PARALLEL WORKS

ACTIVATE

User Guide

The complete guide to running high-performance computing and AI workloads across on-premises and cloud resources.

Version 7.58.0 · June 2026
parallelworks.com

Parallel Works ACTIVATE — User Guide

This book was generated from the online documentation at parallelworks.com/docs on June 2026. It is a point-in-time snapshot; the online documentation is the authoritative and most up-to-date source.

For the latest version, visit parallelworks.com/docs.

Contents

About ACTIVATE

What We Do

The ACTIVATE control plane gives users the ability to create and manage elastic resource pools and high-performance computing (HPC) clusters on virtually any cloud service provider. ACTIVATE is designed to be accessible and universally useful: users can run scripts—called *workflows*—that are defined in a variety of programming languages across multiple pools and clusters.

ACTIVATE customers use their own cloud accounts to deploy cloud HPC resources via the ACTIVATE control plane, meaning that HPC resources are our customer-deployed solution. While ACTIVATE can be deployed in different locations (i.e., in the cloud or on-premises), it is not deployed by customers directly.

Our goal is to make high-performance simulation and modeling computation as simple as possible, and we're working towards that goal every day.

Working With Cloud Resources

The process for using cloud resources typically follows these steps:

- Log in to ACTIVATE.
- Create or start a cluster with the configuration that best suits your work.
- Transfer your data to the cloud from an on-premises location or from an object storage, such as Amazon Web Services (AWS) S3 buckets, Google Cloud Storage (GCS) buckets, or Microsoft Azure Blob containers.
- Complete your computations.
- Transfer your data back to an on-premises location or to object storage.
- Shut down the cluster.

If you've never used ACTIVATE before, you can check out our **Getting Start** tutorial to learn more about using clusters.

Essential Cloud Concepts

Clusters are standard HPC clusters well-suited for supporting the execution of a wide range of parallel applications, including MPI, OpenMP, and GPU-based applications as well as various hybrid combinations. Cloud-based clusters can run a variety of job schedulers, providing an environment that's familiar to users of traditional HPC clusters. On ACTIVATE, a cluster has one controller node that sends jobs to the compute nodes, which are grouped into partitions.

Workflows are scripts written in a variety of languages and programming models. Typically, workflows orchestrate the execution of applications relevant to your domain, whether that's computational fluid dynamics for mechanical engineers or molecular modeling for biochemists. Workflows are run on either pools or clusters, depending on the need of the workflow. There are several demonstration and template workflows available in the ACTIVATE Marketplace (see **Marketplace** for more information).

About This Guide

We constantly update the ACTIVATE control plane, adding features to make it easier to complete large-scale projects. Our documentation is a work in progress: the following instructions and information will be updated regularly as we work to improve ACTIVATE and our users' experience.

We would love to hear your feedback. Feel free to share your requests for additional features and thoughts about ACTIVATE at feedback@parallelworks.com.

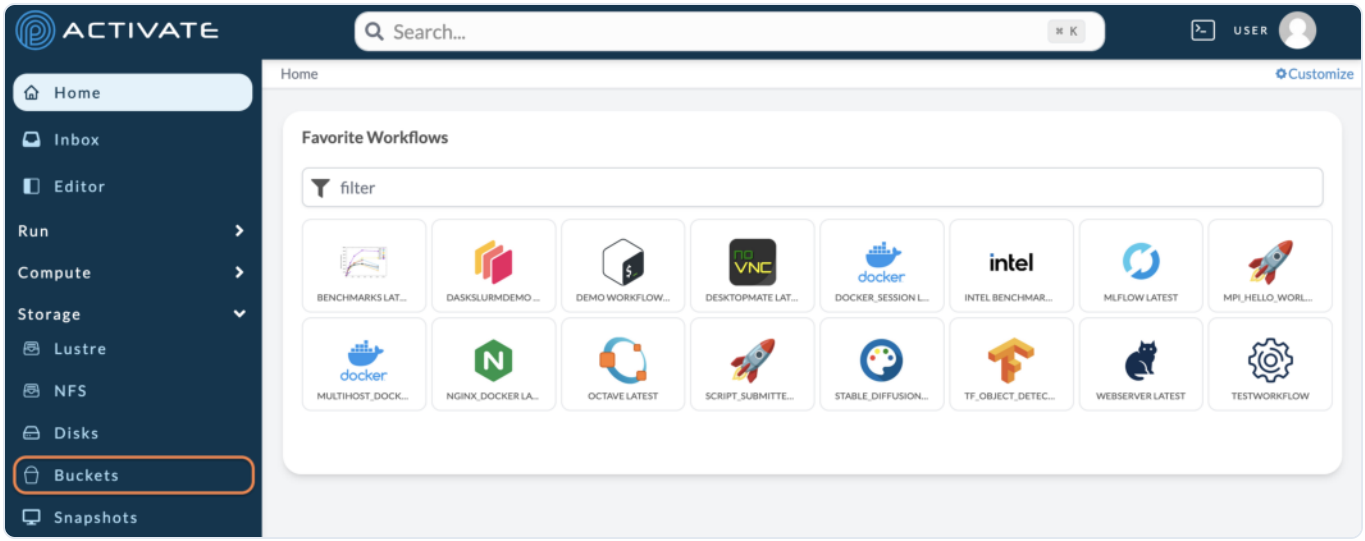
Getting Started

Because there are many uses for ACTIVATE, this tutorial's aim is to familiarize users with the platform and typical steps a user might follow. You will learn how to provision elastic HPC clusters, manage buckets, transfer data, and run computations.

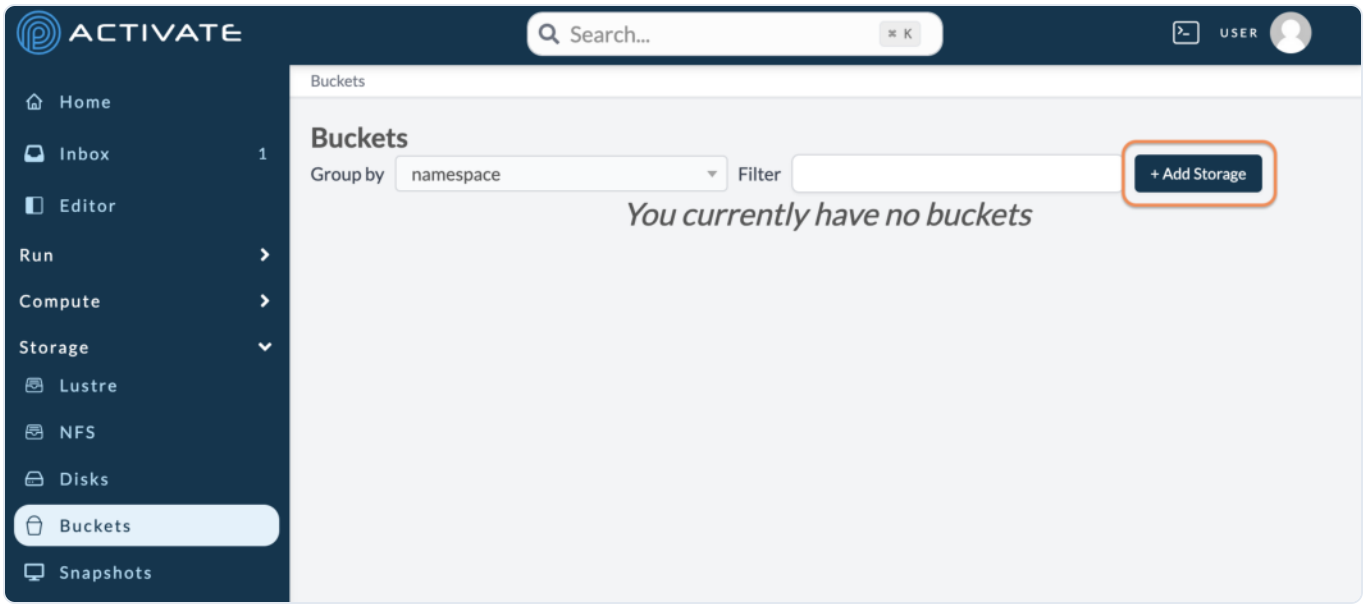
Provision a Bucket

You can use several different types of storage on ACTIVATE. For this guide, we'll be using bucket storage. Buckets are efficient, inexpensive, and and easy to use.

Navigate to **Buckets**.



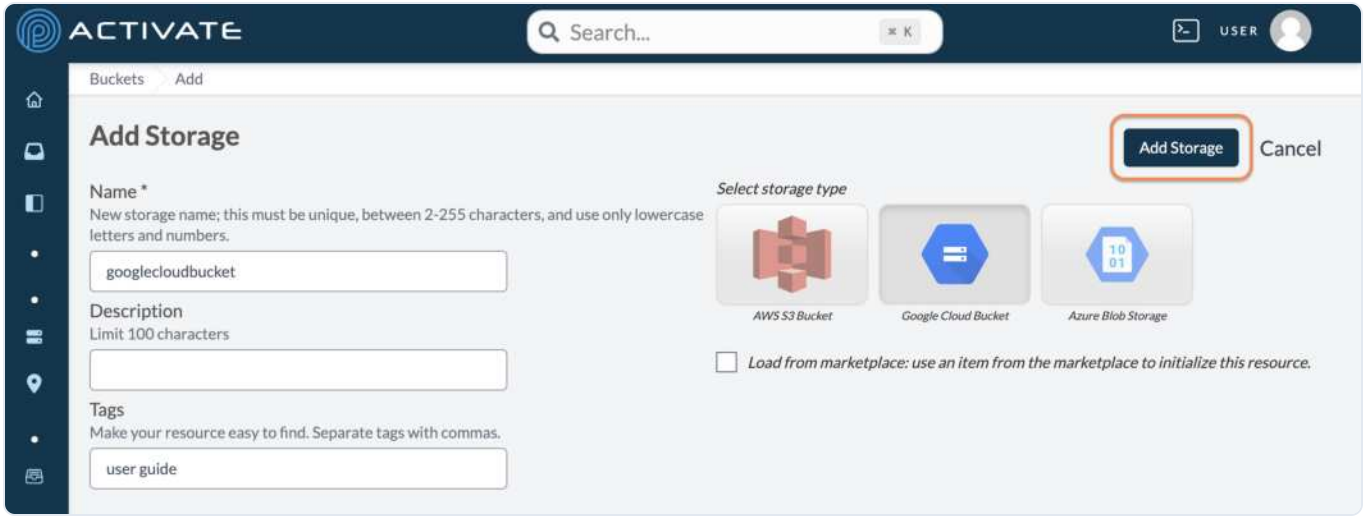
Click **+Add Storage**.



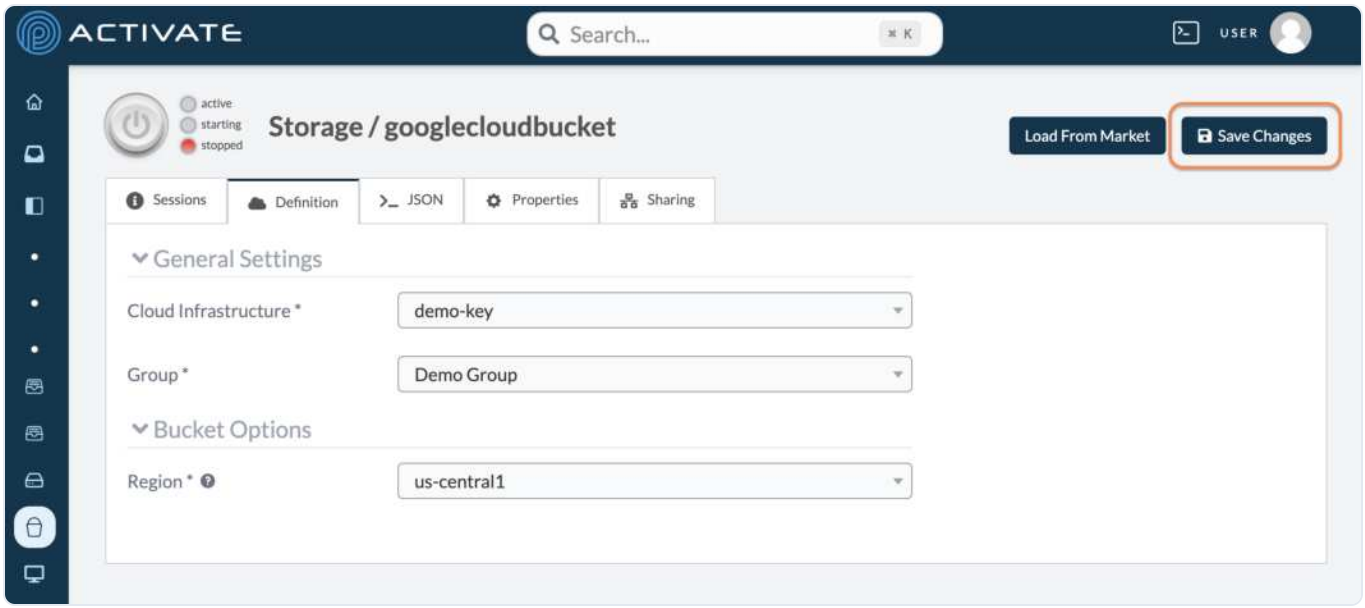
Choose which type of bucket you'll create.

Give your bucket a **Resource Name**. Optionally, add a **Description** and **Tags**.

Click **Add Storage**.



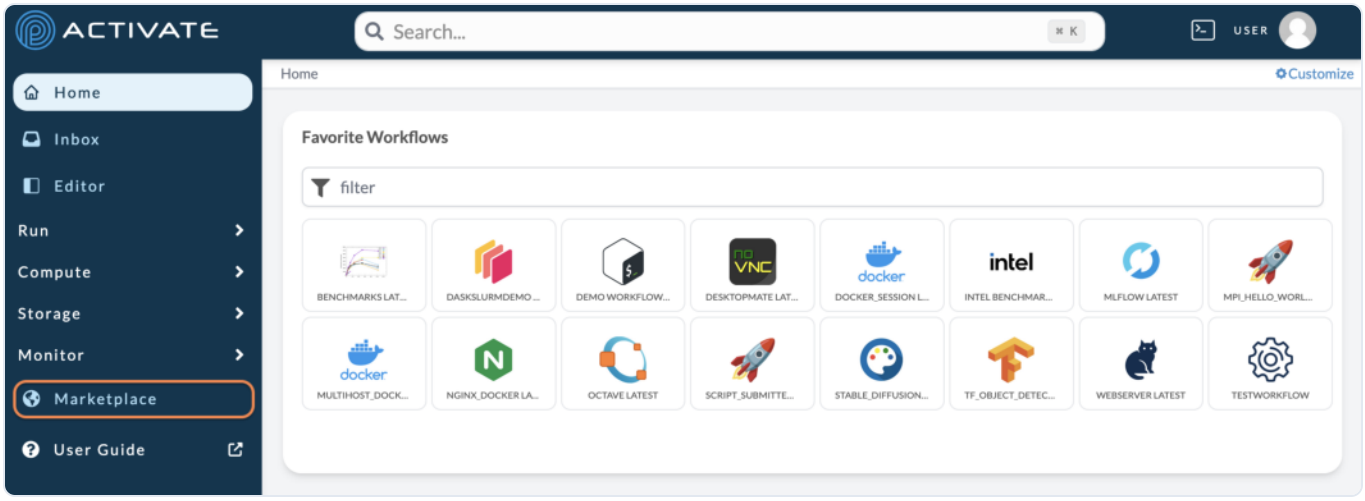
On the next page, the bucket will be pre-configured with your organization's default **Cloud Infrastructure** and **Group**. If you need to change these parameters, click **Save Changes** after doing so.



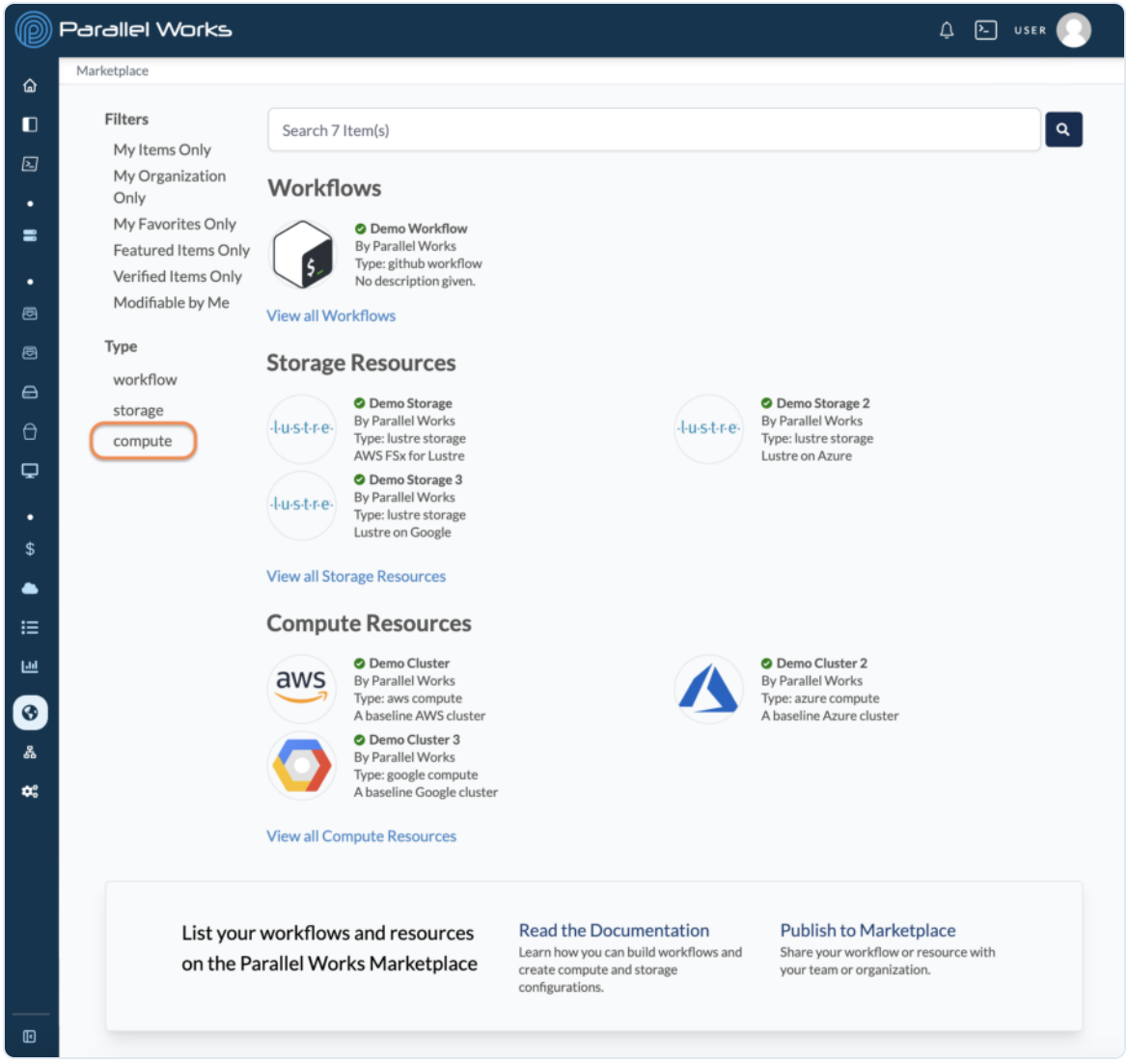
Click the bucket's power button. The bucket will take a few minutes to start. You can proceed to the next step while you wait.

Provision a Cluster

Navigate to the **Marketplace**.

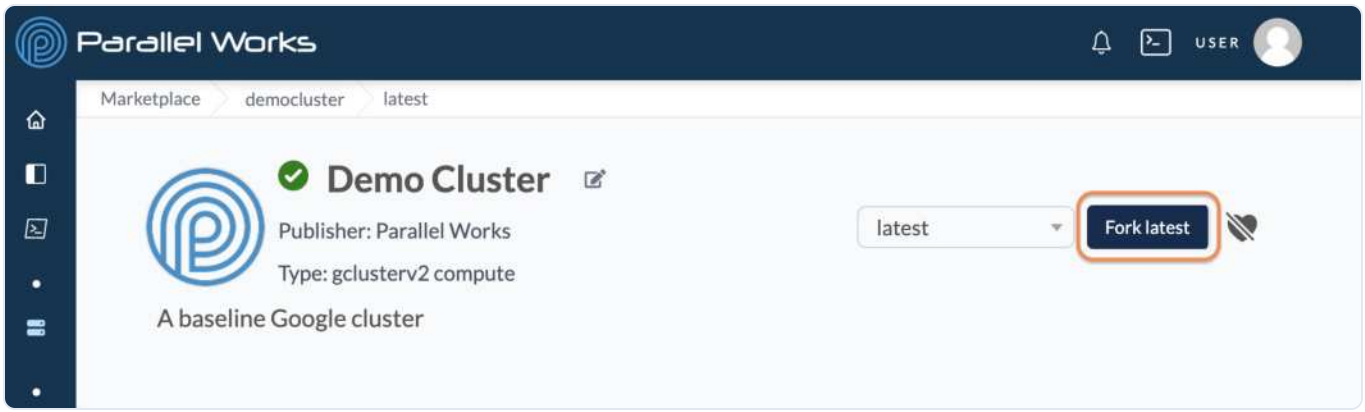


Select **Compute** to see pre-configured clusters.

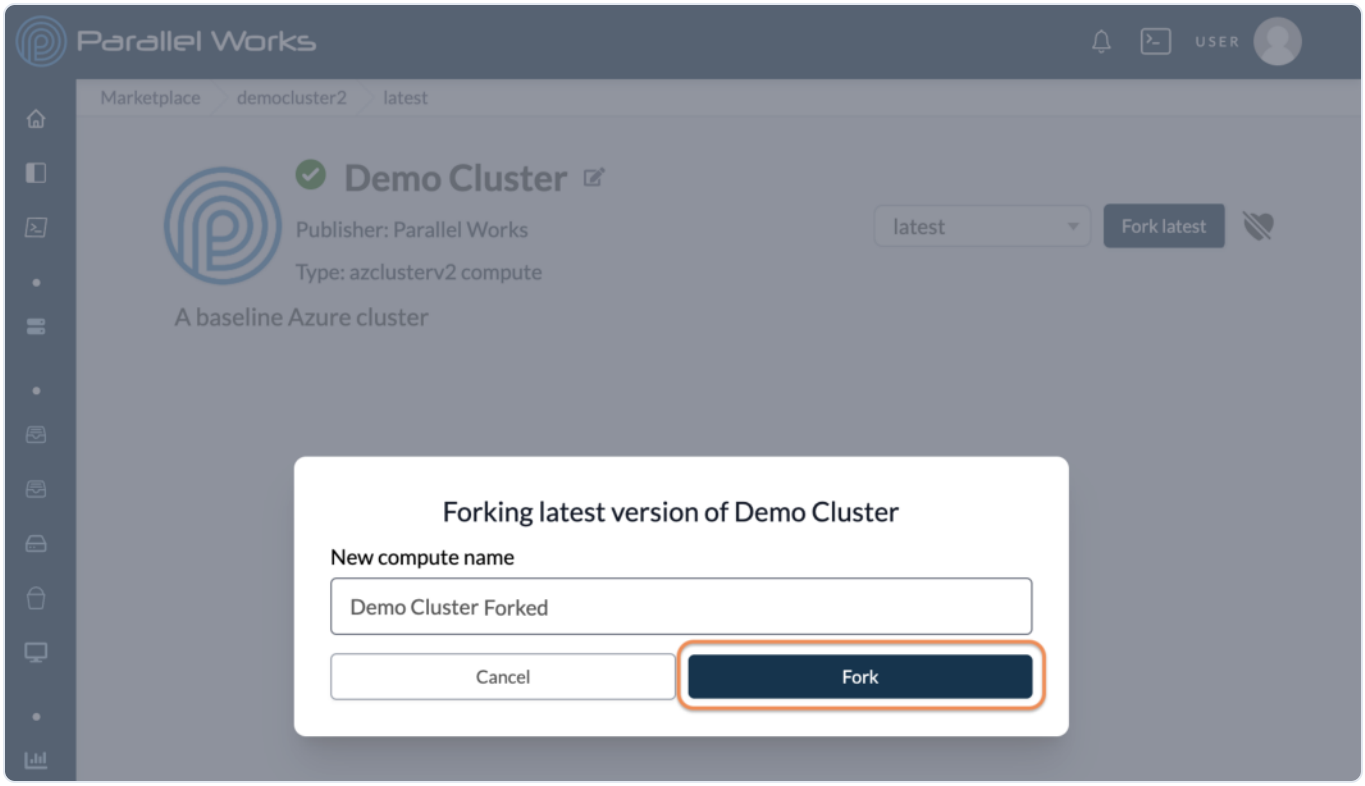


Select the cluster that best suits your needs.

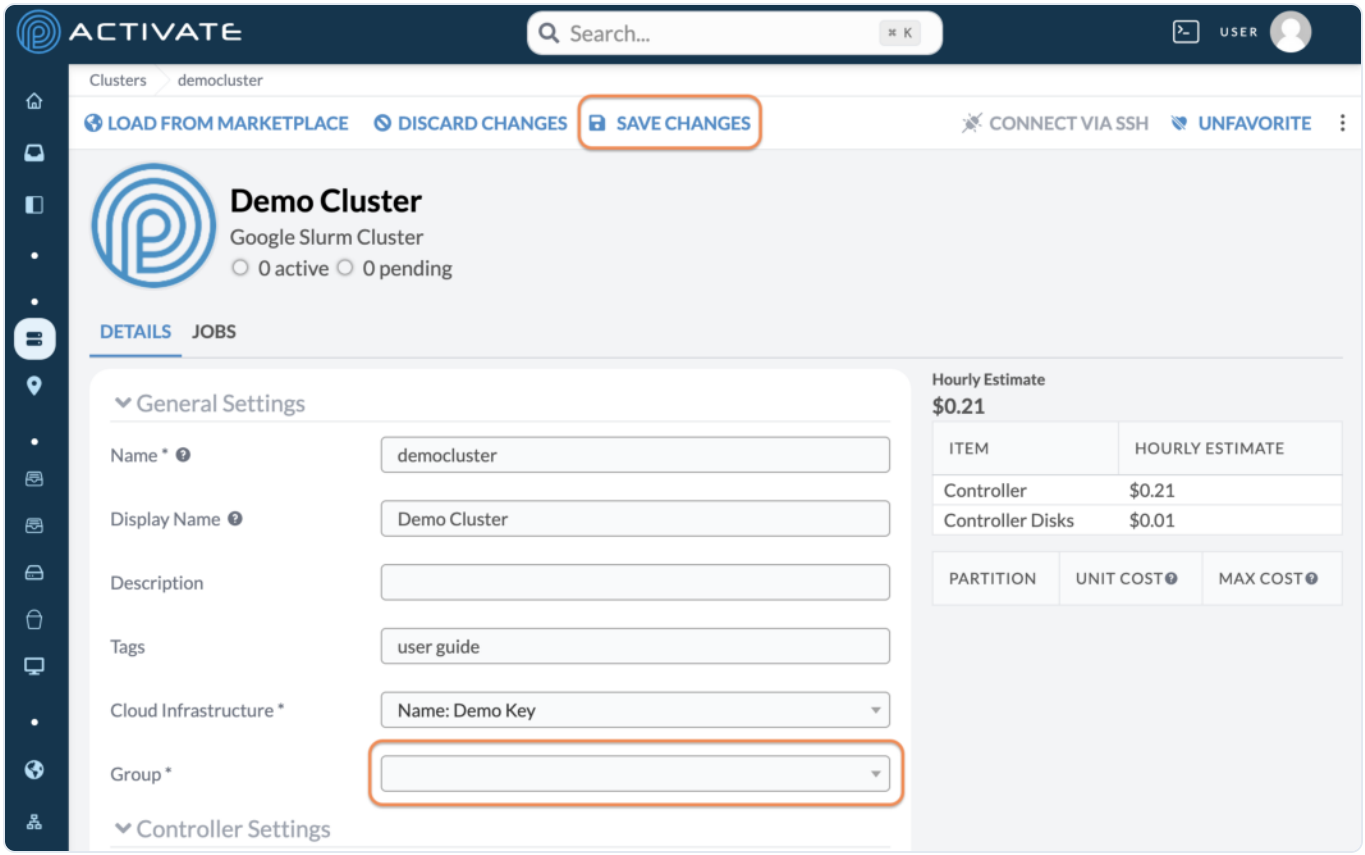
Click **Fork latest**.



Enter a name for your cluster under the **New Compute Name** option and click **Fork**.

