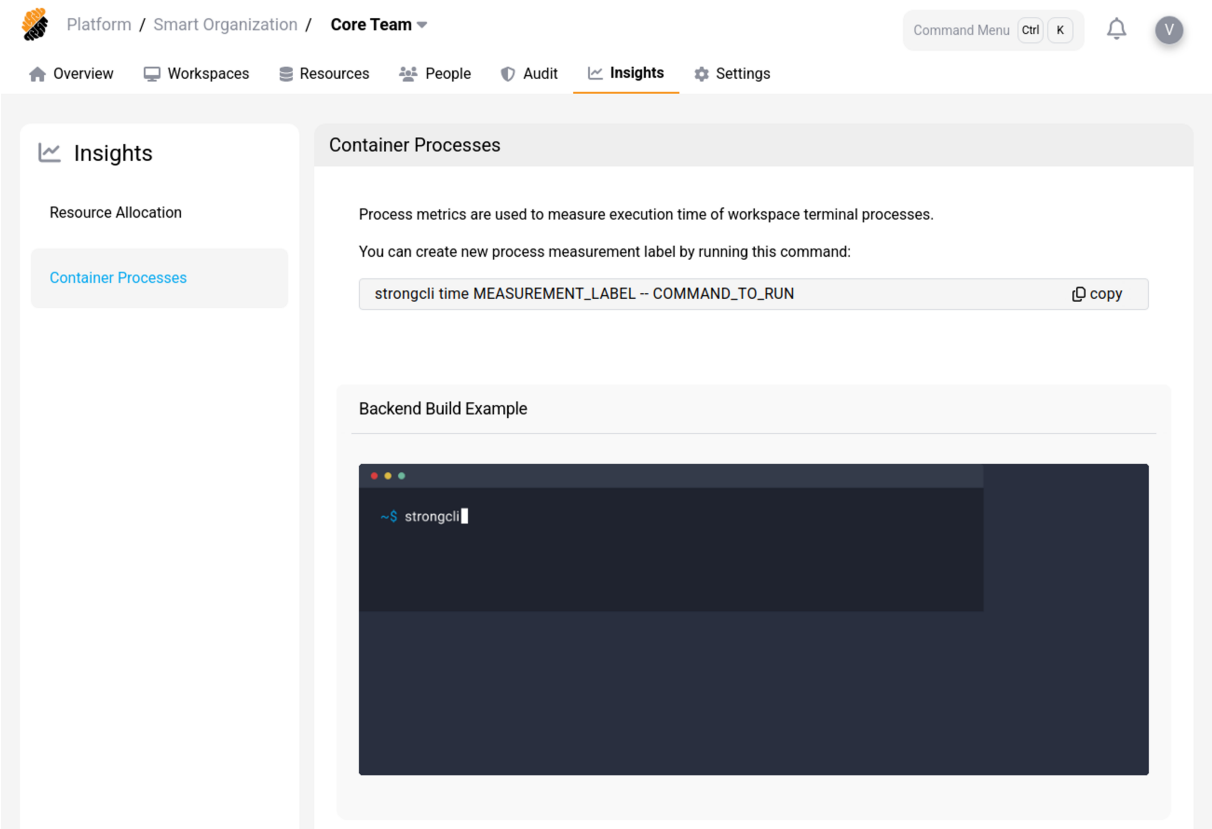
> **Note**
> Each workspace is assigned a CPUs/Memory/Memory specification. You can see the current level of usage for workspaces in the project
> in the Workspace Consumption list.

## Container Process Metrics

October 2, 2025

The section **Container Process** displays time metrics registered using the platform Command Line Interface (CLI) **strongcli** available in developers workspaces.
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:

```
> strongcli time LABEL -- COMMAND_TO_RUN
```

Where:

- **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.

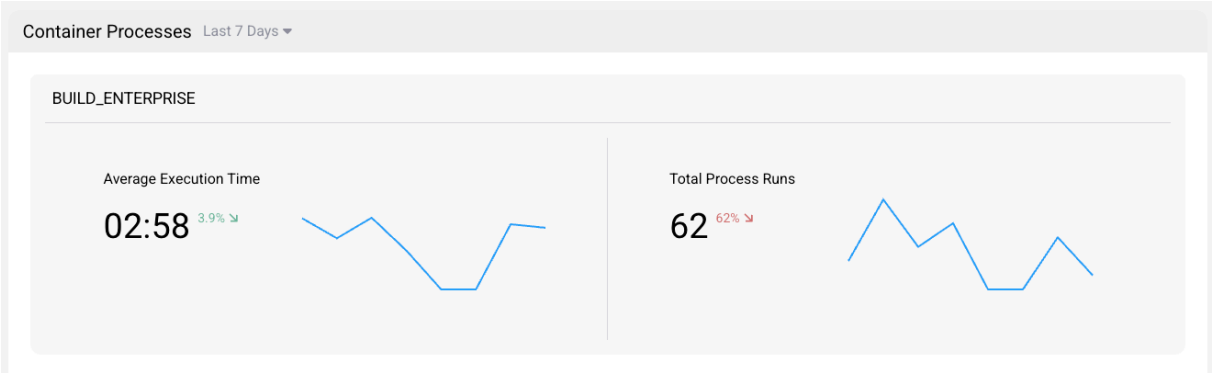
 

This registers a new process for your workspace among the **container processes** and measures its execution time.

## 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.

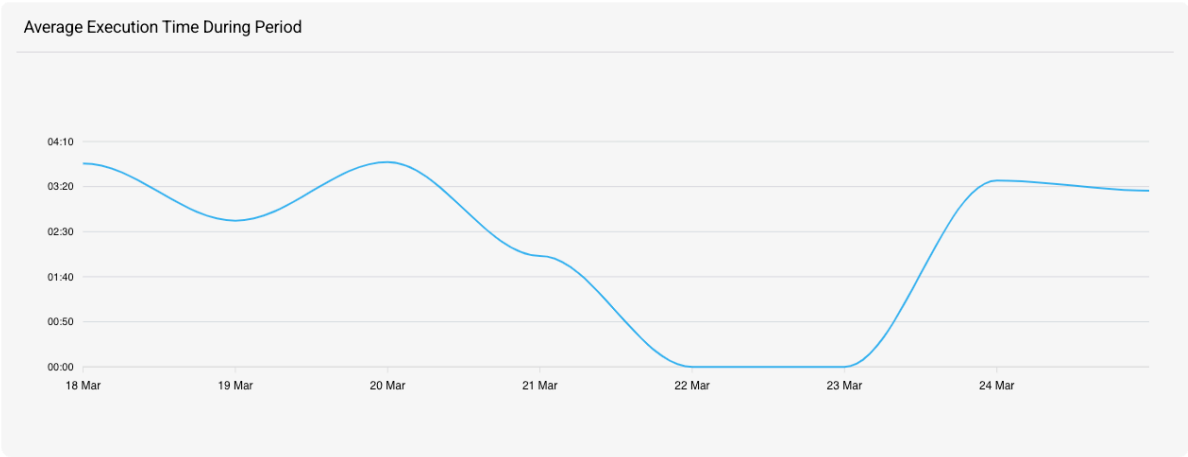



Based on selected period, the graph scale will be adapted accordingly.

## Container Process Insights

### Average

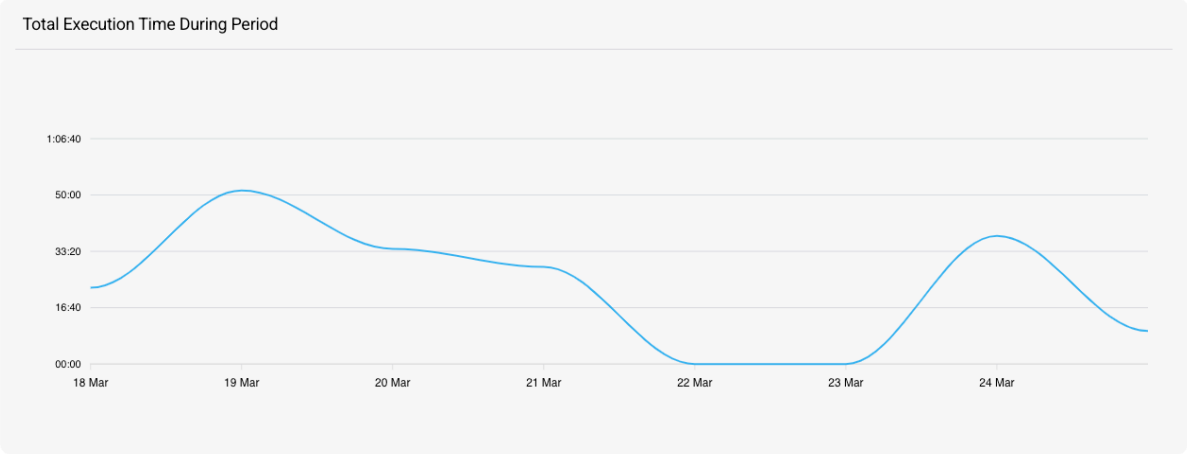
The "average execution time"graph in the container Process section of the Insight dashboard shows the average amount of time it took for a command to be executed within a developer's workspace, as recorded by the platform's Command Line Interface (CLI). The period of time displayed on the average execution time graph can be adjusted, allowing you to view metrics for a specific date range.



### 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.




## Profile and Account Settings

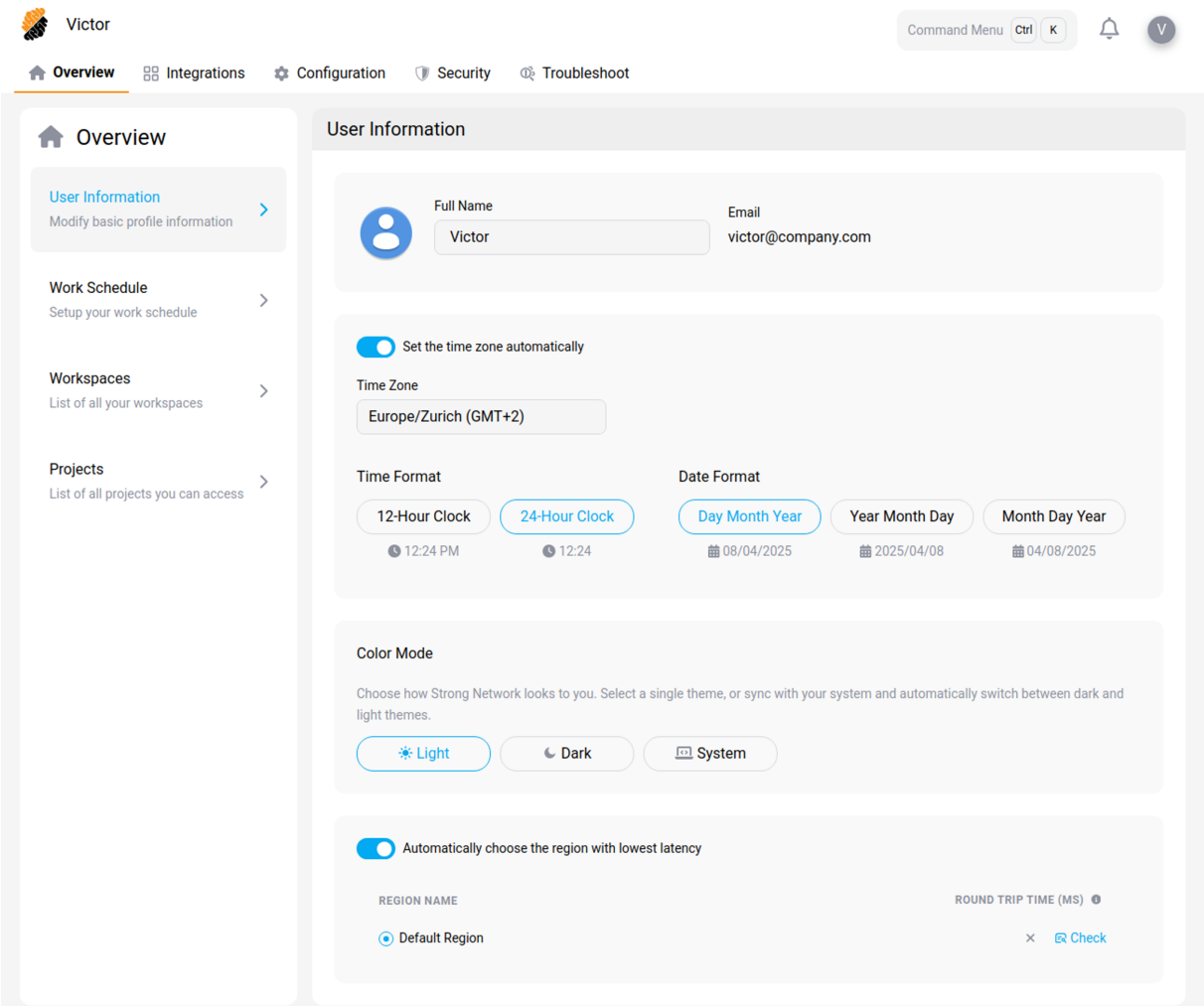
The **Profile and Account Settings** pages lets you manage personal data and set preferences around your work habits. For example, you can set-up a work schedule such that your workspace is automatically deployed at pre-set hours.

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.

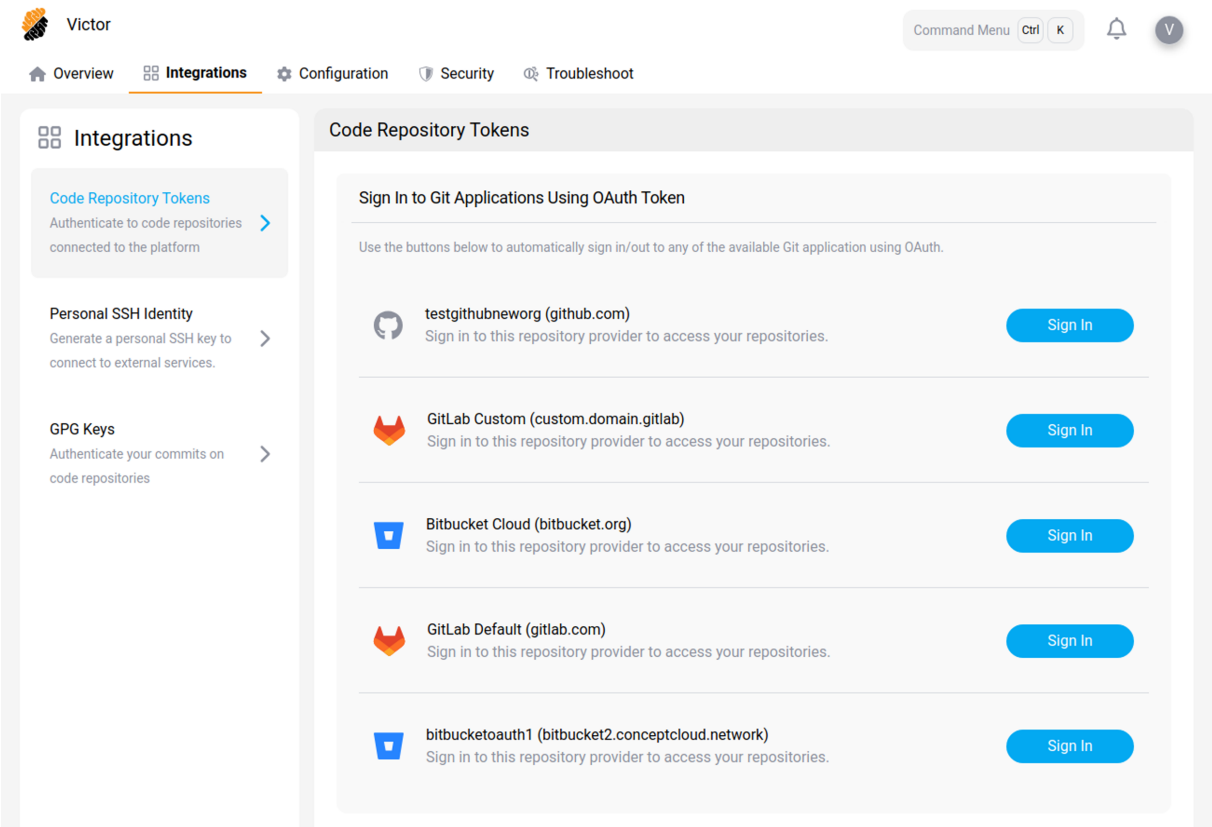
The **Overview Page** allows you to edit personal information, define a work schedule, view owned workspaces and project membership.



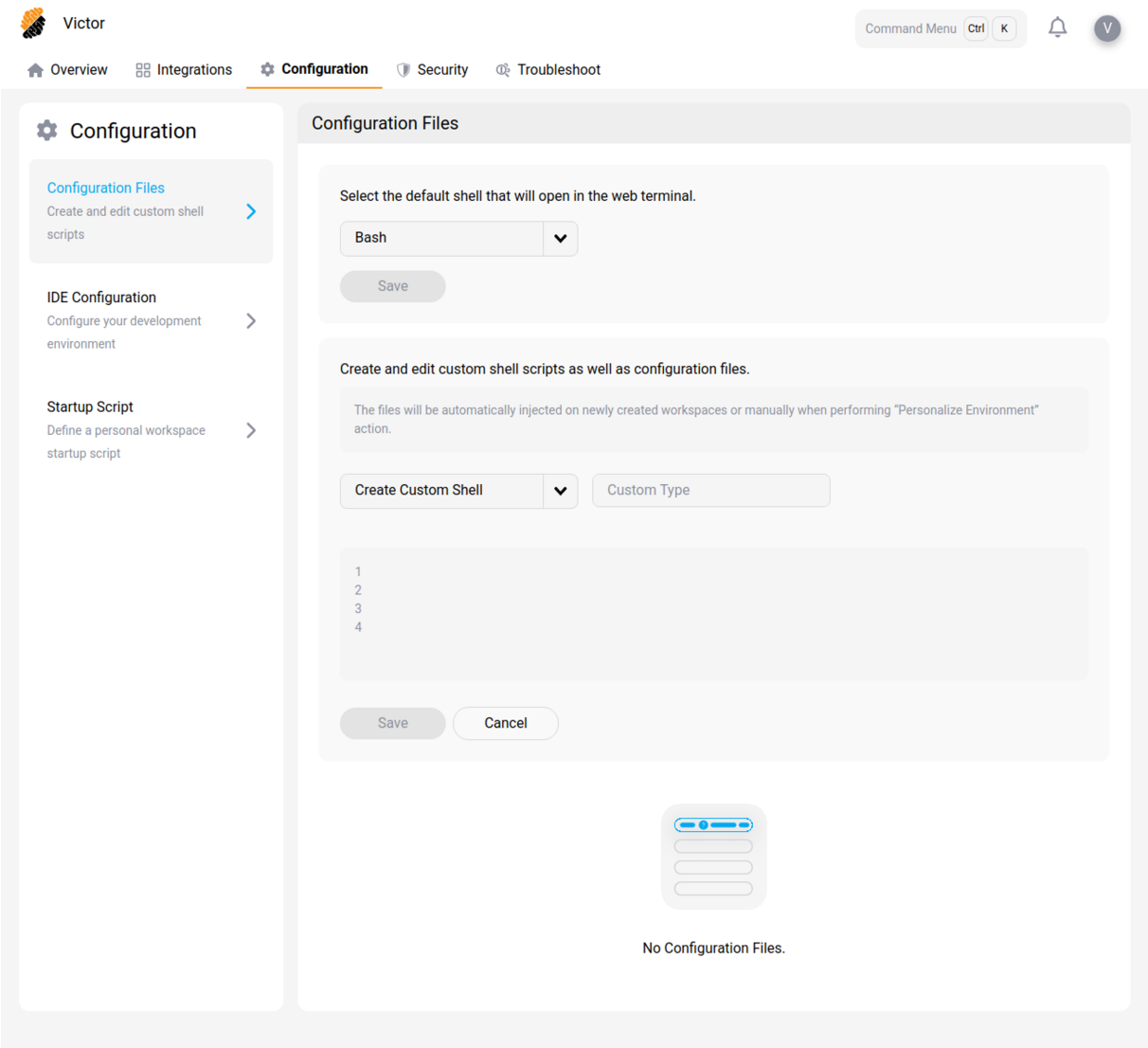

The **Integration Page** allows you to create and edit different authentication tokens, personal SSH identity and GTG keys.



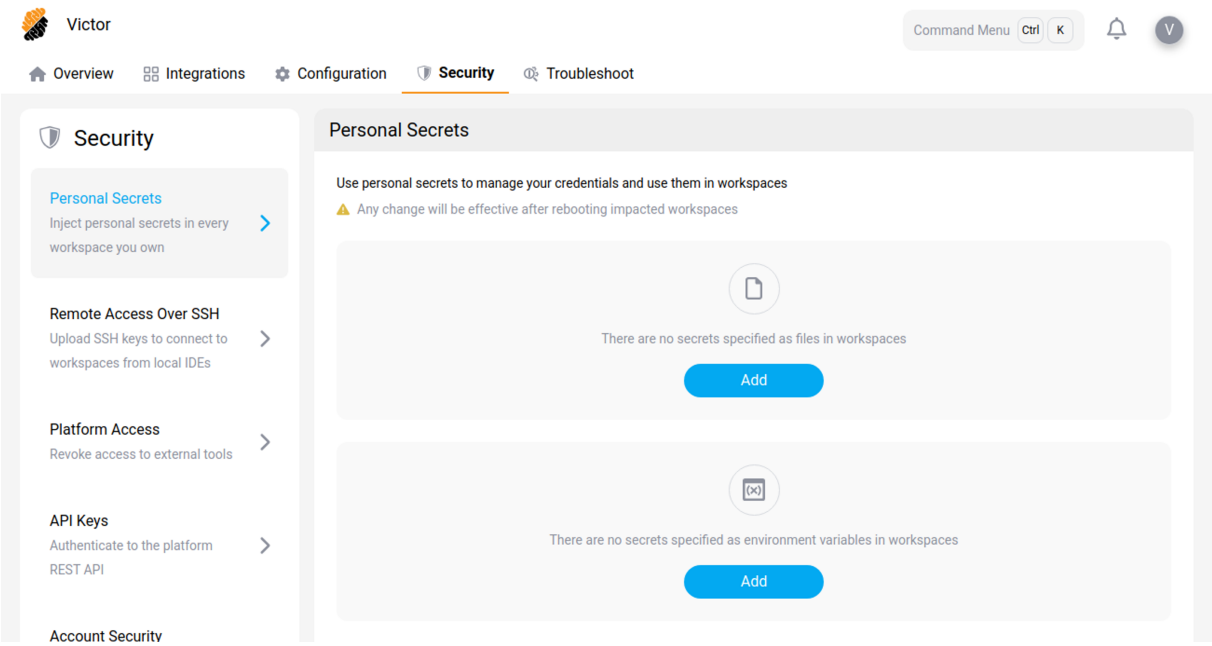

The **Configuration Page** allows you to create and edit custom configuration files, IDE configurations and workspace startup scripts.




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




### Content

- **Overview Page**
- **Integration Page**
- **Configuration Page**
- **Security Page**

## Profile Overview

October 2, 2025

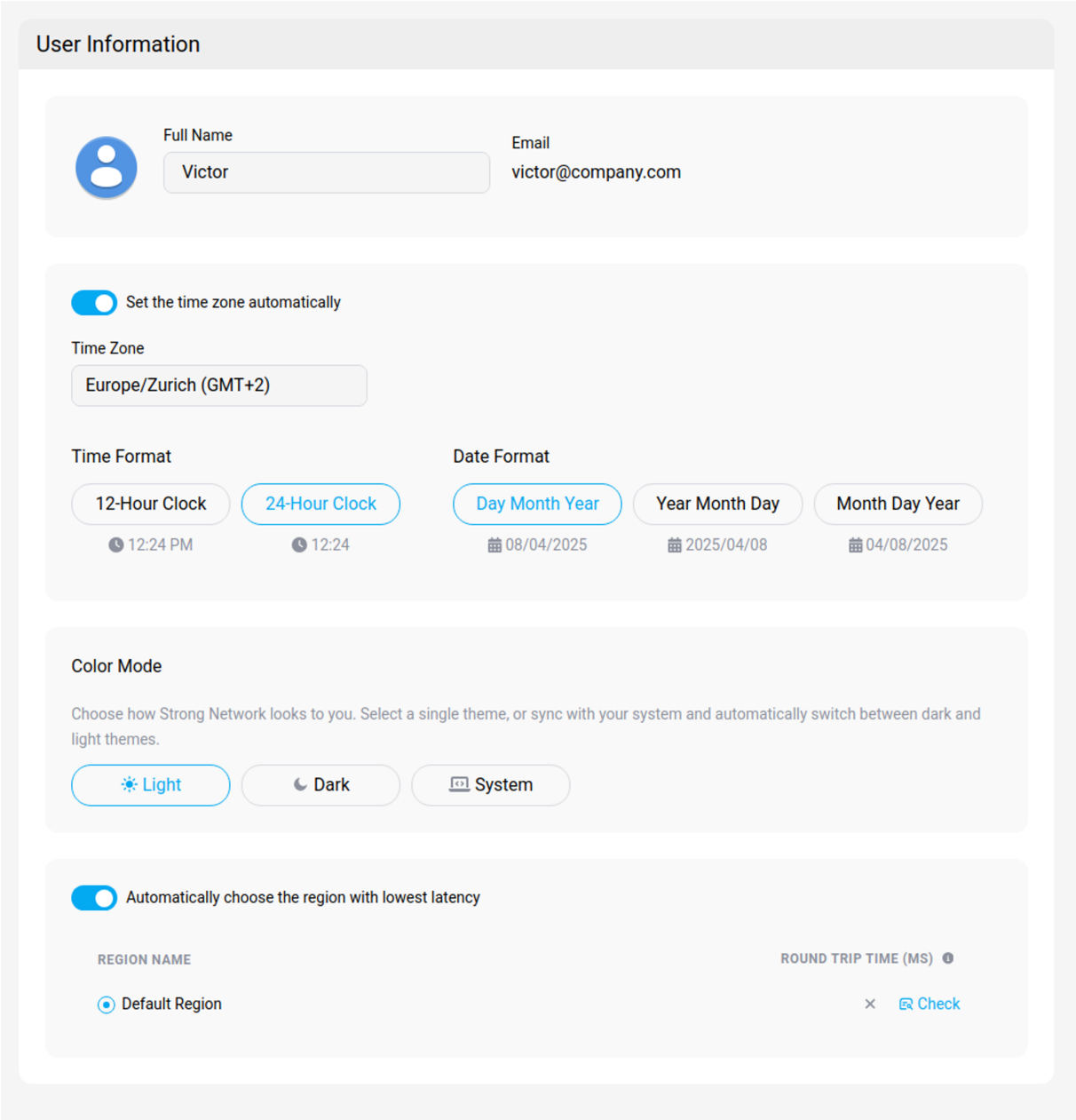
The **Profile Overview Page** serves as a comprehensive summary of the user's information, their workspace ownership and project membership.

- User Information
- Work Schedule
- Workspaces
- Projects

## User Information

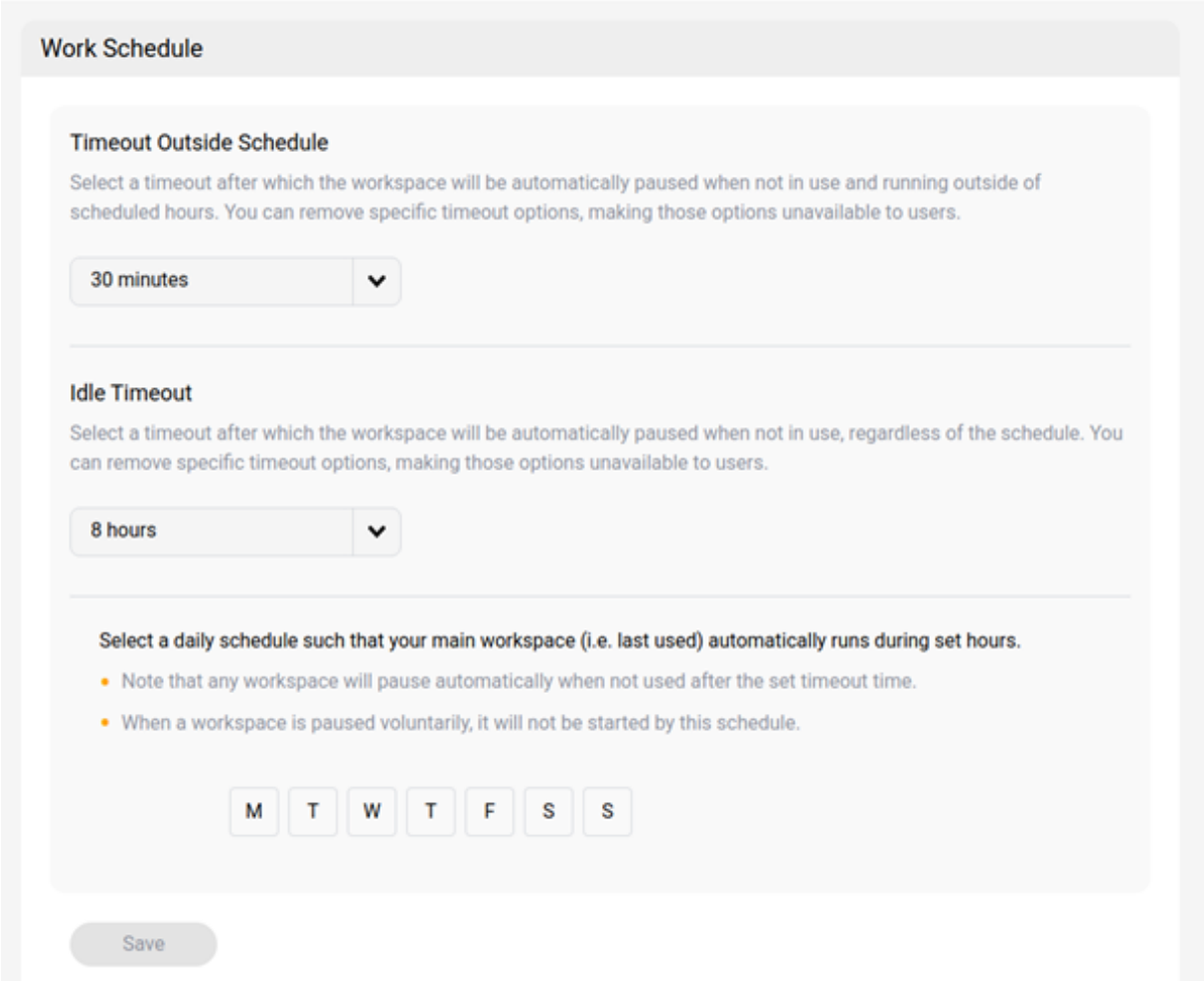
In the **User Information** section you can modify your user's name and time zone.
The email linked to your profile cannot be modified.
The profile picture is retrieved from your identity provider when available.




## 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.

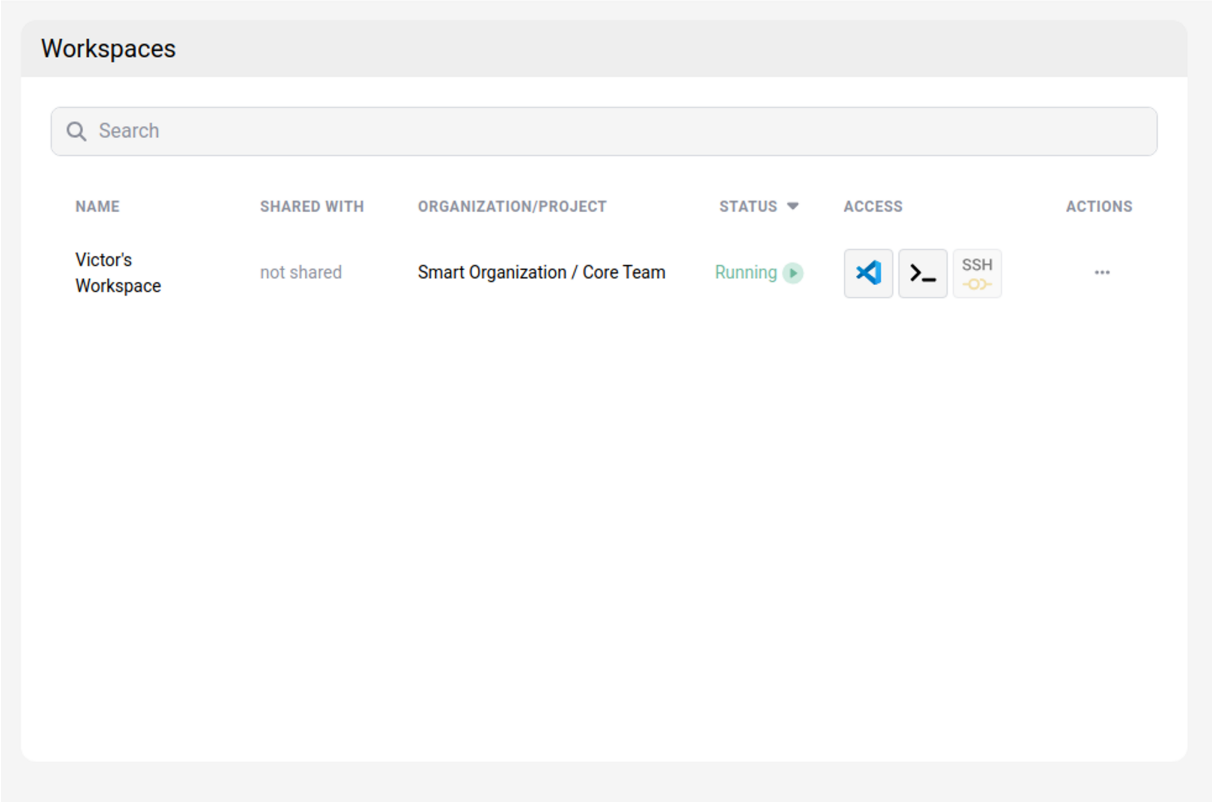



> Tip:
>
> Workspaces will pause automatically when not used for over a pre-set time, typically 60 minutes, depending on the setup of your platform.
>
> When a workspace is paused voluntarily, it will not be impacted by the schedule.

## 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.

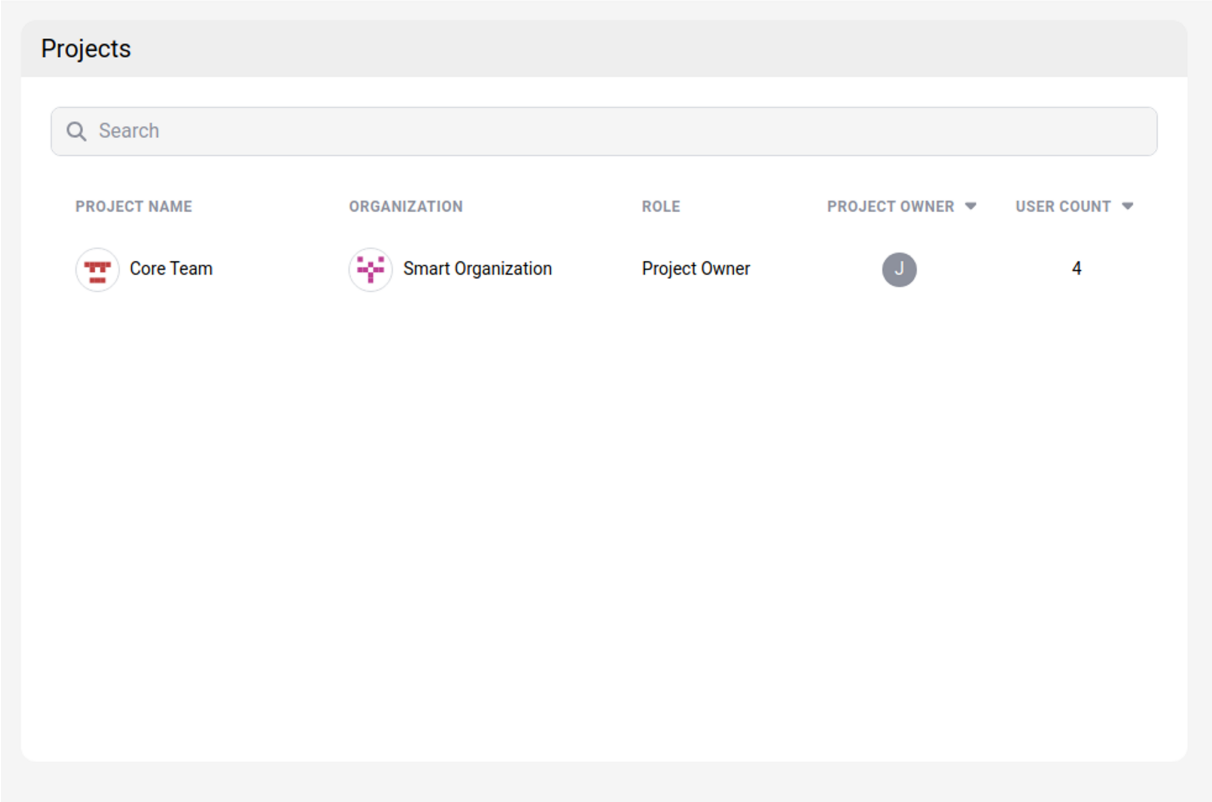
 




## Projects

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.




## Integration

October 2, 2025

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.

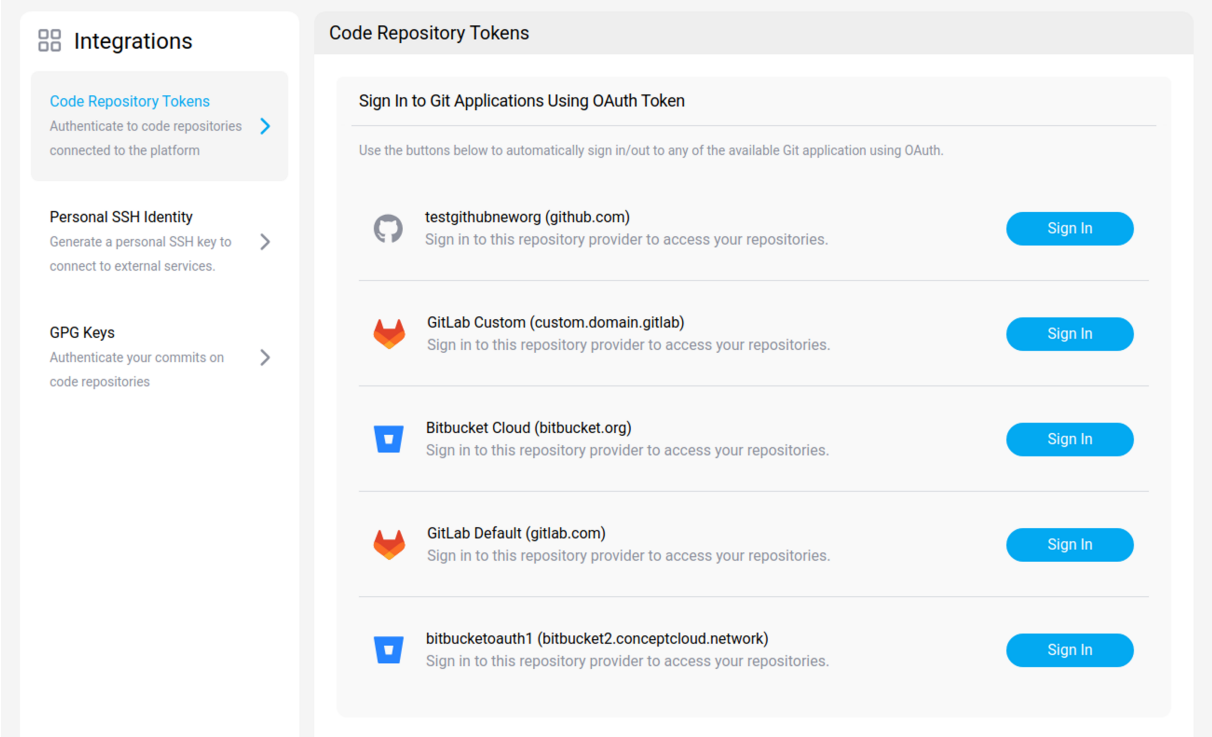
- Code Repository Tokens
- Personal SSH Identity
- GPG Keys

## Code Repository Tokens

Under **Code Repository Tokens**, you can configure authentication, using **OAuth Authentication Tokens** or **Personal SSH Keys**, to the following git providers:

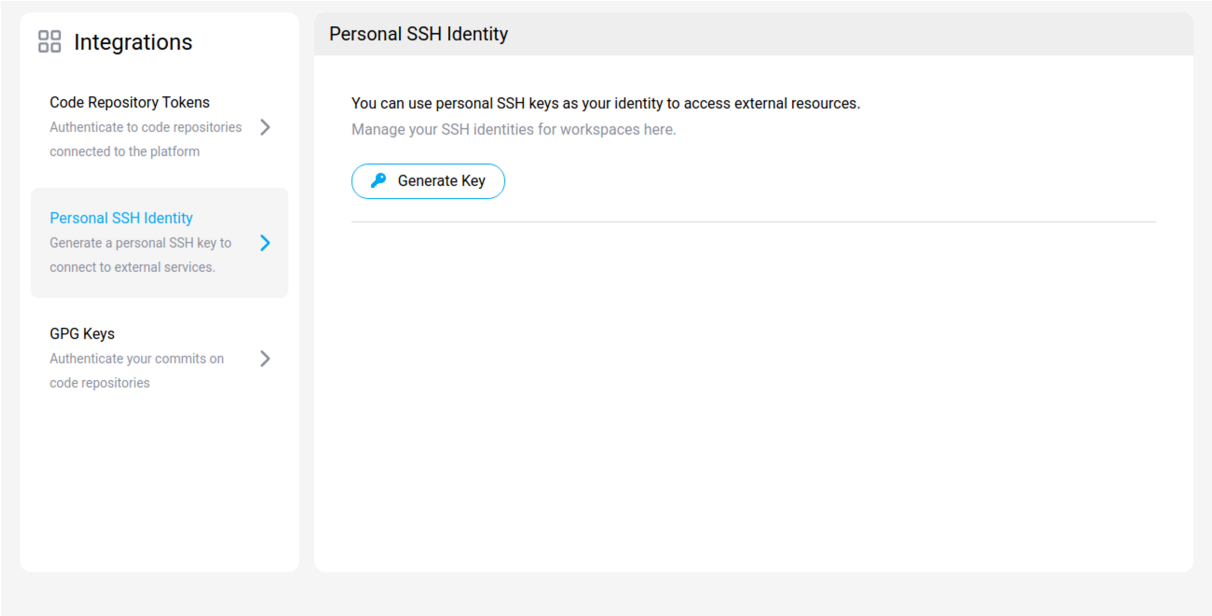
- **GitHub**,
- **GitLab**,
- and **Bitbucket**.
  For certain of these git providers, you have the option to choose between the 'Default'or 'Internal'options. An 'Internal Service'is self-hosted, whereas a 'Default Service'is hosted on the cloud.




## Personal SSH Identity

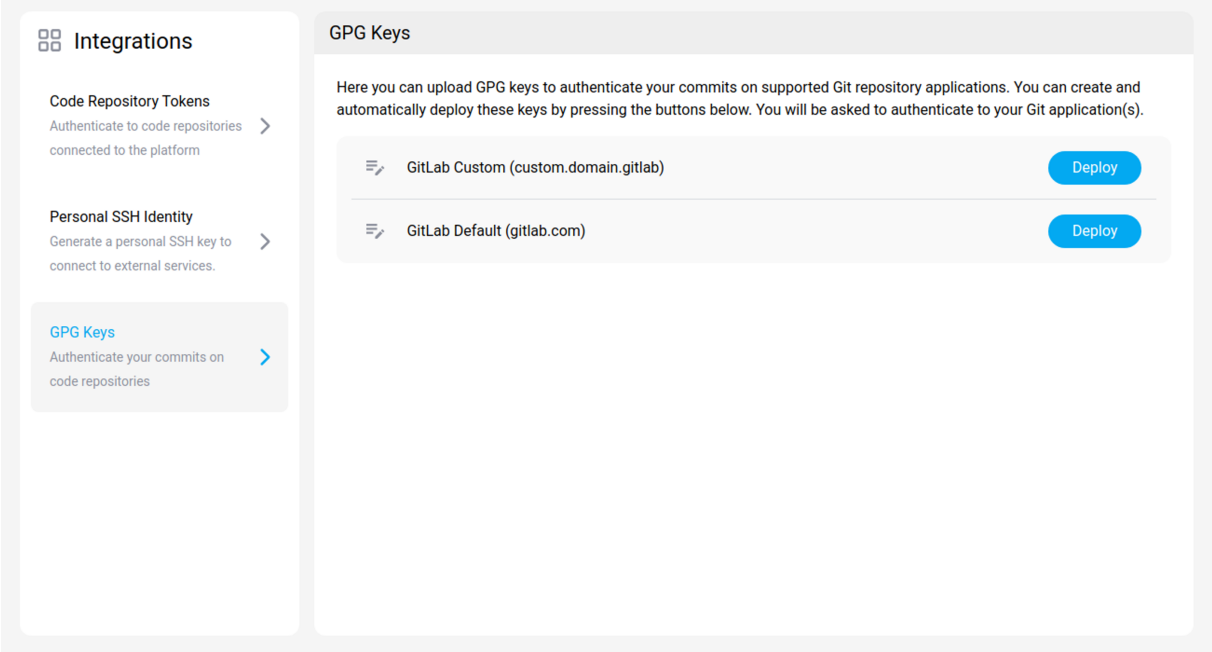
You can generate a personal SSH key to authenticate yourself when accessing external resources. The SSH key will be applied to new or existing workspaces.




## GPG Keys

You can generate and automatically deploy GPG keys to authenticate your commits on supported Git repository applications (i.e. GitHub).



# Configuration

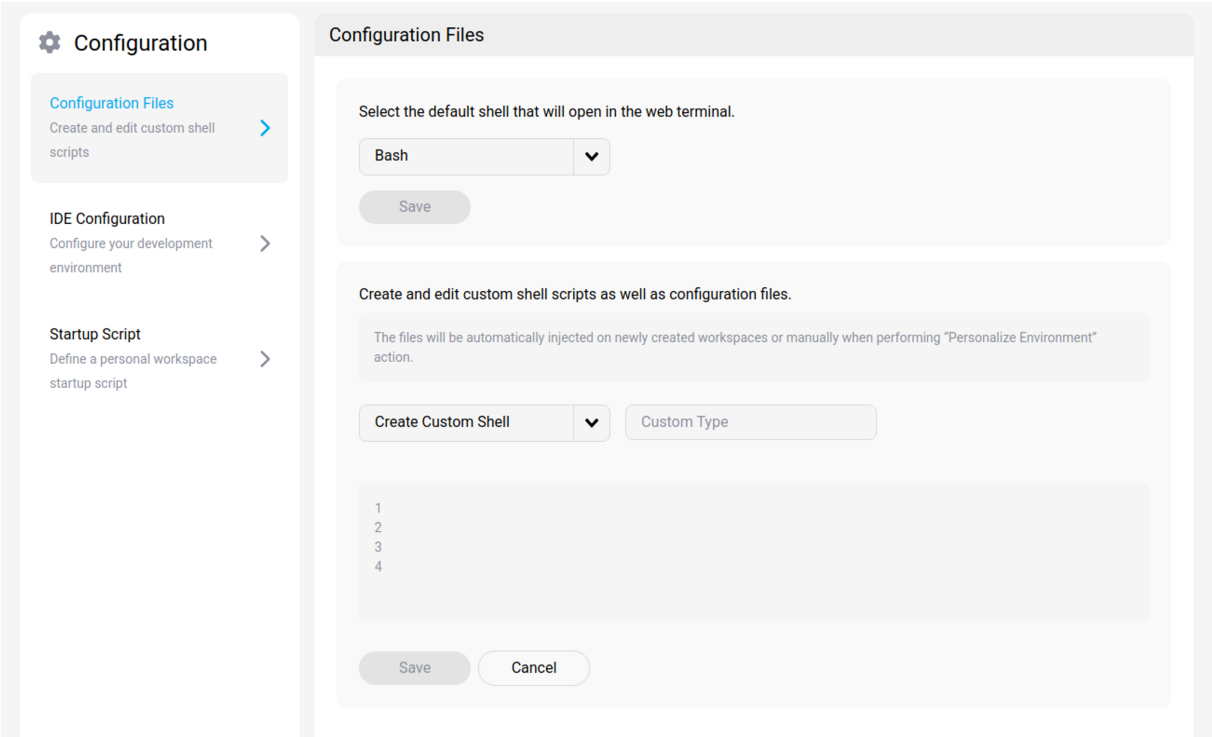
October 2, 2025

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.

- Configuration Files
- IDE Configuration
- Startup Script
- Theme
- Language

## Configuration Files

Personal configuration files can be managed from the **Profile Settings**.
These files can be injected in any new or existing workspaces automatically using the **...** icon and the option **personalize environment**.