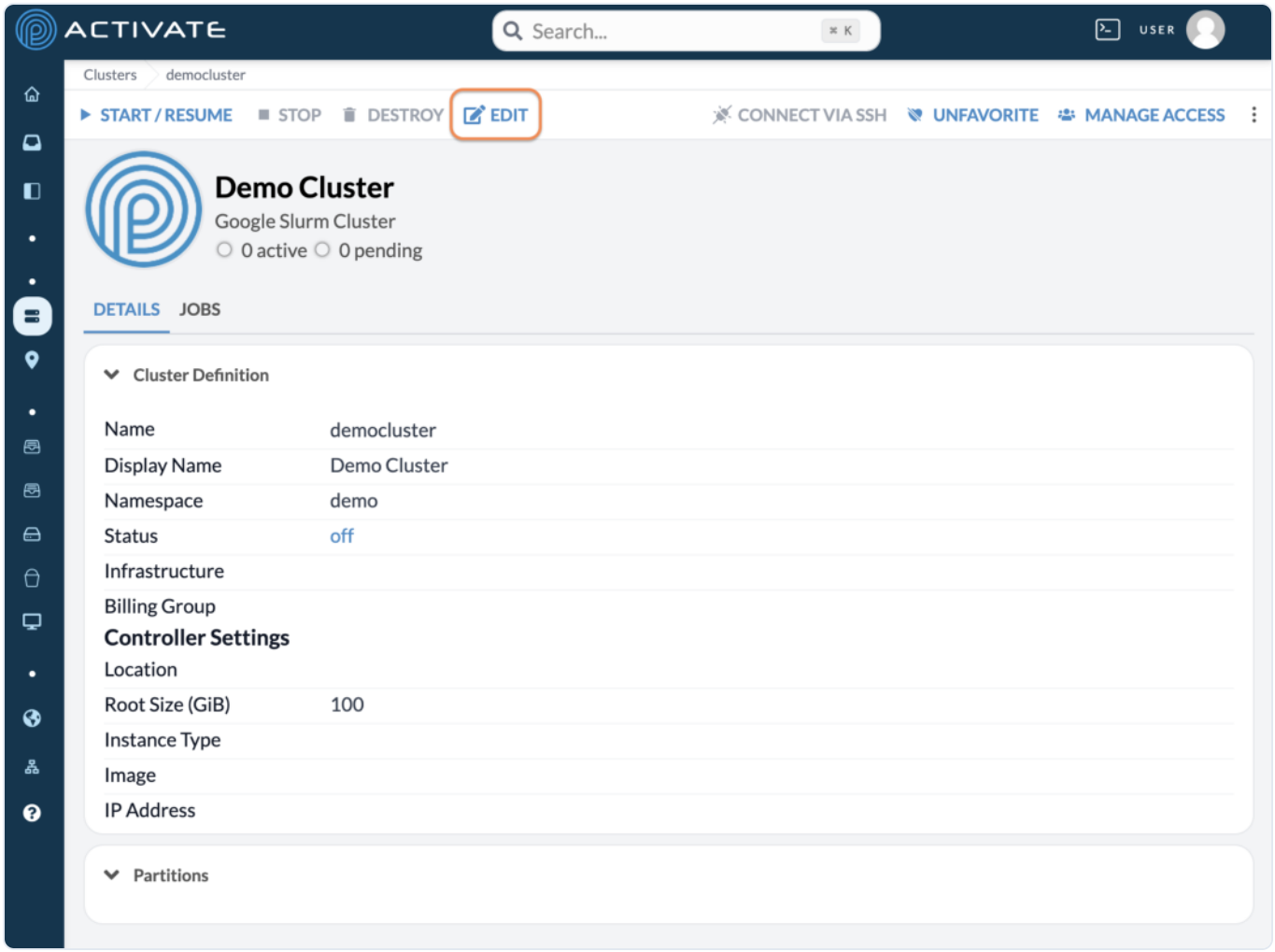


Use the **Group** dropdown menu to select the group that your organization uses to allocate costs. The remaining cluster settings have already been configured. Click **Save Changes**.

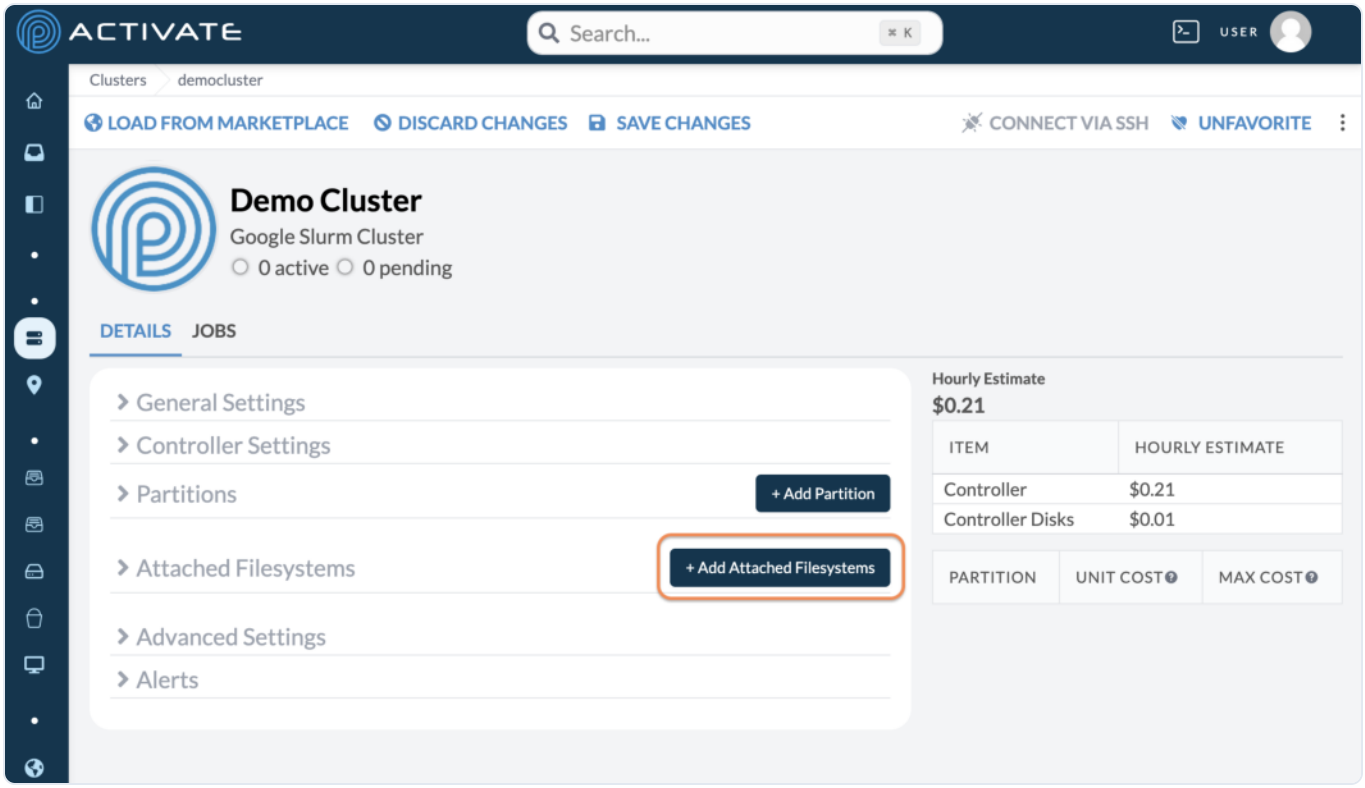


Attach a Bucket

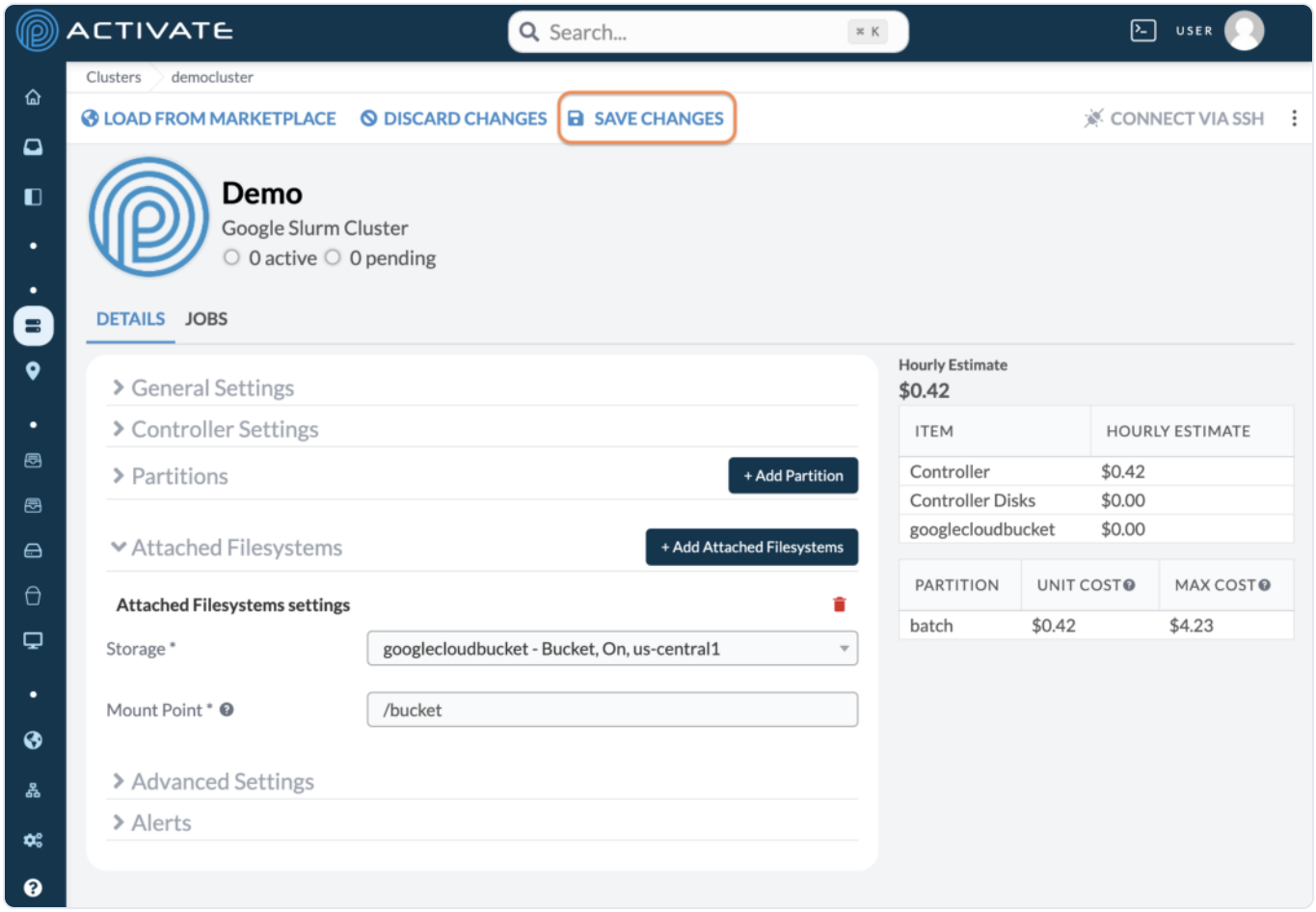
Click **Edit** on your cluster's page.



At the bottom of the page, click **+Add Attached Filesystems**.

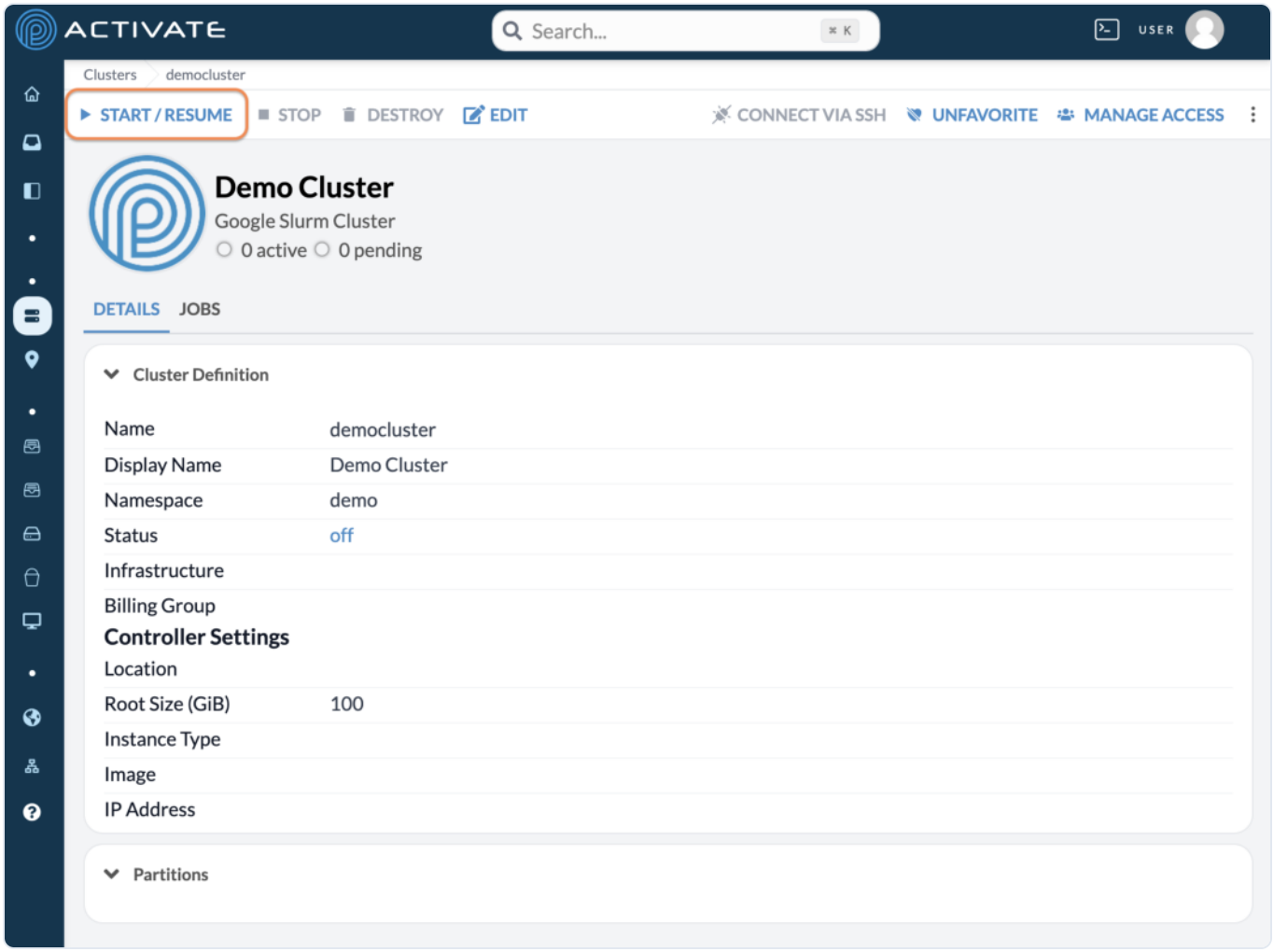


Select the bucket you want to use from the **Storage** dropdown menu. Enter `/bucket` in the **Mount Point** option. Click **Save Changes**.



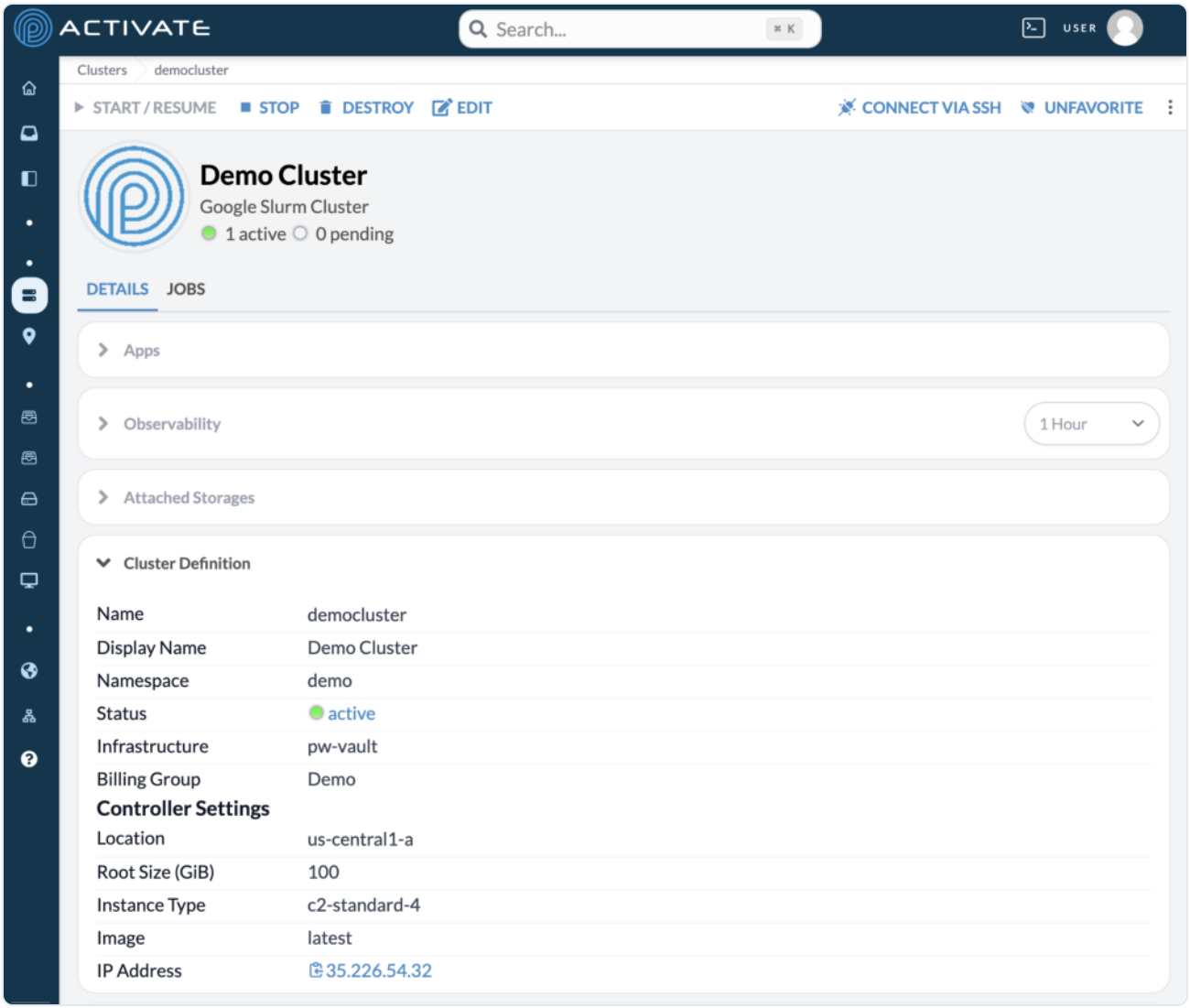
Start the Cluster

Start the cluster by clicking **Start/Resume**.



You'll see the message *Starting [Cluster Name]*. While the cluster starts, the **Status** will show **provisioning** with a yellow bubble.

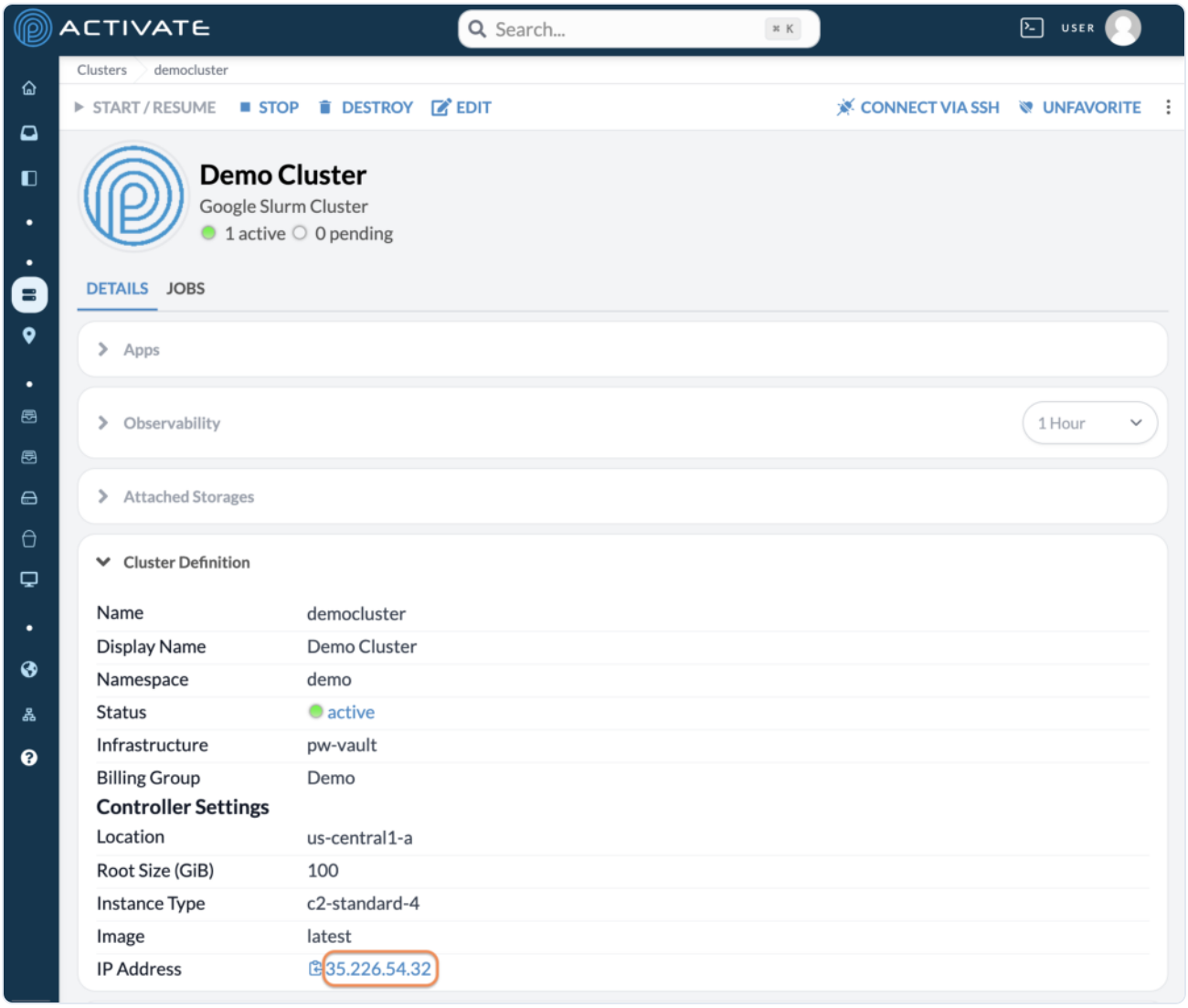
When your cluster is online, both the **Status** will show **active** with a green bubble.



Log In to the Cluster

Clusters use one node, called a controller, to delegate tasks to compute nodes. You'll need to log in to the controller to interact with the cluster.

While on your cluster's page, click its **IP Address**.



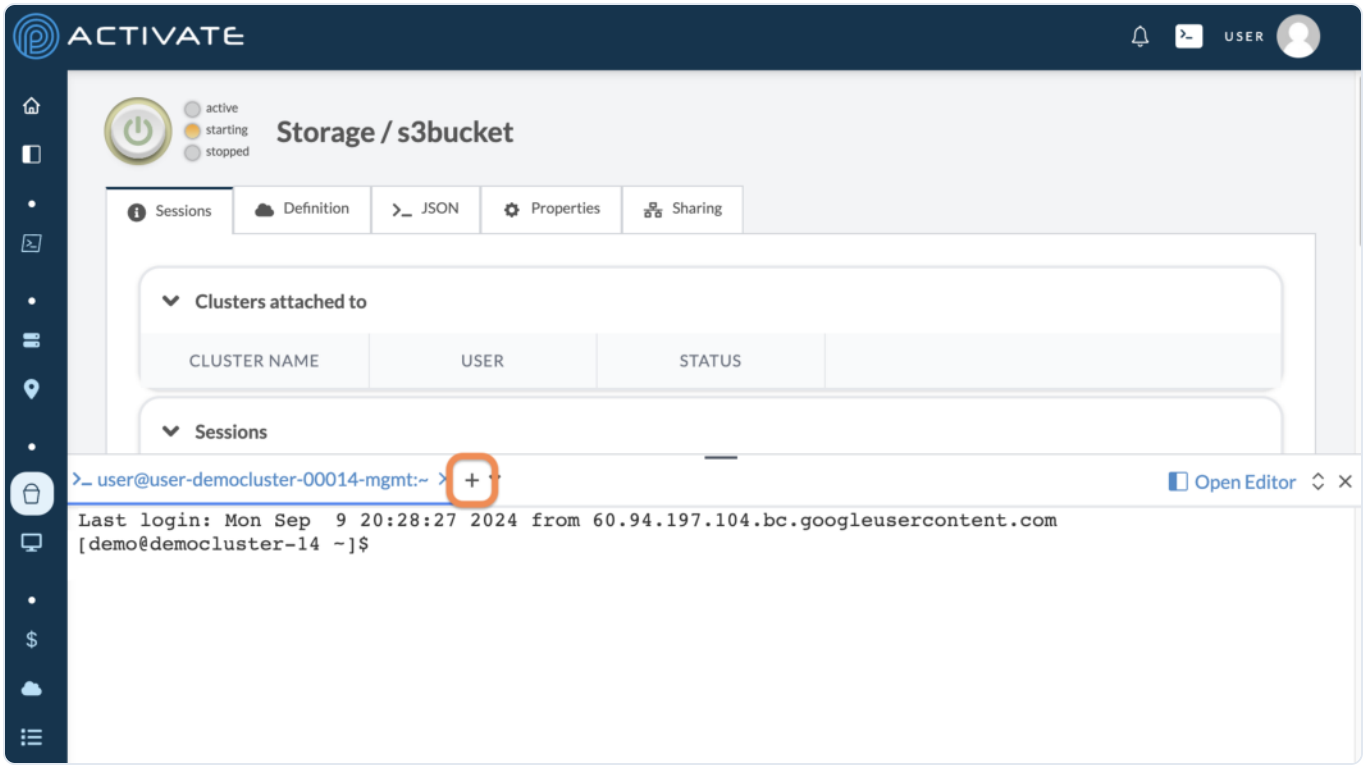
A terminal will pop up on the bottom of the screen, showing the following message:

```
Warning: Permanently added '34.60.48.100' (ED25519) to the list of known hosts.
[username@cluster-name-01 ~]$
```

Create a File

Next, you'll create a file inside your user container, then transfer it to your bucket. The instructions in the following sections follow this MPI tutorial.

In the terminal, click the plus icon to open another tab.



In the new terminal tab, create a file with this code:

```
cat << EOF >> hello-world.c
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {
    // Initialize the MPI environment
    MPI_Init(NULL, NULL);

    // Get the number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    // Get the name of the processor
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    // Print off a hello world message
    printf("Hello world from processor %s, rank %d out of %d processors\n",
           processor_name, world_rank, world_size);

    // Finalize the MPI environment.
    MPI_Finalize();
```

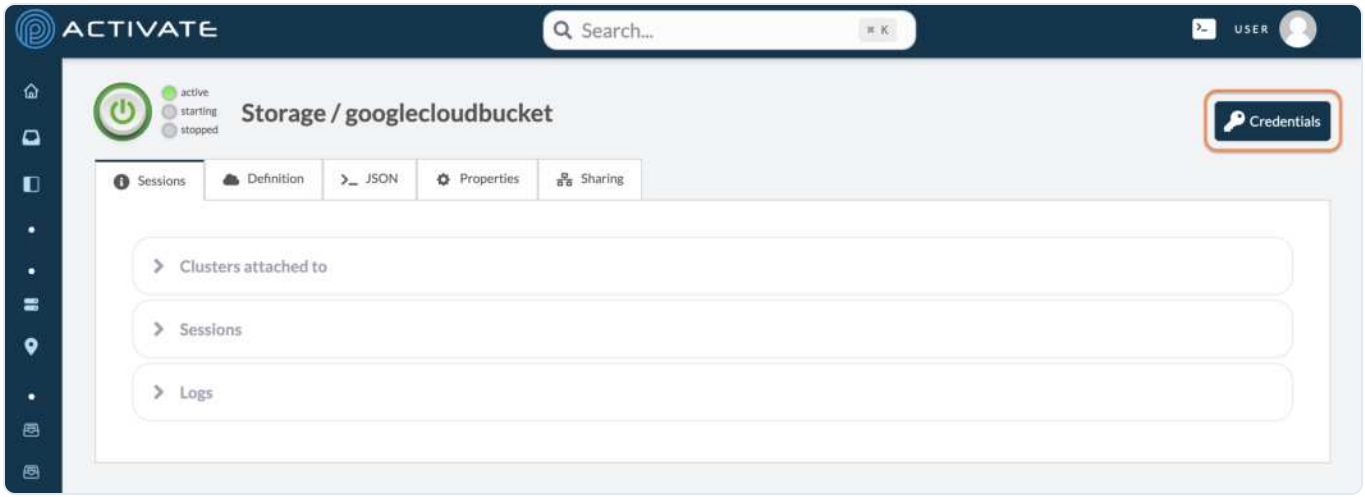
```
}
EOF
```

Press **Enter**.

Retrieve Bucket Credentials

All buckets use short-term credentials, making it easy to transfer data between a bucket and other resources. You'll need those credentials to manage access to the bucket.

Navigate to your bucket's page. Click **Credentials**.



A dialog box will appear with CSP-specific commands for your bucket.

AWS

```
<img src="/content-images/docs/getting-started/aws-bucket-credentials.png" alt="Screenshot of AWS bucket credentials."/>
```

Azure

```
<img src="/content-images/docs/getting-started/azure-bucket-credentials.png" alt="Screenshot of Azure bucket credentials."/>
```

Google

```
<img src="/content-images/docs/getting-started/google-bucket-credentials.png" alt="Screenshot of Google bucket credentials."/>
```

Copy and paste the first command into the same terminal tab where you created the file above. Press **Enter**.

Upload File to Bucket

Next, you'll use the PW CLI to transfer the file you created above.

In the same terminal tab, enter this commmand to copy `hello-world.c` to your bucket:

```
pw buckets cp hello-world.c $BUCKET_URI/hello-world.c
```

Use the list command `pw buckets ls` to check that your file transferred:

```
username@pw-user-username:~/pw$ pw buckets ls pw://username/bucketname
pw://username/bucketname/hello-world.c
username@pw-user-username:~/pw$
```

Authenticate the PW CLI

Before you transfer the file from the bucket to your cluster, you'll need to authenticate the PW CLI on the cluster.