You can edit and manage typical shell scripts attached to Linux-based environments such as:

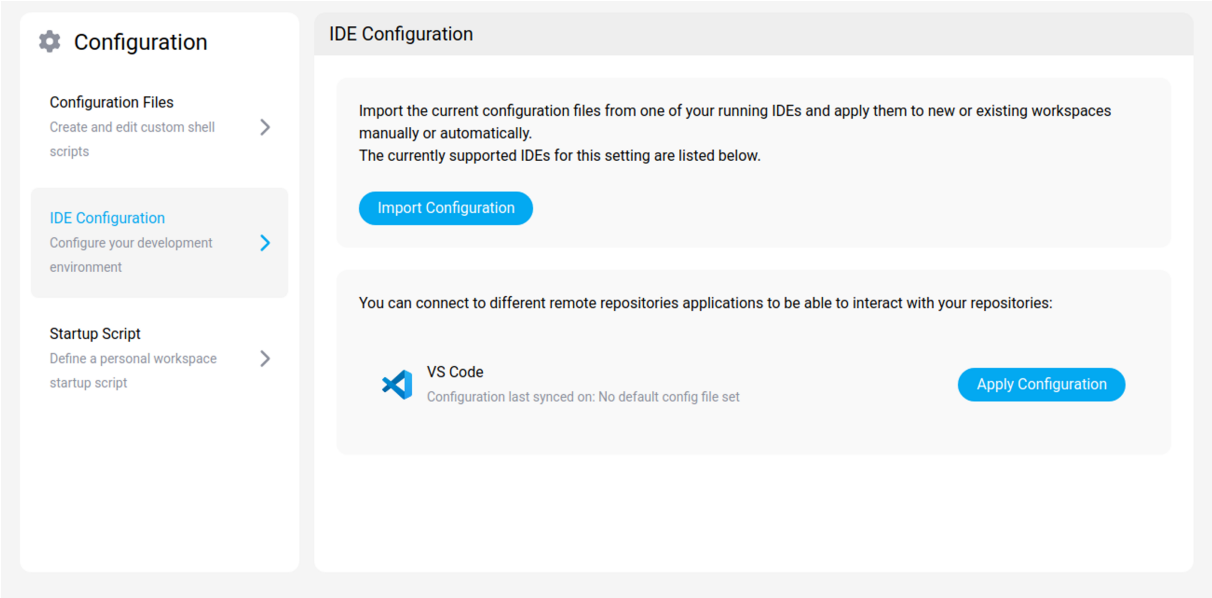
- **.bashrc**,

- **.zshrc**,
- **.profile**, or any file of your choice using the
- **custom**: option.

## 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:

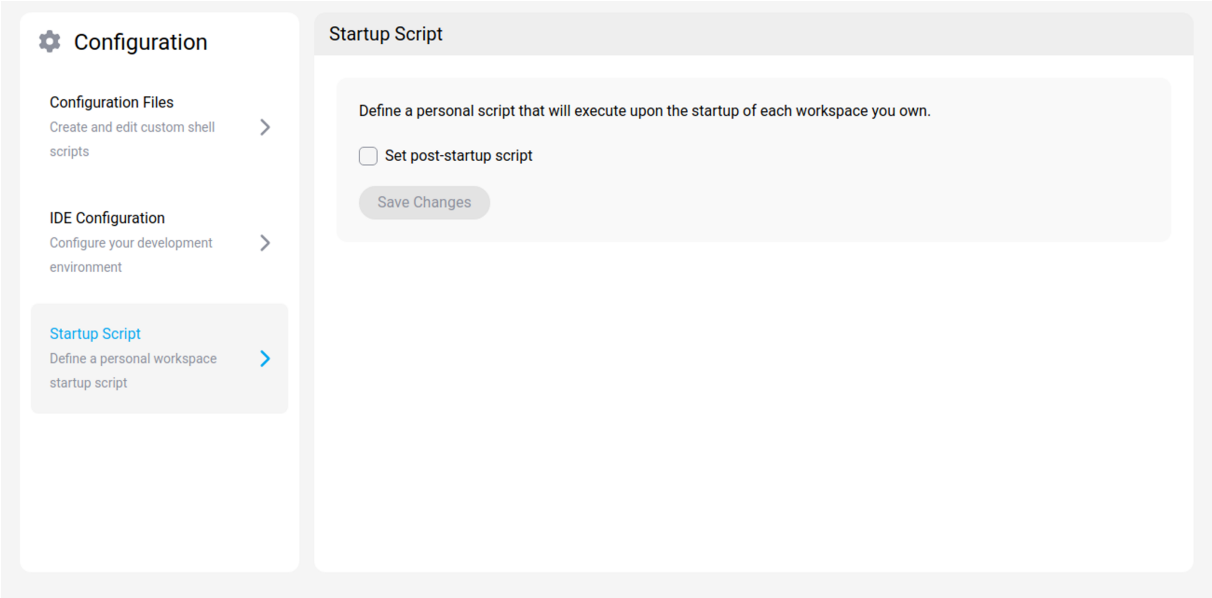
- **VSCode**,
- **any IDEs from Jetbrains**.

## Startup Script

You can define a personal script that will be executed upon each startup of the workspaces that you own

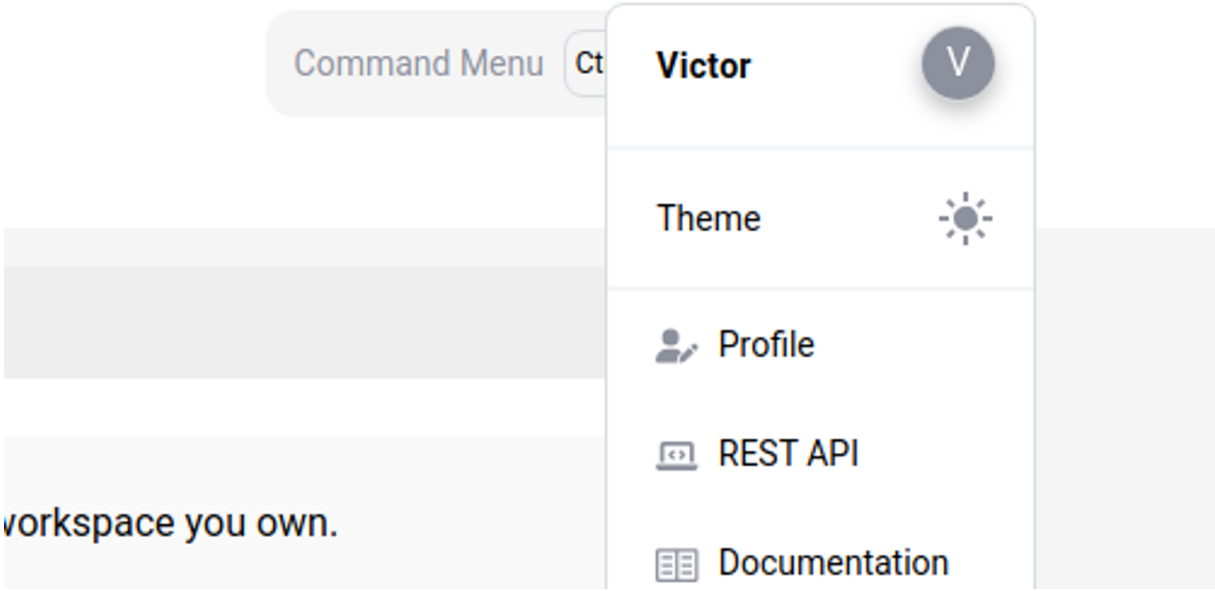
> Tip:
>
> Note that if you defined a startup script for a given workspace, then it will override this one




## 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**,

- **French**.

# Security

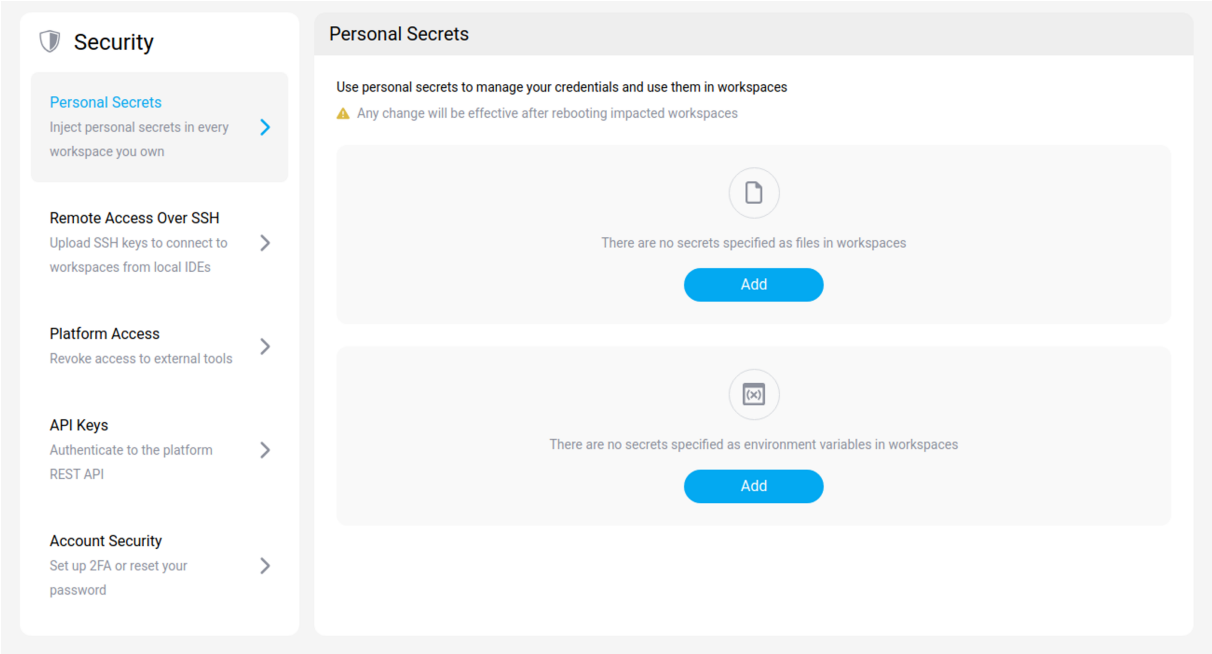
October 2, 2025

In the **Security Page** you can manage the different access keys, secrets and tokens that are linked to the user's profile.
This includes **Personal Secrets**, **Remote Access Over SSH** keys, **API Keys** and **GPG Keys**. 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. This helps to ensure that only authorized users have access to the necessary resources and services, and that access is revoked when necessary.

- Personal Secrets
- Remote Access Over SSH
- API Keys

## Personal Secrets

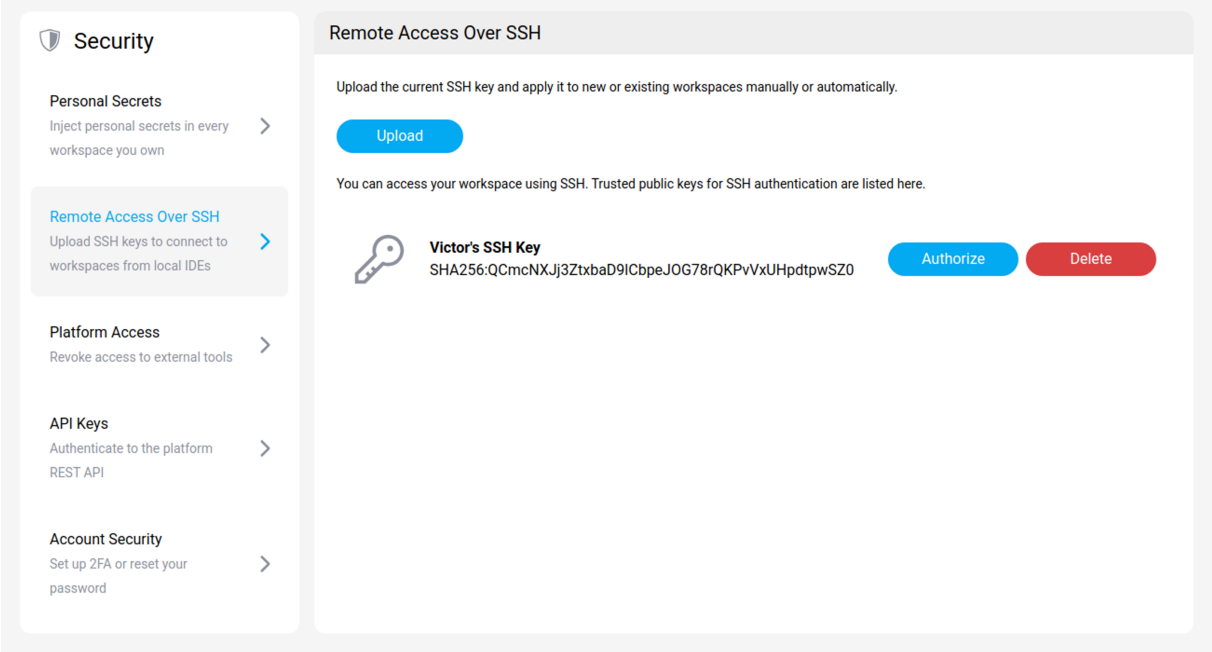
Under **Personal Secrets**, you can manage your secrets.
You add secrets that appear as files in your workspace, or add them as environment variables.



## Remote Access Over SSH

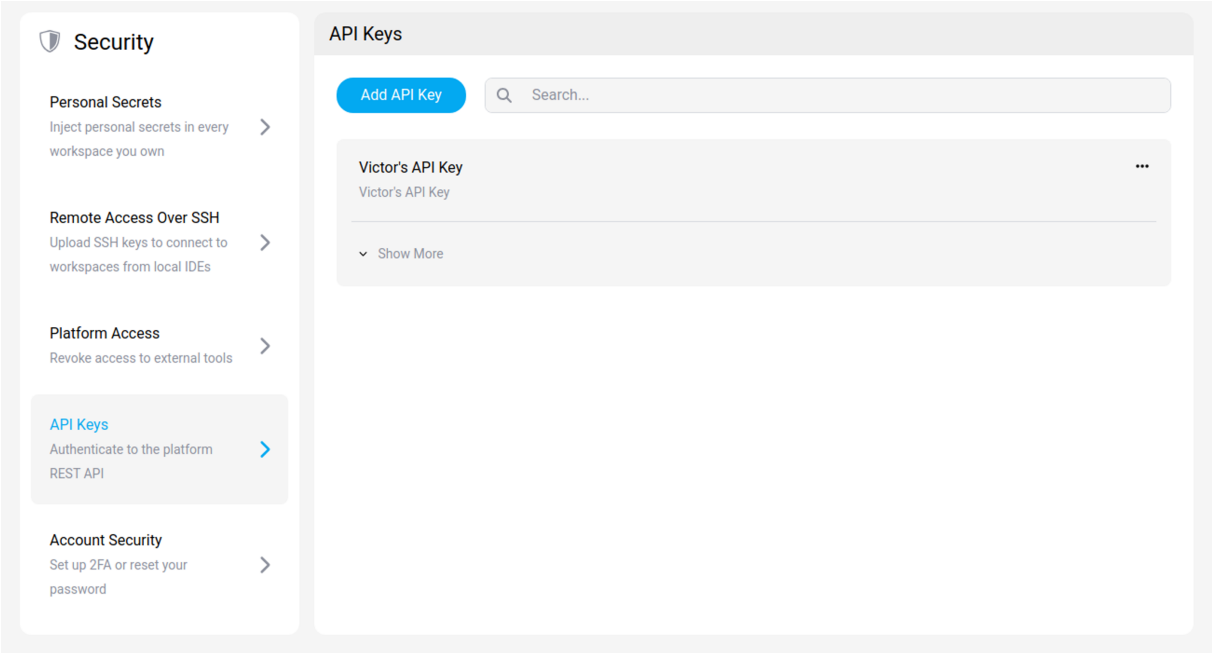
You can access your workspace using SSH, which allows you to run VSCode locally. Trusted public keys for SSH authentication are displayed in this section. Each key is linked to your profile.
One benefit of accessing your workspace using SSH is flexibility. By allowing you to run VSCode on your local machine, you can still leverage the powerful hardware of the remote machine and still not give up on security. View SSH Into Your Workspace to set it up.




## API Keys

An **API key** is a unique identifier used to establish a connection to an API call. Once connected, the API service will be available in your workspaces.
API keys are used to authenticate the source of a request and make sure that the API is only used as intended. API keys are often used by web and mobile apps to connect to web-based services and retrieve or update data.




## Organization General Settings

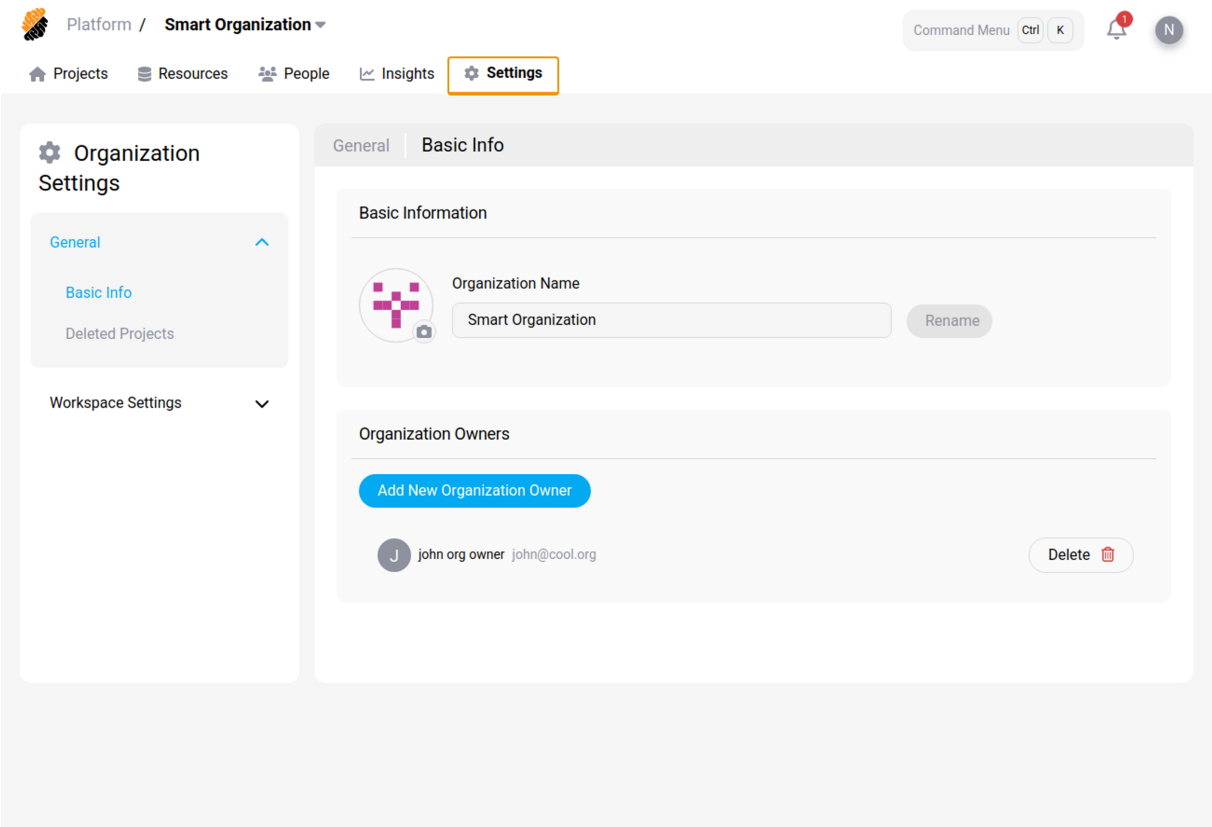
October 2, 2025

### Admin

The Organization Settings serve as the overarching control center for administering and standardizing configurations across all projects within the organization. By defining settings at the organizational level, you can enforce a consistent set of protocols, security measures, and resource limitations that will automatically apply to each new and existing project. This ensures uniform compliance and operational efficiency throughout the organizational ecosystem.

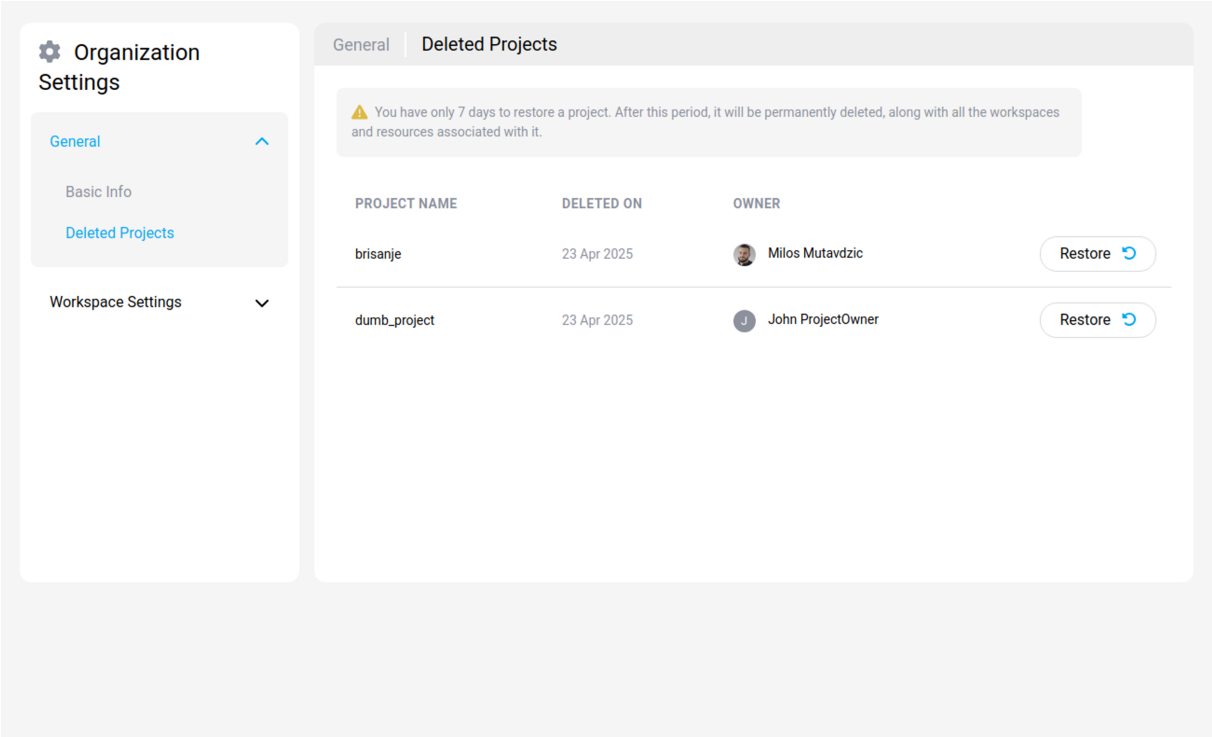
For detailed configurations at the project level, please refer to the Project Settings page.




Additionally, Organization Settings provide a safeguard against accidental deletions by allowing you to recover deleted projects for up to 7 days. After this period, the projects are permanently deleted. This recovery window helps prevent the permanent loss of project data.




## Workspace Settings

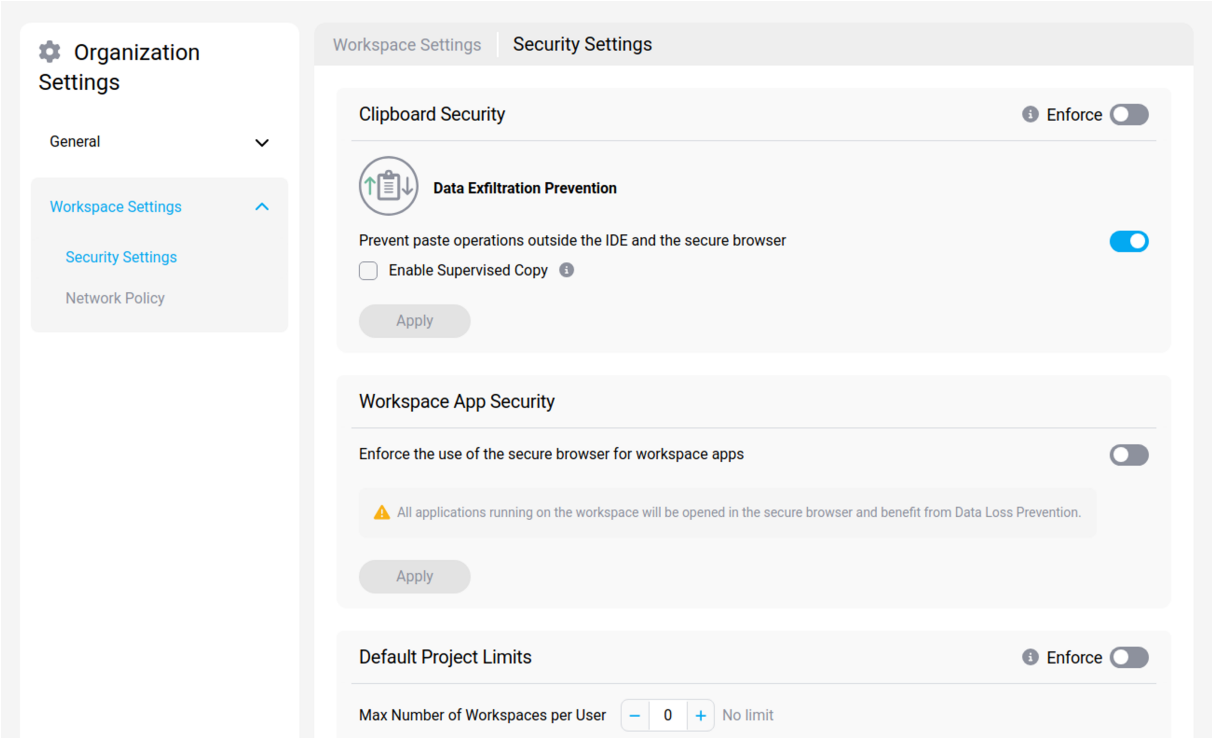
October 2, 2025

This section focuses on configuring settings for workspaces that apply across the entire organization. Define organization-wide security policies governing aspects like data handling and access, and establish network policies to control workspace traffic consistently for all projects within the organization.

### Security Settings

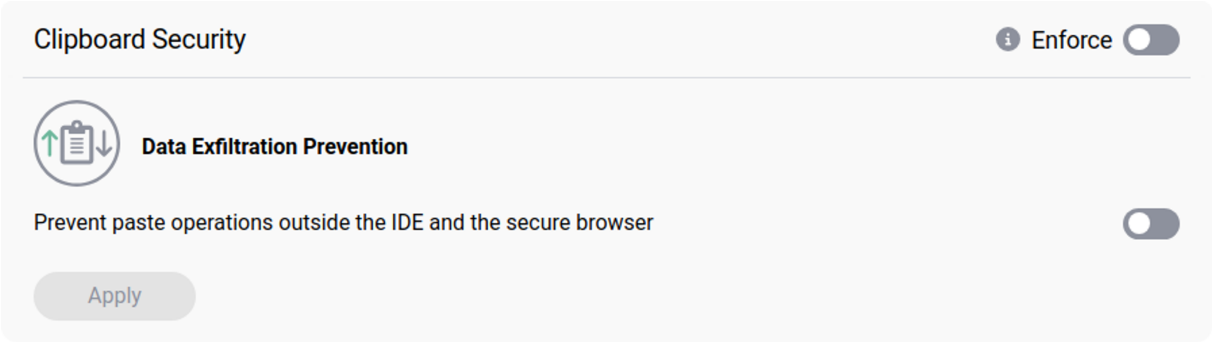
In the “Workspace Settings”section, the “Security Settings”enable you to implement multiple policies including Clipboard Monitoring, Workspace App Security, and Default Project Limits. These policies can be enforced to establish a foundational level of security across all workspaces within your project.




### Clipboard Security

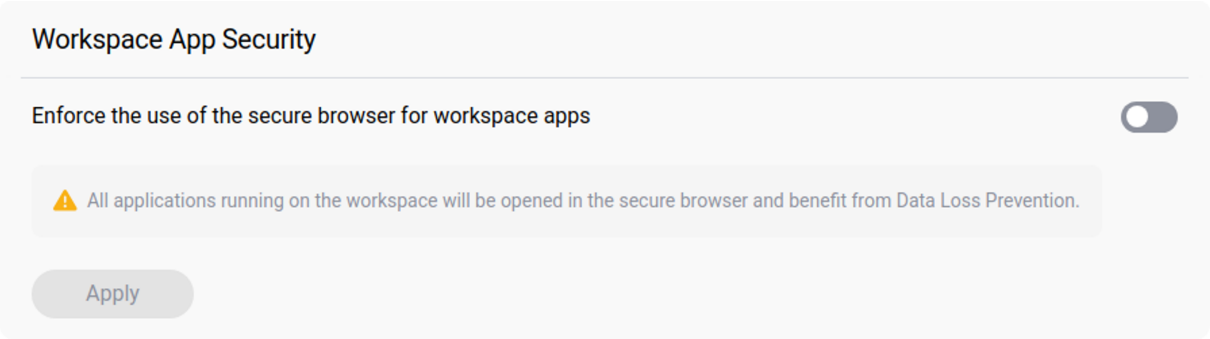
Clipboard Security implements Data Loss Prevention policies to safeguard against data leaks by disabling the ability to paste content from the IDE and secure browser into external applications.




### Workspace App Security

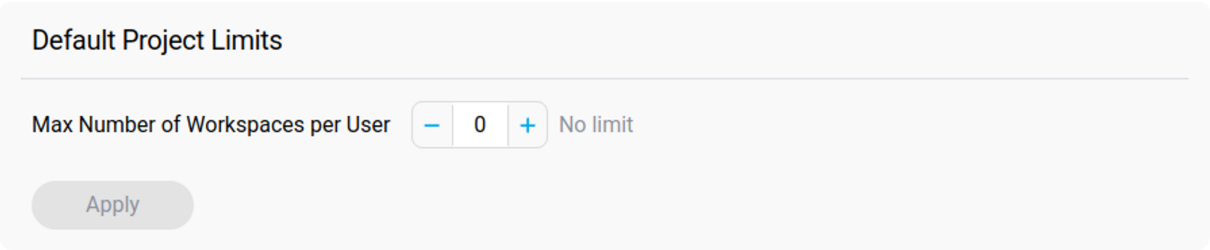
Workspace App Security allows you to mandate the use of a secure browser for workspace applications, ensuring that developers can share the applications they are developing in a protected environment. When used in conjunction with the Clipboard Security policy, this feature helps to prevent any potential data exfiltration from workspace applications.



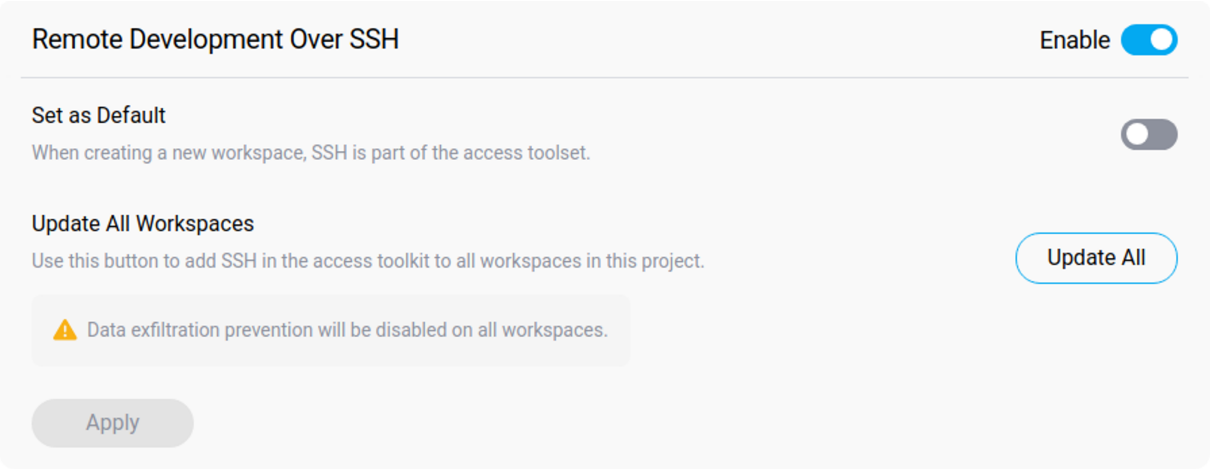

**Default Project Limits**

Default Project Limits can be set to cap the number of workspaces a user can create. This not only aids in resource monitoring and reduces unnecessary workspace proliferation but also contributes to cost efficiency by avoiding the operation of unused workspaces.




**Enable Remote Development Over SSH**

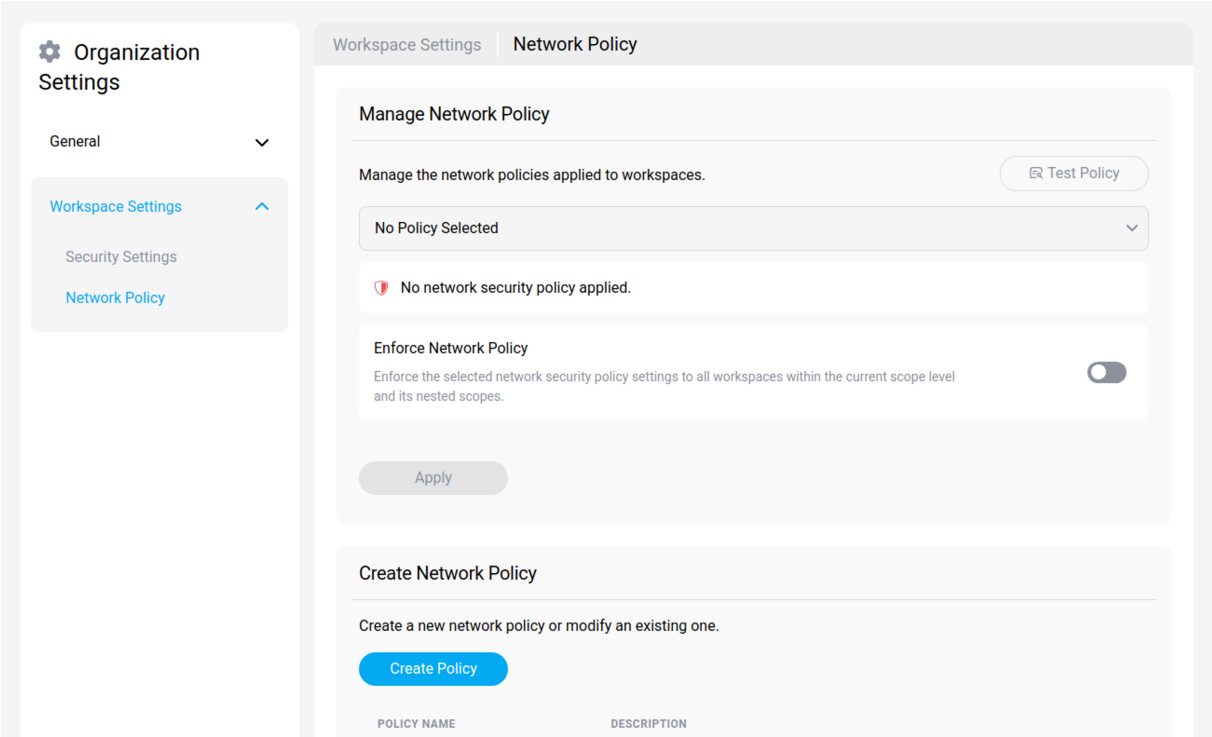
Remote Development Over SSH gives you the option to permit or deny developers the ability to connect to their workspaces via SSH. While convenient for certain tasks, this feature must be used judiciously as it can reduce the effectiveness of local IDE data loss prevention measures.



## Network Policy

Network policies are attached to workspace and enable fine-grained network traffic control. Network traffic is identified using combinations
of IP addresses, port and domain names. Once a network policy is attached to a workspace, all **outbound** traffic is enforced by the rules in
the policy and the workspace's user cannot circumvent the restrictions.




### Default Network Policies

Three default policies are available in a project. An administrator can create a new Network Policy if needed.

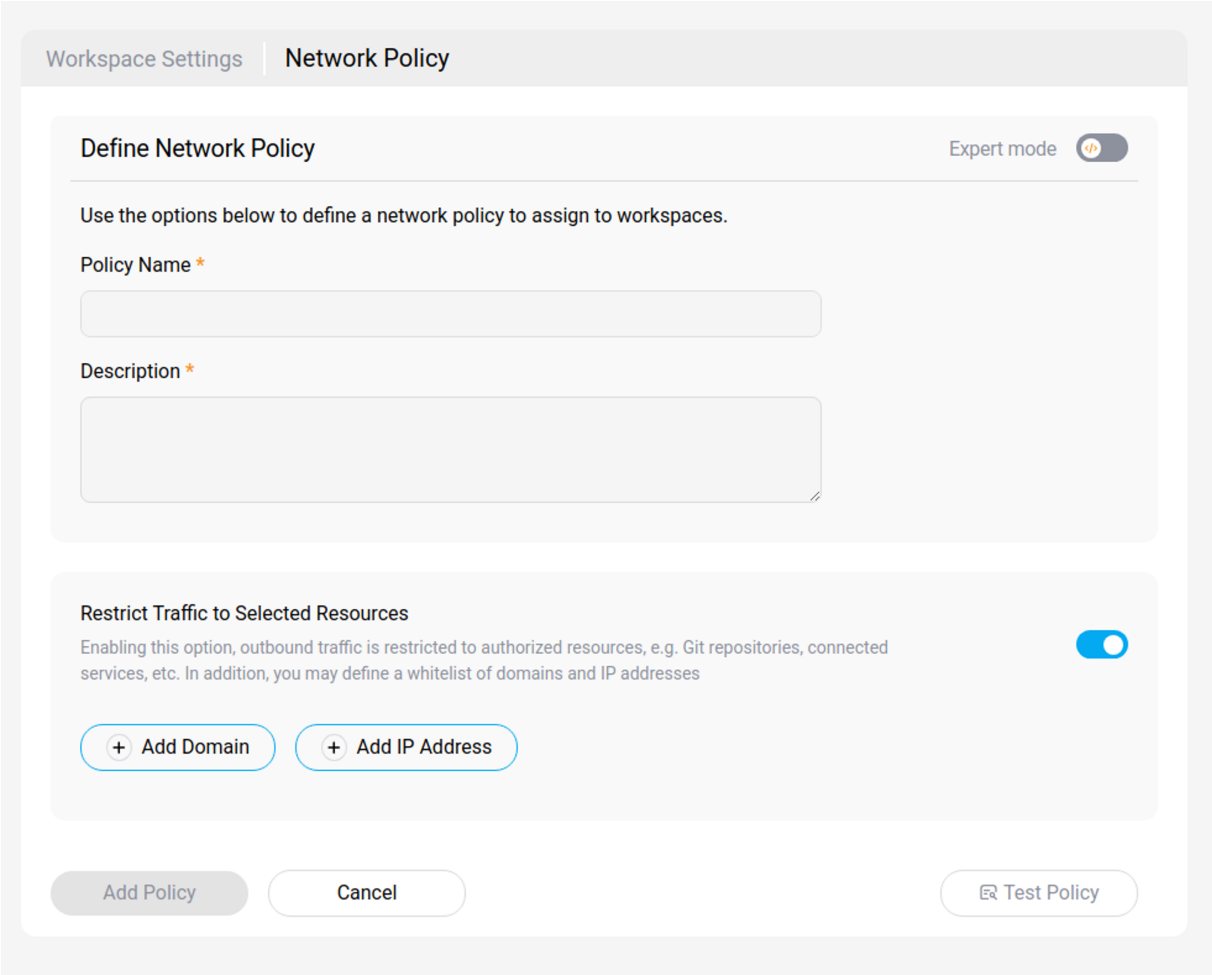
| Name | Scope | Description |
|---|---|---|
| **Monitor Traffic** | Project | This is a standard policy to monitor the outgoing traffic to the workspace. It will cause the generation of log events in the Audit dashboard. |

| Name | Scope | Description |
|---|---|---|
| **Restrict Traffic** | Project | This is a standard policy to restrict outgoing traffic from the workspace. It will block all traffic except to attached repositories and domains. Failed network requests are shown in the log events in the Audit dashboard. |

### Add a Network Policy

You can create a Network Policy by pressing the "**Create Policy**"button.




You will need to enter the following information:

1. **Name**, a name to identify the policy,
2. **Description**,

> **Warning**
>
> Be careful when naming and describing a new policy. A misleading name can end up in giving too many permissions to a user.

1. **Log and record outbound network traffic** (default),
2. **Restrict Traffic to Selected Resources** (optional),
   All traffic will be restricted, except for end systems added to your **whitelist**

- Add each application that you want to whitelist
- Add Domains that you want to whitelist, and indicate whether to include subdomains
- Add IPs that you want to whitelist

#### Edit or Delete a Network Policy

You can edit or delete a Network Policy by clicking on the "**...**"icon next to its class level.

## General Settings

October 2, 2025

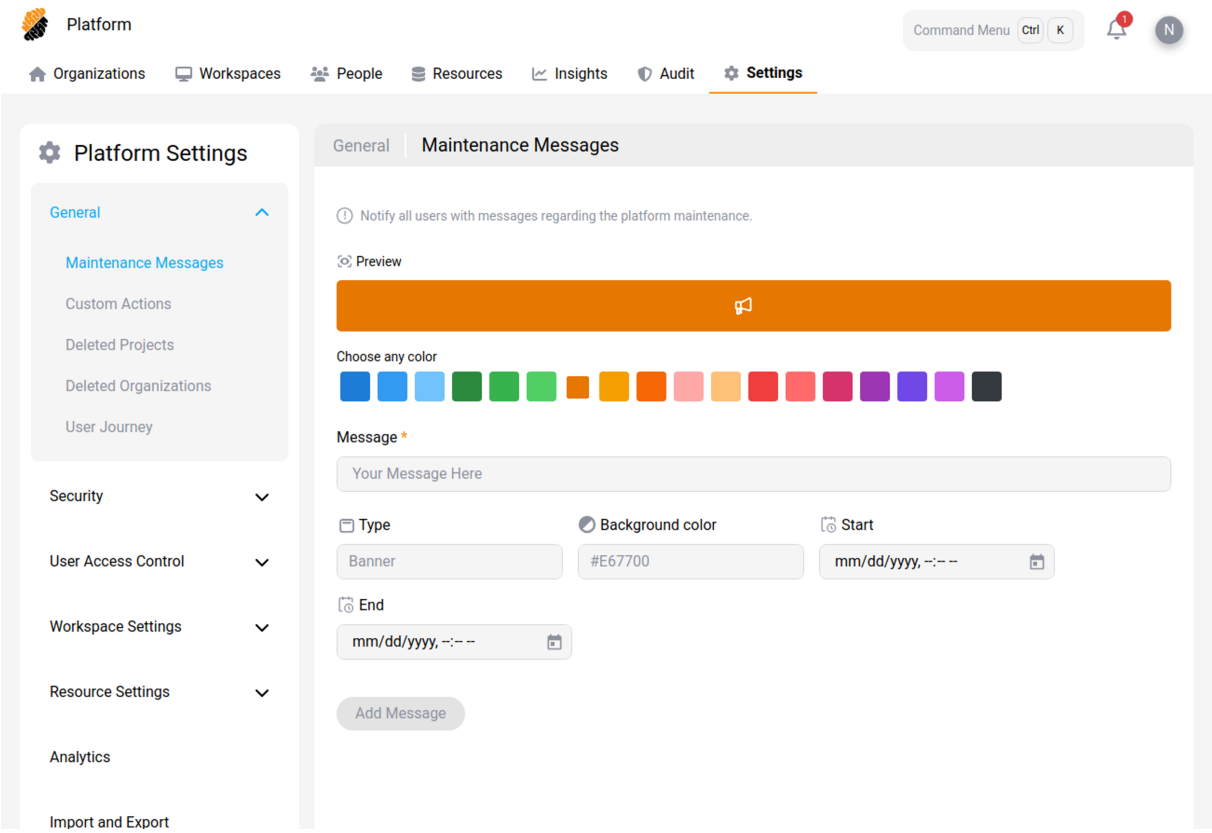
This section covers fundamental platform-wide configurations. Here, administrators can manage **Maintenance Messages**, configure **Custom Actions**, handle the recovery of **Deleted Projects** and **Deleted Organizations**, and adjust settings related to the initial **User Journey**. These settings govern the overall operational aspects and user experience defaults of the platform.

- Maintenance Messages
- Custom Actions
- Deleted Projects
- Deleted Organizations
- User Journey

### Maintenance Messages

You can configure and display maintenance messages to users. These messages can inform users about scheduled downtime, ongoing maintenance activities, or other important platform-wide notifications.

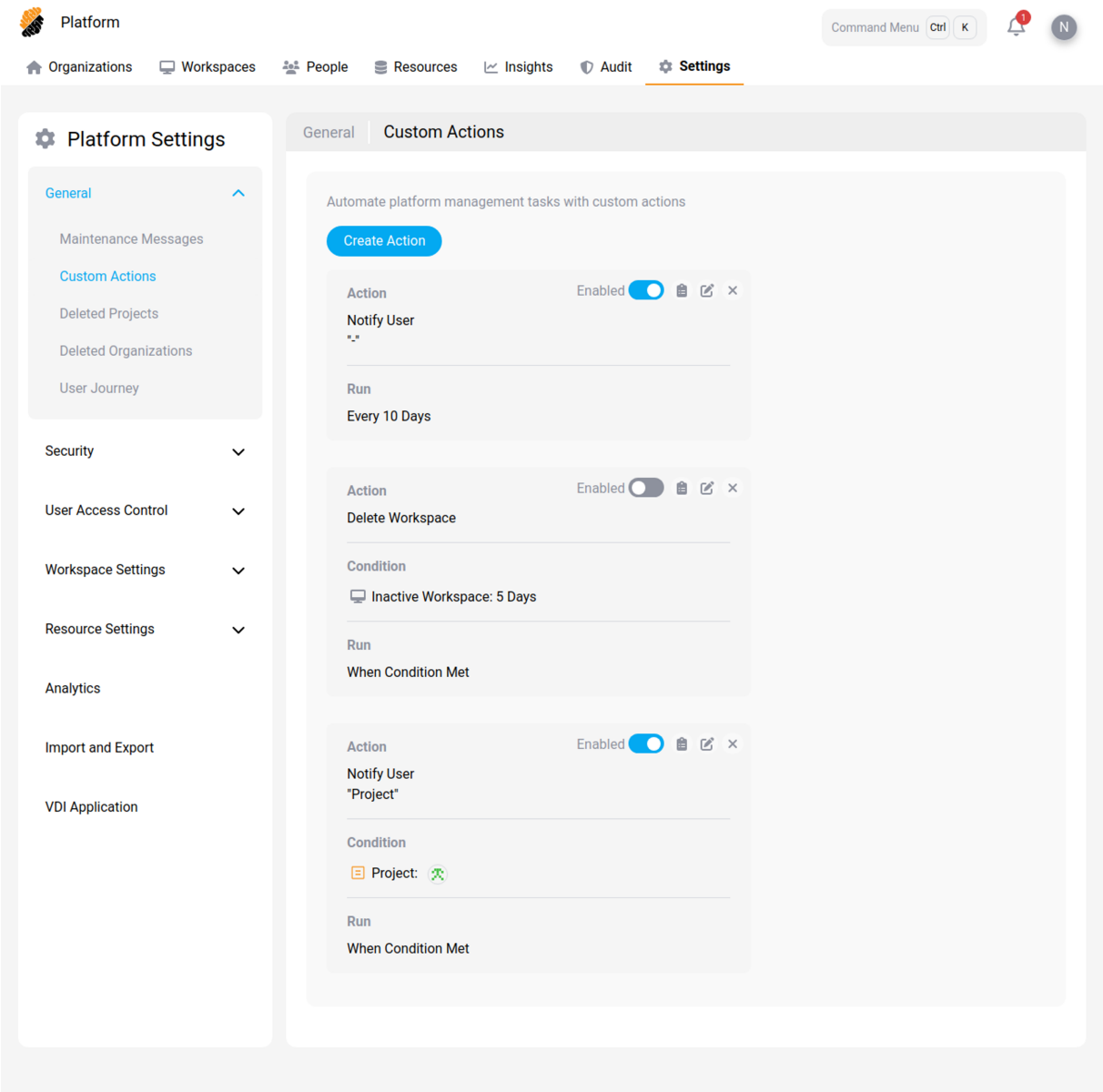
 




## Custom Actions

Configure custom actions that can be triggered within the platform. This allows for extending platform functionality with specific automated tasks or integrations tailored to your organization's workflows.