First, create a short-term token on ACTIVATE. You can follow our instructions here. Copy the token.

Log in to the cluster again (this step above).

Enter `pw auth token` and paste your token. Press Enter. You'll see the following message:

```
2025-03-31T16:23:19Z [INFO] Authenticated as username
[username@usernamecluster-3 ~]$
```

Download File to Cluster

Copy and paste your bucket's credentials again (this step above).

Copy and paste the exact command below. The value for `BUCKET_URI` will be auto-populated with your bucket's URI.

```
pw buckets cp $BUCKET_URI/hello-world.c hello-world.c
```

Compile and Run MPI Test

Enter this command:

```
```

Enter this command:

```
mpicc -o hello hello-world.c
```

Next, you'll create a test job using a script. Enter the following:

```
cat << EOF >> hello-world.sbatch
#!/bin/bash

#SBATCH -N 2
#SBATCH --ntasks-per-node=2
```

```
mpirun -np 2 -N 1 $HOME/hello
EOF
```

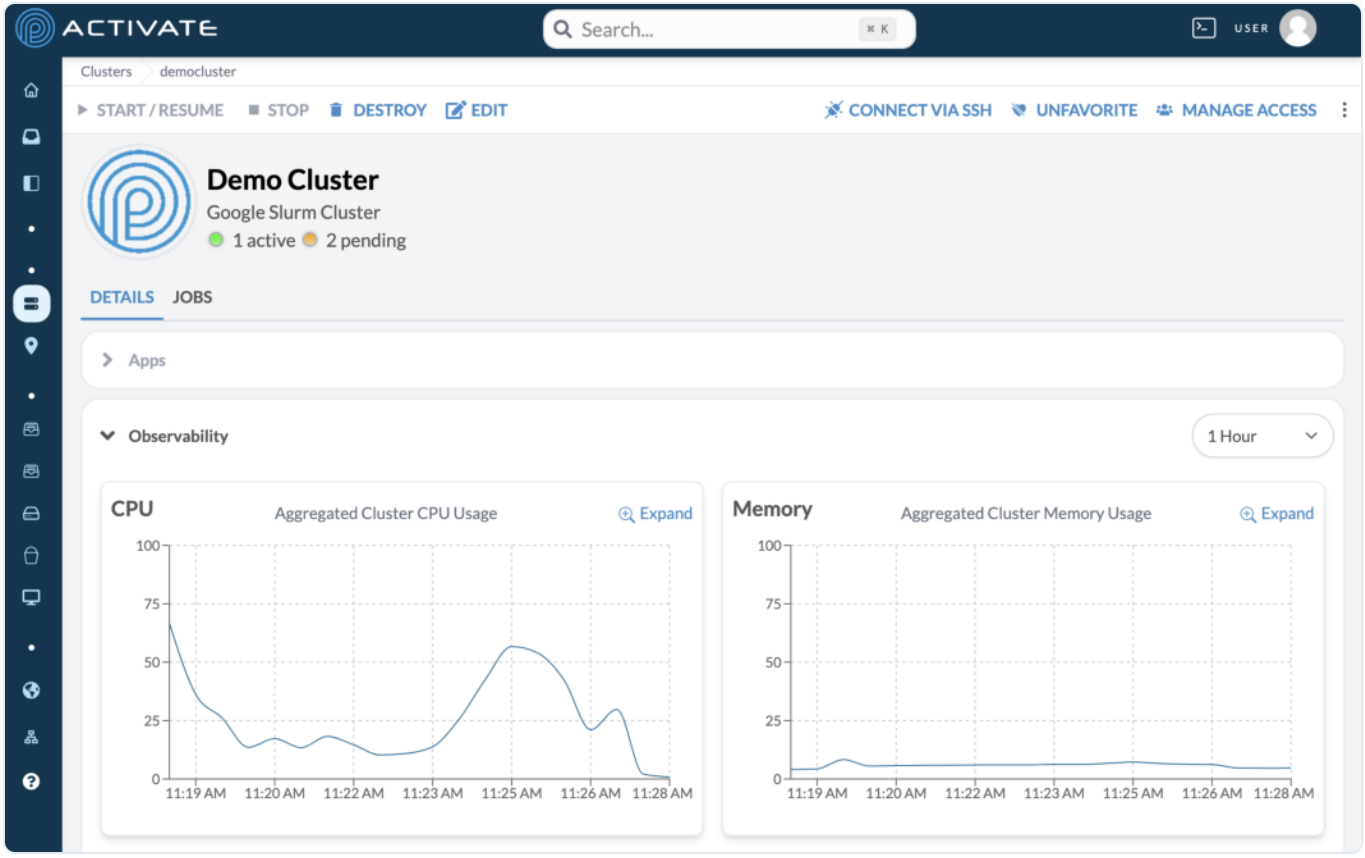
Press Enter.

Run the job with this command:

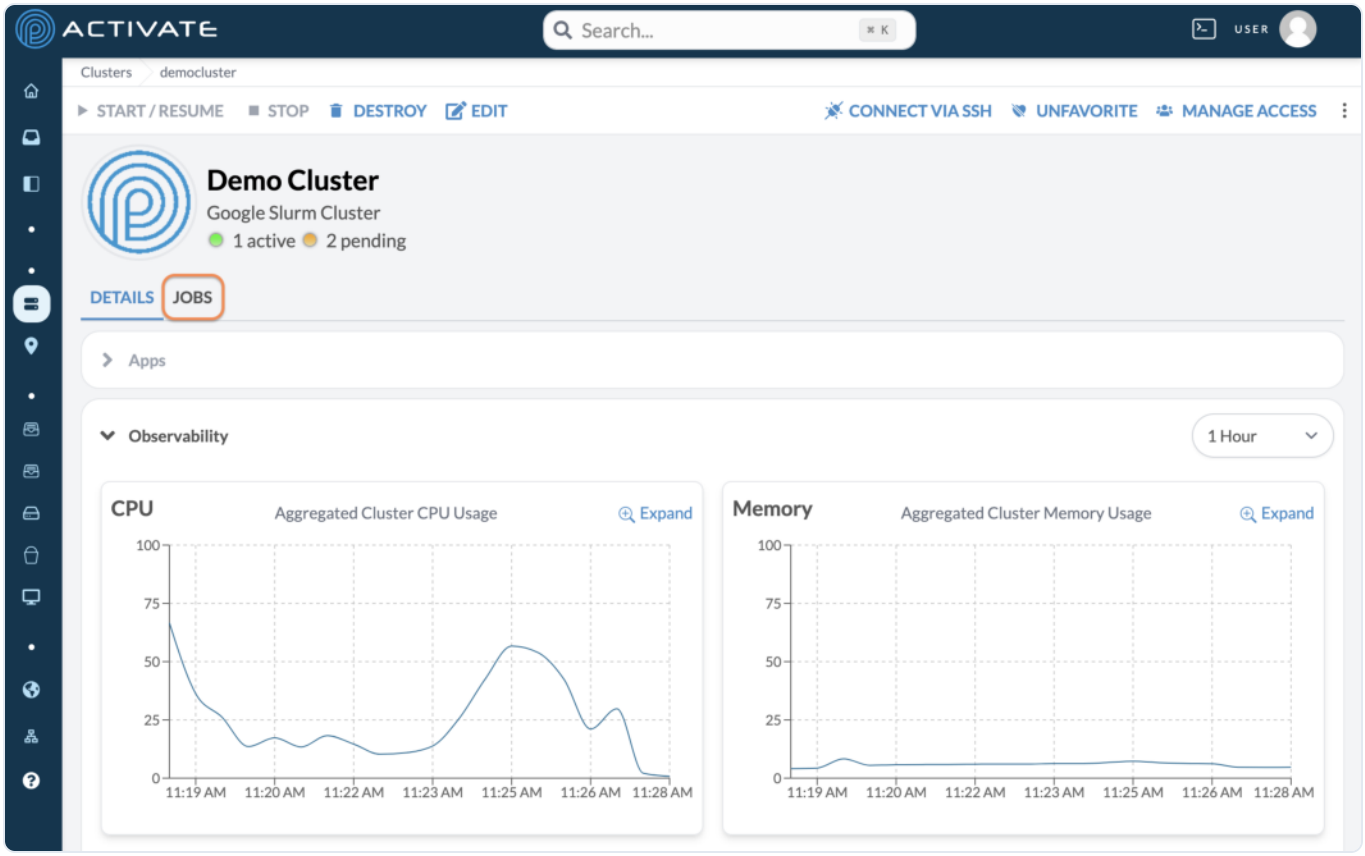
```
sbatch hello-world.sbatch
```

Monitor Job

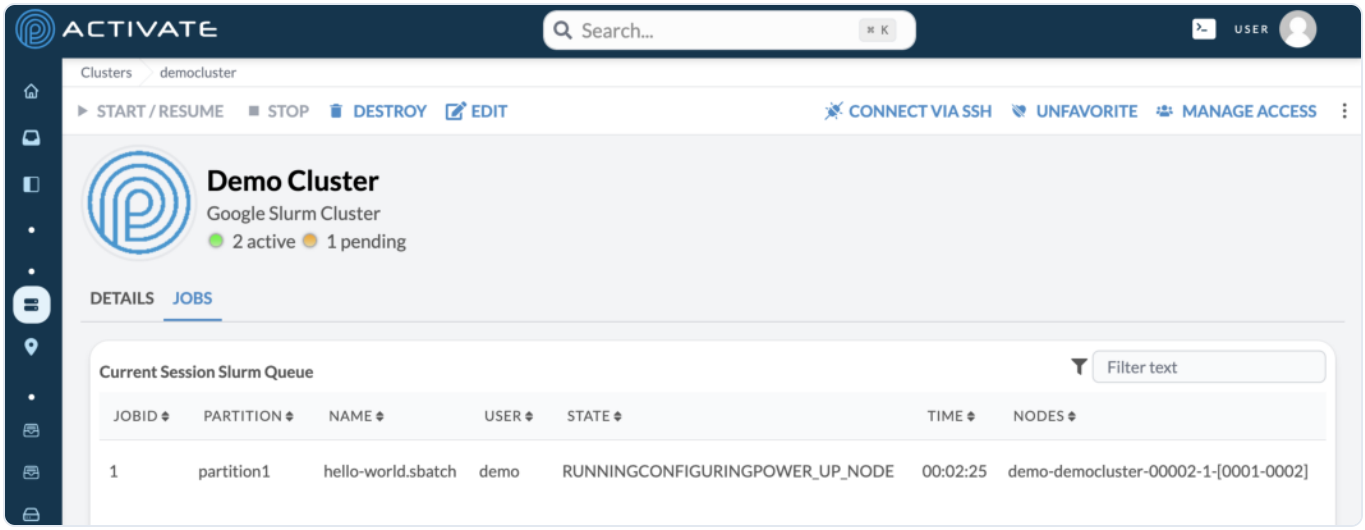
Go to your cluster's page. In the **Observability** module, you'll see your cluster's CPU activity change as it runs the job you just submitted.



Click **Jobs**.



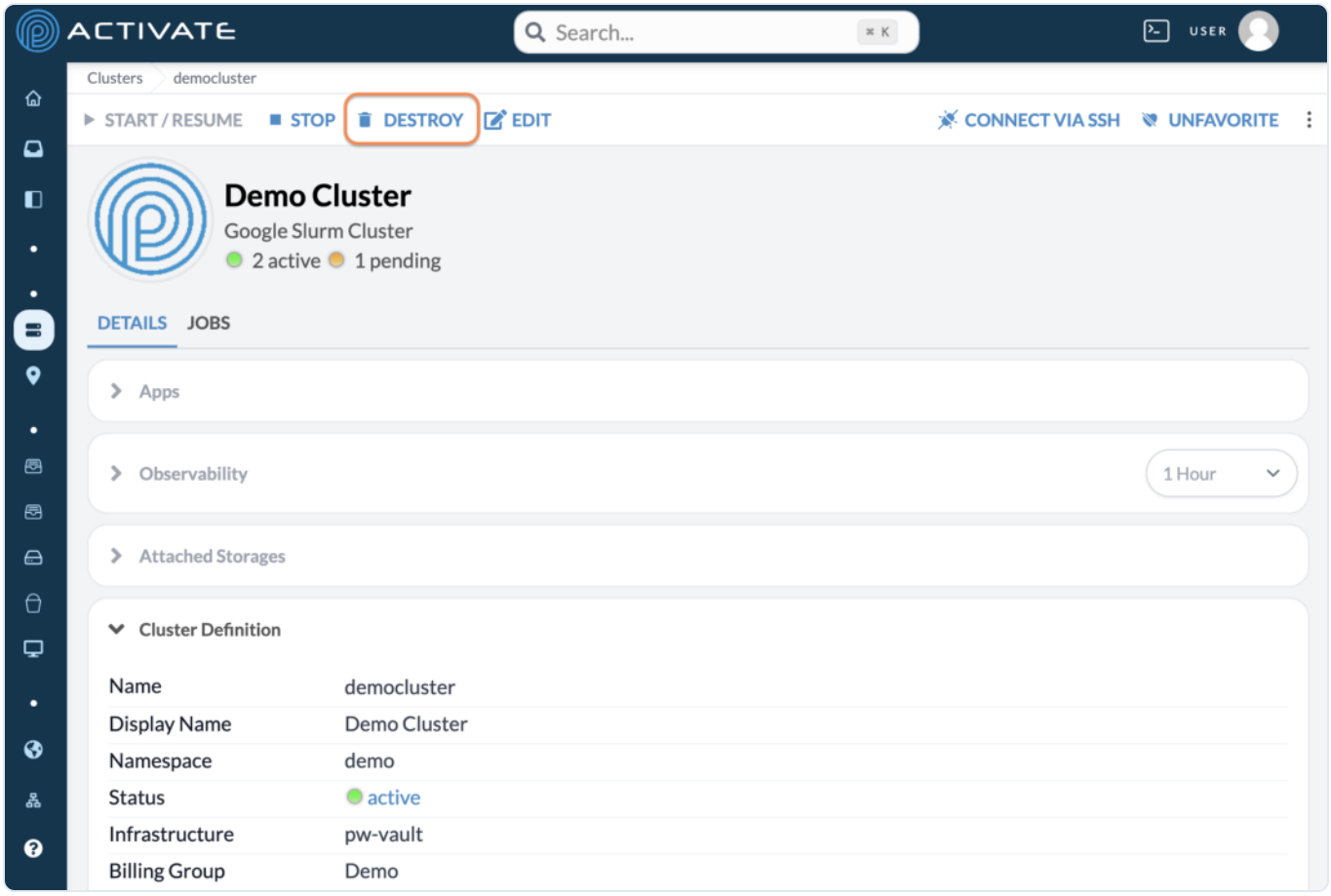
You'll see the state of your partition.



Shut Down Resources

Navigate to the configuration page for your bucket. Click the power button. A module will appear. Click **Deprovision**.

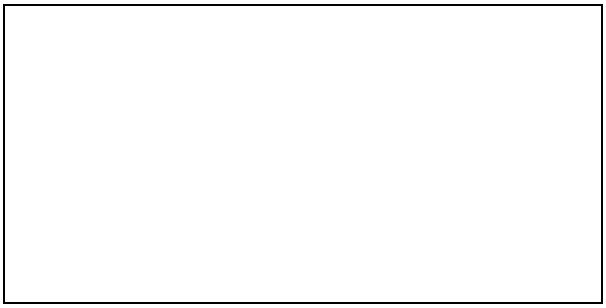
Next, navigate to the configuration page for your cluster. Click **Destroy**.



A module will appear. Click **Destroy**.

Navigating ACTIVATE

This page introduces the major features of ACTIVATE to help new users get oriented.



Your Workspace

Your personal workspace includes tools for managing files and interacting with ACTIVATE. For detailed documentation on each component, see:

- **Dashboard** - Your central hub for monitoring workflows, sessions, and clusters
- **Editor** - Full-featured IDE for editing files and code
- **Explorer** - File browser for managing files
- **Terminal** - Command-line access to your workspace
- **Inbox** - Notifications and messages

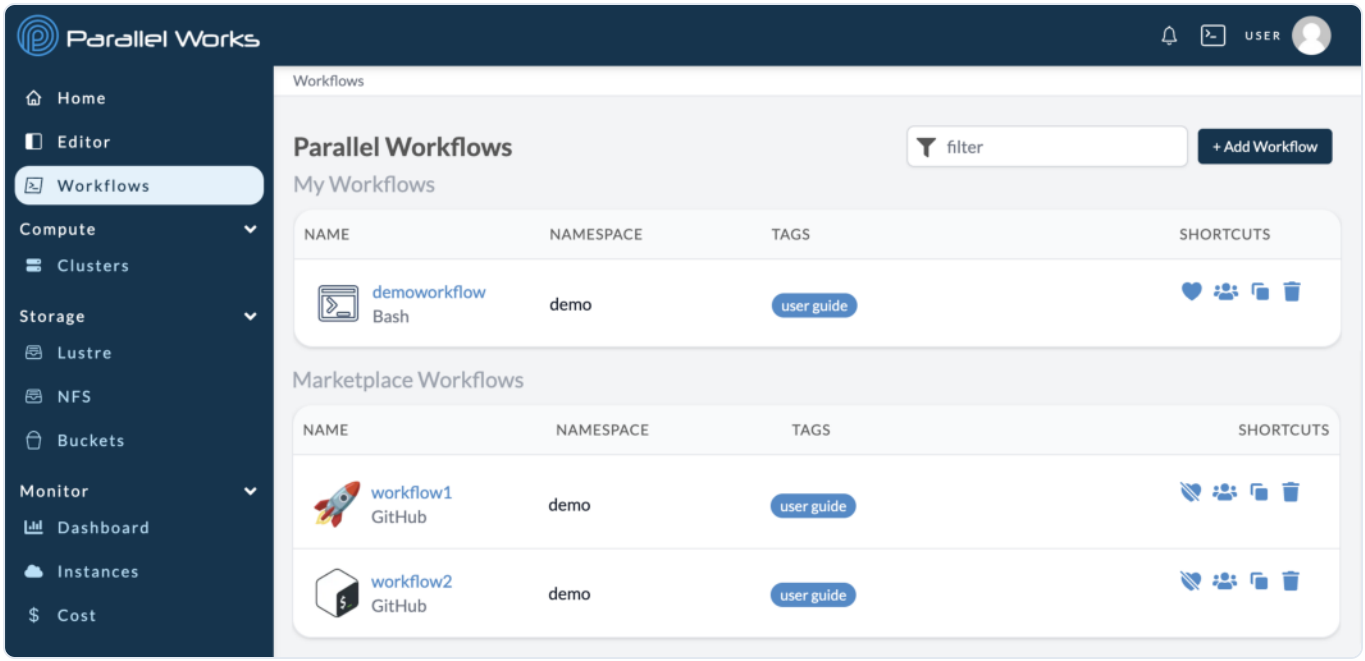
If you want to customize your workspace layout, see **Customizing Layouts**.

Run

The **Run** tab houses the pages for different options to complete work on ACTIVATE.

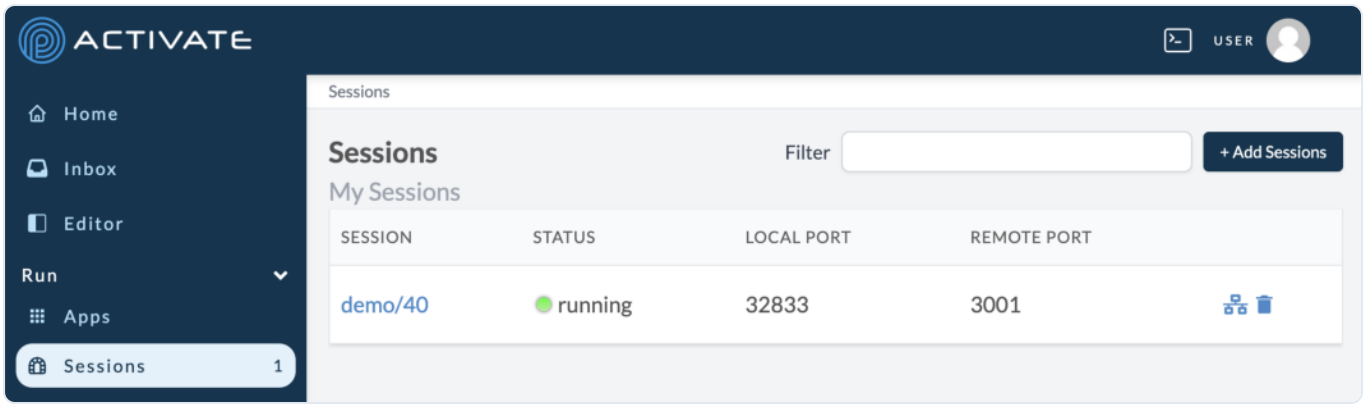
Workflows

On the **Workflows** page, you can create and manage your workflows. Any workflows you add from the **Marketplace** will appear here automatically.



For more information, please see **About Workflows**.

Sessions



On the **Sessions** page, you can access running sessions. A session is a secure, shareable connection between a cluster and a workflow. Sessions can also generate interactive applications.

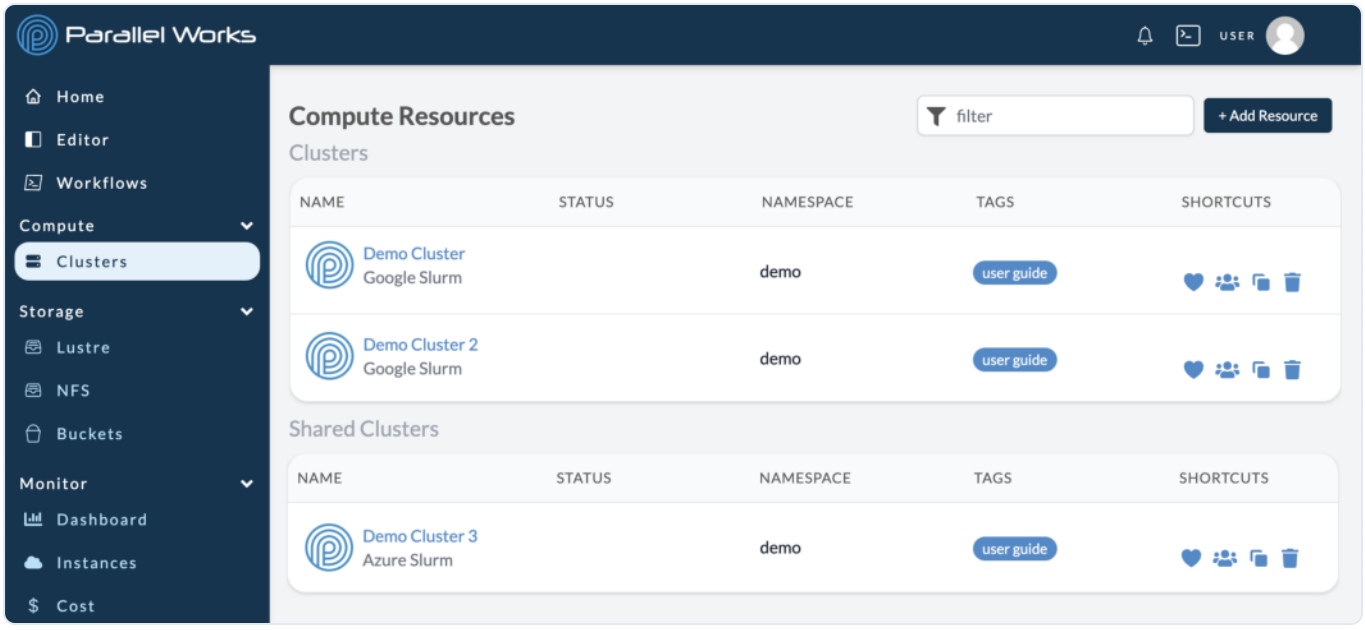
For more information, please see **About Sessions**.

Compute

The **Compute** tab houses the pages for different computing resources.

Clusters

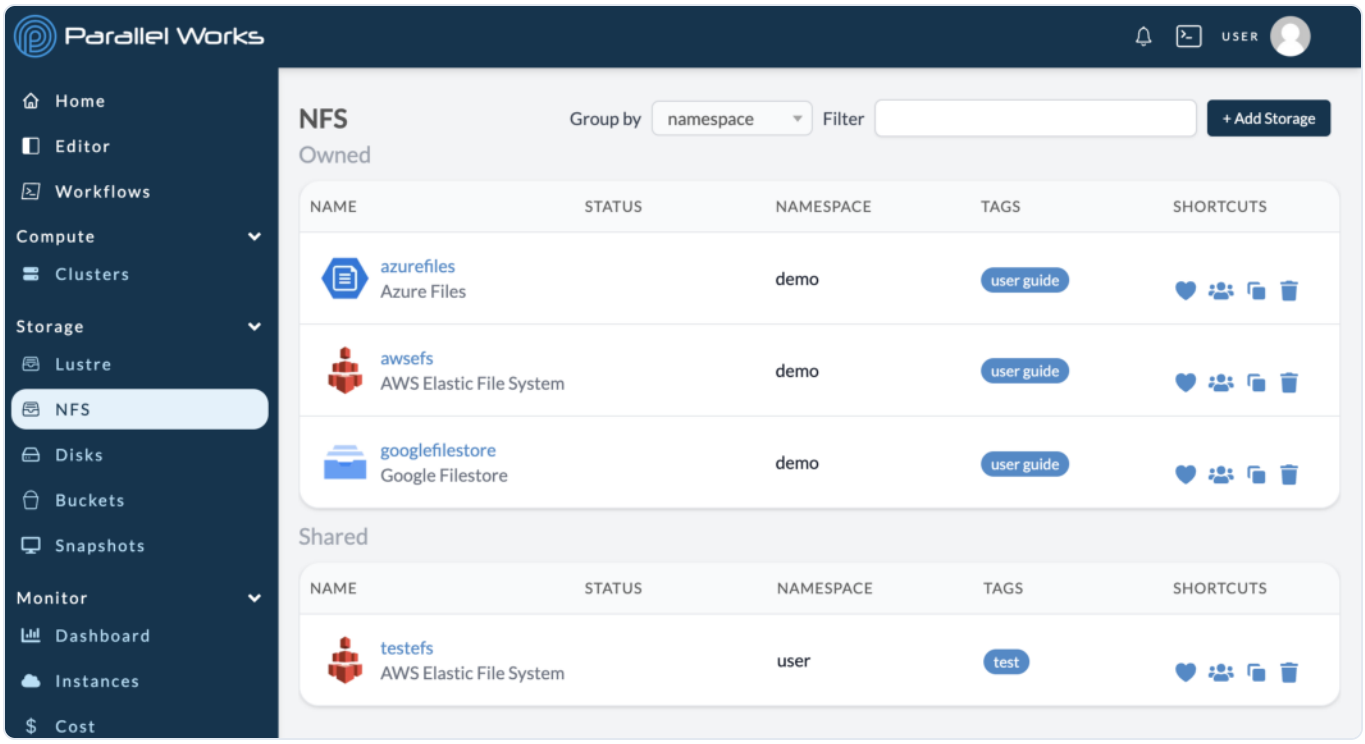
On the **Clusters** page, you can create and manage clusters as well as access any shared clusters.



For more information on adding and configuring new resources, please see **Creating Clusters** and **Configuring Clusters**.

Storage

The **Storage** tab houses the pages for different storage types: **Lustre**, **NFS**, **Disks**, **Buckets**, and **Snapshots**. For more information about these options, please see **Storage**.

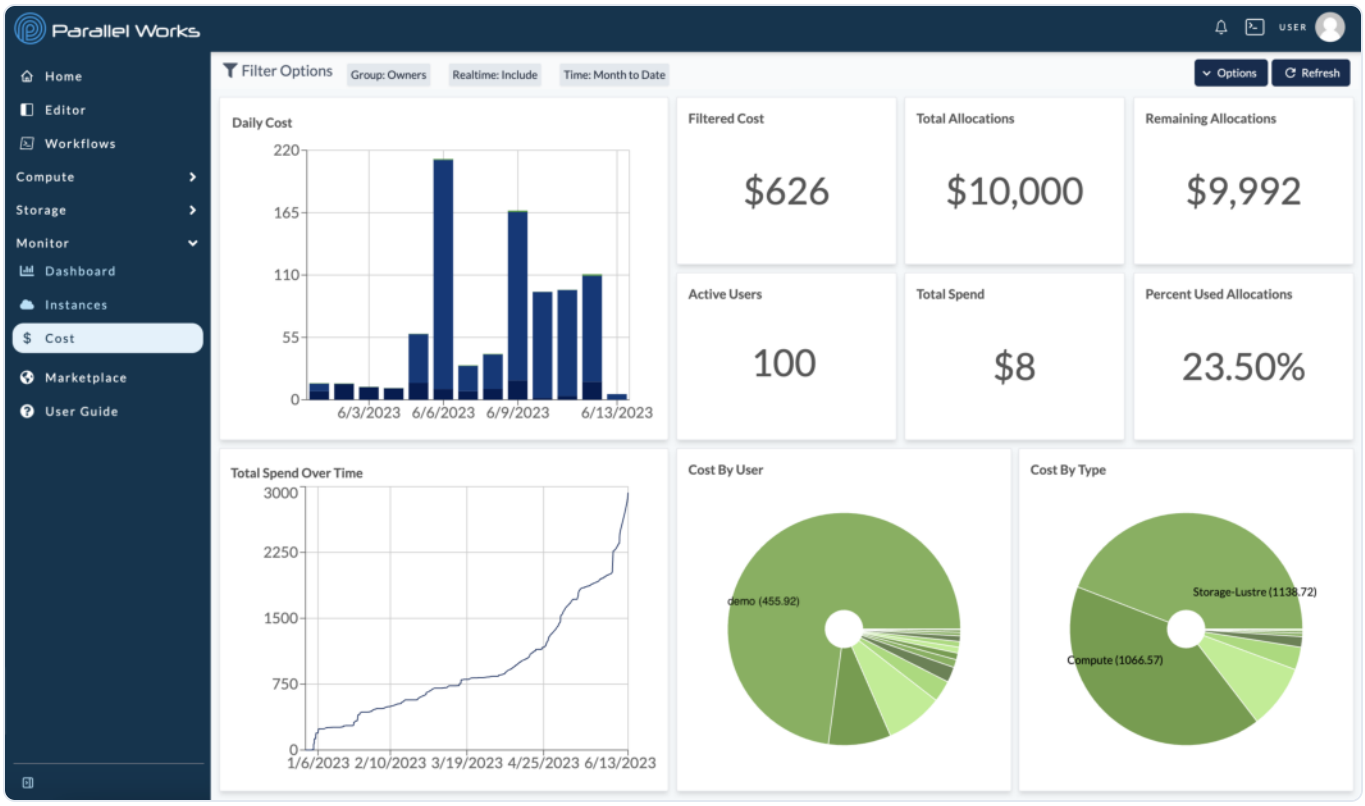


Monitor

The **Monitor** tab houses pages that show important information about your resources and ACTIVATE platform usage.

Cost

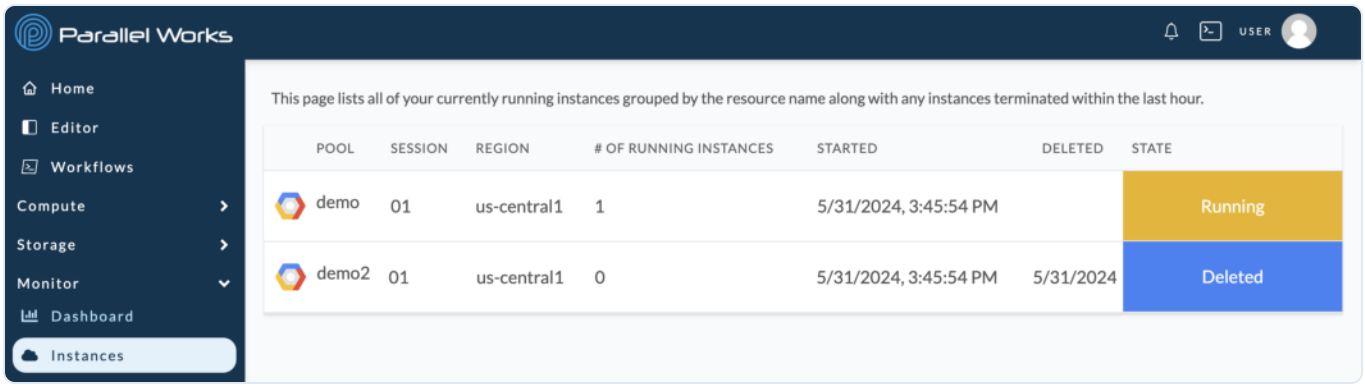
The **Cost** page displays cost tracking information for your organization, such as total spend over time and allocation usage.



For more information, please see **Monitoring Costs**.

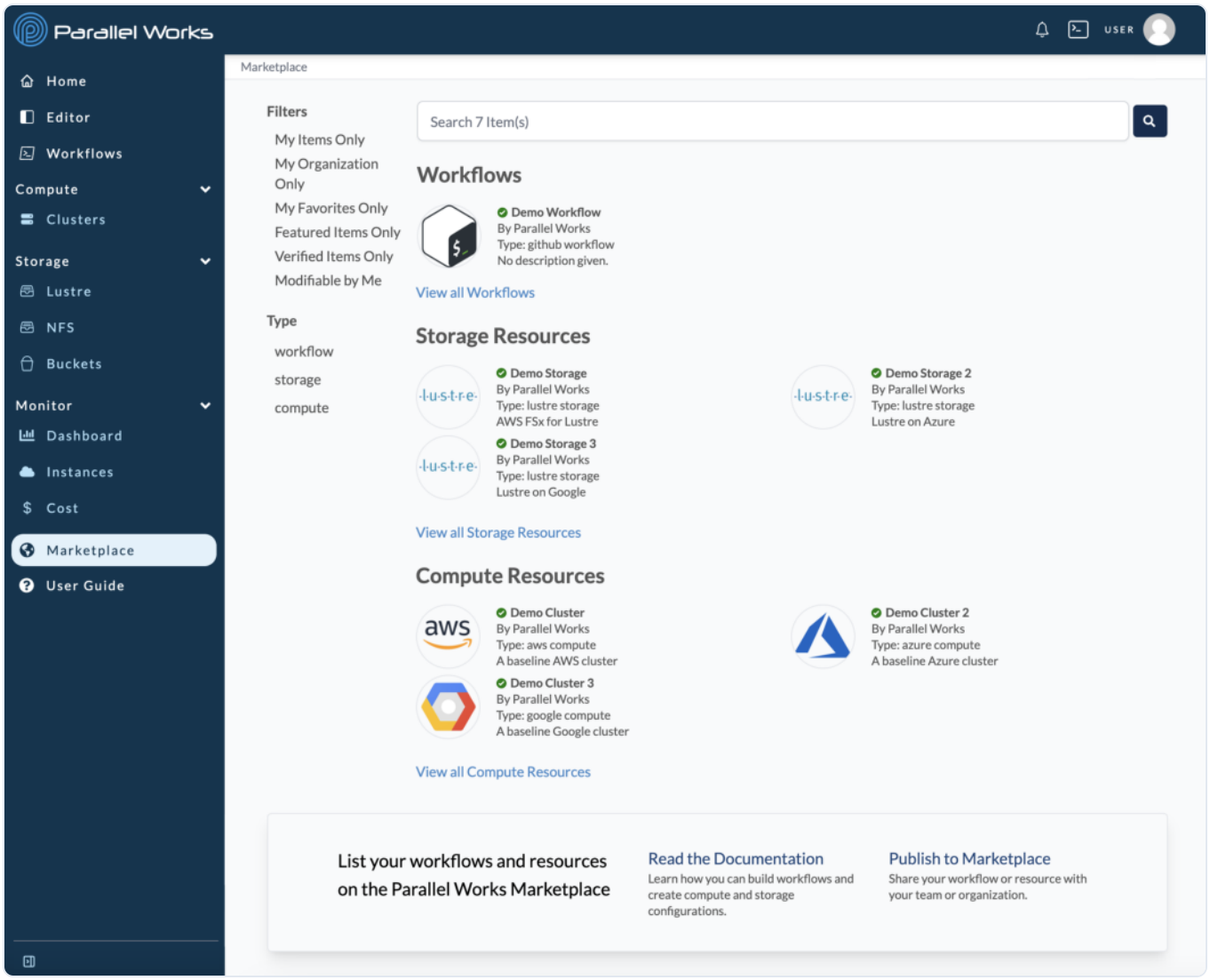
Instances

The **Instances** page displays any running clusters and clusters that have been deleted within the last hour. For more information, please see **Instances** in **Monitoring Your Work**.



Marketplace

On the **Marketplace** page, you can select pre-configured workflows, storage resources, and compute resources for your projects.

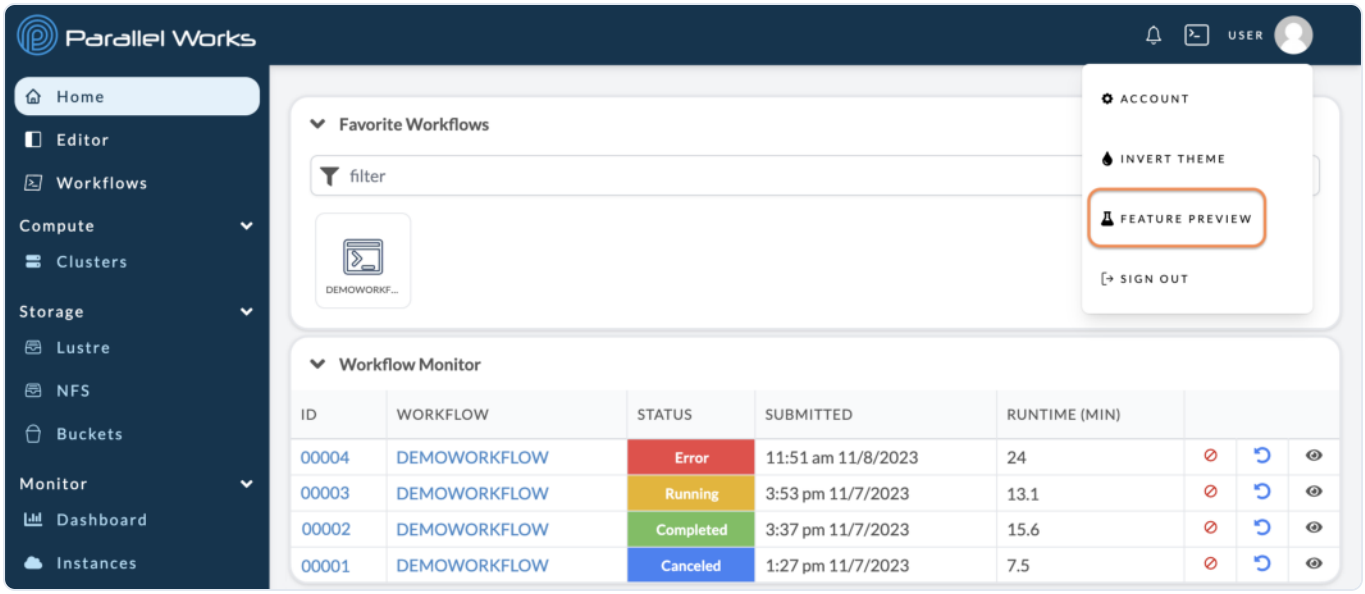


For more information, please see **Marketplace**.

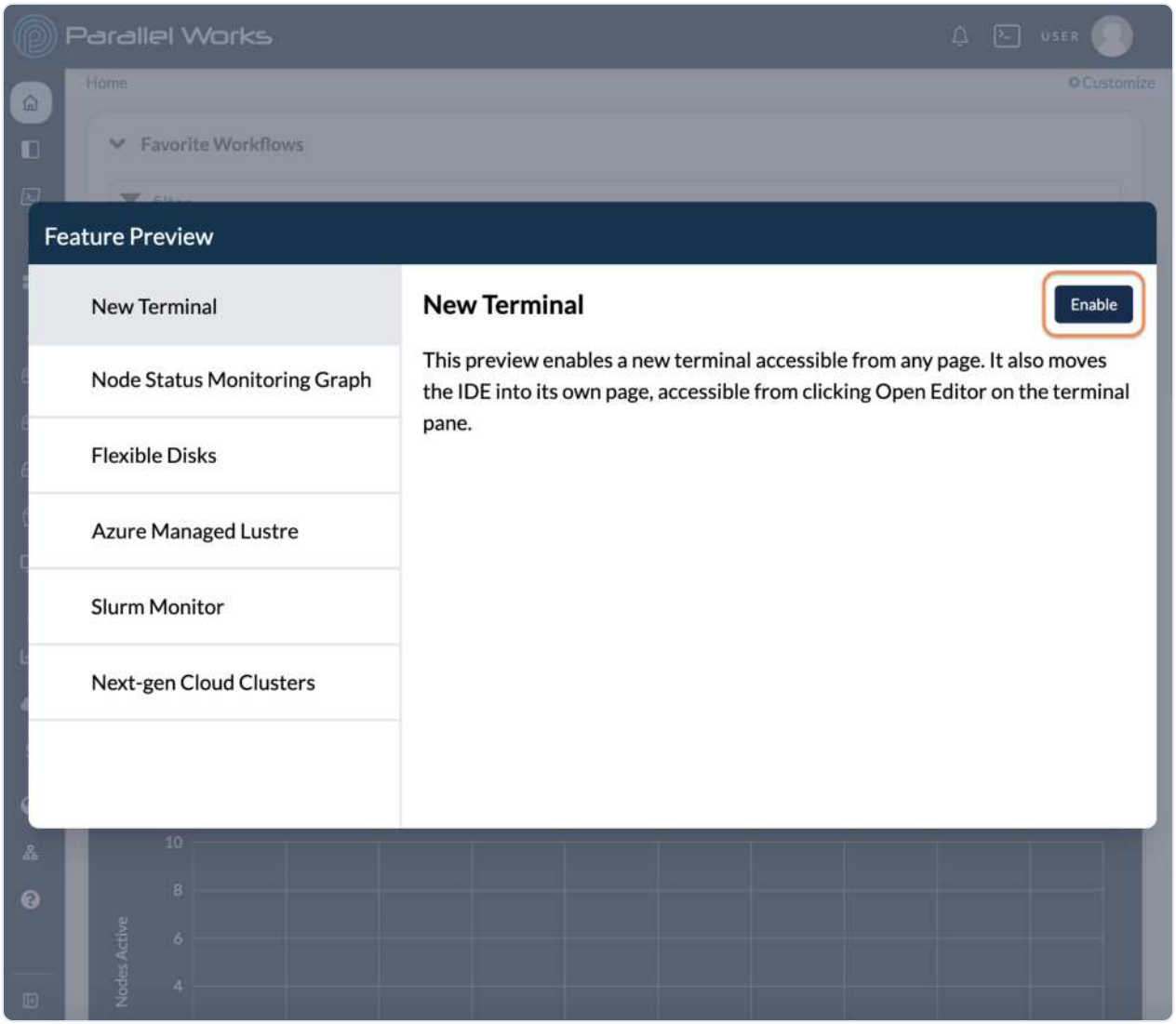
Feature Preview

With the Feature Preview, you can test upcoming features before they're integrated into ACTIVATE.

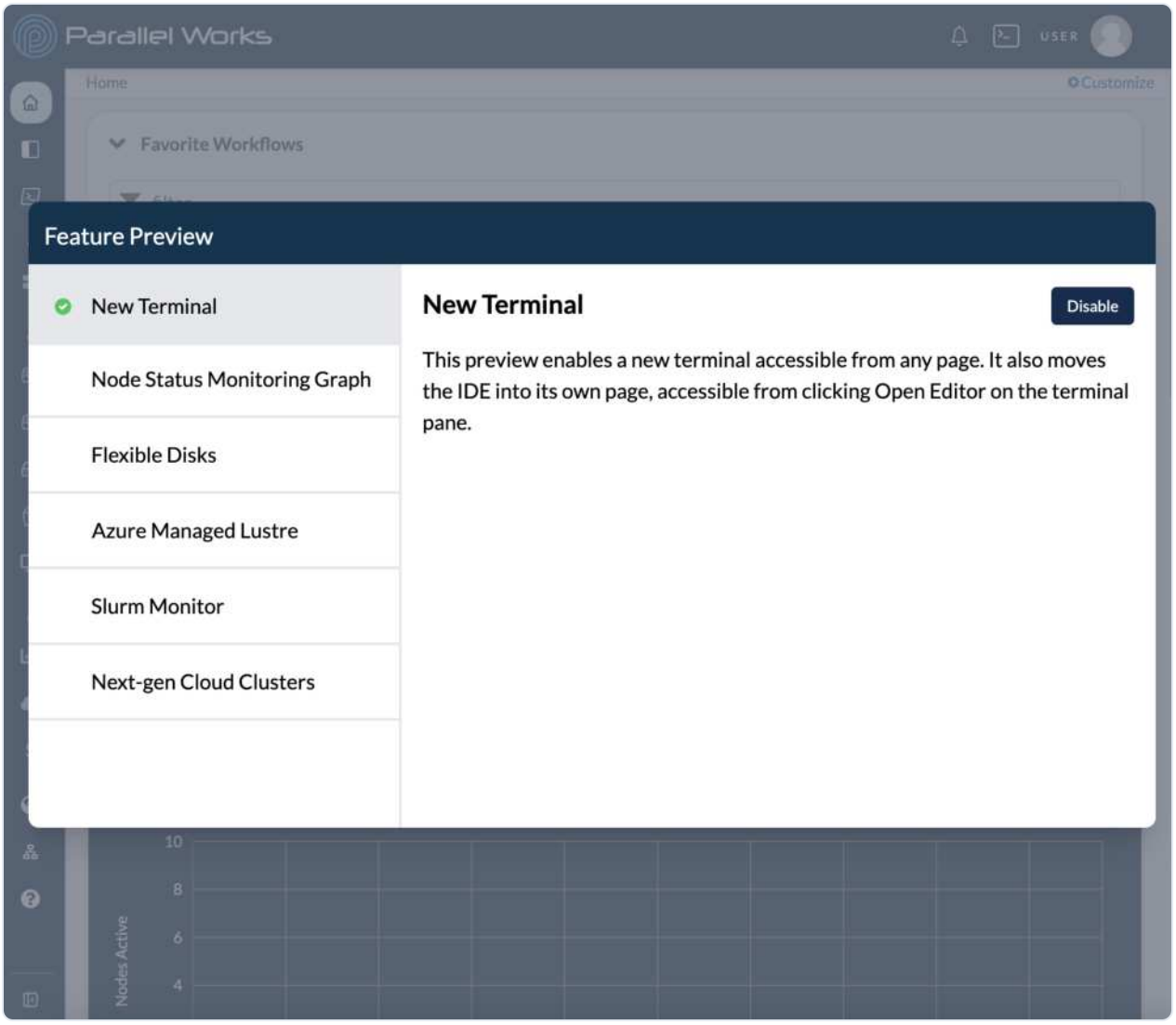
Click your username, then select **Feature Preview**.



A new dialog box will appear with a list of features. Select a feature, then click **Enable**.



Enabled features will display a green checkmark next to their names.



You can remove features at any time by clicking **Disable**.

Please note that the items listed in the Feature Preview are controlled by your organization's administrator.

Your Workspace

Your workspace is your personal environment in ACTIVATE where you can manage files, write code, and interact with the platform. This section covers the core tools available in your workspace.

Workspace Components

Dashboard

The Dashboard is your central hub in ACTIVATE. It displays customizable widgets showing your workflows, sessions, clusters, and other resources. You can personalize the layout to focus on what matters most to you.

Editor

The Editor provides a full-featured IDE (Integrated Development Environment) based on Theia. Use it to edit files, write code, debug workflows, and access an integrated terminal.

Explorer

The Explorer is a file browser for navigating and managing files in your workspace and connected storage. Create, rename, move, upload, and download files with ease.

Terminal

The Terminal gives you command-line access to your workspace. Use it to run commands, manage files, and interact with the PW CLI while navigating ACTIVATE.

Inbox

The Inbox shows notifications from your clusters, workflows, and organization administrators. Stay informed about important events and messages.

Getting Started

If you're new to ACTIVATE, start by exploring the Dashboard to see an overview of your resources. From there, you can:

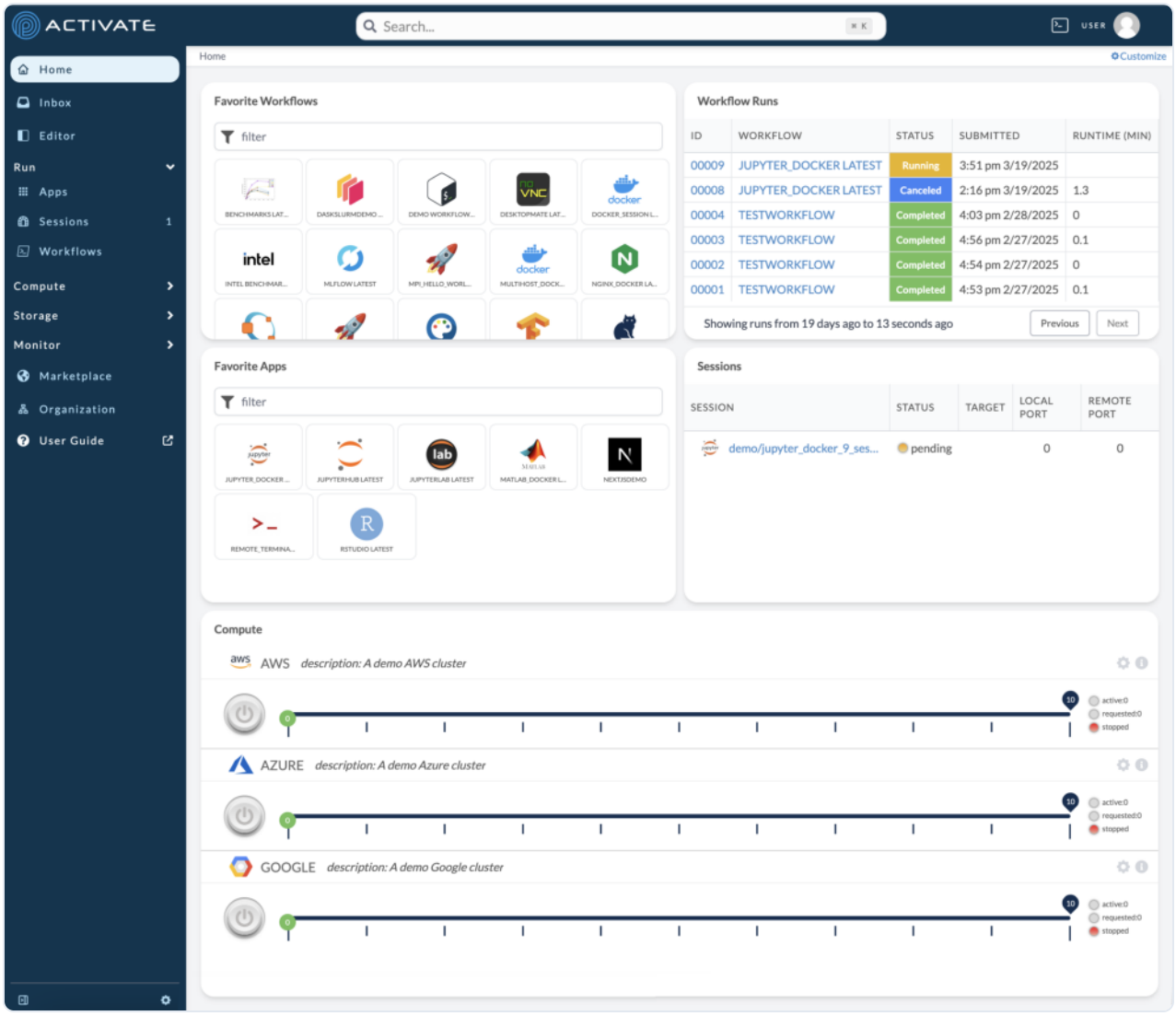
- Use the Editor to work with files and code
- Use the Terminal to run commands
- Check the Inbox for notifications

Related Documentation

- Account Settings - Manage your account preferences
- Getting Started - Tutorial for new users

Dashboard

When you log in to ACTIVATE, you'll see the **Dashboard** by default. The Dashboard serves as your central hub for monitoring and managing your work.



Overview

The Dashboard displays customizable widgets that give you quick access to your most important resources and activities. You can:

- View and run your favorite workflows
- Monitor workflow runs and their status
- Access running sessions
- Start and stop clusters
- Track recent activity

Default Widgets

The Dashboard includes five default widgets:

Favorite Workflows

Click a workflow to run or configure it. Add workflows to your favorites from the Workflows page for quick access.

Workflow Runs

Track current and past workflow runs. From here you can:

- View run status and progress
- Cancel running workflows
- Re-run completed workflows
- Access run logs and output

Sessions

Monitor active workflow sessions. Click a session to connect to it.

Compute

View and manage your favorited clusters:

- Start or stop clusters
- Configure cluster settings
- Click a resource name to collapse or expand it

Customizing Your Dashboard

You can personalize the Dashboard to show the information most relevant to you:

- **Add widgets**: Click the widget selector to add new widgets
- **Remove widgets**: Click the close button on widgets you don't need
- **Rearrange widgets**: Drag and drop widgets to reorder them
- **Resize widgets**: Drag widget edges to resize

For detailed customization instructions, see Customizing Layouts.

Quick Actions

From the Dashboard, you can quickly:

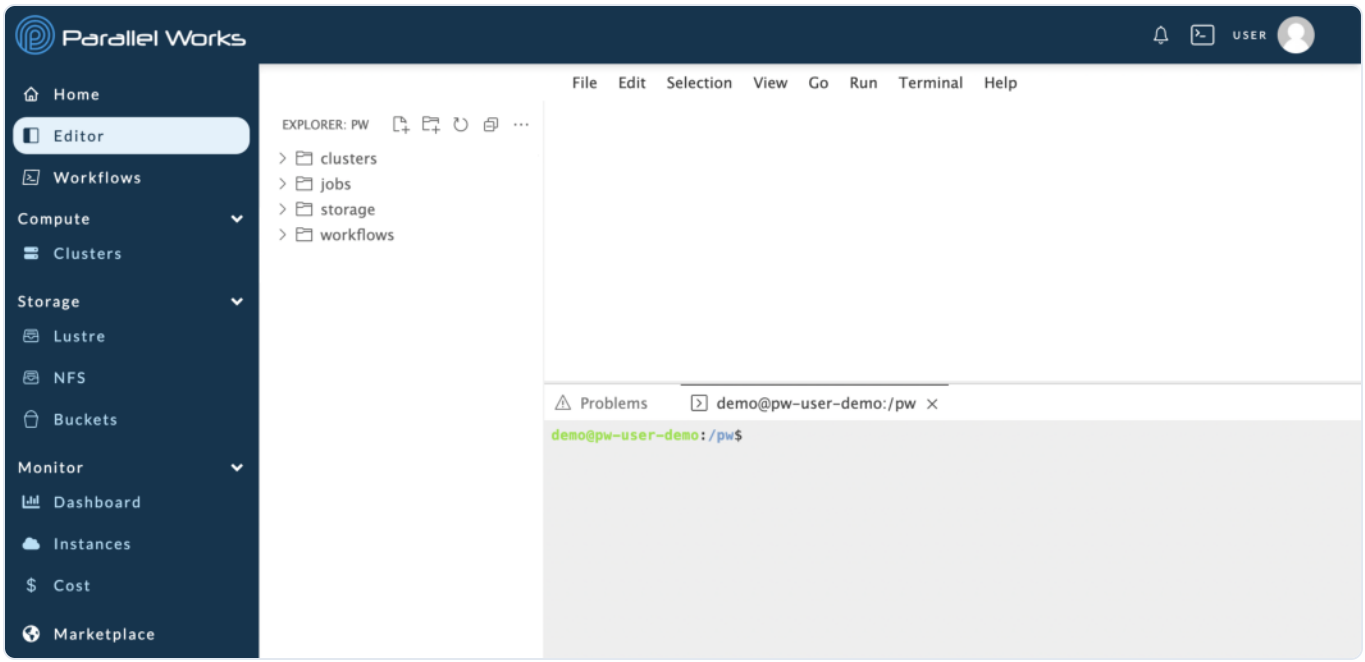
- **Run a workflow**: Click a favorite workflow
- **Start a cluster**: Click start on a favorite cluster
- **Open a session**: Click an active session
- **View details**: Click any item to see more information

Related Documentation

- Customizing Layouts - Personalize your workspace
- Running Workflows - Learn about workflows
- Creating Clusters - Set up compute resources

Editor

The Editor provides a full-featured integrated development environment (IDE) within ACTIVATE. Based on Theia, it offers a familiar VS Code-like experience for editing files, writing code, and debugging workflows.



Opening the Editor

Click **Editor** in the sidebar navigation to open the IDE.

Interface Overview

The Editor interface consists of several key areas:

Activity Bar (Left)

The vertical bar provides access to different views:

- **Explorer**: File and folder navigation
- **Search**: Find text across files
- **Source Control**: Git integration
- **Extensions**: Manage extensions

Sidebar

Shows content for the selected activity (files, search results, git status).

Editor Area (Center)

The main area where you edit files:

- Open multiple files in tabs
- Split the editor for side-by-side viewing
- Syntax highlighting for many languages

Panel (Bottom)

Contains terminal, problems, output, and debug console.

Status Bar

Shows current file information and editor state.

Working with Files

Opening Files

- **From Explorer**: Click a file in the file tree
- **Quick Open**: Press `Ctrl+P` (or `Cmd+P` on macOS)
- **From Terminal**: Use `code filename`

Creating Files

1. Right-click in the Explorer
2. Select **New File**
3. Enter the filename and press Enter

Saving Files

- **Save**: `Ctrl+S` (or `Cmd+S`)
- **Save All**: `Ctrl+K S` (or `Cmd+K S`)

Code Editing Features

Syntax Highlighting

The Editor supports many languages including Python, JavaScript, YAML, JSON, Bash, and more.

IntelliSense

- Type to see suggestions
- Press `Ctrl+Space` to trigger suggestions
- Press `Tab` or `Enter` to accept

Find and Replace

- **Find**: `Ctrl+F`
- **Replace**: `Ctrl+H`
- **Find in Files**: `Ctrl+Shift+F`

Integrated Terminal

Press `Ctrl+`` to open the integrated terminal, or select **Terminal** > **New Terminal**.

The terminal gives you command-line access for:

- Running scripts
- Using the PW CLI
- Managing files
- Connecting to clusters

Keyboard Shortcuts

Action	Windows/Linux	macOS
Command Palette	Ctrl+Shift+P	Cmd+Shift+P
Quick Open	Ctrl+P	Cmd+P
Save	Ctrl+S	Cmd+S
Find	Ctrl+F	Cmd+F
Toggle Terminal	Ctrl+`	Cmd+`
Go to Line	Ctrl+G	Cmd+G

Related Documentation

- Terminal - Command-line access
- Explorer - File browser
- Building Workflows - Workflow development

Explorer

The Explorer provides a file browser interface for navigating and managing files in your ACTIVATE workspace and connected storage.