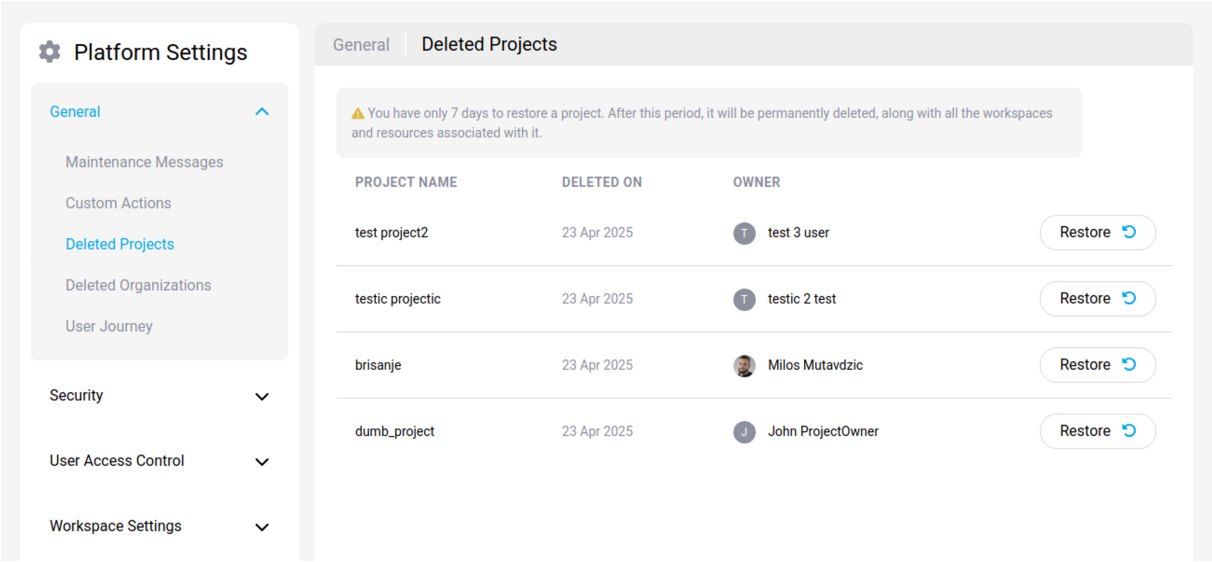
 




## Deleted Projects

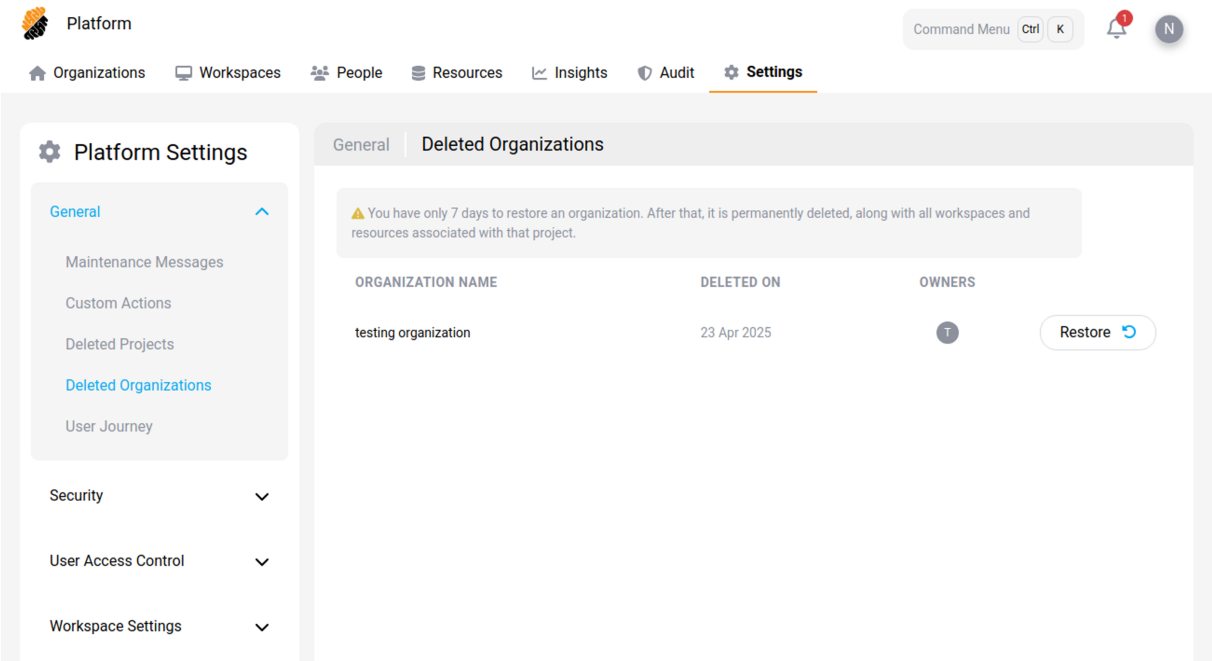
You can recover a deleted project for a period of 7 days on the **Deleted Projects** tab. Simply press the **Recover** button the right of the project you want to restore.




## Deleted Organizations

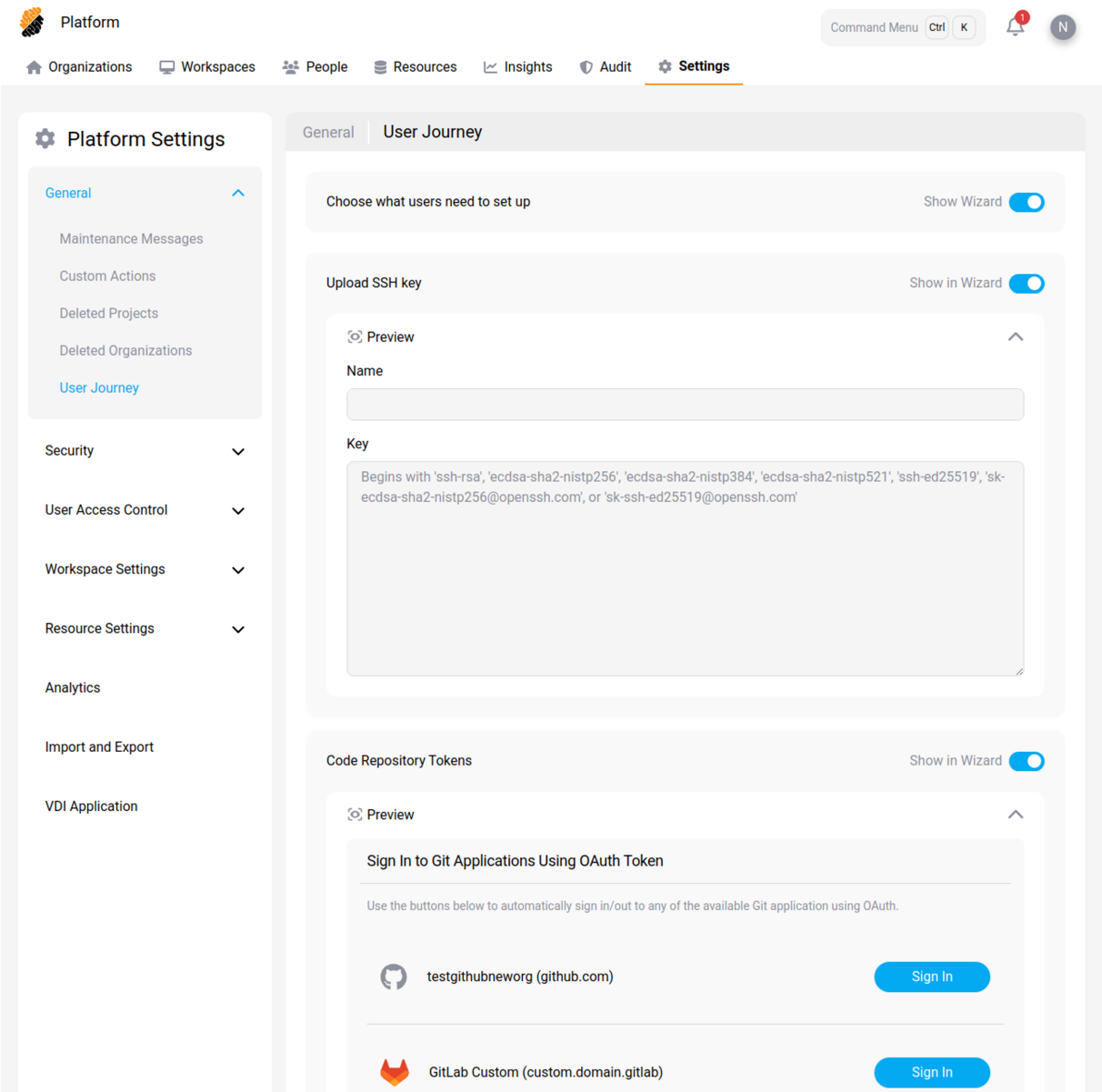
You can recover a deleted organization for a period of 7 days on the **Deleted Organizations** tab. Simply press the **Recover** button to the right of the organization you want to restore.




## User Journey

This section allows administrators to configure the initial setup wizard presented to users upon their first interaction with the platform.




## Security Settings

October 2, 2025

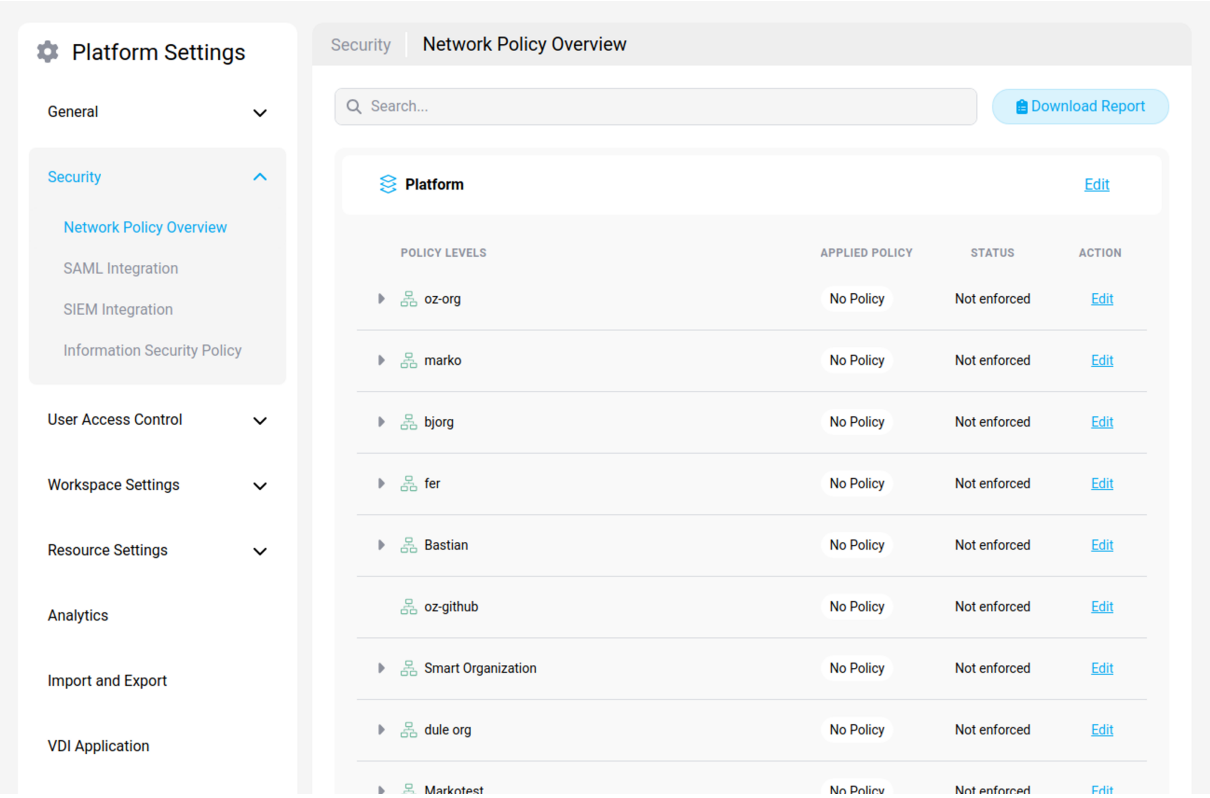
Configure critical security parameters for the entire platform. This includes managing **SAML Integration** for secure web application access via RBI, setting up **SIEM Integration** for centralized logging, getting a **Network Policy Overview**, and establishing platform-wide **Information Security 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.

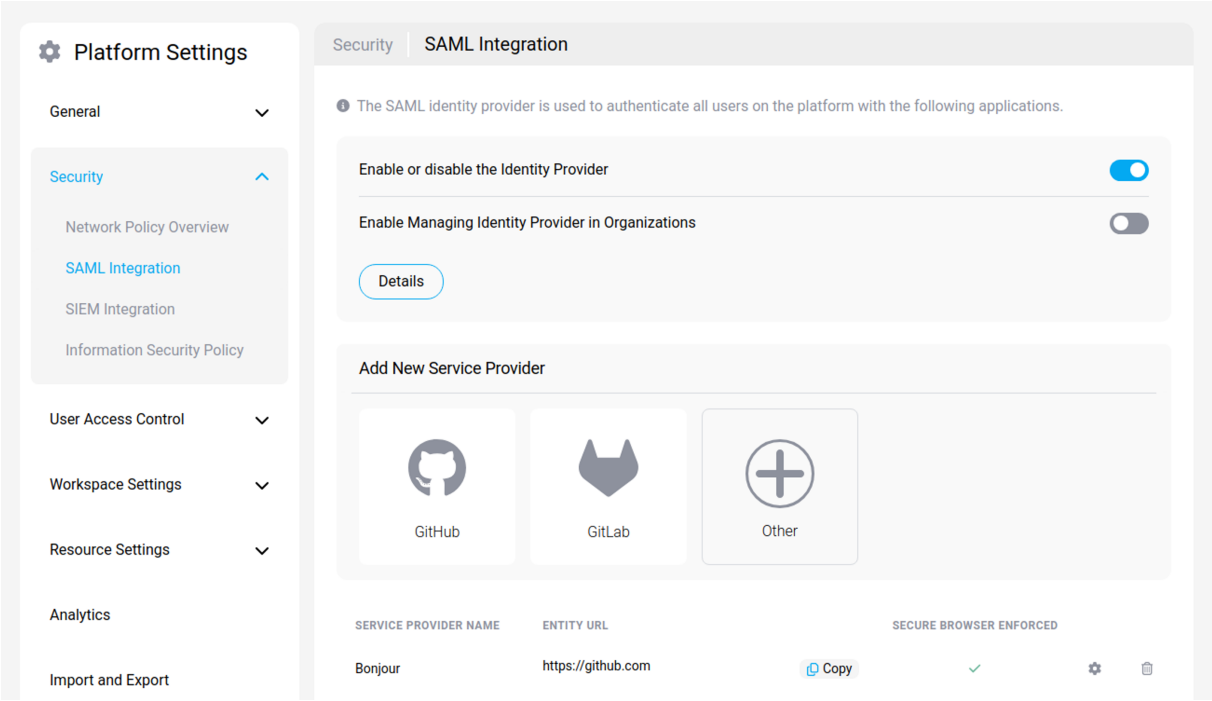



## SAML Integration

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), 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.

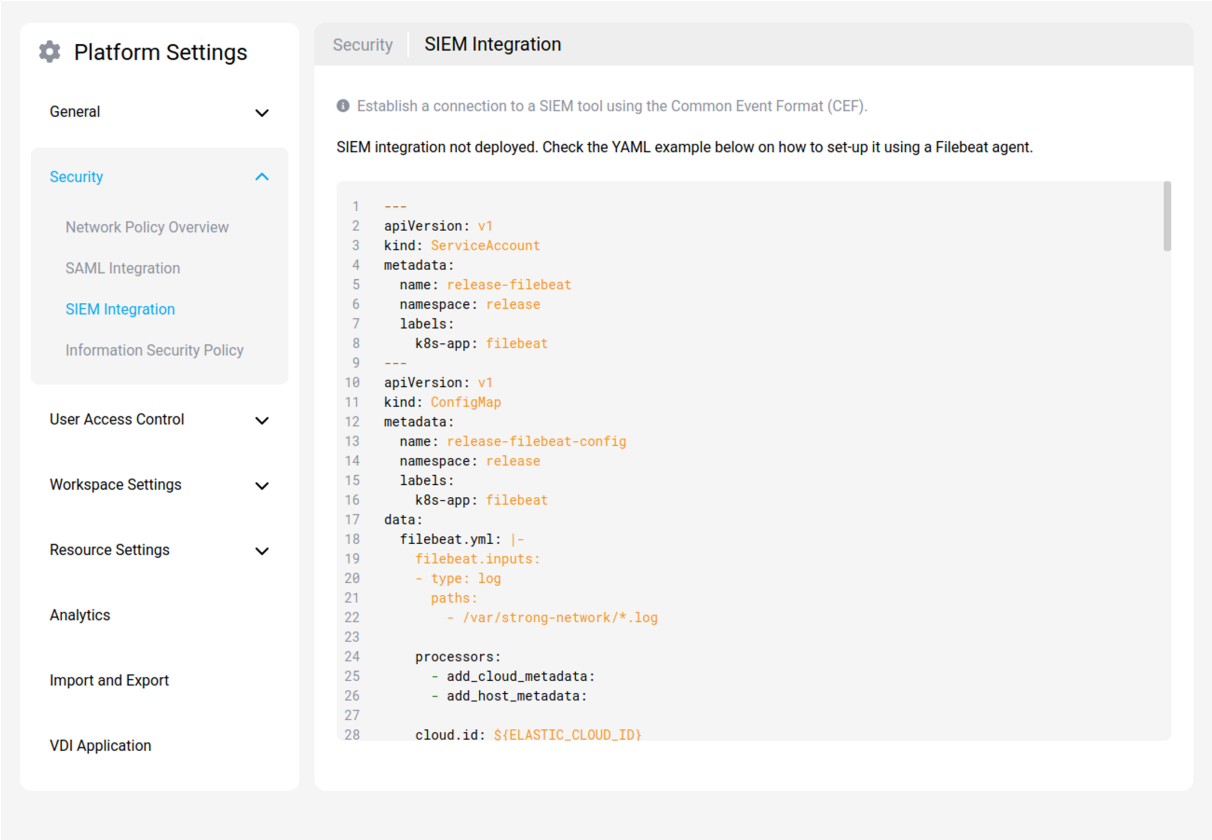
 




## SIEM Integration

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.

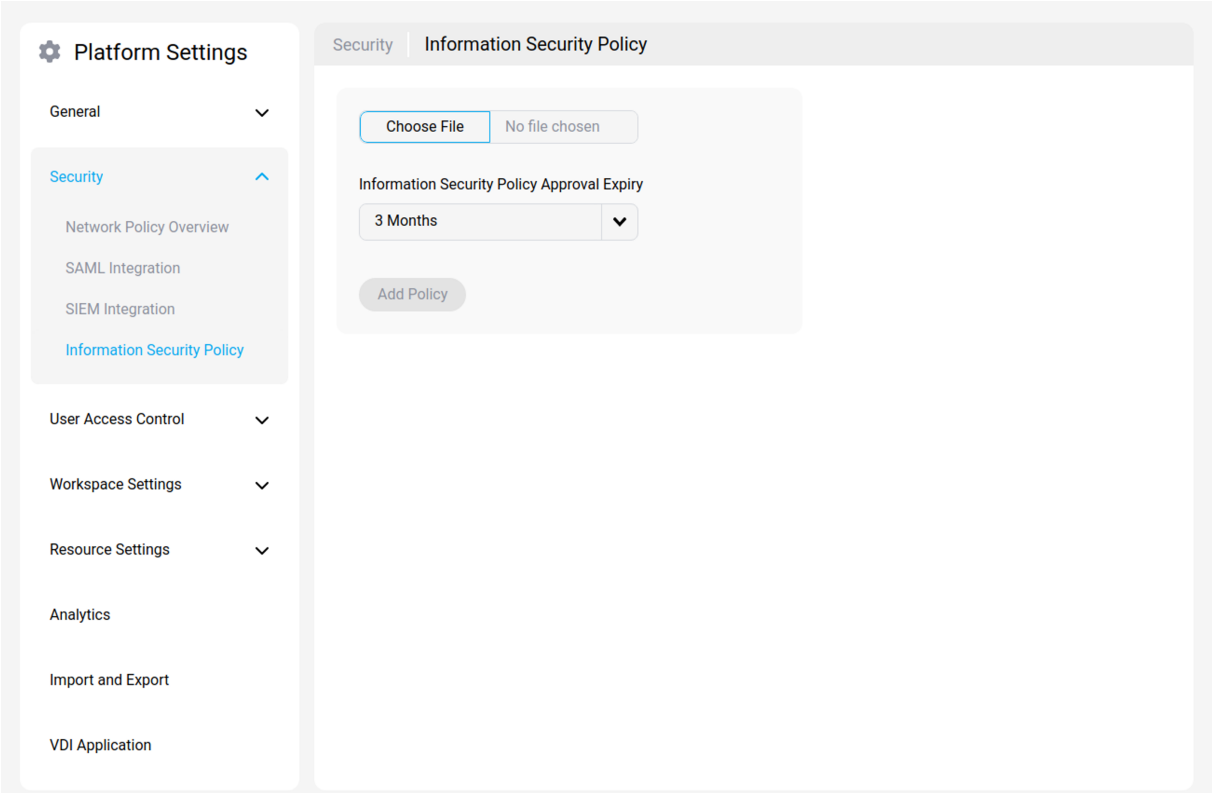
 




## Information Security Policy

Define and manage the information security policies enforced by the platform. This section may include settings related to data handling, access controls, and compliance standards that users and the system must adhere to.




## User Access Control

October 2, 2025

Manage how users authenticate and what they can access at the platform level. This involves configuring **Registered Domains and Identity Providers** (IDPs), including multi-factor authentication, and setting platform-wide rules via **User Access Control Settings** which encompass compliance features, platform constraints, and container image URL constraints.

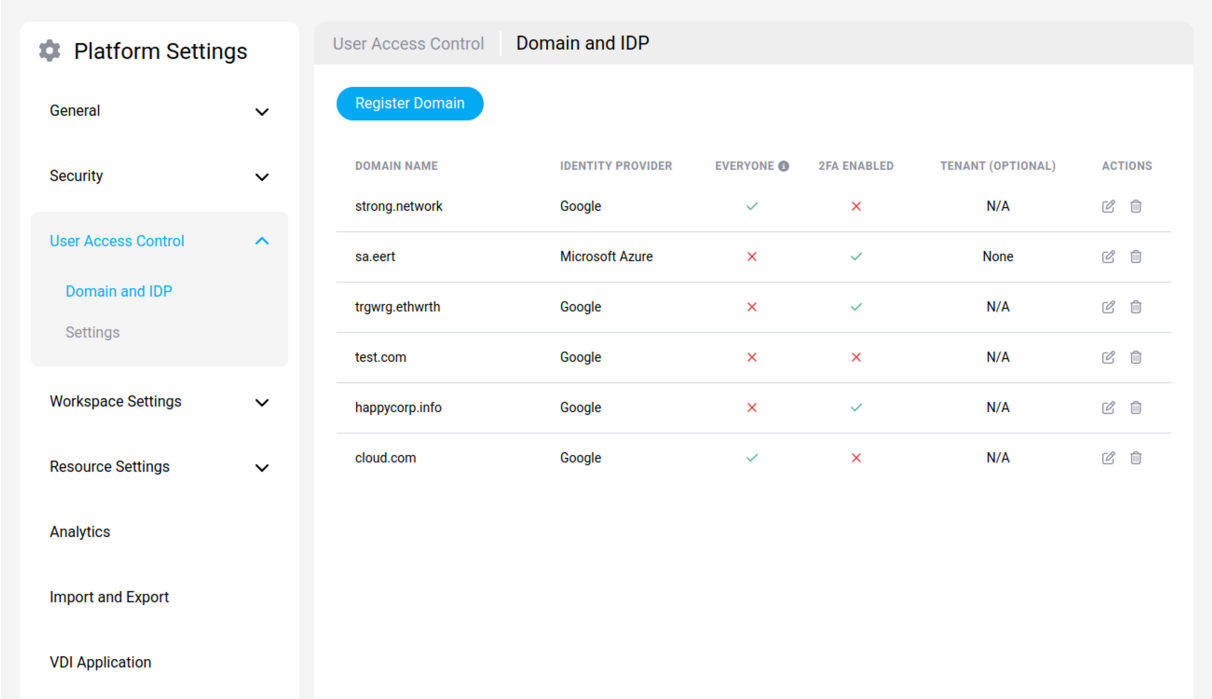
- Domain and IDP
- User Access Control Settings

### Domain and IDP

The **Registered Domains and Identity Providers** section offers a centralized control over user authentication processes. By defining specific domain names from which your users originate, you can associate them with a corresponding identity provider (IDP). As a result, users from the designated domain will be authenticated using the chosen IDP.