Opening the Explorer

Click **Explorer** in the sidebar navigation, or access it through the Editor's file tree.

Navigating Files

The Explorer shows your files and folders in a hierarchical tree structure:

- Click a folder to expand or collapse it
- Click a file to select it
- Double-click a file to open it in the Editor

Navigation Controls

- Use the **breadcrumb path** to navigate to parent directories
- Click **refresh** to reload the current directory
- Use **search** to filter files by name

File Operations

Creating Files and Folders

1. Navigate to the destination directory
2. Right-click and select **New File** or **New Folder**
3. Enter a name and press Enter

Renaming

1. Right-click the item
2. Select **Rename**
3. Enter the new name and press Enter

Moving and Copying

Move:

1. Right-click and select **Cut** (or Ctrl+X)
2. Navigate to destination
3. Right-click and select **Paste** (or Ctrl+V)

Copy:

1. Right-click and select **Copy** (or Ctrl+C)
2. Navigate to destination
3. Right-click and select **Paste** (or Ctrl+V)

You can also drag and drop files between folders.

Deleting

1. Right-click the item
2. Select **Delete**
3. Confirm the deletion

> **PERMANENT DELETION**
>
> Deleted files cannot be recovered. Back up important files before deleting.

Uploading Files

1. Navigate to the destination folder
2. Click **Upload** or drag files from your computer

3. Wait for upload to complete

Downloading Files

1. Right-click the file
2. Select **Download**
3. Choose where to save it

Accessing Cluster Storage

When clusters are running, you can browse their file systems through the Explorer. Storage volumes attached to clusters appear at their configured mount paths.

Keyboard Shortcuts

Action	Windows/Linux	macOS
Copy	Ctrl+C	Cmd+C
Cut	Ctrl+X	Cmd+X
Paste	Ctrl+V	Cmd+V
Delete	Delete	Delete
Rename	F2	F2

Related Documentation

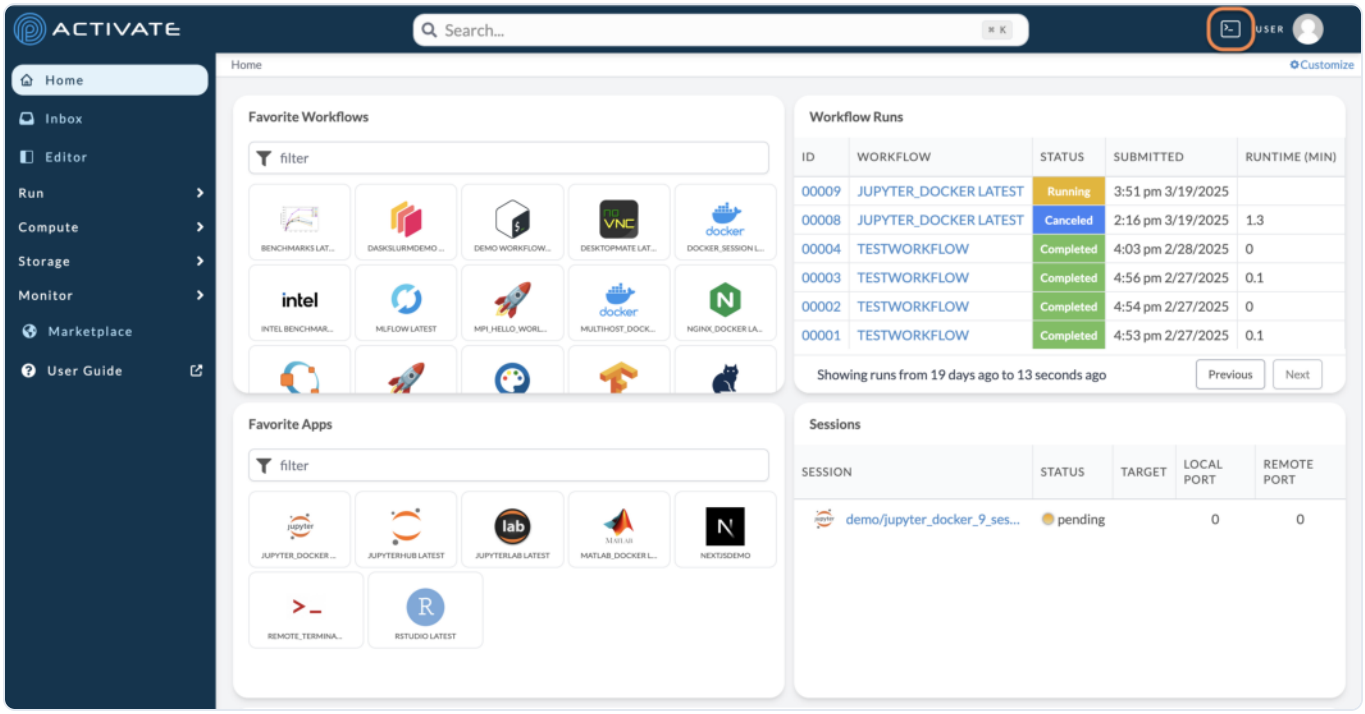
- Editor - Full IDE experience
- Storage - Storage options

Terminal

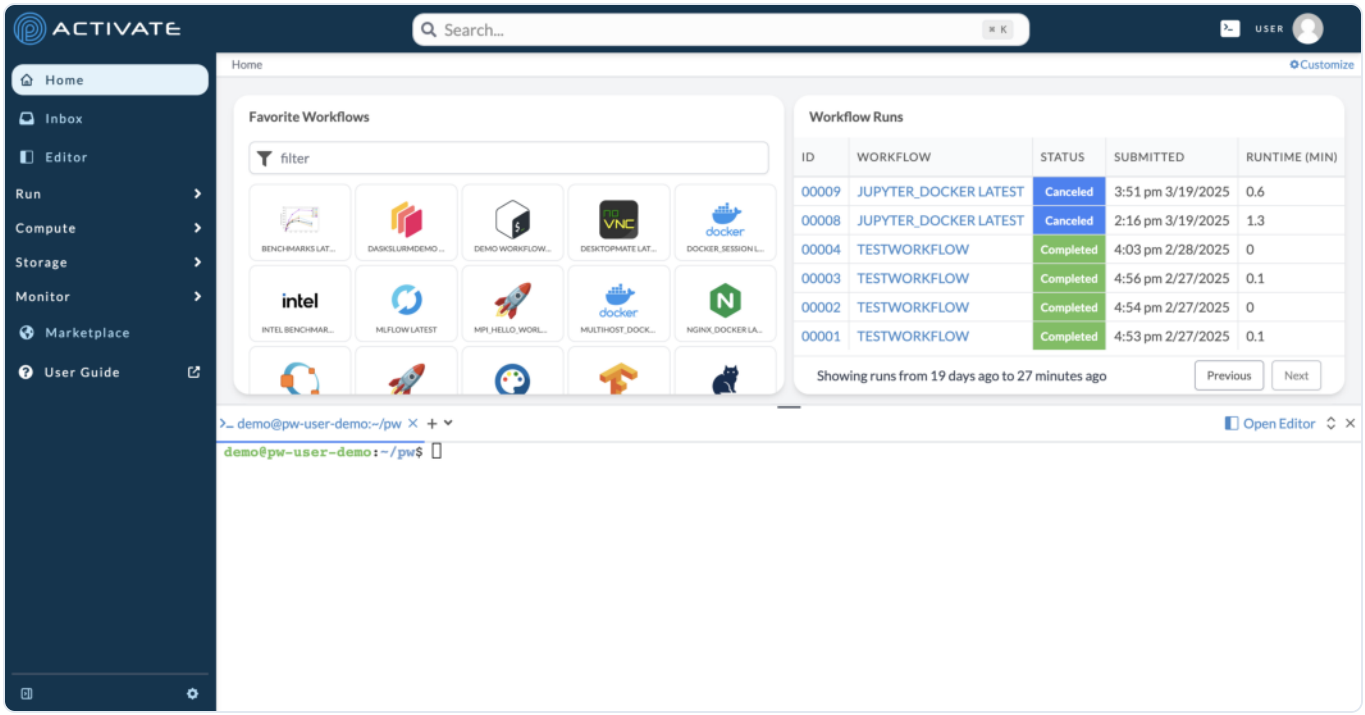
ACTIVATE provides a built-in terminal that you can use while navigating the platform. The terminal gives you command-line access to your user workspace and allows you to run PW CLI commands, manage files, and connect to clusters.

Opening the Terminal

Click the terminal icon in the top navigation bar to open a pop-up terminal at the bottom of your screen.



A terminal will appear at the bottom of your screen.



The terminal remains visible as you navigate between different pages in ACTIVATE.

Closing the Terminal

Click the terminal icon again to close the terminal. Your terminal session remains active in the background.

Using the Terminal

Your Workspace

When you open the terminal, you're in your user workspace where you can:

- Create and edit files
- Install software and packages
- Run scripts
- Access the PW CLI

Running PW CLI Commands

The PW CLI is pre-installed in your workspace:

```
# List your clusters
pw cluster ls

# List your workflows
pw workflow ls

# List your buckets
pw buckets ls
```

Connecting to Clusters

Use SSH to connect to a running cluster:

```
pw ssh my-cluster-name
```

To exit the SSH session:

```
exit
```

Terminal Features

Multiple Tabs

Open multiple terminal tabs for different tasks:

1. Click the + button next to existing tabs
2. Click between tabs to switch sessions
3. Click **x** to close a tab

Resizing

Drag the top edge of the terminal panel to resize it.

Keyboard Shortcuts

Action	Shortcut

Clear screen	Ctrl+L
Cancel command	Ctrl+C
Search history	Ctrl+R
Move to line start	Ctrl+A
Move to line end	Ctrl+E

Tips

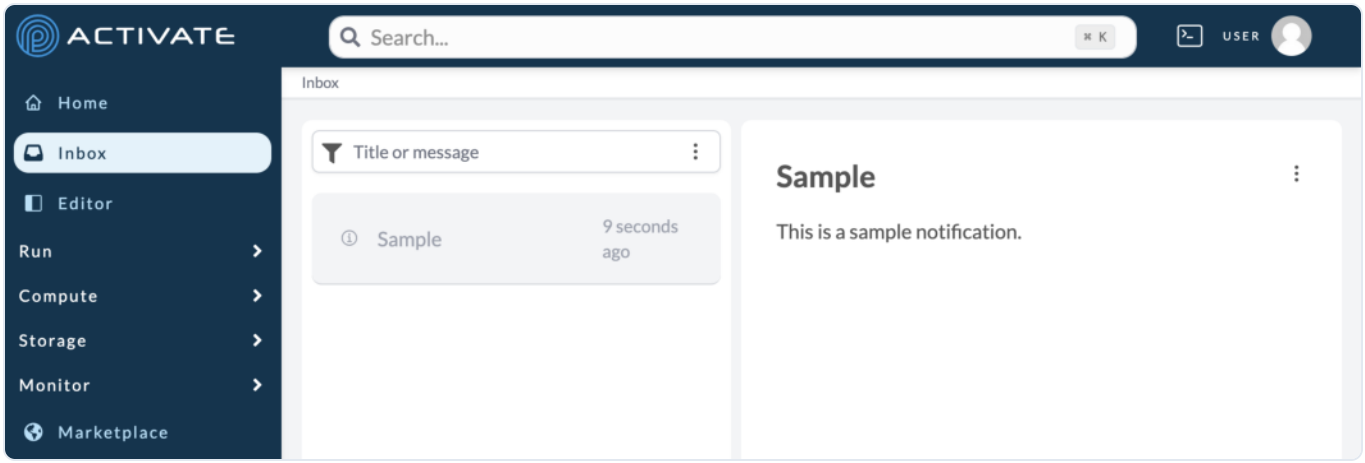
- Use **up/down arrows** to navigate command history
- Press **Tab** for auto-completion
- Use **Ctrl+C/Ctrl+V** (or Cmd+C/Cmd+V) for copy/paste

Related Documentation

- CLI Reference - Complete CLI documentation
- Editor - IDE with integrated terminal

Inbox

The Inbox shows notifications from your clusters, workflows, and organization administrators. Stay informed about important events and messages.



Accessing the Inbox

Click **Inbox** in the sidebar navigation to view your notifications.

Types of Notifications

Cluster Notifications

- Cluster started or stopped
- Provisioning complete
- Cluster errors or warnings

Workflow Notifications

- Workflow run completed
- Workflow run failed
- Important workflow events

Administrator Messages

Organization administrators can send messages to users about:

- System maintenance
- New features
- Policy updates
- Important announcements

Managing Notifications

Reading Notifications

Click a notification to view its full details. Unread notifications are highlighted.

Marking as Read

- Click a notification to mark it as read
- Use "Mark all as read" to clear all unread indicators

Notification Preferences

Configure which notifications you receive in Account Settings.

Related Documentation

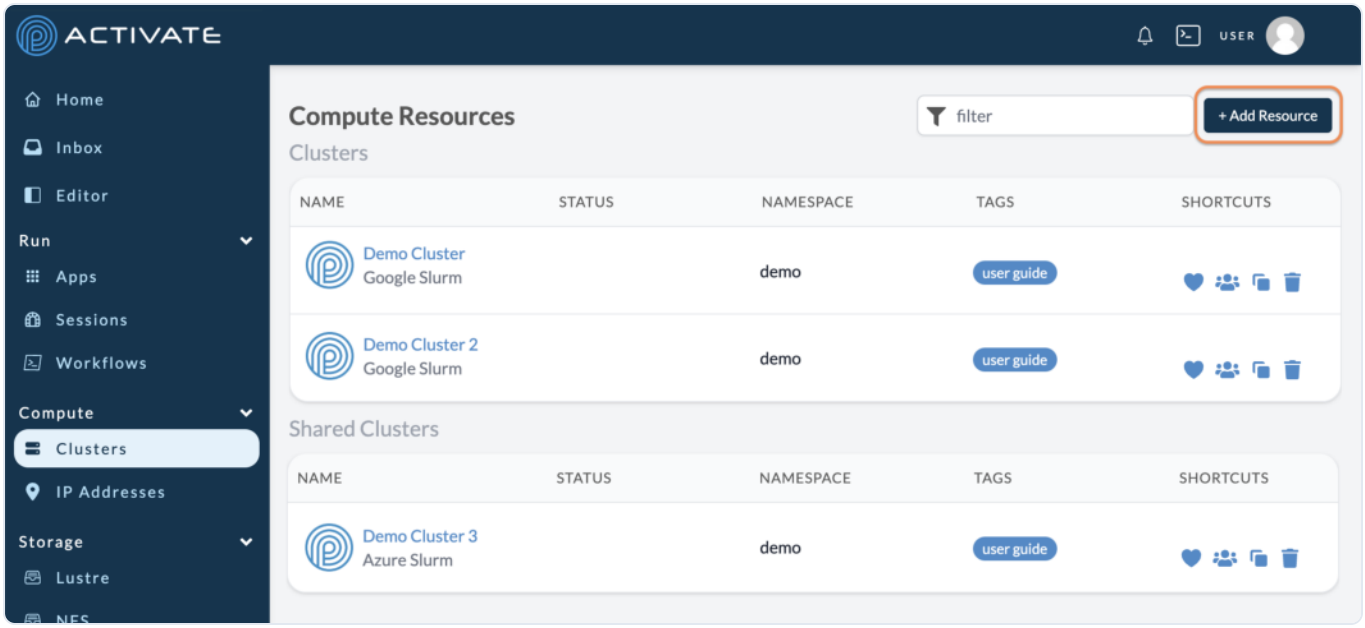
- Managing Notifications - Configure notification preferences
- Dashboard - View activity overview

Compute

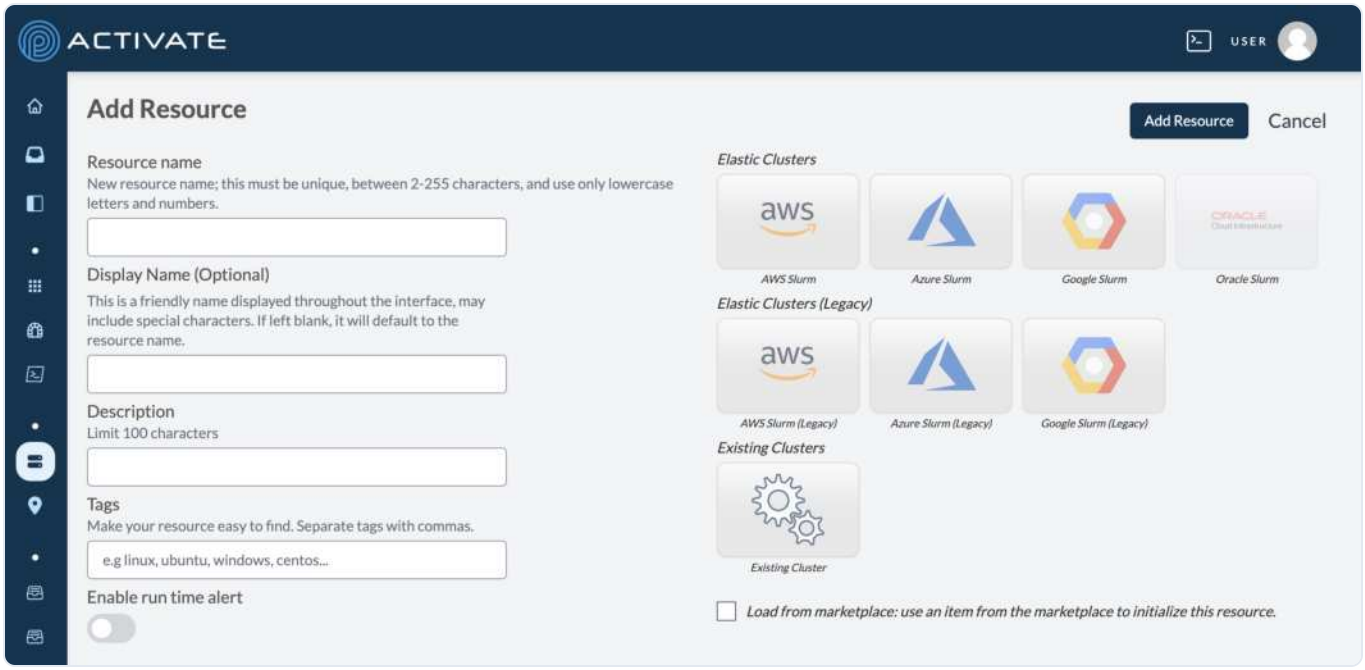
Creating Clusters

Any work you do on ACTIVATE will need to be completed on a compute resource. If you don't have any pre-configured resources that fit your needs for the project at hand, you can create a new one.

Navigate to the **Clusters** page and select **+ Add Resource**.



Now you can select the type of resource you need from the available elastic clusters. Once you choose your type of resource, you can create a cluster on a specific cloud service provider (CSP). For example, if you want to create an elastic cluster, you can do so with resources from AWS, Azure, or Google.

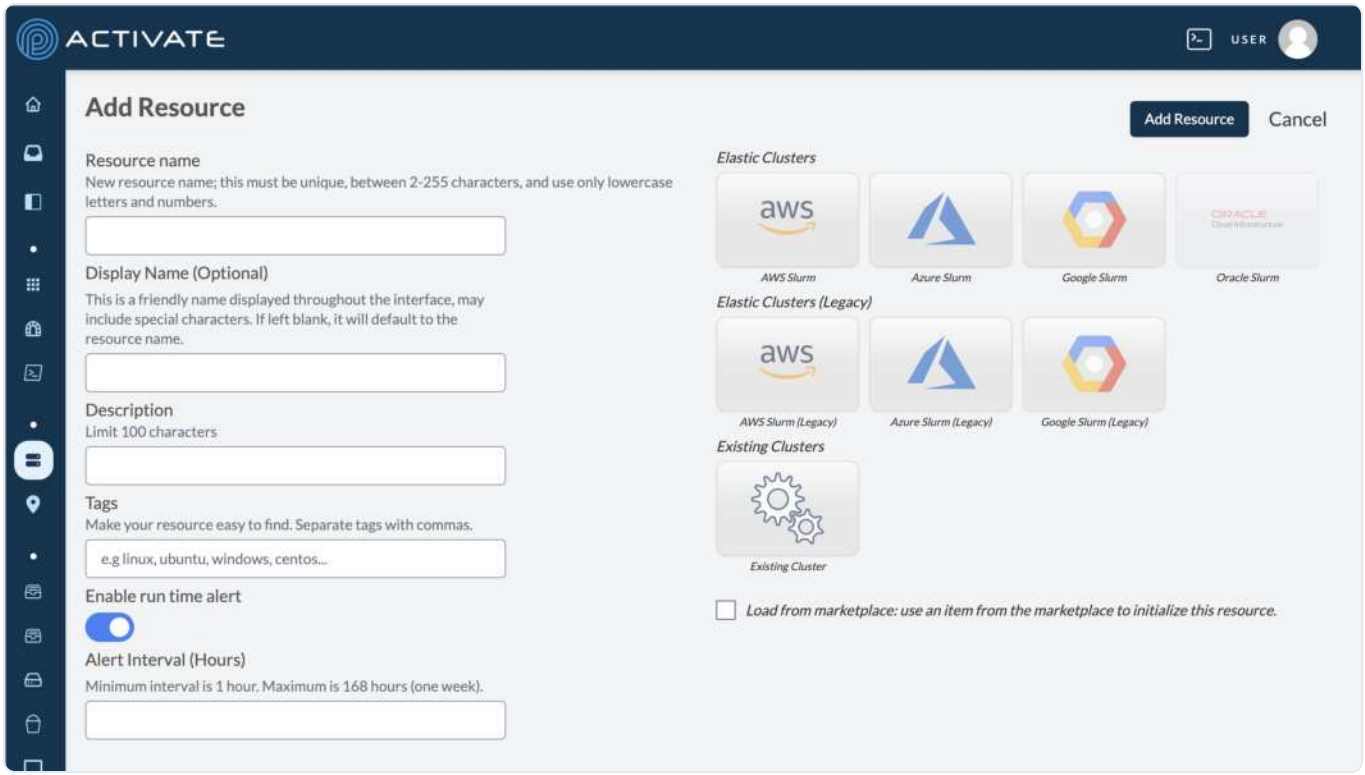


You must enter a **Cluster Name**. The **Display Name**, **Description**, and **Tags** are optional. If you leave **Display Name** blank, the cluster will default to the **Cluster Name** text.

Optionally, you can use the toggle button for **Enable run time alert**. When enabled, you'll receive an automated email from `noreply@parallelworks.com` at a set hourly interval, reminding you that your cluster is still running:

```
Dear {username},

This is an automated message to inform you that your cluster {cluster_name} has been running for
{time} hours. Please take note of the following details:

Cluster name: {cluster_name}
Cluster start time: {start_date}
Total run time: {total_time}
Next alert time: {next_alert}

This alert can be turned off in the resource's properties page.

Sincerely,

Parallel Works
```

When you click the toggle button, the field for **Alert Interval (Hours)** will appear. The value you enter here determines how often you'll receive run time alerts.



When you're ready, click **Add Resource**. Your resource creation will be confirmed on the next page with the message *Resource created*. You can configure your resource from this page. For more information on configuring clusters generally or what you need to configure clusters on specific CSPs, please see **Configuring Clusters**.

Your new cluster will also appear in the **Computing Resources** module on the **Home** page.

Which type of resource should I use?

If you're creating a new resource, an elastic cluster is typically the best choice. Users who are familiar with Slurm will be more comfortable using elastic clusters instead of elastic pools. Additionally, newer ACTIVATE workflows run exclusively on elastic clusters. For these reasons, we consider elastic pools to be a legacy feature and discourage users from choosing them to create new resources.

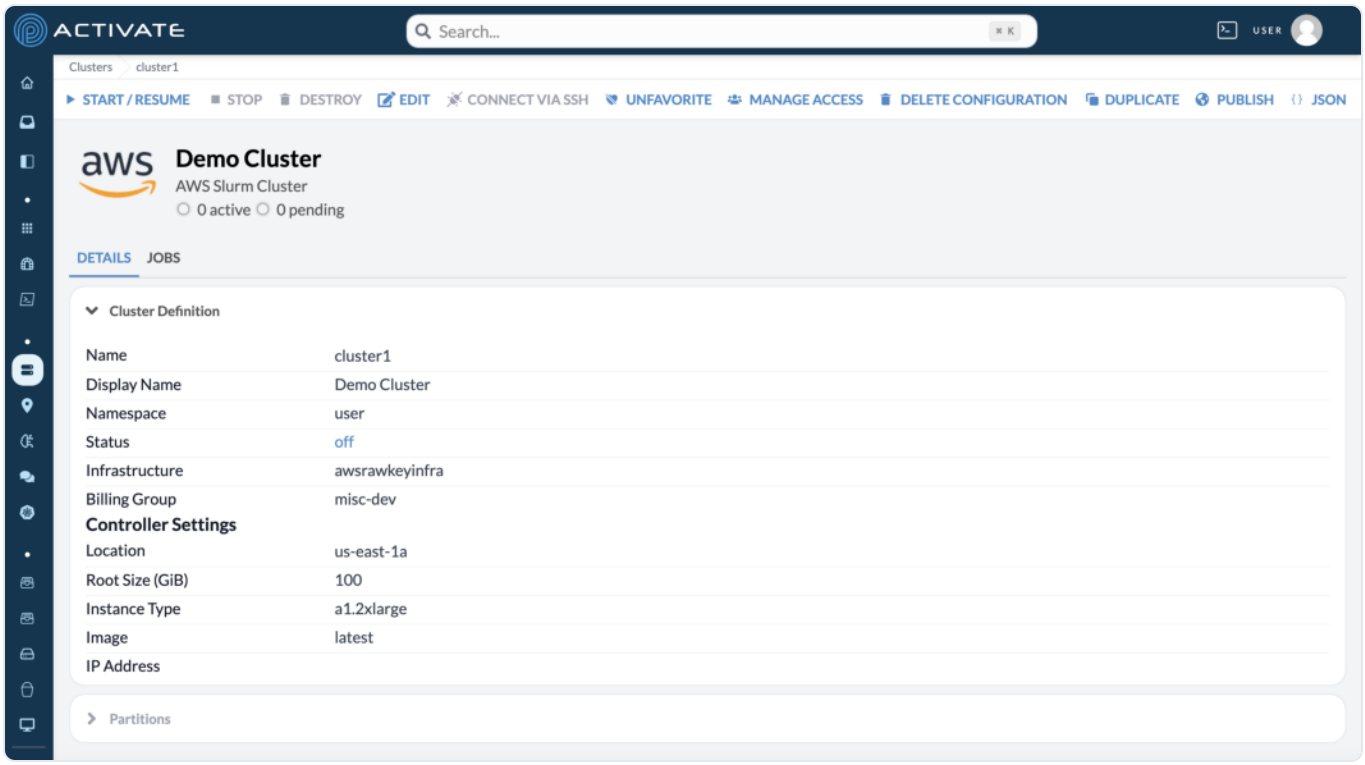
The CSP you choose depends on your organization, workflow needs, and whether you already have a cloud account from a specific CSP. Otherwise, any elastic cluster is a good choice to start with.

How should I configure my resource?

If you're simply testing resources or if your organization has not provided specific configuration settings for your project, we recommend using a configuration from the Marketplace. For more information, please see **Configuring Clusters > Load From Market**.

Navigating Clusters

Details



The **Details** tab displays information about your cluster's configuration settings.

Workflows*

**Only available on active clusters*

This module lets you run workflows directly from your cluster.

Observability*

**Only available on active clusters*

This module displays usage graphs for CPU, memory, and root disk storage on your cluster.The x axis represents time by the minute. The y axis represents the percentage of the resource you've used. Mouse over a graph line to display exact percentages and amounts in GiB.

Click **Expand** on any graph to see a larger version. Click the time dropdown menu to change the interval of data between 1, 3, and 8 hours. Please note that data older than 8 hours is discarded.

Attached Storages*

**Only available on active clusters*

This module displays any storage resources that are attached to your cluster, including the root disk. To add attached storage, please see **Configuring Storage** and **Attaching Storage**.

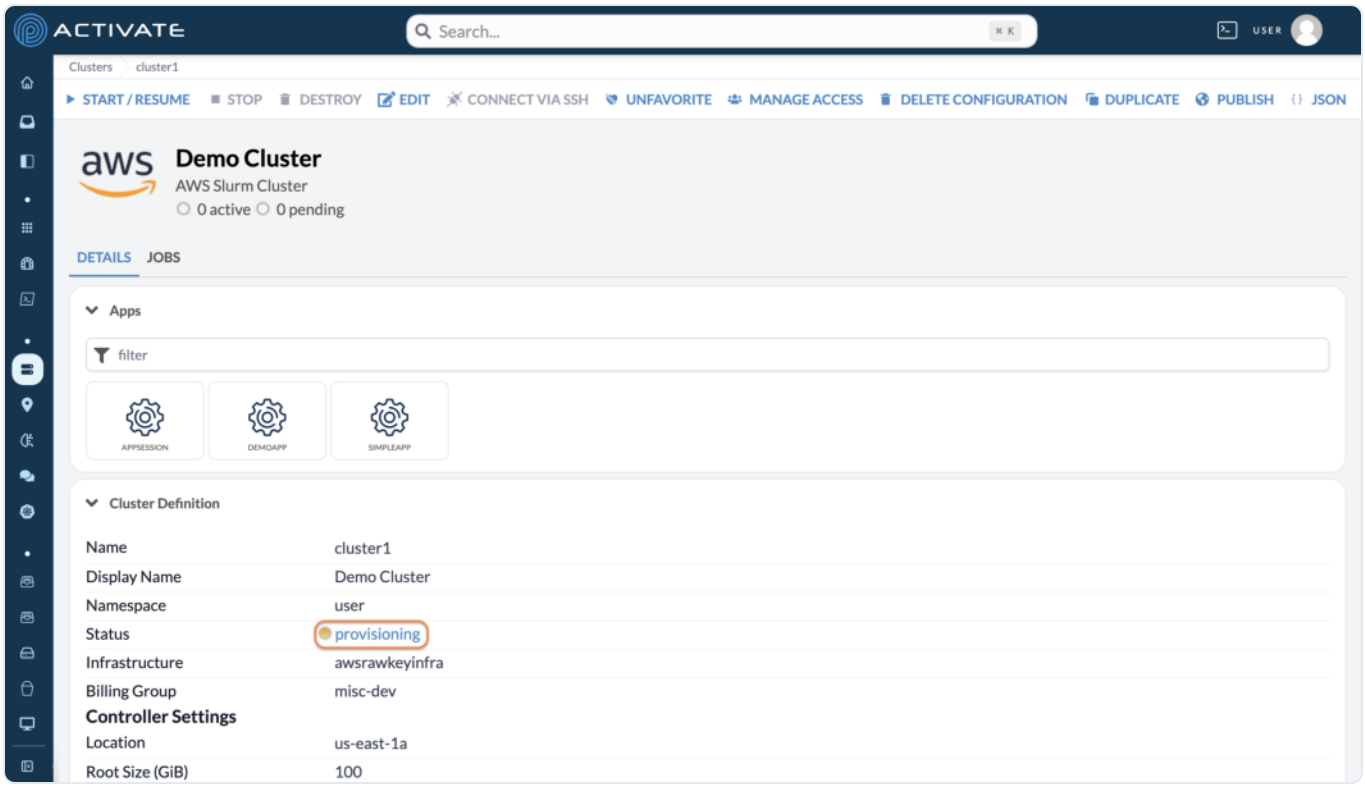
Click the monitor icon to create a snapshot of the storage resource. For more information, please see **Working With Disk Snapshots**.

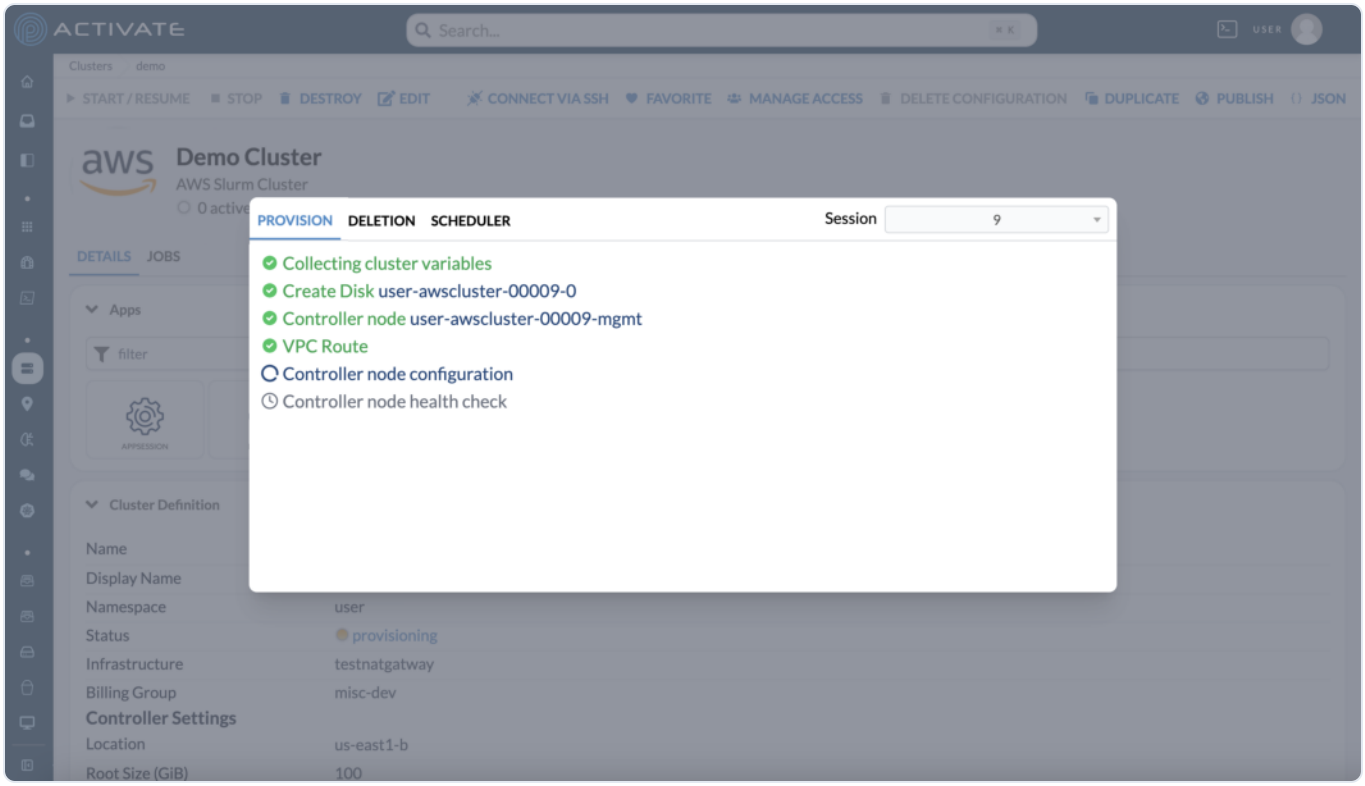
Cluster Definition

The **Cluster Definition** module lists the parameters and status of the controller node on your cluster. Please note that if you've just created your cluster, many parameters in **Cluster Definition** will be blank.

Status

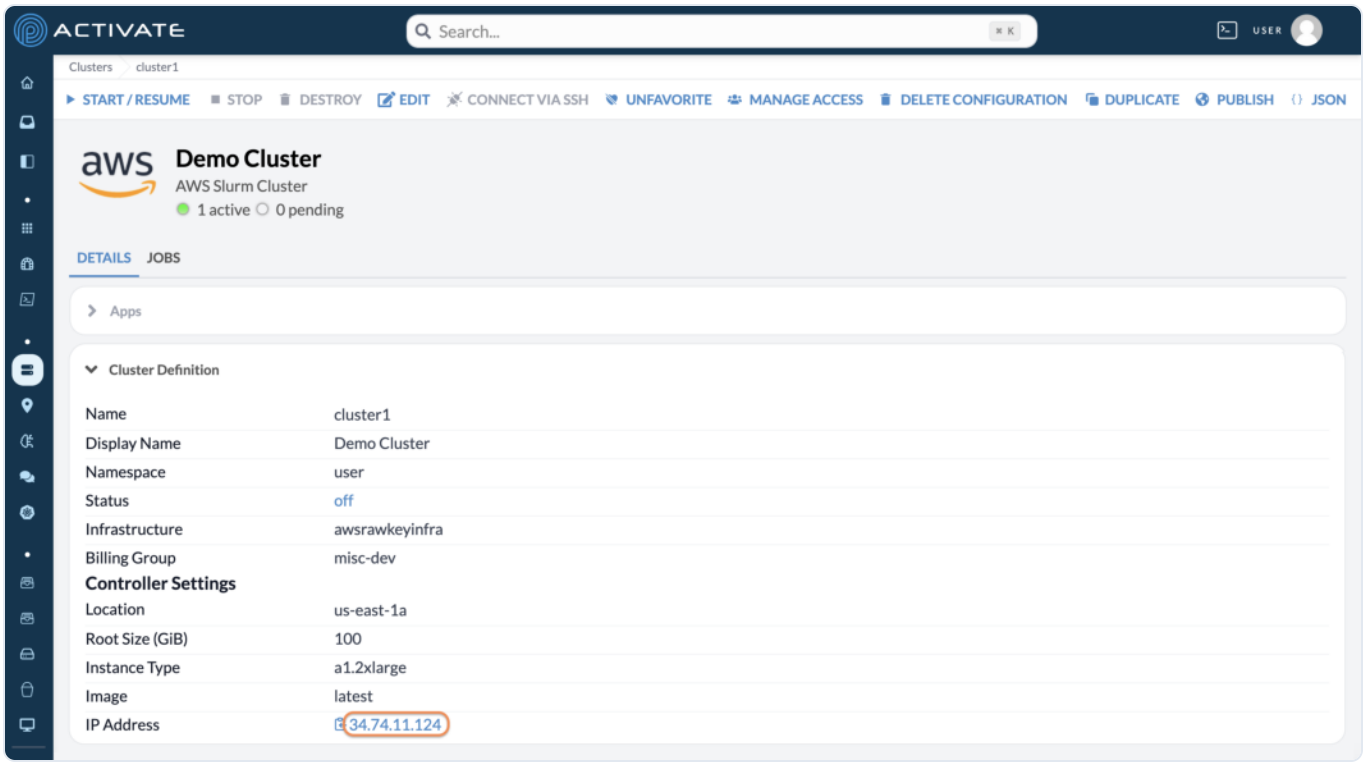
You can click a cluster's **Status** to see detailed provisioning logs.





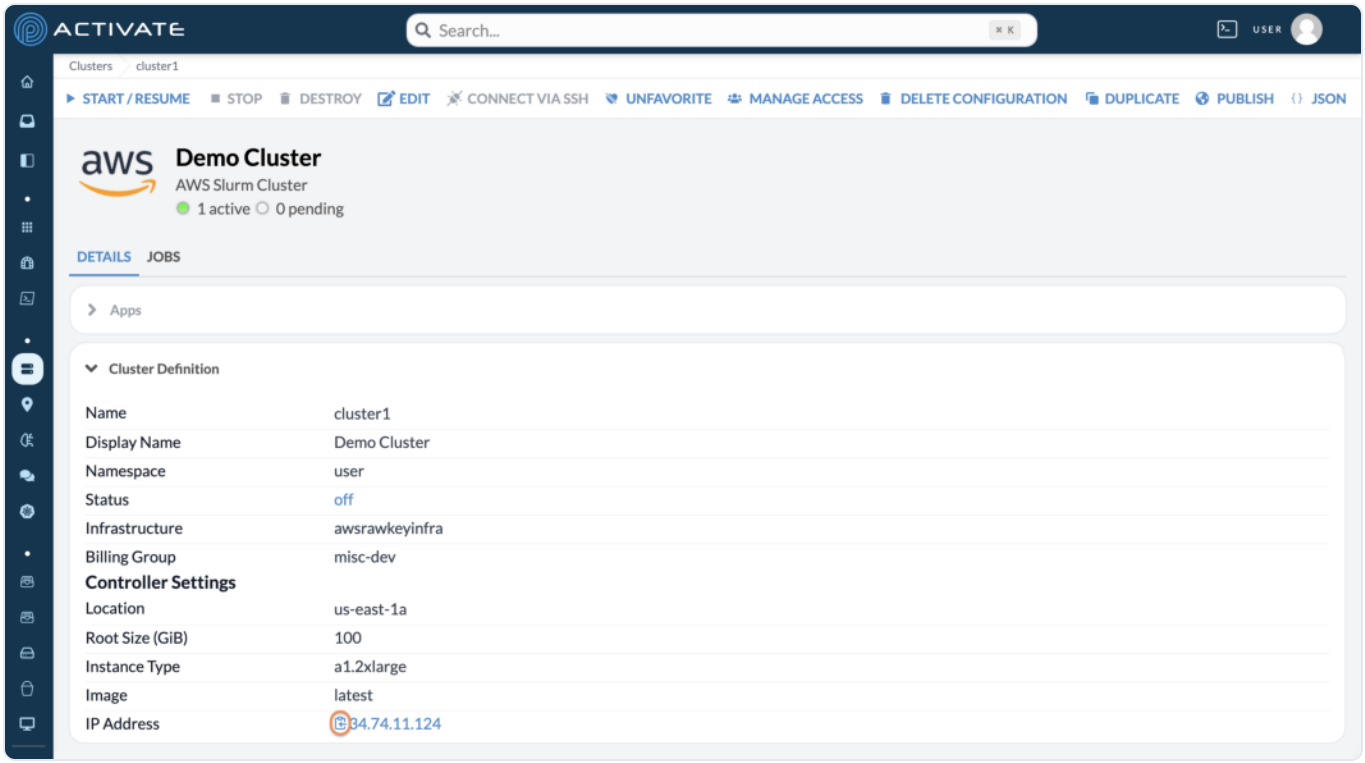
IP Address

When a cluster is active, you can click its IP Address to connect via SSH.



A terminal will appear on the bottom half of your screen, where you'll be connected to the controller node.

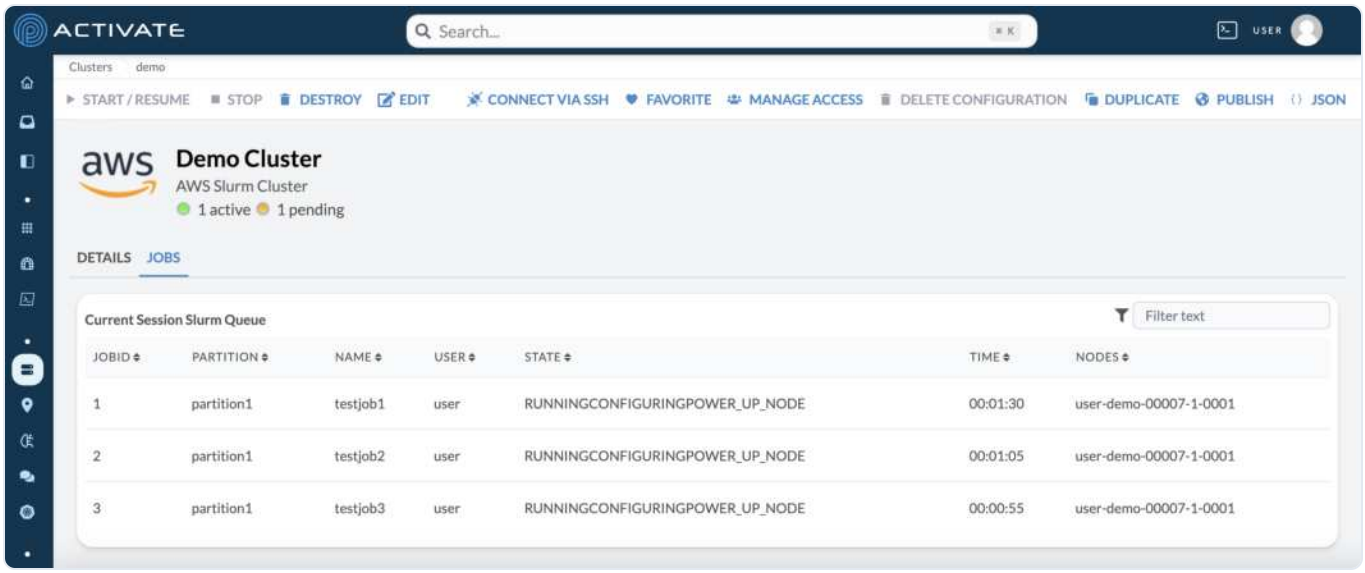
You can also click the copy icon to copy your IP address.



Partitions

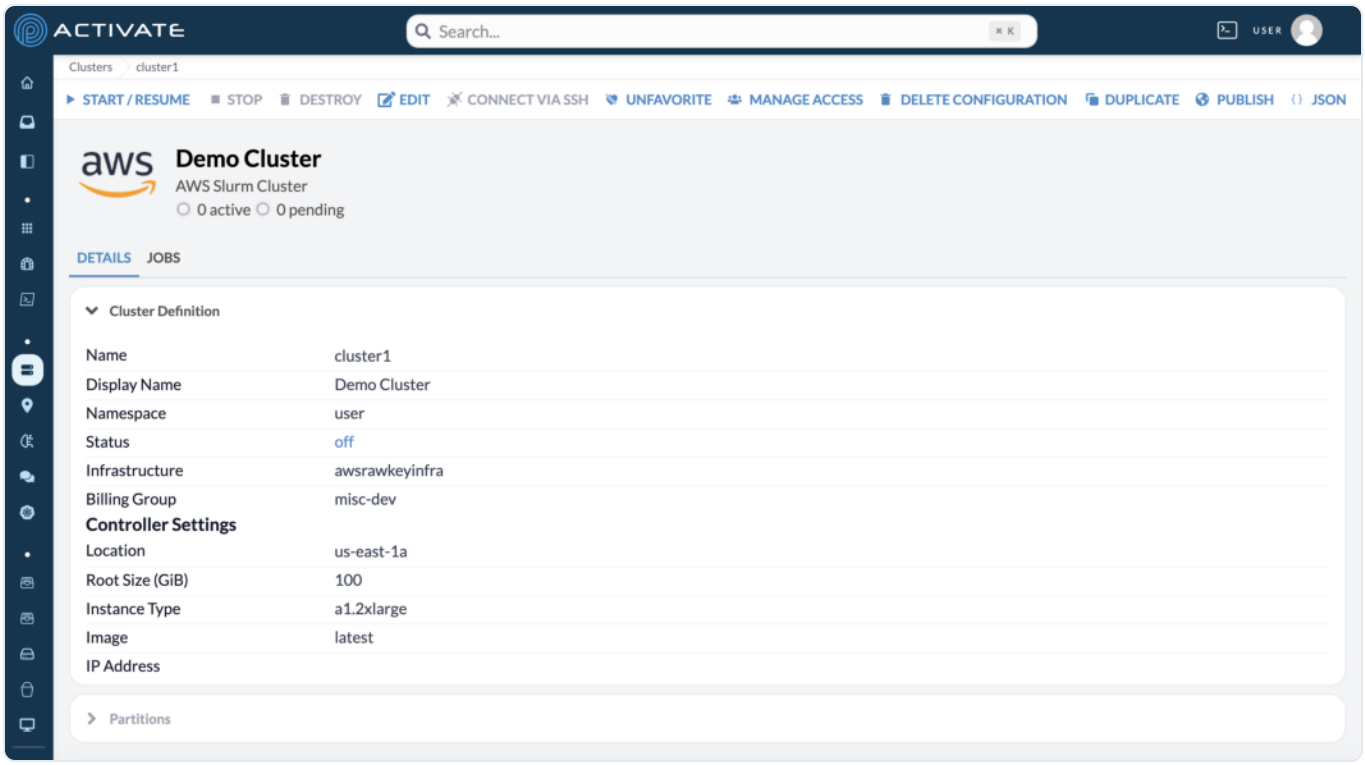
If you have any partitions on your cluster, details about them will be listed in **Partitions**.

Jobs



The **Jobs** tab displays a monitor for any running Slurm jobs on your cluster.

Action Bar



The action bar displays the major actions you can perform with your cluster.

Start/Resume

Use this button to start your cluster. You can only click **Start/Resume** when a cluster isn't provisioned or after you've stopped it by clicking **Stop**.

Stop

Use this button to stop your cluster's controller. This action is equivalent to pressing a physical power button or running the `shutdown` command. You can click **Start / Resume** to resume the controller while it's stopped.

When stopped, clusters continue to accrue costs only for attached storage resources and IP addresses. You will not be charged for compute resources.

Destroy

Use this button to destroy your cluster, which deprovisions all nodes and ephemeral filesystems. If you want to use the cluster again, it will have to start the provisioning process from the beginning when you click **Start / Resume**.

Edit

Use this button to edit your cluster's configuration settings. For more information, please see **Configuring Clusters**.

Connect Via SSH

Use this button to open a terminal and automatically connect to your cluster's controller node via SSH.

Favorite

Use this button to favorite your cluster. Favorited items appear on your Dashboard in the **Compute** module.

After favoriting, this button will change to **Unfavorite**. Click it to remove the cluster from your Dashboard.

Manage Access

Use this button to manage which groups in your organization can access your cluster. Your group name(s) will be specific to your organization.

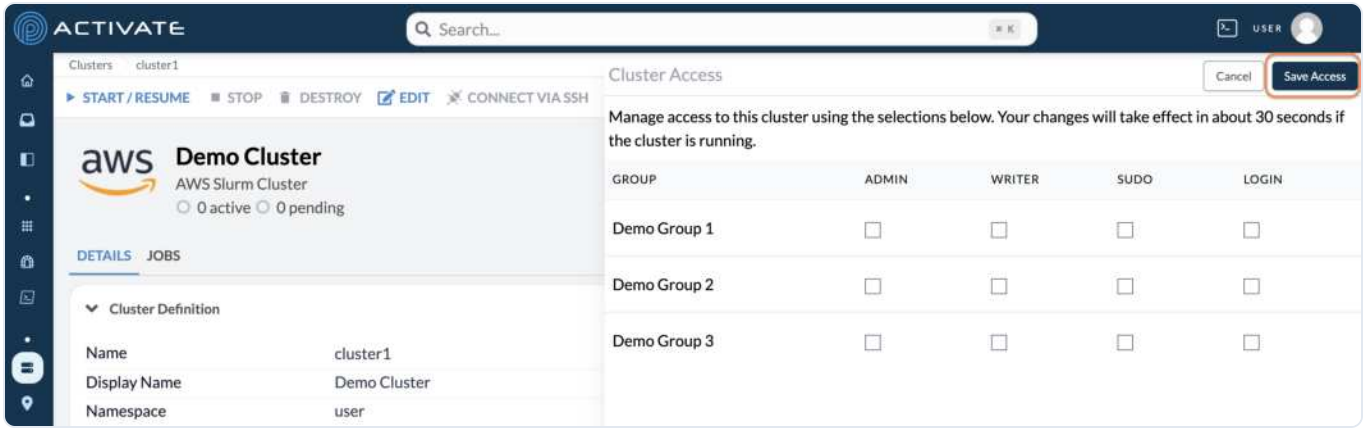
There are four levels of access:

- **Admin** grants users the same level of access as the owner of the cluster, including the ability to delete it.
- **Writer** grants users the ability to edit the cluster's configuration as well as log in to the cluster.
- **Sudo** grants users the ability to log in to the cluster with root access via the `sudo` command. Root access allows users to do anything inside both the controller and compute nodes.
- **Login** grants users the ability to log in to any compute node on the cluster by using the `ssh` command. Users' home directories are created automatically on the first login. Removing this permission revokes access after 30 seconds and kills any active sessions.

If you're not sure which group to select, you can contact us or your organization's ACTIVATE administrator.

> **ABOUT SUDO ACCESS**
>
> Although you can remove **sudo** access, it is best to reprovision a cluster to ensure access is revoked. This is because once a user has **sudo** access, they can install software and make changes to the cluster which may not be easily reversible.

After selecting or unselecting a group, click **Save Access**.



Delete Configuration

Use this button to delete your cluster. After deletion, it will no longer appear on your **Clusters** page.

Duplicate

Use this button to make a copy of your cluster. When you duplicate a cluster, you'll be asked to name it. The copy will appear on your **Clusters** page.

Publish

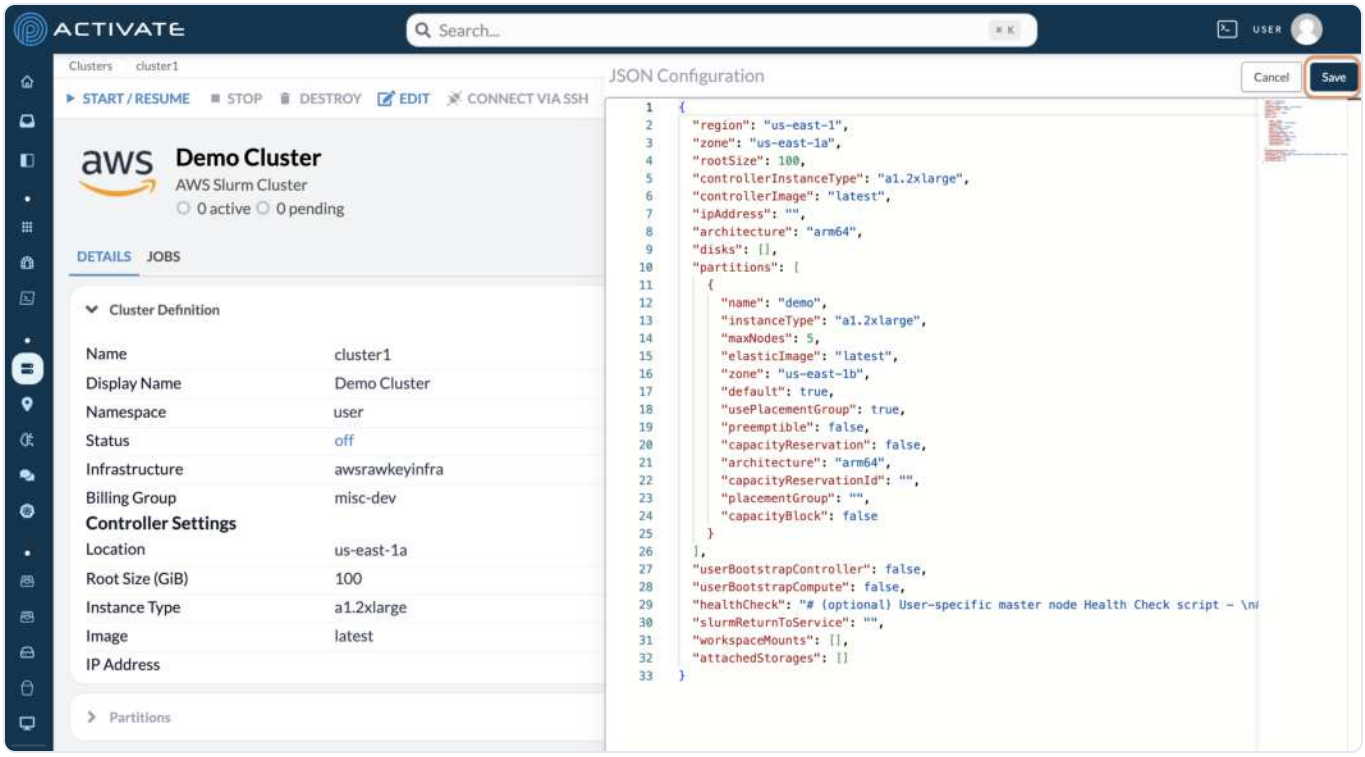
Use this button to publish a copy of your cluster in the Marketplace. Other users in your organization will be able to use the published version in their own ACTIVATE accounts.

For more information, please see **Publishing an Item**.

JSON

This button shows the JSON code version of your resource's configuration settings.

You can manually adjust these settings, then click **Save**.



What's the difference between stopping, destroying, and deleting clusters?

Stopping a cluster shuts down the controller node, preventing some costs from accruing. While stopped, you will not pay for the controller node compute costs, but you will continue to pay for any filesystems and storage.

If you spent time customizing the controller but do not need to use it for a period, you can stop the cluster instead of destroying it. As an alternative, you may consider snapshotting the controller and then when you're ready to resume, you can start the controller back up.

Another reason to stop the controller can be for troubleshooting. Anytime the system would benefit from rebooting, you may stop and then resume the controller to achieve that effect.

Destoying a cluster completely deprovisions it from the cloud. This deletes all ephemeral filesystems, all disks, the controller node, and all compute nodes are immediately deleted. Any running or pending jobs will be lost as resources are deleted.

Deleting a cluster configuration removes it entirely from your account. There are no costs incurred for having cluster configurations in your account.

More succinctly:

- Stopping temporarily shuts down the controller.
- Destroying completely removes all provisioned resources, including ephemeral filesystems and compute nodes.
- Deleting a configuration completely removes that configuration from our platform.

Configuring Cloud Clusters

This page explains the parameters of a cluster's configuration page.

Most ACTIVATE users will work exclusively with elastic clusters. These clusters are made up of a controller node and compute nodes, with the controller delegating tasks to the compute nodes.