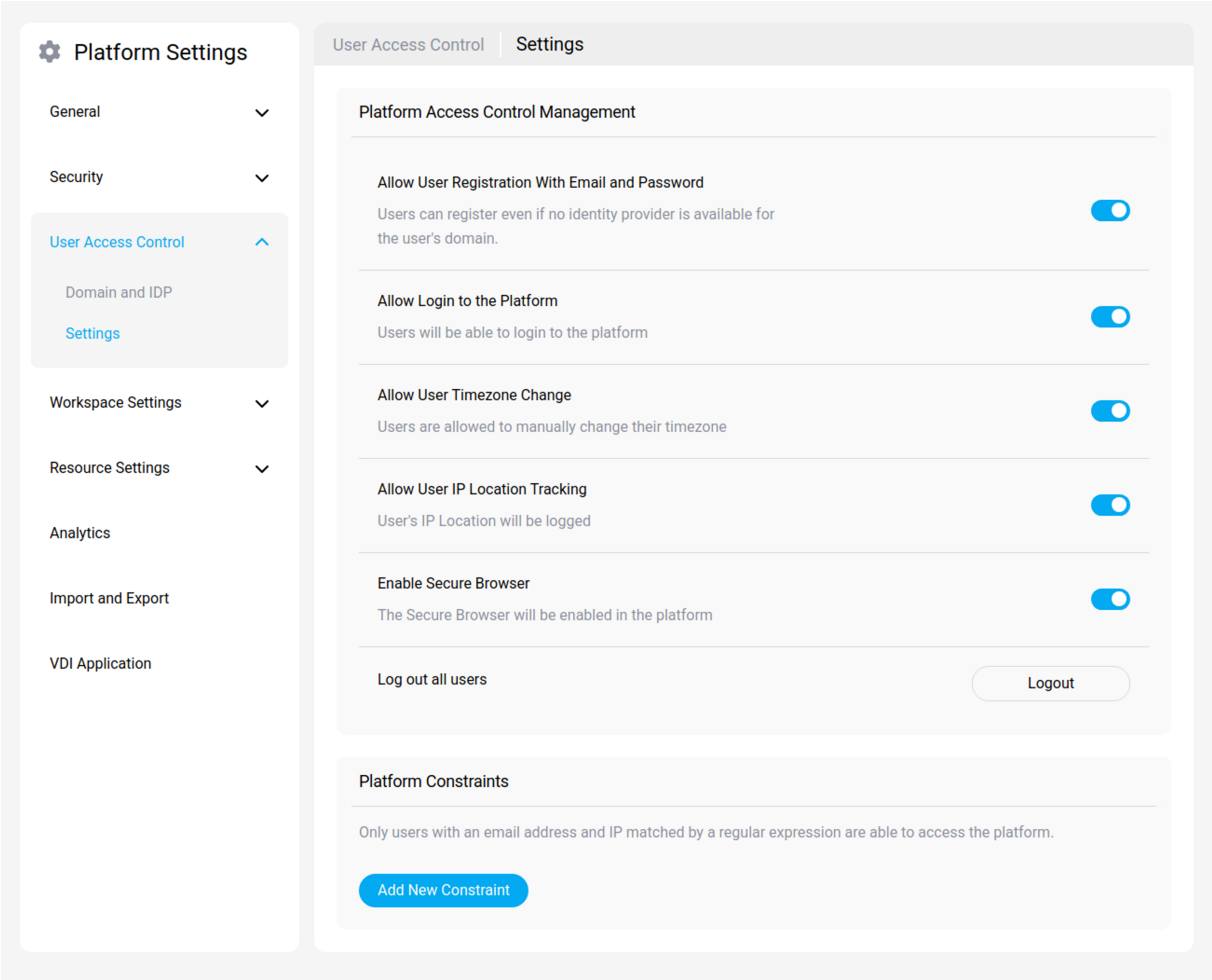
This section allows you to set access permissions based on specific domains and also offers the option to enable two-factor authentication, enhancing overall security.




## User Access Control Settings

The **User Access Control Settings** section offers features essential for meeting compliance requirements. These features encompass *Platform Access Control Management* and *Platform constraints*.




## Workspace Settings

October 2, 2025

Define the rules and defaults that govern individual workspaces created within the platform. Configure workspace-specific **Security Settings** like clipboard control and SSH access, manage **Schedule Settings** for workspace uptime, set policies via **Workspace Apps Settings**, define allowed **Workspace Specification** options (CPU/RAM), control workspace **Network Policy**, and manage workspace-specific **Registry Access**.

- Security Settings
- Schedule Settings
- Workspace Apps Settings
- Workspace Specification
- Network Policy

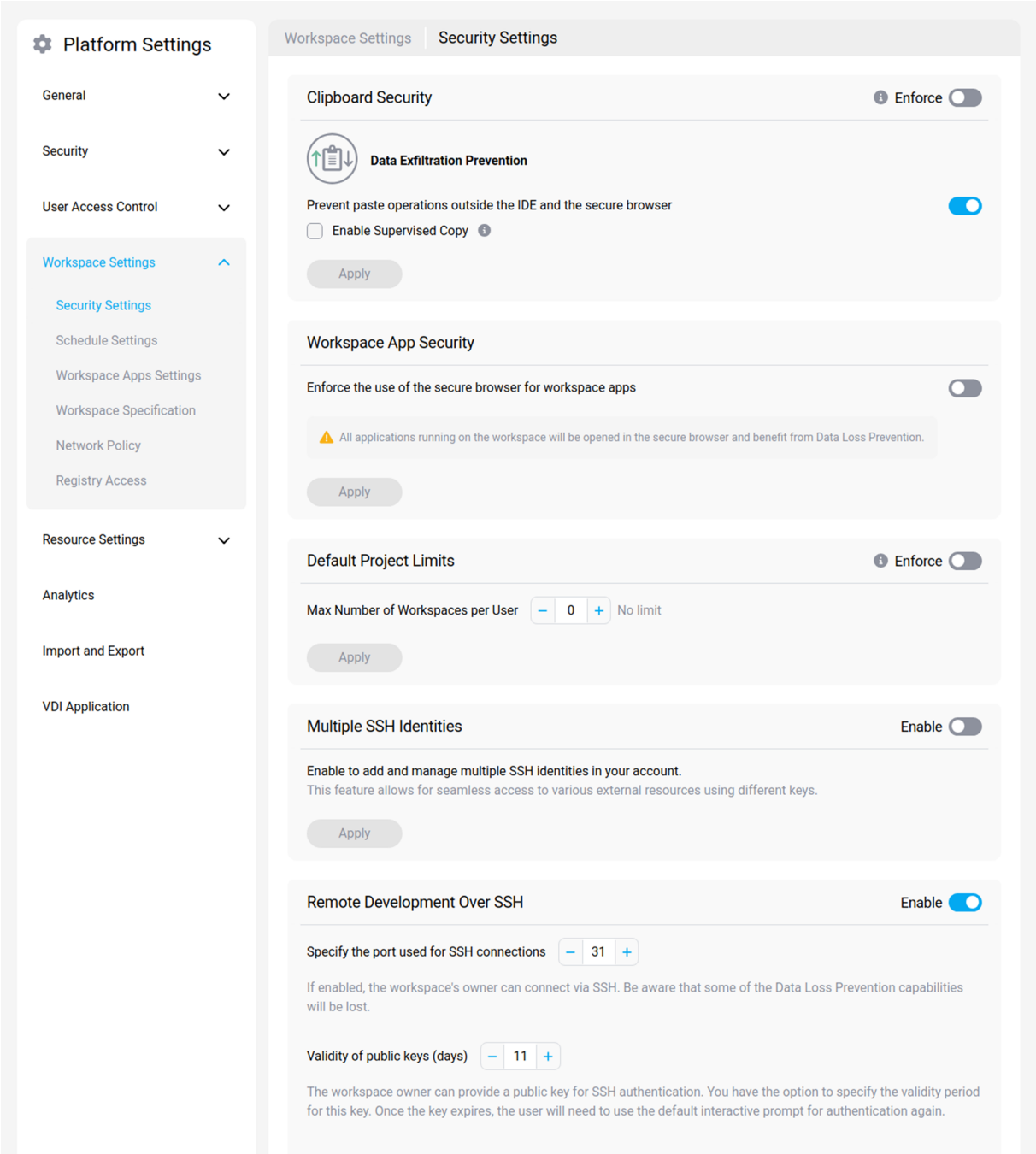
- Registry Access

### Security Settings

The **Security Settings** let you enforce security rules within underlying organizations and projects.

1. Clipboard Security: If enabled, users are prevented from pasting content outside of the IDE and the Secure Browser.
2. Personal Key Settings: If enabled, it permits workspace owners to use their personal OAuth tokens to authenticate with external repositories.
3. Default Project Limits: If enabled, it restricts users to a specified maximum number of workspaces, ensuring resource conservation.
4. Connect via SSH: If enabled, it grants the workspace's owner permission to connect via SSH. However, it's crucial to note that certain Data Loss Prevention functionalities might be compromised.

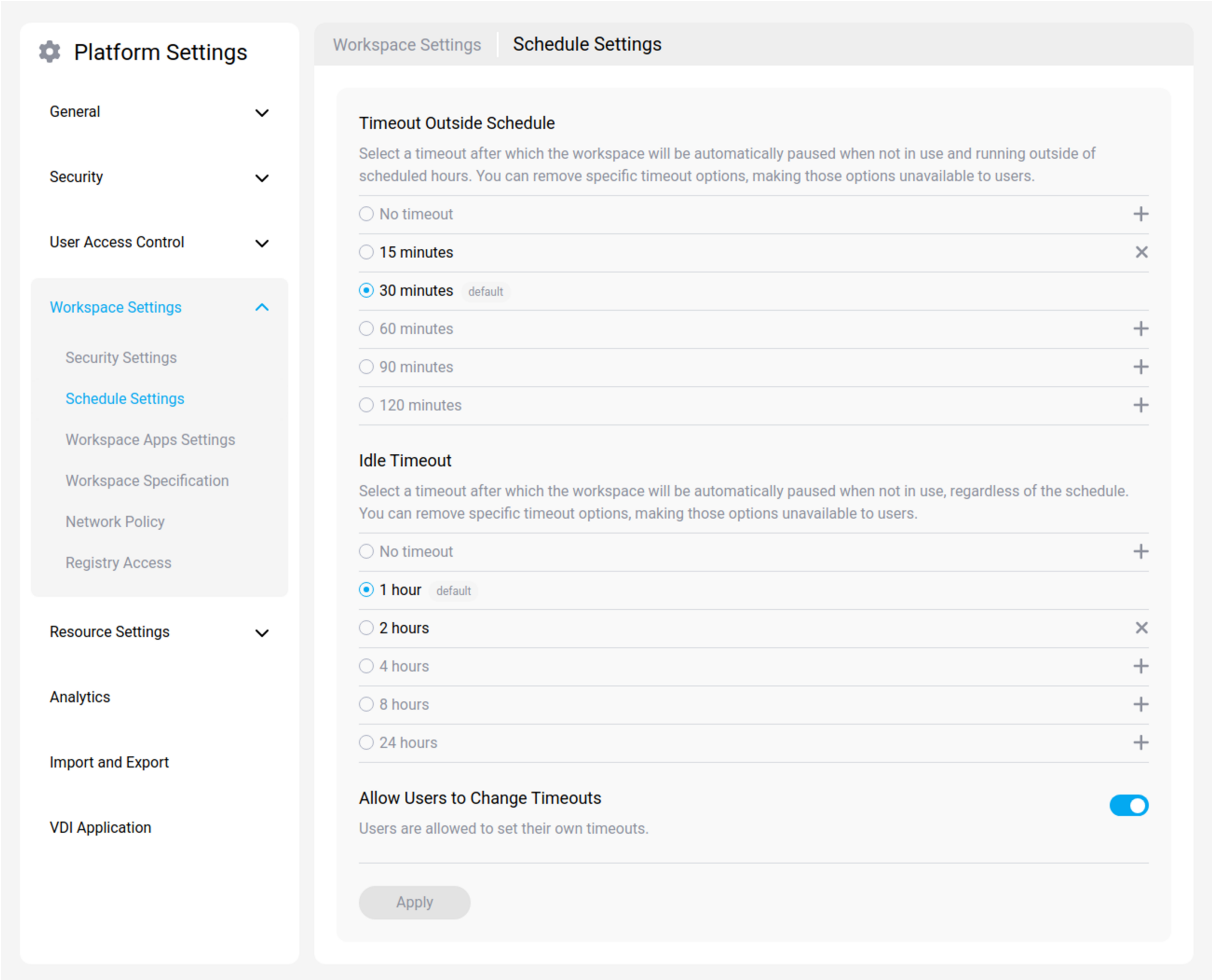
 




## Schedule Settings

Configure automatic scheduling for workspaces, such as setting operational hours or defining auto-shutdown policies. This helps manage resource consumption and ensures workspaces are only running when needed.

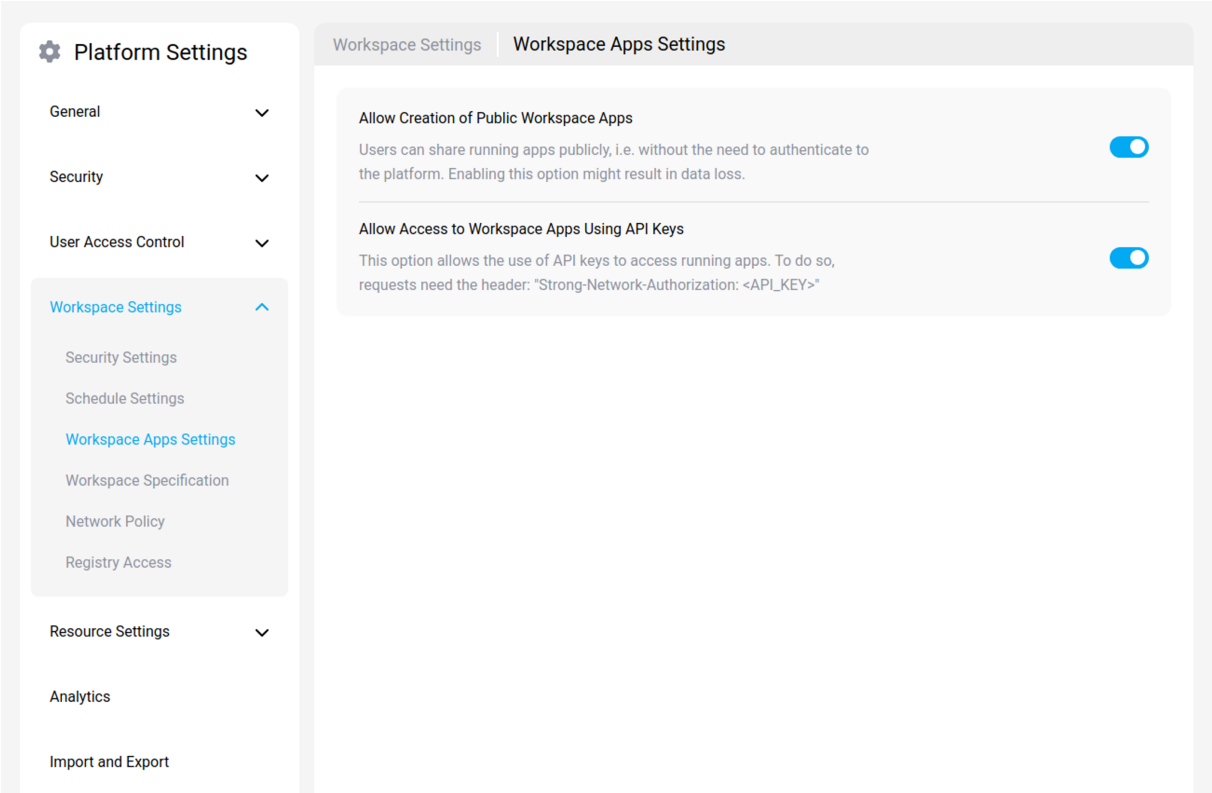



## Workspace Apps Settings

The **Workspace Apps Settings** section establishes guidelines for Workspace Apps within underlying organizations and projects.

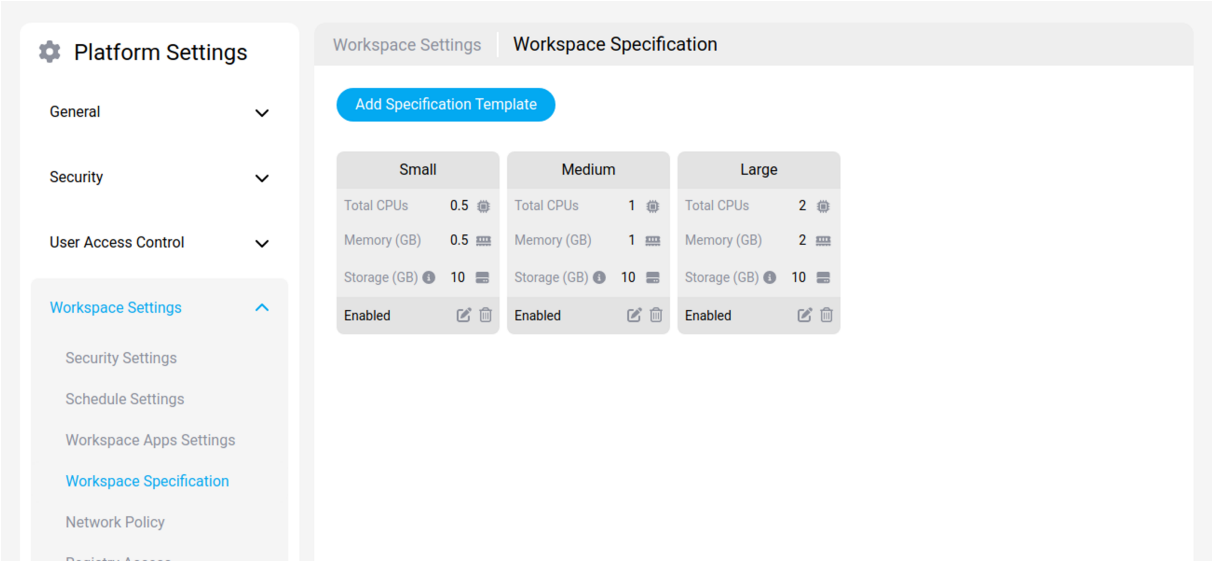
- Allow Creation of Public Workspace Apps: This feature permits users to share active apps with the public, meaning there's no requirement for authentication to the platform. However, activating this option may lead to potential data loss.
- Allow Access to Workspace Apps Using API Keys: This option grants users the ability to utilize API keys for accessing active apps. When doing so, requests should include the header: "Strong-Network-Authorization: ".




## Workspace Specification

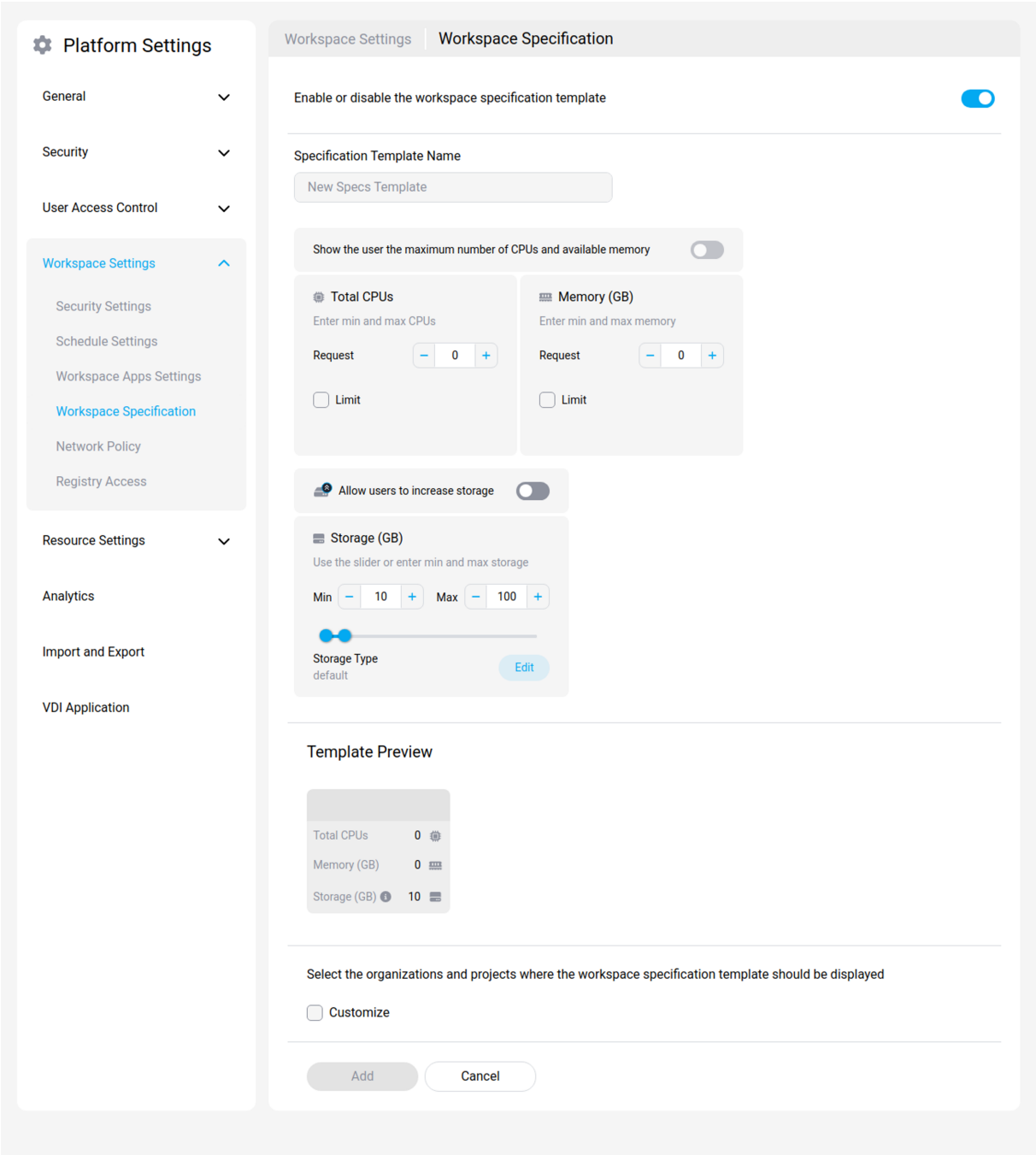
The **Workspace Specification** section allows administrators to create predefined templates that define resource allocations for workspaces.




When creating a template, you can set both initial 'request' values and maximum 'limit' values for CPU, RAM, and storage. You can also customize template availability, restricting specific templates to cer-

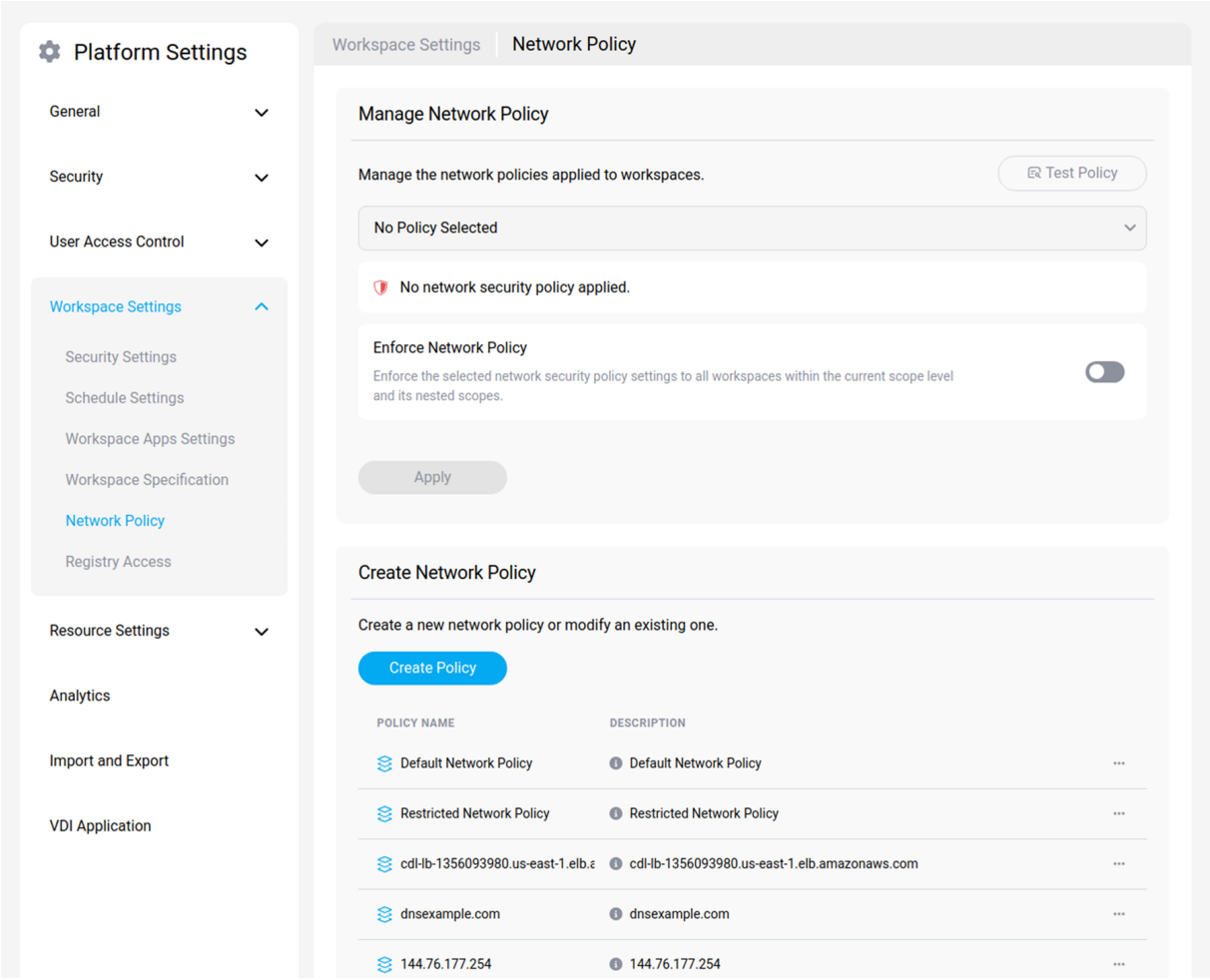
tain organizations or projects. When users later create a new workspace, they will only see the templates applicable to their context.




## Network Policy

Define specific network policies that apply to workspaces created within the platform. This allows administrators to control network traffic flow, segment networks, and enforce security rules at the workspace level.

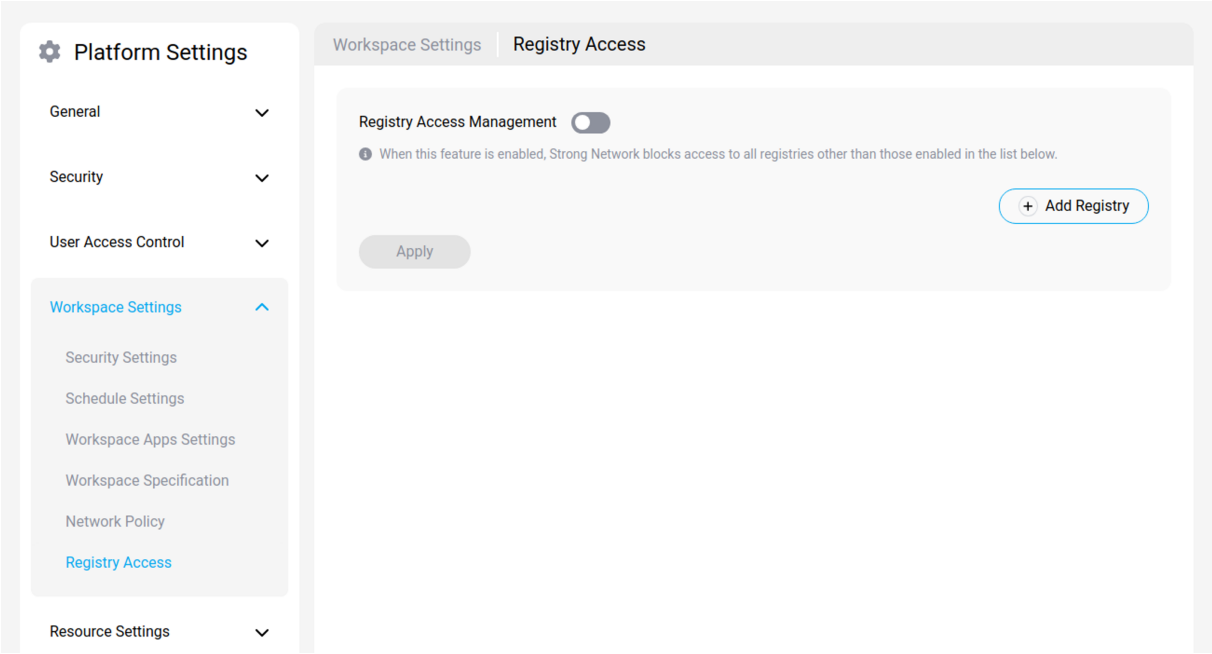
 




## Registry Access

Manage and control which container image registries workspaces are allowed to pull images from. This enhances security by ensuring that only trusted and approved image sources are used within development environments.




## Resource Settings

October 2, 2025

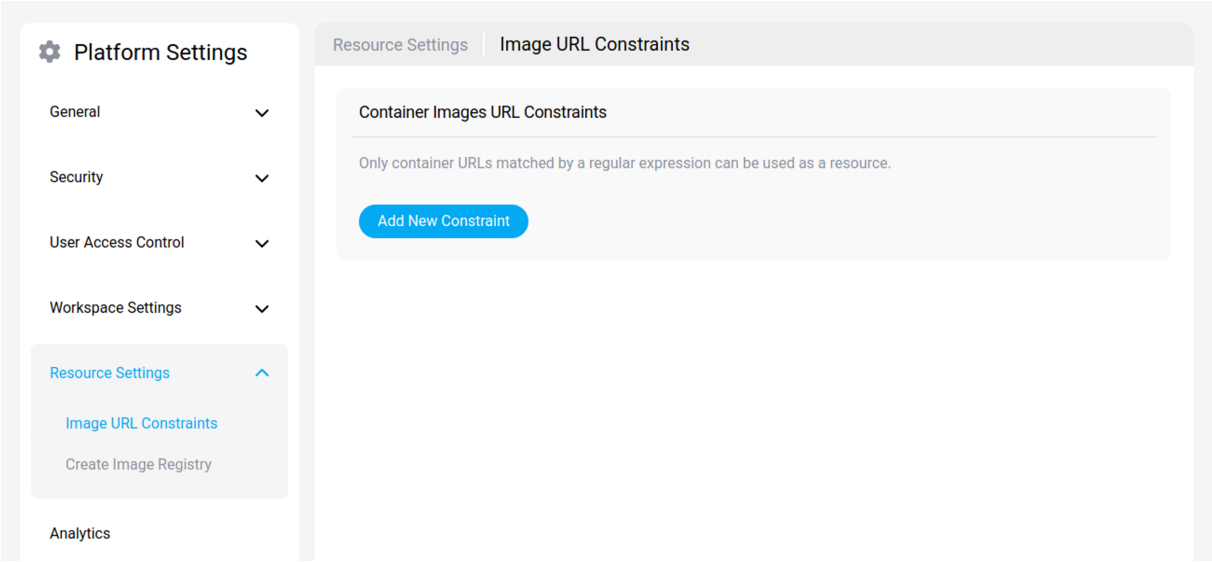
Control access to external resources used by the platform and workspaces. Primarily, this involves **Registry Access Management** (restricting allowed registries) and configuring connections to private registries via **Create Image Registry**.

- Image URL Constraints
- Create Image Registry

### Image URL Constraints

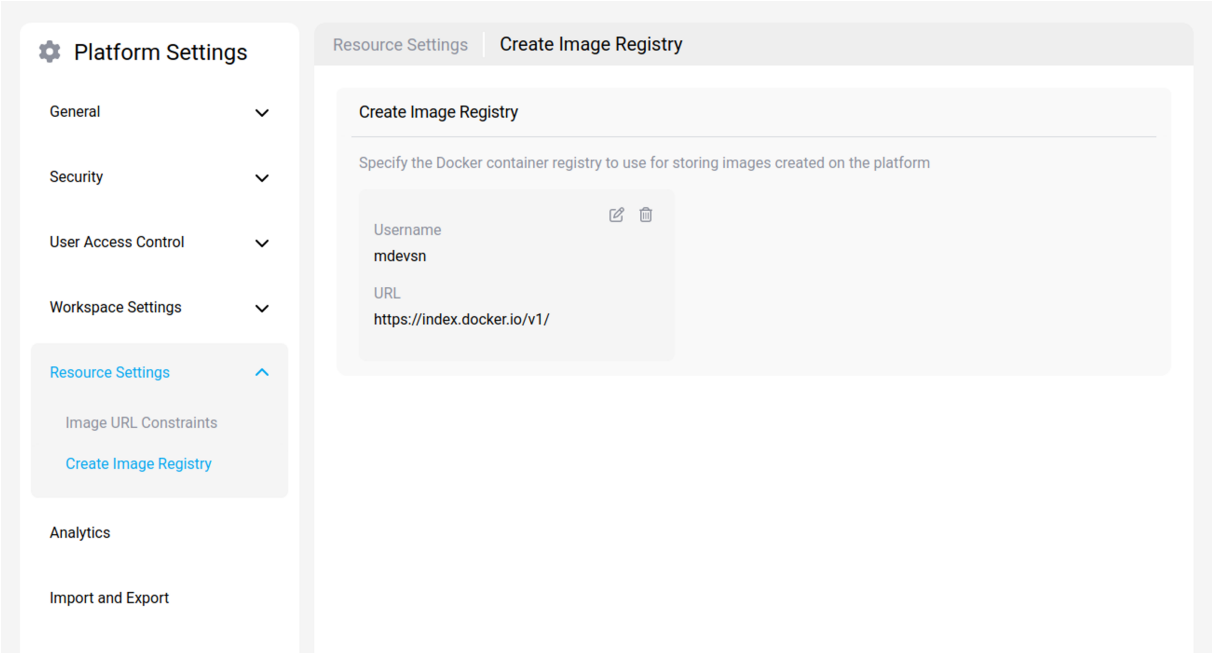
The **Image URL Constraints** section lets administrators ensure that their developers only access registries that are allowed. When this feature is enabled, Strong Network™ restricts access to all registries except those explicitly permitted in the list provided.




### Create Image Registry

Configure and manage connections to private or custom container image registries. This section allows you to add new registry credentials and endpoints for use across the platform.



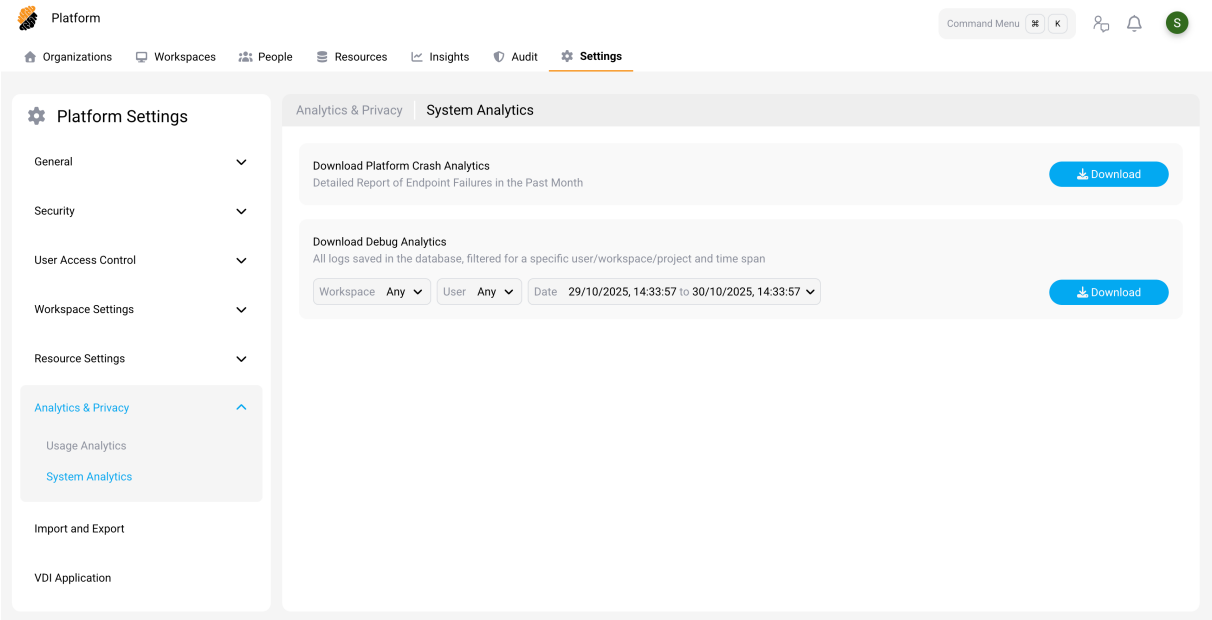

## Analytics

October 30, 2025

## System Analytics

Use the System Analytics section to download detailed reports and logs for the Citrix Secure Developer Spaces™ (SDS) platform. These reports include API and endpoint failure data from the past 30 days, along with comprehensive system logs.
You can filter the data by Workspace, user, or time range to support targeted troubleshooting and analysis.




## Usage Analytics

The SDS 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.

### Data Collection Preferences

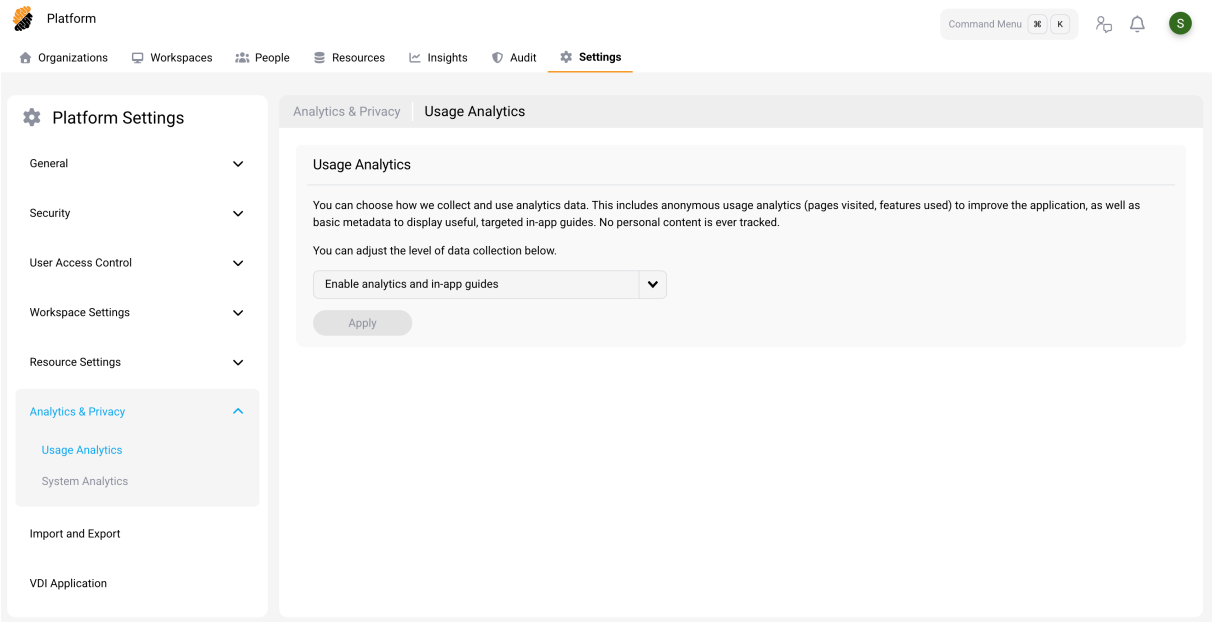
You can choose how analytics data is collected and used. This includes anonymous usage data (such as pages visited and features used) to improve the application, and basic metadata to enable targeted in-app guides. **No personal content is ever tracked.**

Available configuration options:

- **Enable analytics and in-app guides**
  This is the default configuration providing access to all Pendo-based functionality.

- **Disable analytics, keep in-app guides (basic metadata only)**
  No product usage information is shared with Citrix, but in-product guidance remains available.
- **Disable all analytics and guides**
  All Pendo components are disabled and no information is share with Citrix. In-product guidance, notifications, and the ability to submit feedback are not available.




### Connectivity Requirements

To ensure you can view Pendo content within the management console, Citrix recommends that the address ‘http://citrix-sds-content.customer.pendo.io’is contactable.

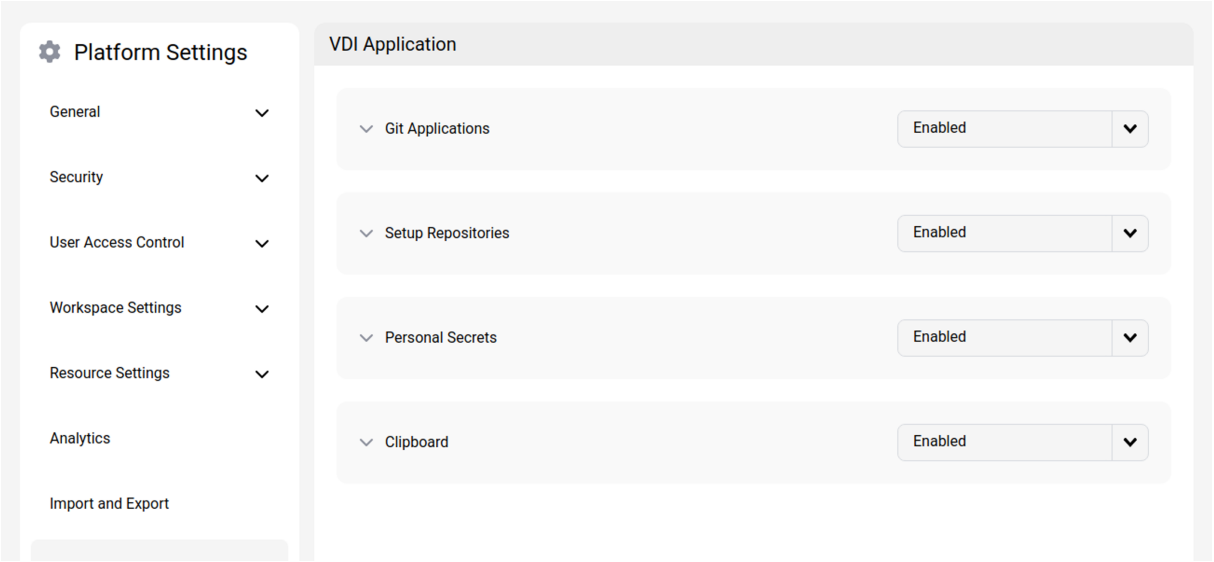
Pendo is a third-party sub-processor that Citrix uses to provide cloud and support services to Citrix customers. For a complete list of these sub-processors, see Sub-Processors for Citrix Cloud & Support Services and Citrix Affiliates

## VDI Application

October 2, 2025

Configure settings related to Virtual Desktop Infrastructure (VDI) Agent accessible through the platform.

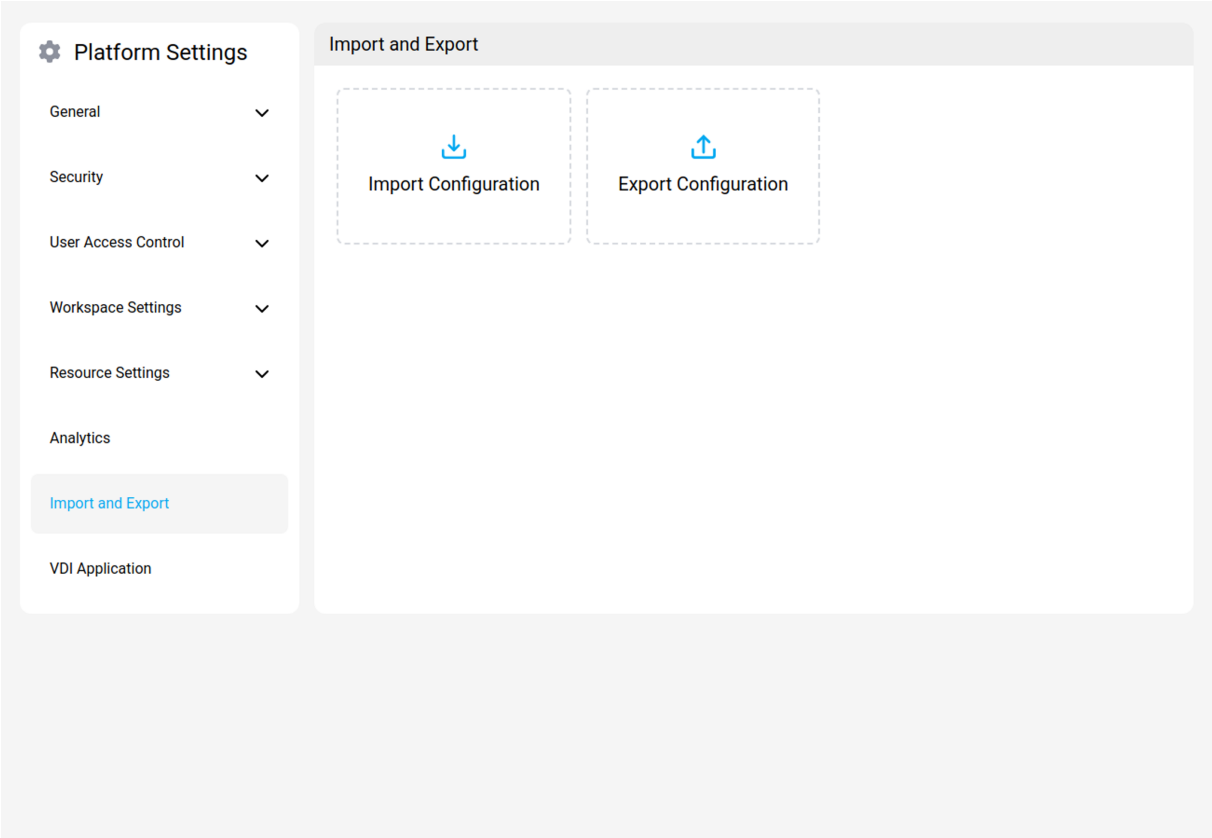



## Import and Export

October 2, 2025

This section provides options for importing and exporting platform configurations or data. This can be useful for backups, migrations, or sharing settings between different platform instances.



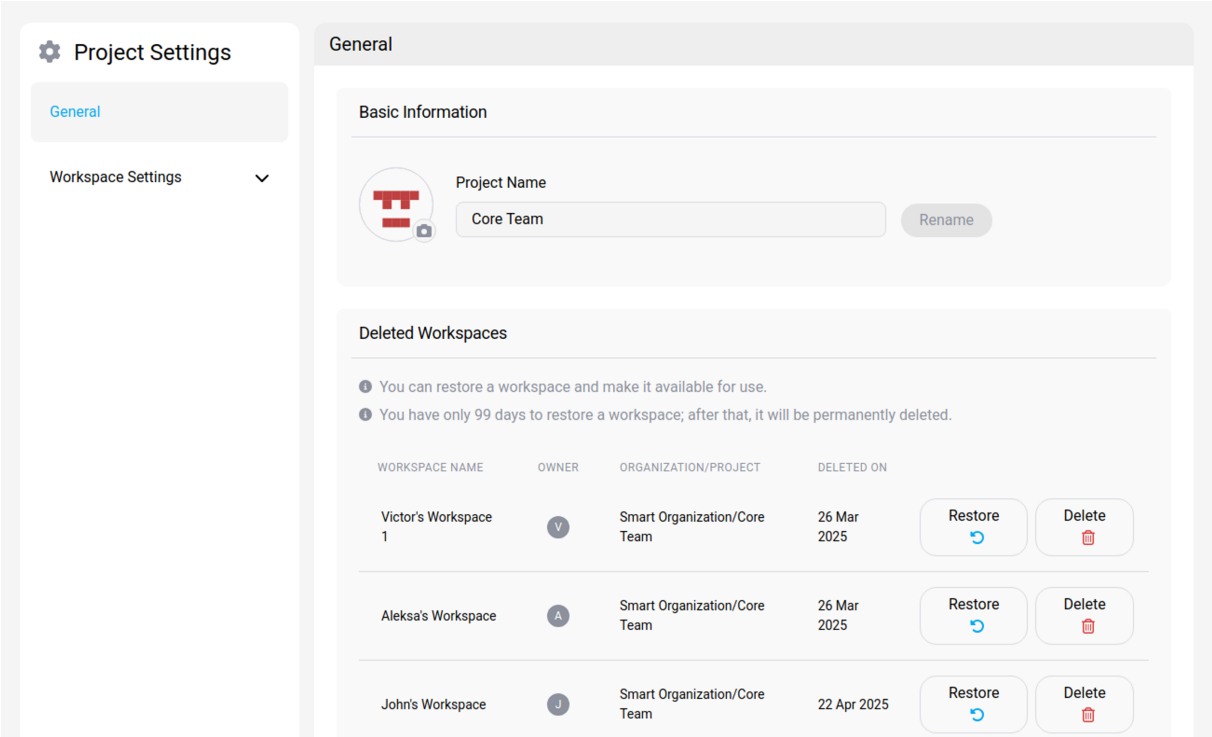

## Project General Settings

October 2, 2025

In the Project General Settings, you can update your project's name within the Basic Information panel.

Additionally, workspaces that have been deleted can be restored within seven days of their deletion. After this period, they will be permanently deleted.