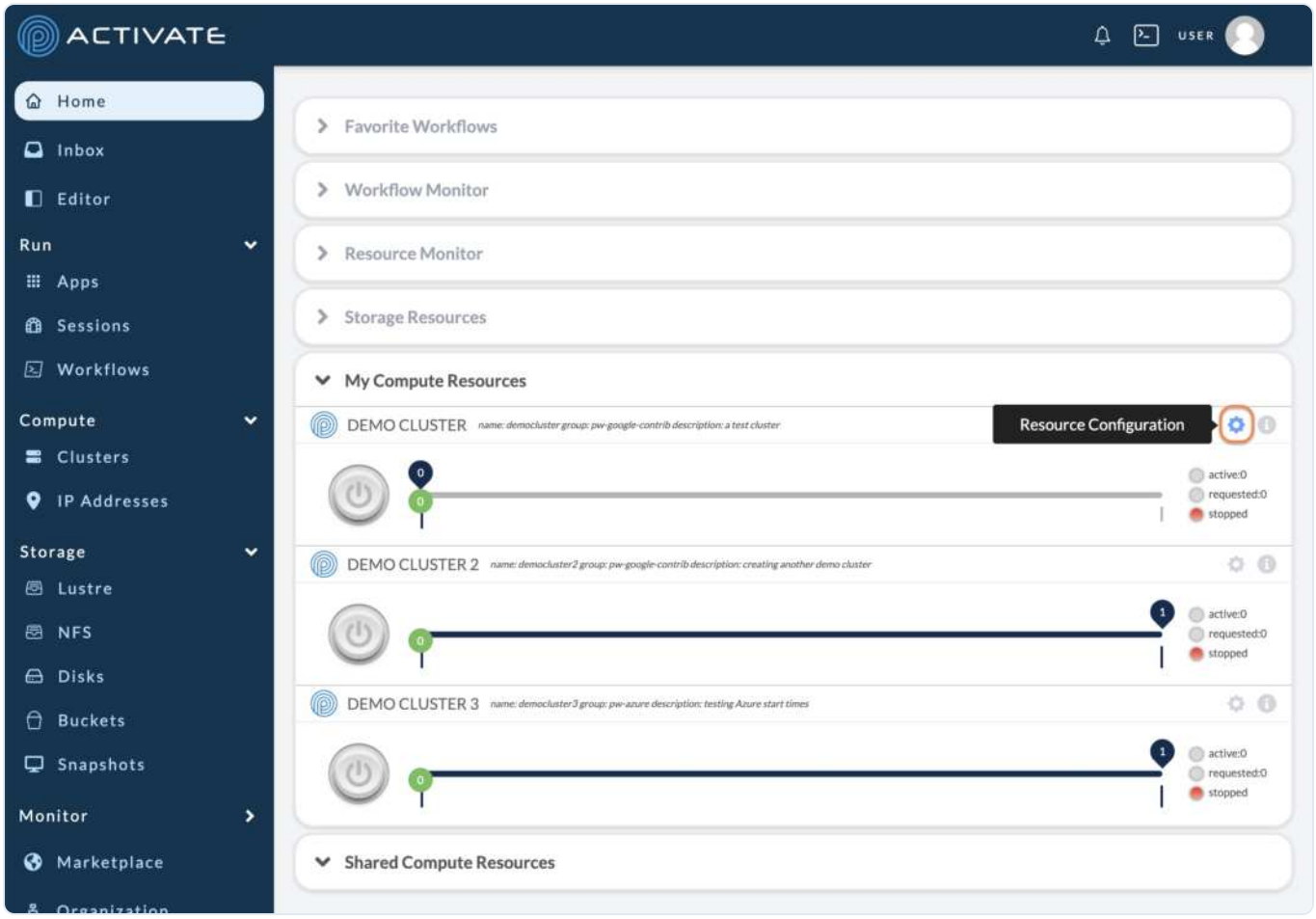
Clusters have several adjustable parameters for both controller and compute nodes, such as compute instance types and node count. Additionally, compute nodes are grouped together in partitions, which have their own settings. For more information, see **Partition Settings** below.

ACTIVATE also supports an optional parallel filesystem, Lustre. For more information on setting up Lustre for your account, see **Configuring Storage**.

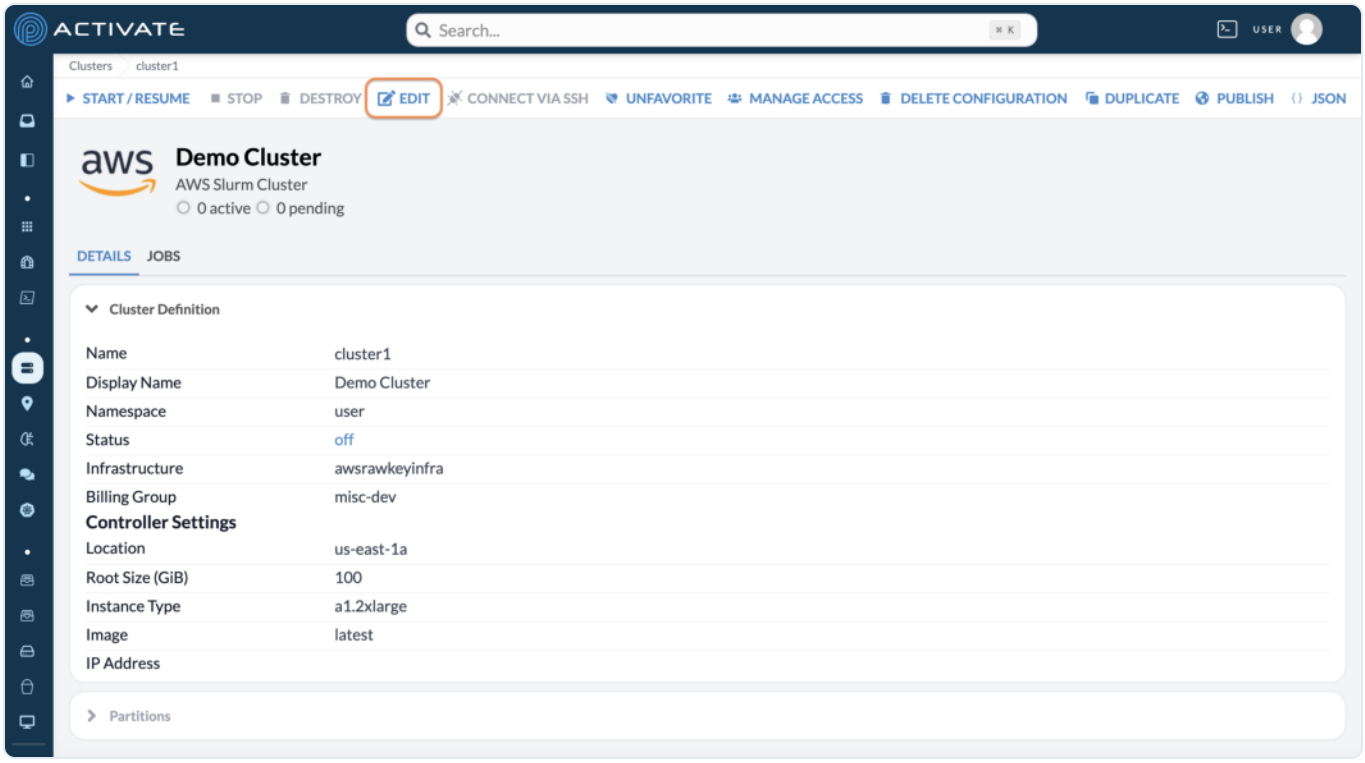
Accessing Configuration Settings

You can access a resource's configuration settings from the **Clusters** page. Click the name of the resource you want to configure.

Alternatively, if you have the **Compute** module on your **Dashboard**, click the gear icon for the resource you want to configure.

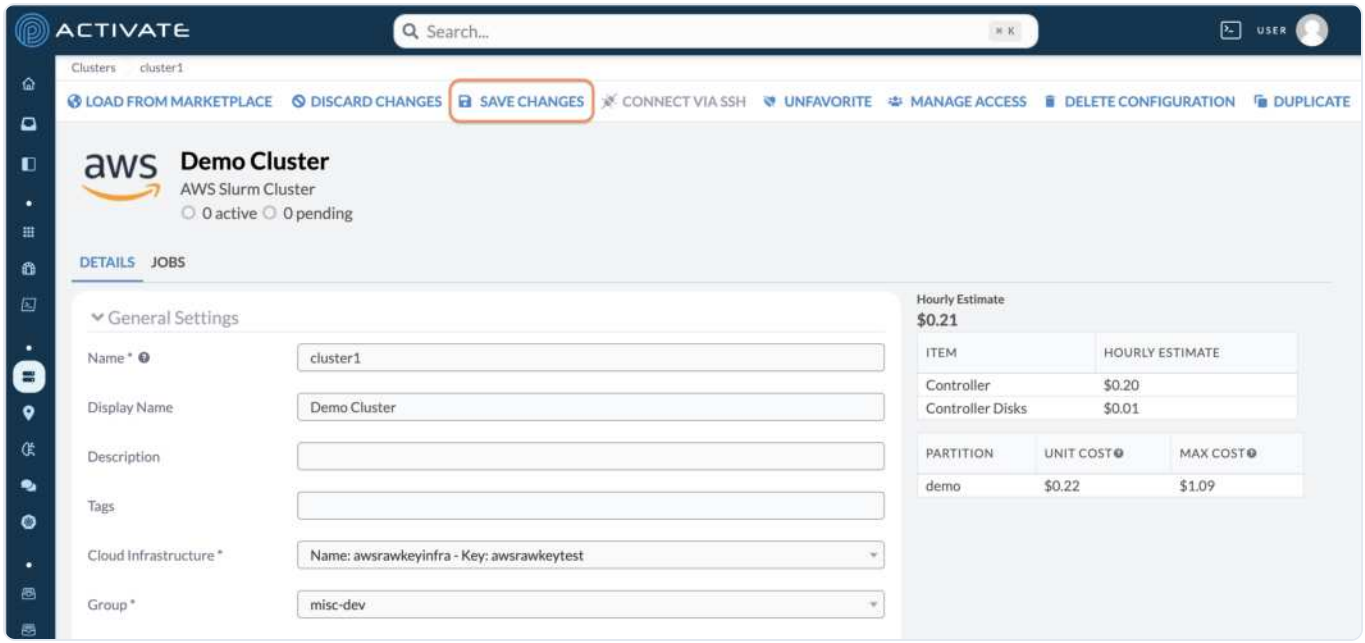


Once you've navigated to your cluster, click **Edit**.



About Saving

When you change settings for a cluster, you must click **Save Changes**.



About Live Updatable Settings

Some settings can be updated while a cluster is running. These settings are marked with a icon on this page.

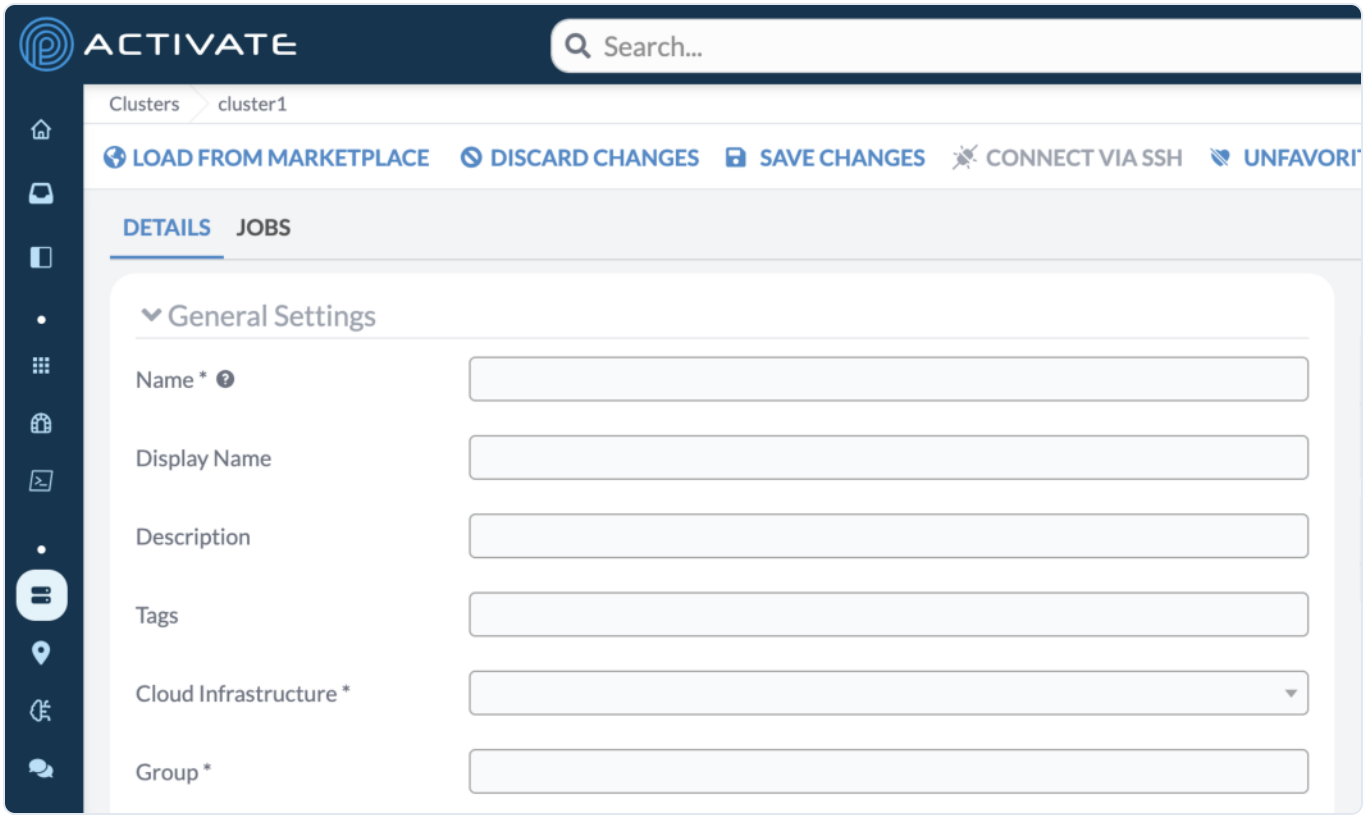
When you change a live updatable setting, click **Save Changes**. A dialog box will appear with the message *Changing configuration while cluster is running will make it unavailable until the changes are complete.*

Are you sure you want to save?

Click **Save**. Once your cluster updates, a notification will appear with the message *Cluster [name] configuration updated successfully*.

General Settings

Clusters will typically have these settings in the **Definition** tab of the configuration page. Settings with an asterisk * are required.



Name

Use this field to name the cluster. The name you choose must be unique across your account. For example, you cannot have both a cluster and a storage resource named "demo".

Display Name

Use this field to give the cluster a display name, which will appear at the top of the cluster's page, on the **Clusters** page, and on the Dashboard in the **Compute** module. If you leave this field blank, the value for the cluster's **Name** will be displayed instead.

Description

Use this field to give the cluster a description of 100 characters or fewer.

Tags

Use this field to add tags to the cluster. Tags appear on the **Clusters** page. They are useful for information at a glance (such as "user guide" tags on the clusters we use for our documentation).

Network

Use this dropdown menu to select the network that your organization uses for a specific cloud service provider. If you're not sure which one to select, please contact your organization's administrator. For more information about networks, see **Networks**.

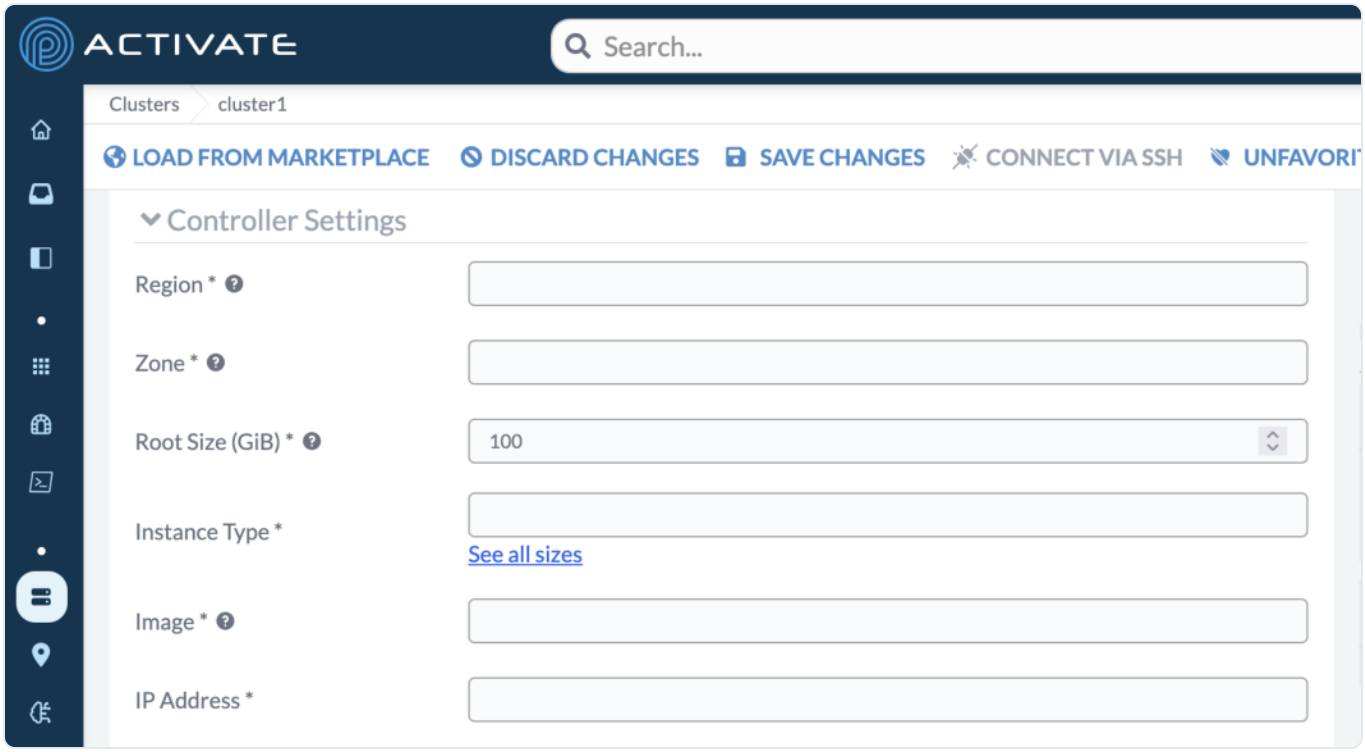
Group

Use this dropdown menu to select the group name that your organization uses to allocate costs. This menu is especially important if your organization is running multiple groups simultaneously.

If you're not sure which group to select, you can contact us or your organization's ACTIVATE administrator.

Controller Settings

These settings define the configuration for the controller node, such as region, instance type, and OS image. Some settings will differ depending on which type of resource you're using. For more information, see **CSP-Specific Settings** below.



Region

Use this dropdown menu to select the region that your cluster will deploy computing resources into.

A region represents a geographic area.

Zone

Use this dropdown menu to select the zone to use for the controller.

A zone refers to an isolated location inside a region.

Root Size (GiB)

Use this field to specify the size of the root disk in gibibytes (GiB).

Instance Type

Use this dropdown menu to select the instance type of the controller. The instance type determines the CPUs and amount of memory available on the machine. Certain instance types may also have specialty hardware, such as GPUs or low-latency networking options.

To see a list of available instance types and their cost per hour, click **See all sizes**. From the instance type list, you can click an option to select it.

For more information about instance types and what their names mean, please see **Choosing Instance Types**.

Image

Use this dropdown menu to select the operating system (OS) image for the cluster's controller node.

If you're not sure which image to pick, we recommend using the latest version because this will ensure you have the most up-to-date software on your cluster; the latest image version includes OS updates and software required to connect to the ACTIVATE control plane.

You can also use this dropdown menu to select custom cloud snapshots.

IP Address

Use this dropdown menu to choose the IP address that will be assigned to the cluster. You can choose an automatically assigned IP address or an IP address that you've created.

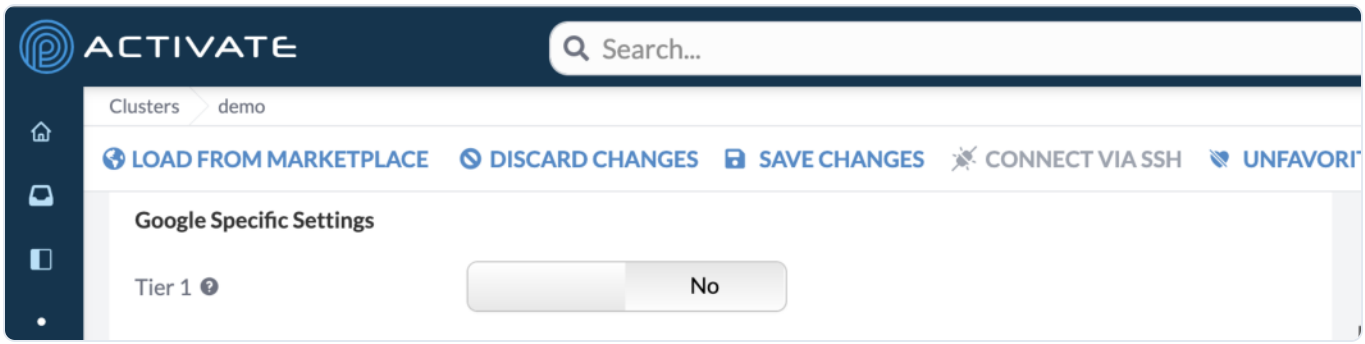
For more information, please see **Managing IP Addresses**.

CSP-Specific Settings

Each cloud service provider (CSP) builds and configures their resources differently. Clusters on ACTIVATE have settings that correspond to each CSP's model of cloud services. The CSP-specific parameters are outlined below.

Please note that these CSP-specific settings will also appear as options inside the partition settings on clusters.

Google



Tier 1

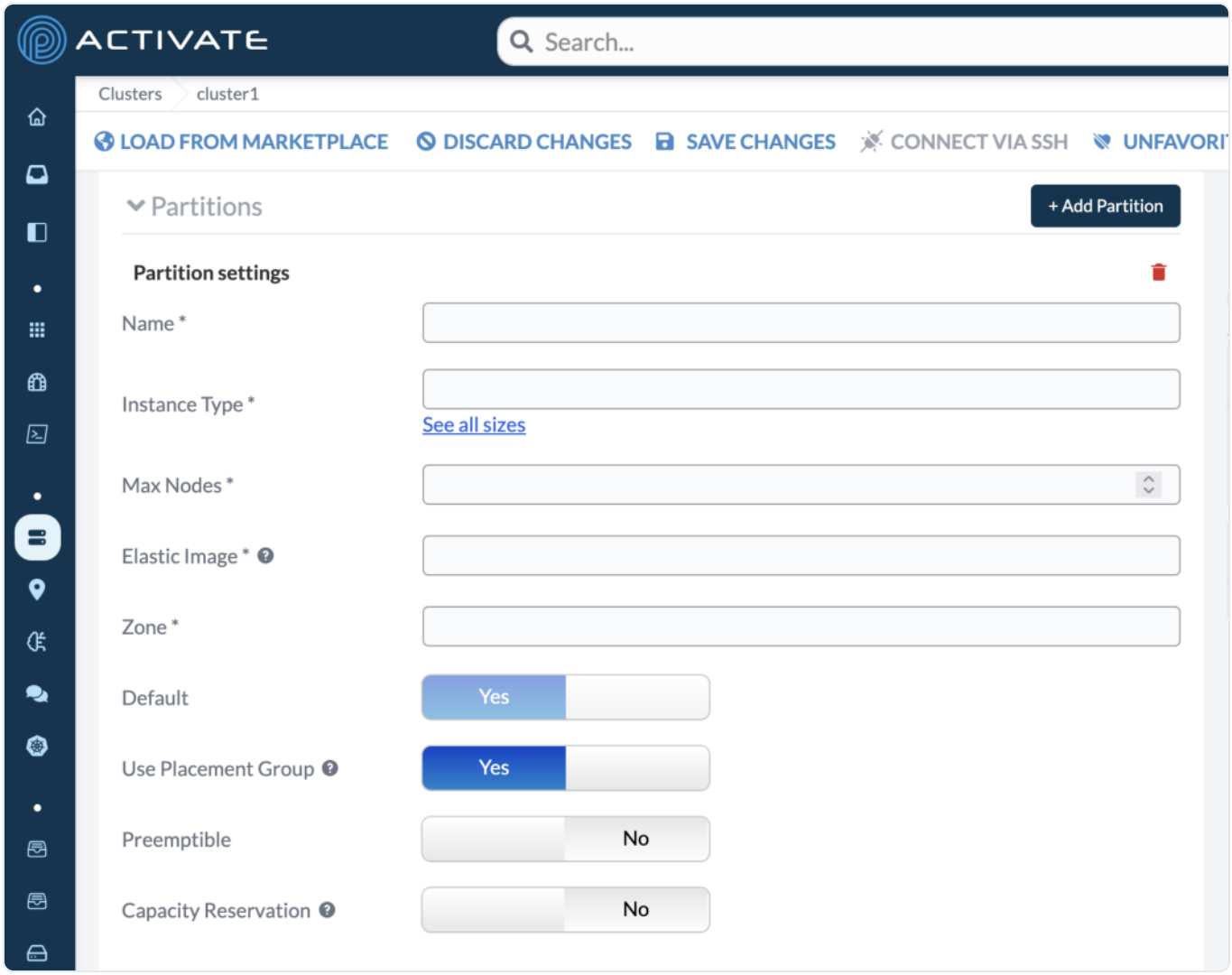
Use this toggle button to enable Tier 1, which increases maximum egress bandwidth (upload speed) to 50–100 Gps, depending on the size of the instance. If Tier 1 is off, the egress bandwidth will range from 10–32 Gbps.

For more information, see the Google documentation on Tier 1.

Partition Settings {#partition-settings}

You can create partitions in clusters to send your work to homogenously configured sets of worker nodes.

If you click + **Add Partition**, a list of new settings will appear. Typically, a partition will have the following configuration options. Some settings will differ depending on which type of resource you're using. For more information, see **CSP-Specific Settings** below.



Name

Use this field to name your partition. You must use a unique name for each partition you create.

Instance Type

Use this dropdown menu to select the configuration of the partition. These options work in the same way that the controller instance types do.

Max Nodes

Use this field to enter the max number of nodes in a partition.

Elastic Image

Use this dropdown menu to select the operating system image for the partition. We recommend using the latest version.

MultiZone

Enable this toggle to provision nodes in multiple zones. If there is not enough capacity in one zone, we will try to provision nodes in a different zone. This can be helpful for obtaining instances that are in high demand.

The exact behavior depends on the cloud provider:

- **AWS**: All nodes in a single request are placed in the same zone. If a zone cannot fit every node in the batch, we move the entire batch to the next zone you selected and try again. When adding nodes to a running cluster, we try the zone that already holds the most of the cluster's nodes first; if the cluster has no nodes yet, we try the controller's zone first when it is one of your selected zones.
- **Google**: A single request is submitted at the region level, along with your list of allowed zones. Google Cloud then chooses zones from that list based on available capacity, so the request can spread or fall back across zones automatically.
- **Azure**: Azure clusters do not use the **MultiZone** toggle. Selecting **Automatic** from the **Zone** dropdown instead lets Azure distribute the partition's compute nodes across the region's availability zones for you.

> **WARNING**
>
> If nodes are provisioned in multiple zones, there may be costs associated with data transfer between zones. For more information, see the **AWS documentation on data transfer costs** or the **Google documentation on data transfer costs**.

Zone(s)

Use this dropdown menu to select which zone(s) nodes will be provisioned in within your selected region. If **MultiZone** is enabled, you can select multiple zones.

Default

Use this toggle button to specify whether a partition is the default location for running jobs. For more information on running jobs on specific partitions, see **Submitting Jobs**.

> **DEFAULT PARTITION**

> This feature is important if you create multiple partitions. If you only create one partition, it will automatically be set to **Default** and cannot be changed, as shown in the screenshot above.

Use Placement Group

Currently available only on AWS and Google clusters.

Use this toggle button to enable the partition's instances to be launched as a placement group.

Typically, partition instances are launched throughout a zone. This strategy minimizes the chances of disprupting users' work: if one of the instances fails, especially from an outage or damage, the other instances are likely to remain unaffected.

With a placement group, the instances are instead launched close together. This strategy can improve network performance and is best suited for jobs that are small, short, or will not be affected by interruptions.

For AWS clusters, this setting is enabled by default.

Preemptible

Use this toggle button to specify whether a partition is a preemptible instance. Preemptible instances can be cost effective because they make use of resources that are already available but currently unused.

However, preemptible instances can be disrupted because another user can take over that available resource at any time. For this reason, we recommend using preemptible instances at your own risk.

Capacity Reservation*

*Before you can use a capacity reservation on ACTIVATE, it must first be configured by an administrator in your organization. After this initial step, your administrator can distribute the values for **Reservation ID** and, if necessary, **Placement Group**.*

Use this toggle button to enable on-demand capacity reservations, which reserve a set amount of compute capacity.

When **Capacity Reservation** is enabled, a **Reservation ID** dropdown menu will appear. For AWS, you will also enter the **Placement Group**.

Attached Filesystems Settings

Use this section to attach any of your configured storage resources. For more information, please see **Attaching Storage**.