## Workspace Settings

October 2, 2025

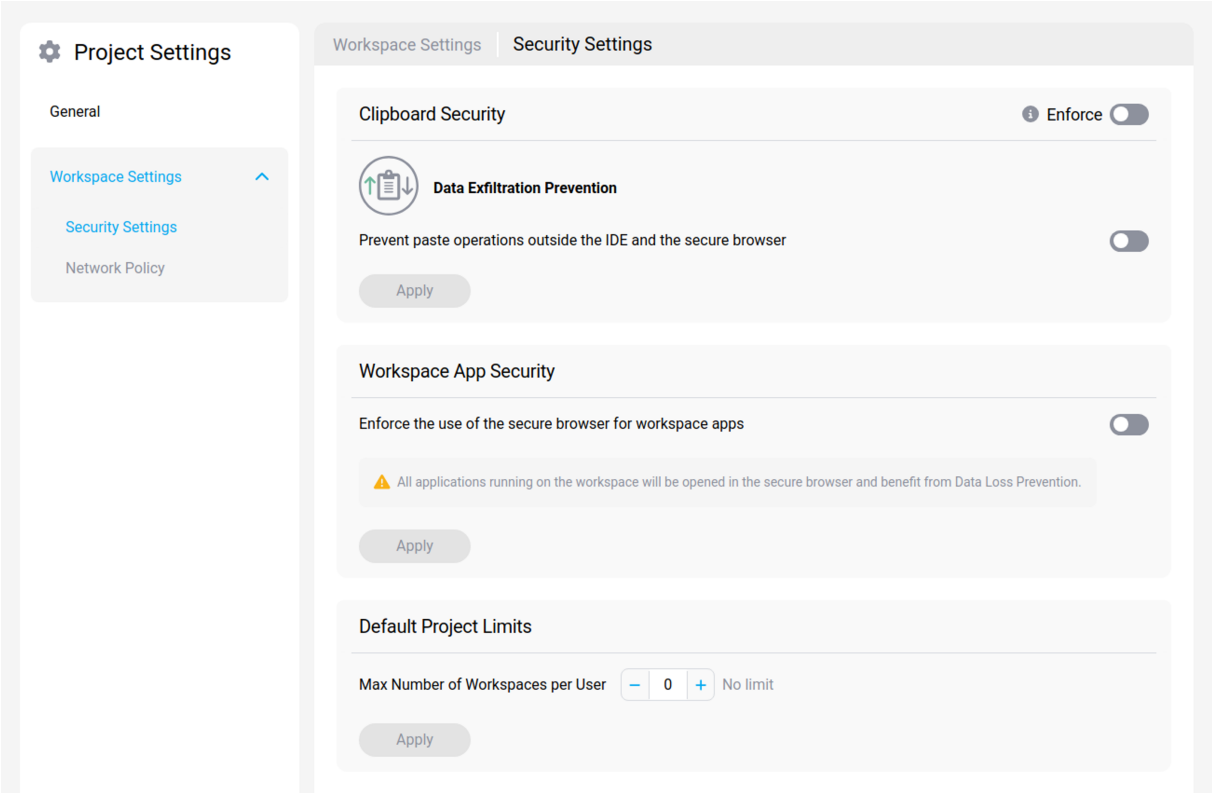
This section allows you to configure workspace settings specifically for this project. Define project-level security policies for data handling and access, and establish network policies to control workspace traffic within the context of this project.

- Security Settings
- Network Policy

### Security Settings

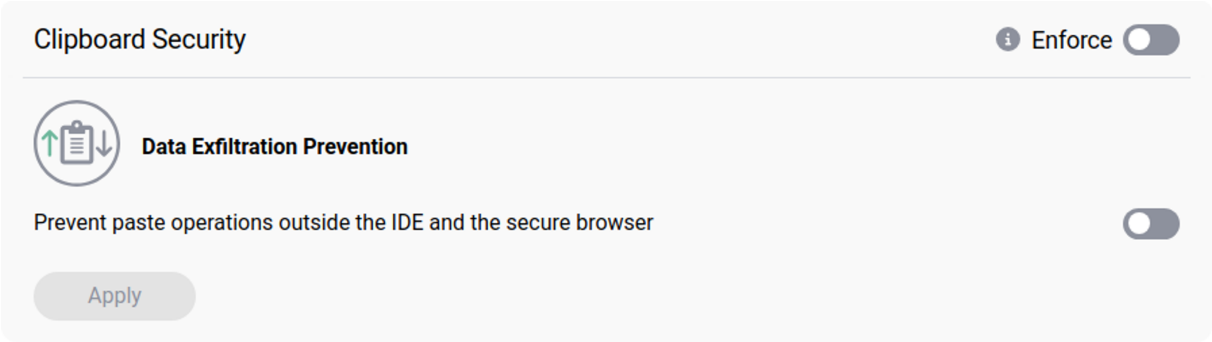
In the “Workspace Settings”section, the “Security Settings”enable you to implement multiple policies including Clipboard Monitoring, Workspace App Security, and Default Project Limits. These policies can be enforced to establish a foundational level of security across all workspaces within your project.




### Clipboard Security

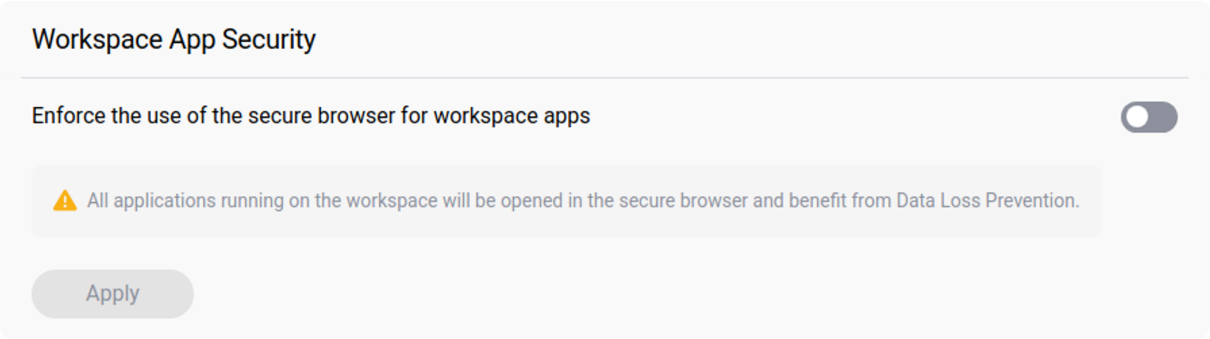
Clipboard Security implements Data Loss Prevention policies to safeguard against data leaks by disabling the ability to paste content from the IDE and secure browser into external applications.




### Workspace App Security

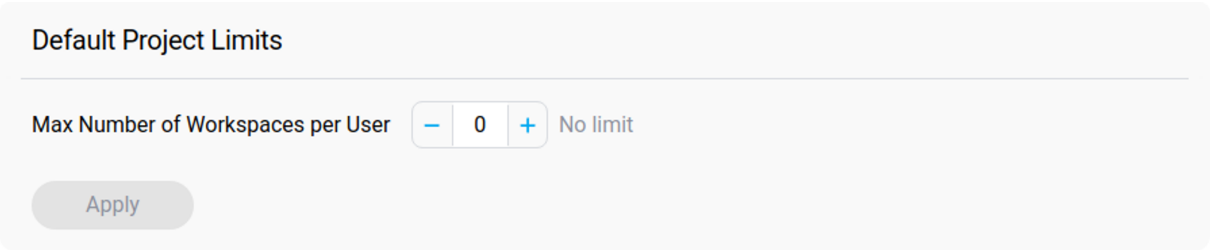
Workspace App Security allows you to mandate the use of a secure browser for workspace applications, ensuring that developers can share the applications they are developing in a protected environment. When used in conjunction with the Clipboard Security policy, this feature helps to prevent any potential data exfiltration from workspace applications.



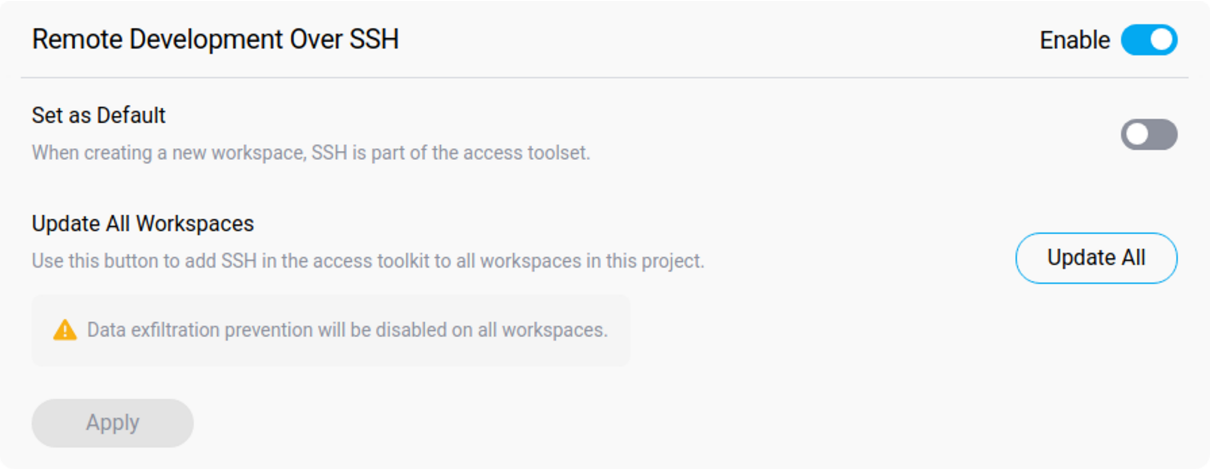

**Default Project Limits**

Default Project Limits can be set to cap the number of workspaces a user can create. This not only aids in resource monitoring and reduces unnecessary workspace proliferation but also contributes to cost efficiency by avoiding the operation of unused workspaces.




**Enable Remote Development Over SSH**

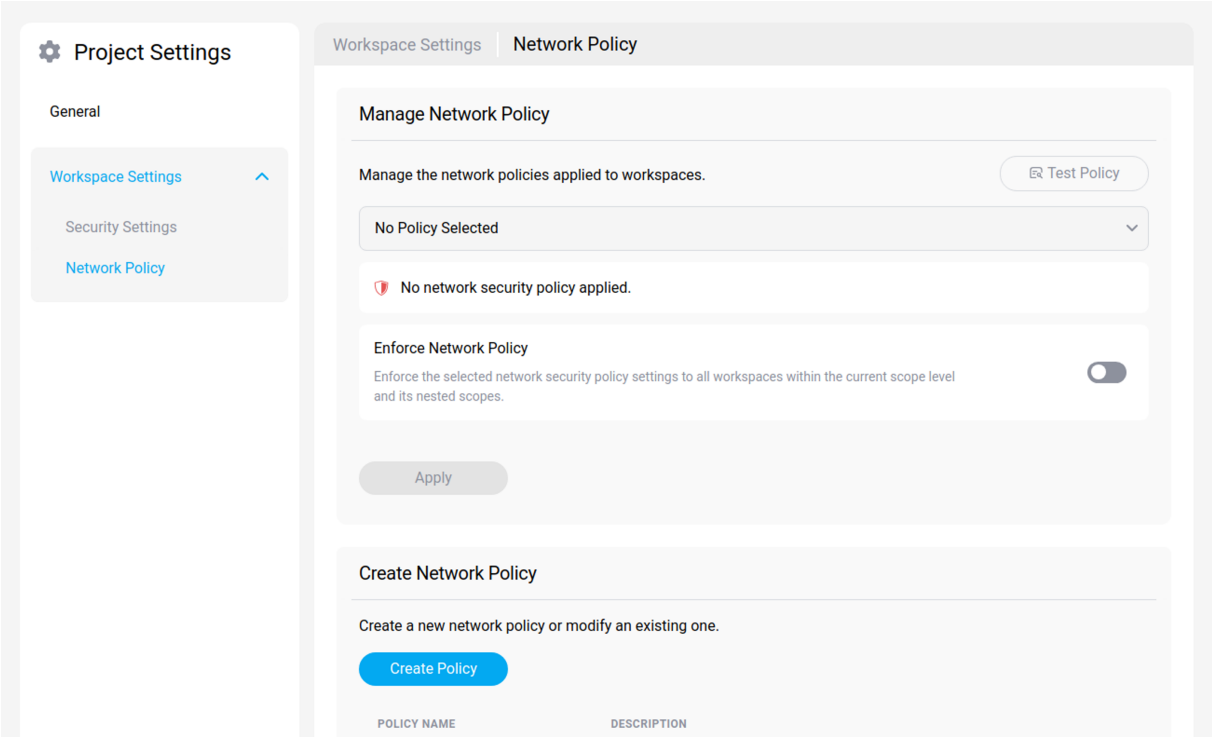
Remote Development Over SSH gives you the option to permit or deny developers the ability to connect to their workspaces via SSH. While convenient for certain tasks, this feature must be used judiciously as it can reduce the effectiveness of local IDE data loss prevention measures.



## Network Policy

Network policies are attached to workspace and enable fine-grained network traffic control. Network traffic is identified using combinations
of IP addresses, port and domain names. Once a network policy is attached to a workspace, all **outbound** traffic is enforced by the rules in
the policy and the workspace's user cannot circumvent the restrictions.




### Default Network Policies

Three default policies are available in a project. An administrator can create a new Network Policy if needed.

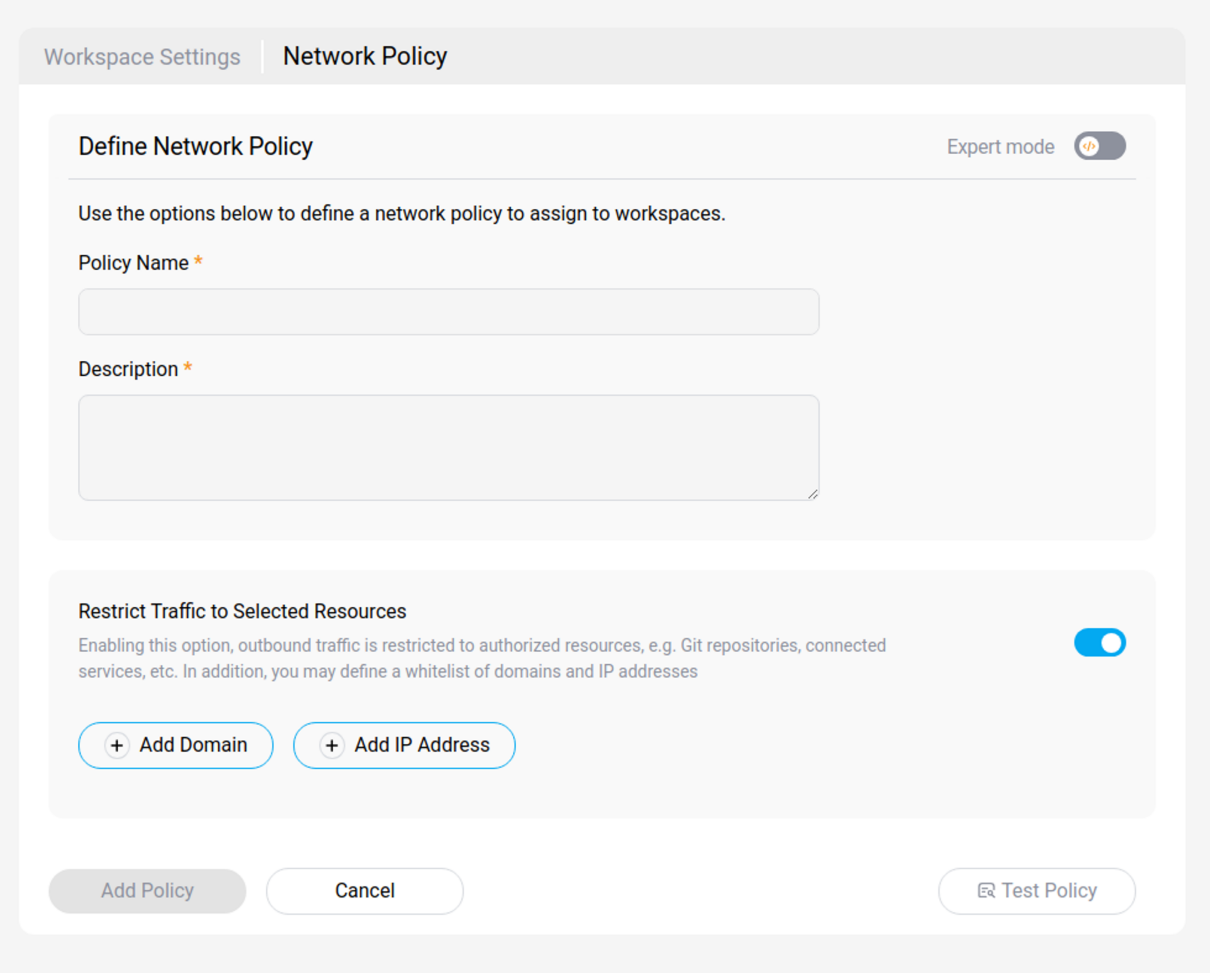
| Name | Scope | Description |
|---|---|---|
| **Monitor Traffic** | Project | This is a standard policy to monitor the outgoing traffic to the workspace. It will cause the generation of log events in the Audit dashboard. |

| Name | Scope | Description |
|---|---|---|
| **Restrict Traffic** | Project | This is a standard policy to restrict outgoing traffic from the workspace. It will block all traffic except to attached repositories and domains. Failed network requests are shown in the log events in the Audit dashboard. |

### Add a Network Policy

You can create a Network Policy by pressing the "**Create Policy**"button.




You will need to enter the following information:

1. **Name**, a name to identify the policy,
2. **Description**,

> **Warning**
>
> Be careful when naming and describing a new policy. A misleading name can end up in giving too many permissions to a user.

1. **Log and record outbound network traffic** (default),
2. **Restrict Traffic to Selected Resources** (optional),
   All traffic will be restricted, except for end systems added to your **whitelist**

- Add each application that you want to whitelist
- Add Domains that you want to whitelist, and indicate whether to include subdomains
- Add IPs that you want to whitelist

#### Edit or Delete a Network Policy

You can edit or delete a Network Policy by clicking on the "**...**"icon next to its class level.

## Help

In the help section, you can find the resources you need to make the most of the platform. Whether you're a beginner or an advanced user and find the documentation unhelpful, there are alternative options to get help.

- You can use the troubleshooting tool in case you experience problems.

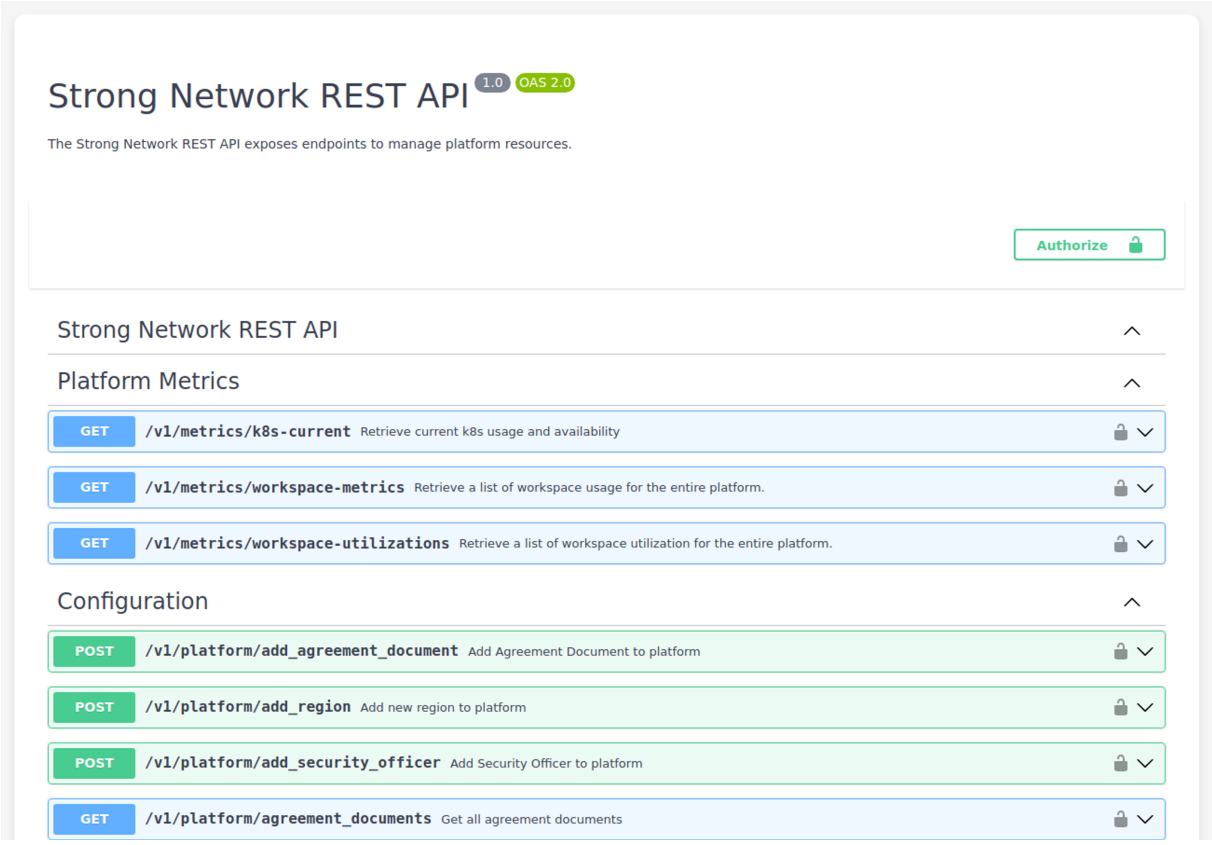
## REST API

October 2, 2025

The Strong Network™ platform can be fully controlled and integrated via an API of over 150 endpoints (detailed on the platform's API page) for complete control of enterprise applications and integration with security and analytics tools such as Splunk, Sumologic, etc.

> Info:
>
> Only users authenticated on the Strong Network Platform can have access to the API documentation.

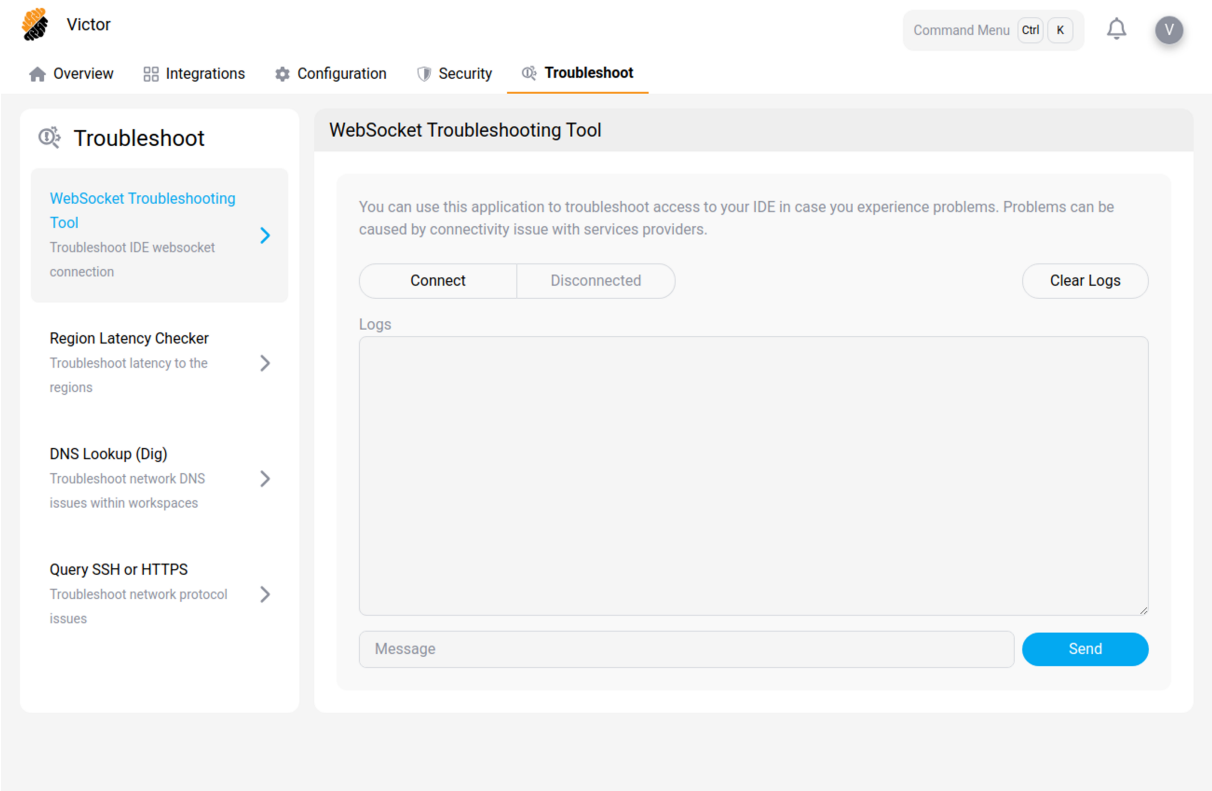



## IDE Troubleshooting Tool

October 2, 2025

In the Profile Settings you can setup the IDE WebSocket Troubleshooting Tool. You can use this application to troubleshoot access to your IDE in case you experience problems. Problems can be caused by connectivity issues with service providers.




You can also troubleshoot latency to regions with the **Region Latency Checker** tool.