Advanced Settings

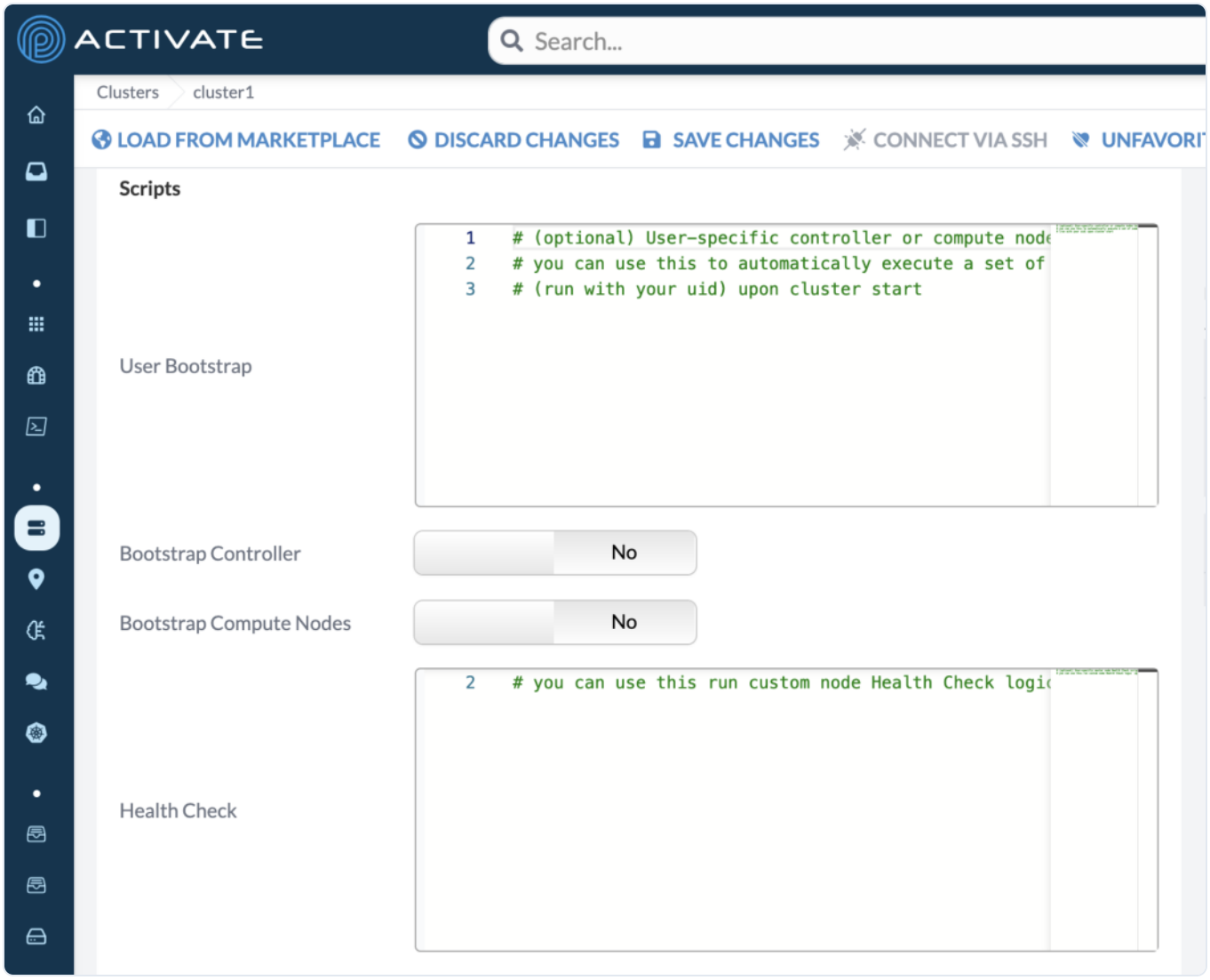
Admin Settings

Debug Mode

Use this toggle button to disable clean-up for provisioning failures. The cluster owner will also have root access to the nodes for debugging purposes (`ssh root@`).

Script Settings

Optionally, you can set scripts to execute when you start a cluster.



User Bootstrap

Use this text box to set a script that executes once a controller node has started. For example, you can set files to automatically move into a specific folder.

Bootstrap Controller

Use this toggle button to specify whether your bootstrap script will run on the controller node.

Bootstrap Compute Nodes

Use this toggle button to specify whether your bootstrap script will run on compute nodes.

Health Check

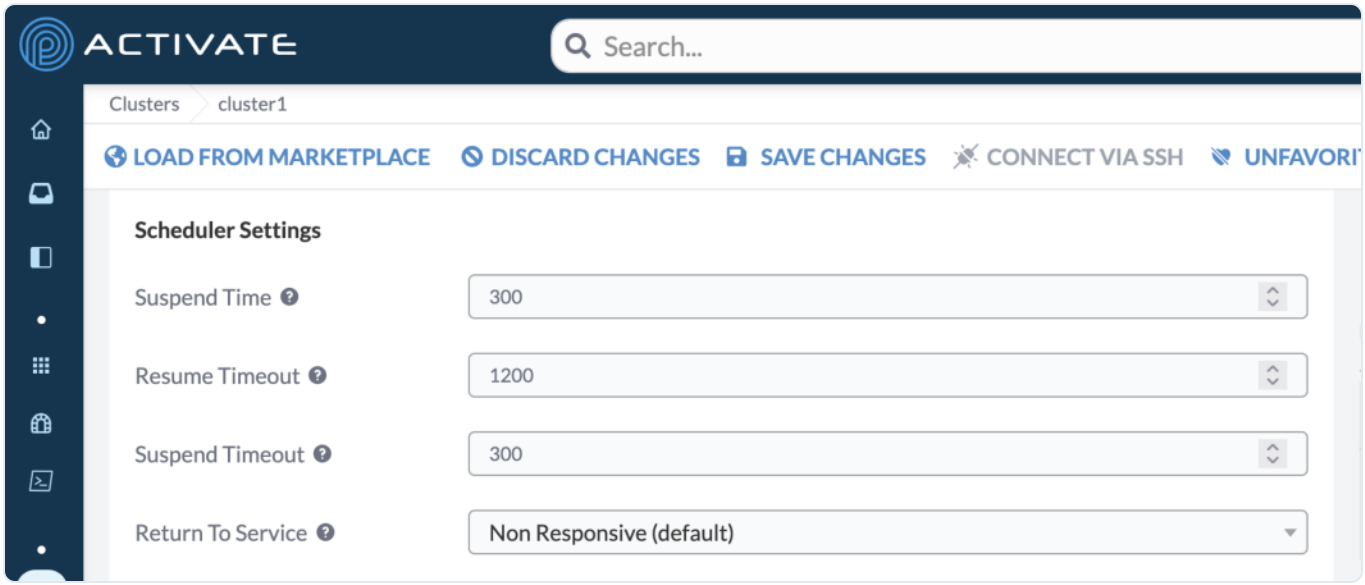
Use this text box to set a script that runs a health check on a controller node. When the script is done running, you'll see any error codes in red or an exit code of `0` in green if there are no errors.

For more information, see **Health Checks** (coming soon).

Scheduler Settings

ACTIVATE uses Slurm to manage jobs on controller and compute nodes. The settings below determine how Slurm behaves for your cluster's nodes.

Please note that numerical values you enter in these fields are measured in seconds.



Suspend Time

Use this field to set how long Slurm will wait before shutting down idle nodes. This field is set to `300` by default.

Resume Timeout

Use this field to set the maximum amount of time Slurm will try to start nodes. If the nodes don't start by the end of the set time, Slurm will end the initialization attempt. This field is set to `1200` by default.

Suspend Timeout

Use this field to set how long Slurm will wait to make nodes available again after shutting them down. This field is set to `300` by default.

Return To Service

Use this dropdown menu to select when down nodes are returned to service.

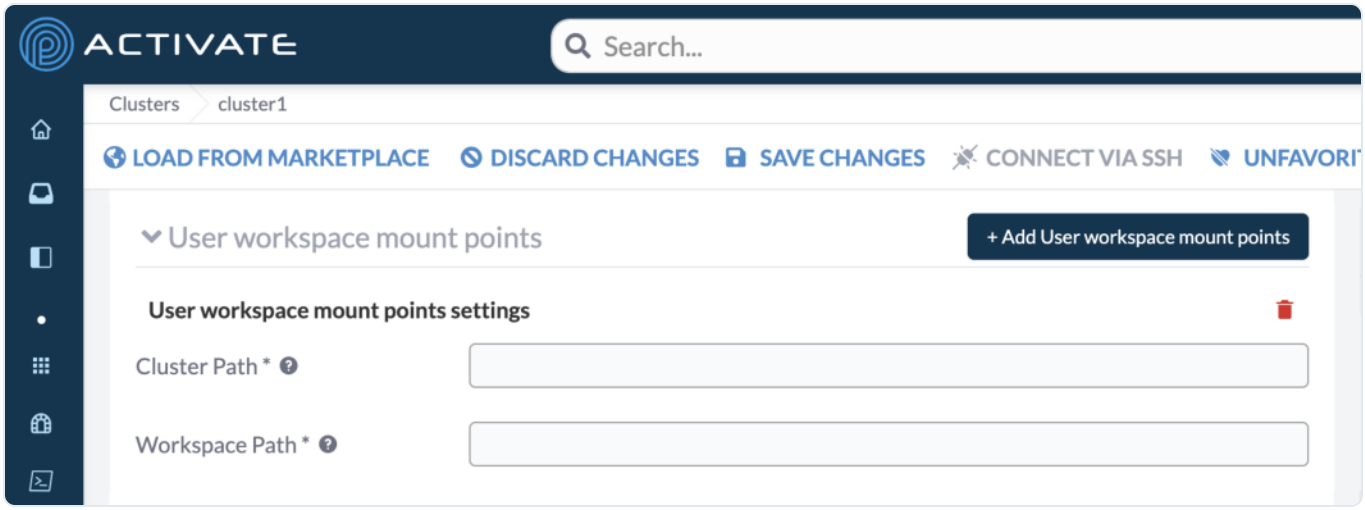
The `Non Responsive` option means that down nodes will become available only if they were set to down because they were non-responsive.

The `Any Reason` option means that down nodes will become available if they were set to down for any reason, including low memory, an unexpected reboot, or being non-responsive.

This field is set to `Non Responsive` by default.

User Workspace Mount Points

You can specify multiple mount points for your cluster, which makes it easier to view, upload, and edit cluster files in the ACTIVATE **Editor** and terminal.



Cluster Path

Use this field to enter the path that will be mounted from your cluster to your user workspace. You can use the following substitutions:

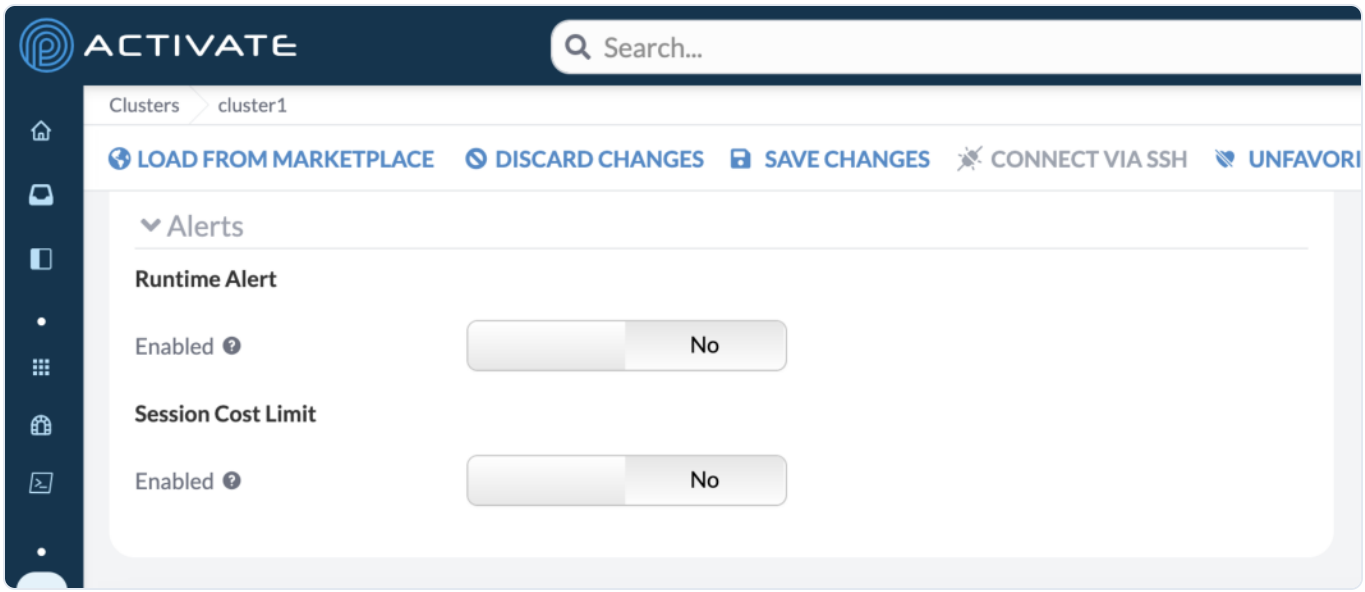
- `__HOME__` is the user home directory on your cluster.
- `__USER__` will be replaced by your cluster's remote username; this username may be the same as your ACTIVATE username in some cases.

Workspace Path

Use this field to enter the path to mount the cluster inside your user workspace. You can use the following substitutions:

- `__HOME__` is the user workspace home directory.
- `__USER__` will be replaced by your ACTIVATE username.
- `__CLUSTER__` will be replaced by your cluster's name.

Alerts



Runtime Alert

Use this toggle button to enable runtime alerts. When enabled, you'll receive an automated email from noreply@parallelworks.com at a set hourly interval, reminding you that your cluster is still running:

```
Dear {username},

This is an automated message to inform you that your cluster {cluster_name} has been running for
{time} hours. Please take note of the following details:

Cluster name: {cluster_name}
Cluster start time: {start_date}
Total run time: {total_time}
Next alert time: {next_alert}

This alert can be turned off in the resource's properties page.

Sincerely,

Parallel Works
```

Session Cost Limit

Use this toggle button to enable a session cost limit. When enabled, your cluster will be destroyed and you'll receive an automated email from noreply@parallelworks.com:

```
Dear {username},

This is an automated message to inform you that the current session for your cluster
{cluster_name} has exceeded the cost limit. Please take note of the following details:

Resource name: {cluster_name}
Resource start time: {start_date}
```

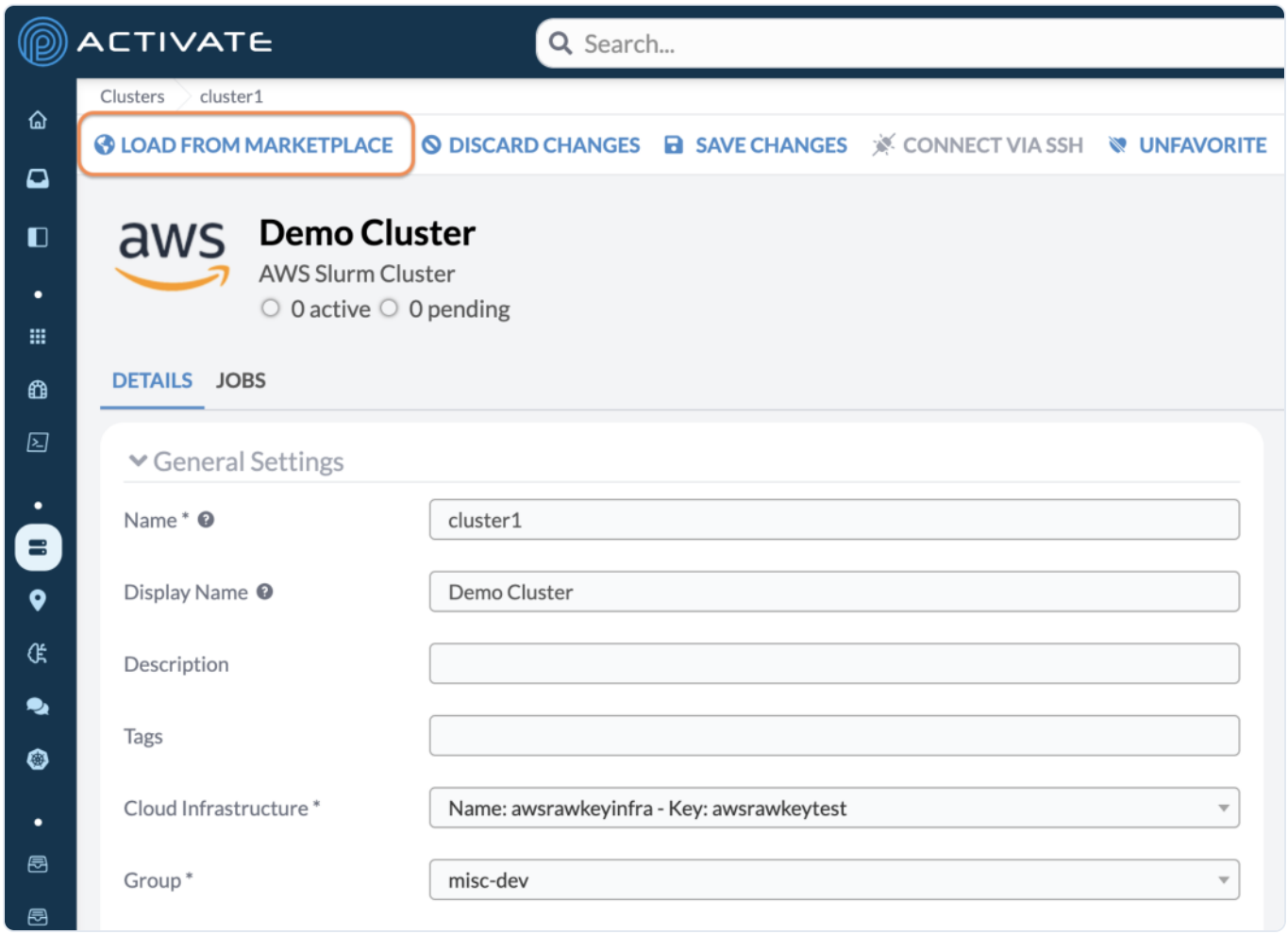
```
Total run time: {total_time}Current session cost: {current_cost}Session cost limit:
{cost_limit}Your cluster will be terminated. Data saved on this resource is not recoverable. This
automated action can be turned off in the resource's properties page under Session Cost
Limit.Sincerely,Parallel Works
```

Load From Marketplace

You can load pre-configured settings from existing resources in the Marketplace.

First, add an item from the Marketplace.

Next, create a new resource or navigate to an existing resource's settings. Click **Edit**, then **Load From Marketplace**.



After you click **Load From Marketplace**, a dialog box will appear with more information: *This will change the cluster settings to a configuration from the Marketplace. Restoring will overwrite all changes you have made to these sections. Filesystems are configured separately. For more information, see the documentation*.

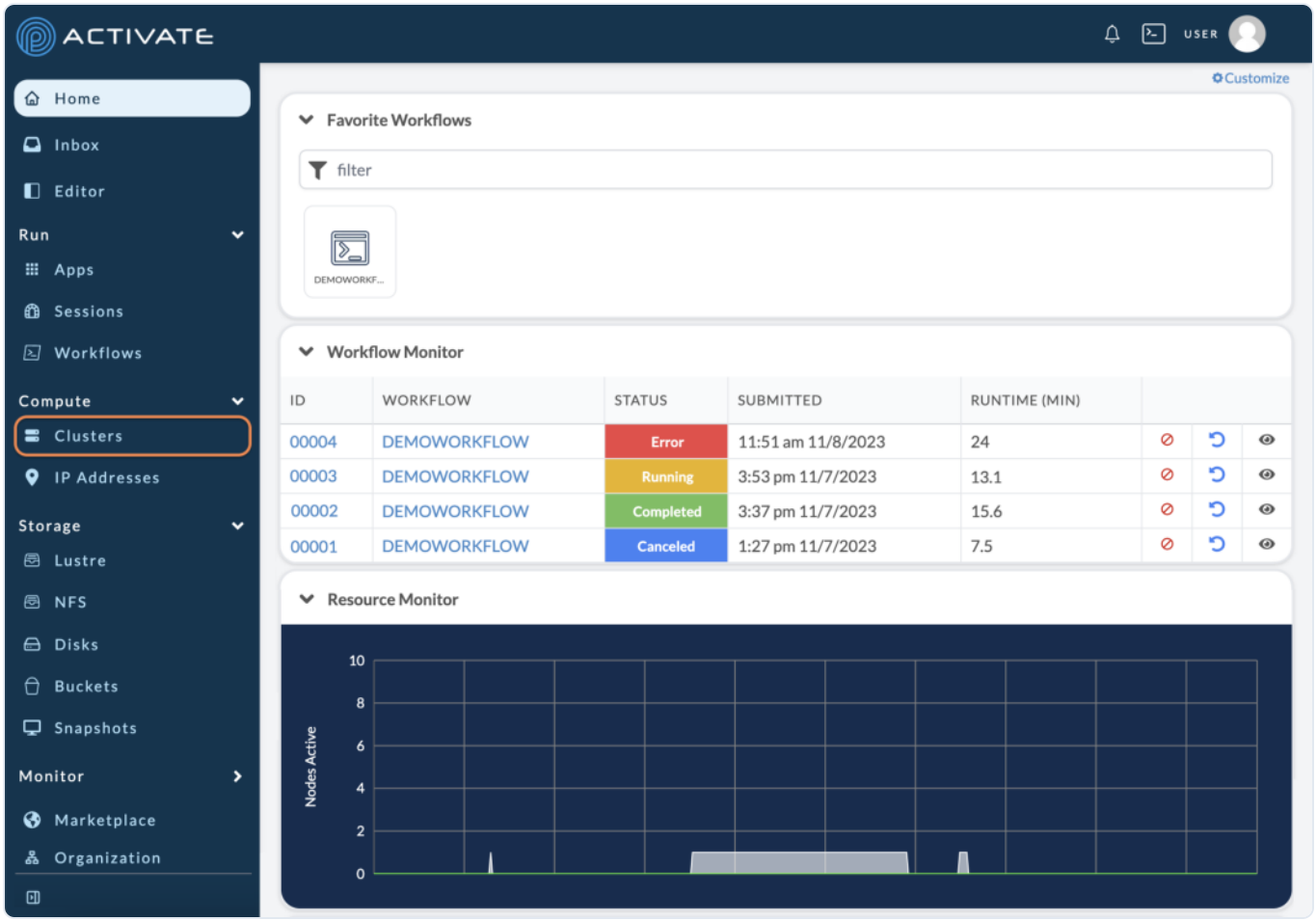
Choose your resource from the dropdown menu, then click **Restore**.

Configuring Existing Clusters

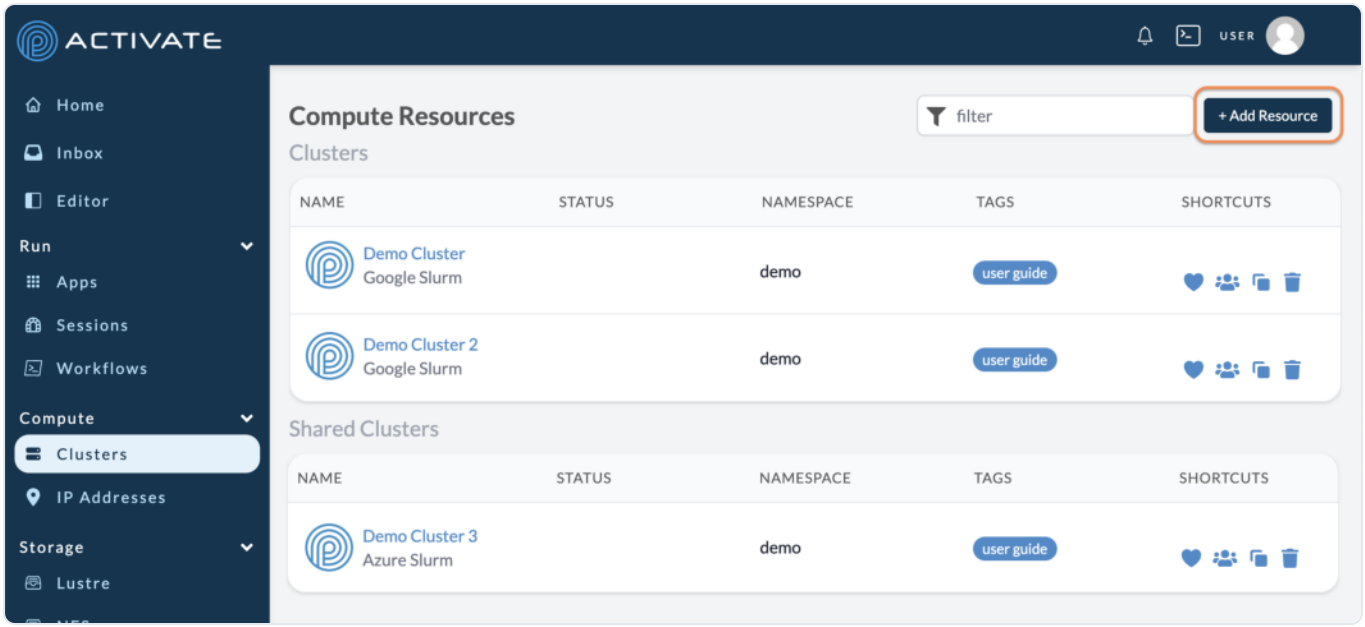
ACTIVATE allows users to access the capabilities of an existing HPC cluster remotely.

Creating a Cluster Definition

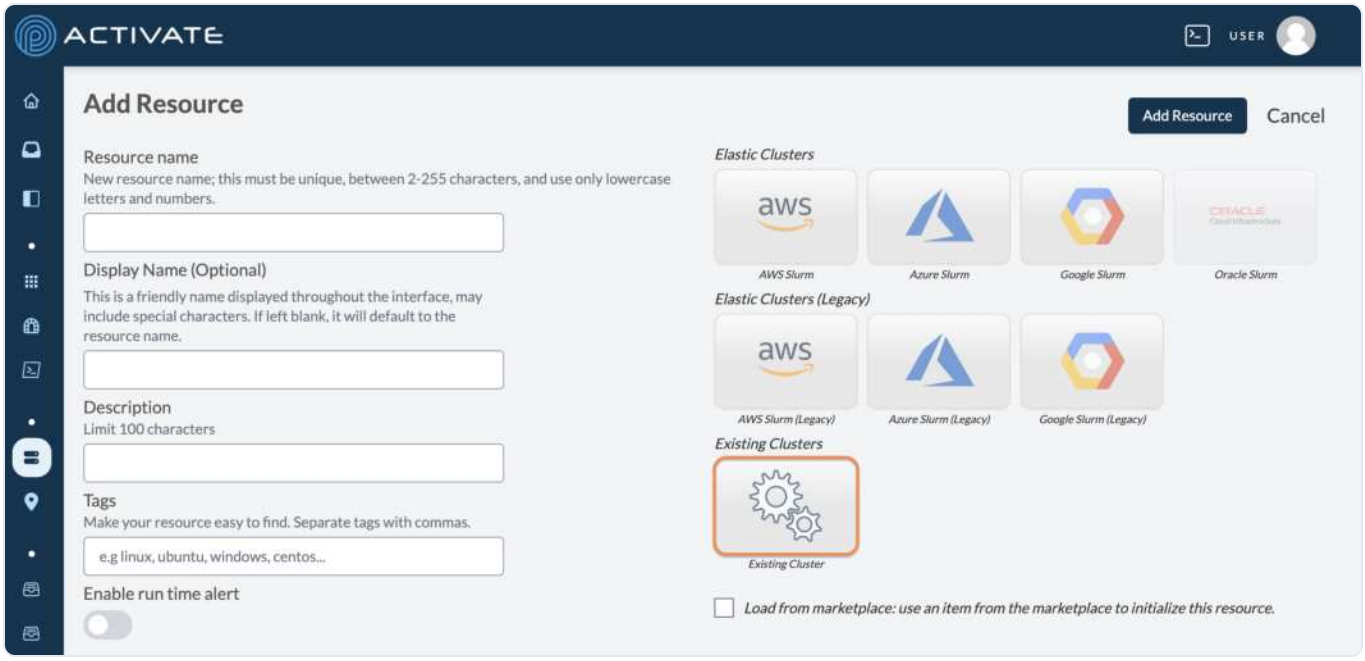
Navigate to the **Clusters** page under the **Compute** tab.



Click the + **Add Resource** button

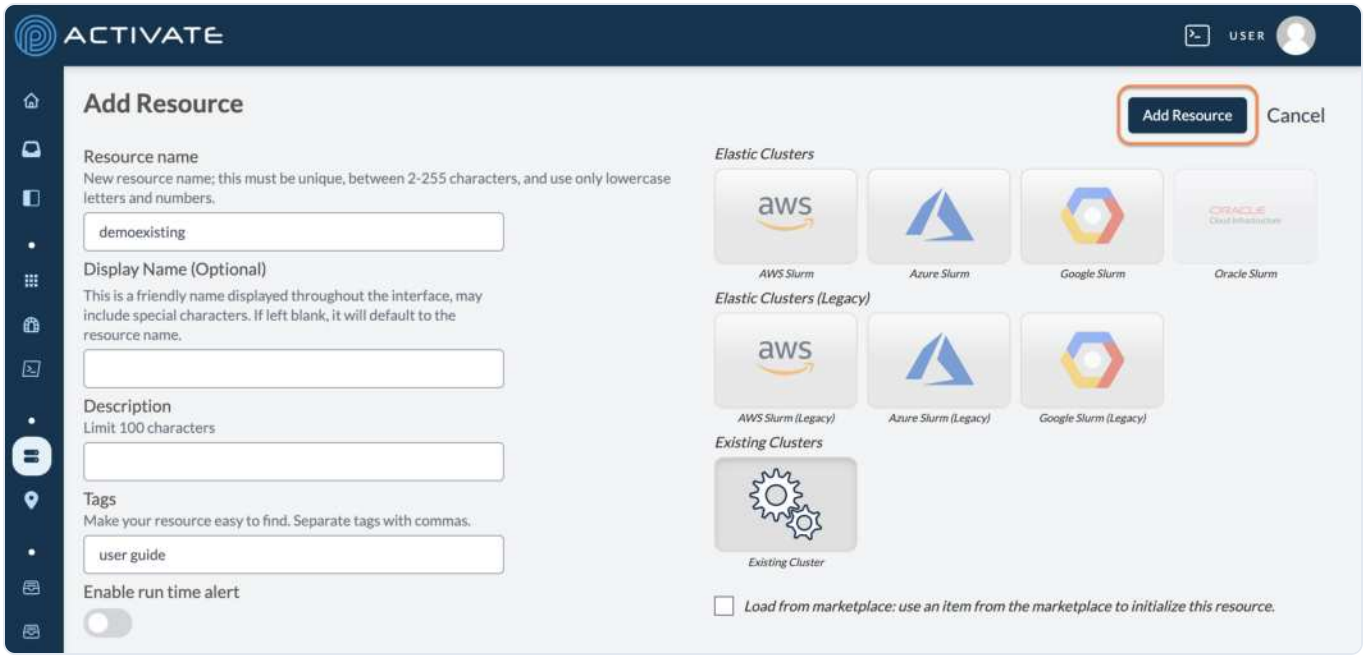


Select **Existing Cluster**.

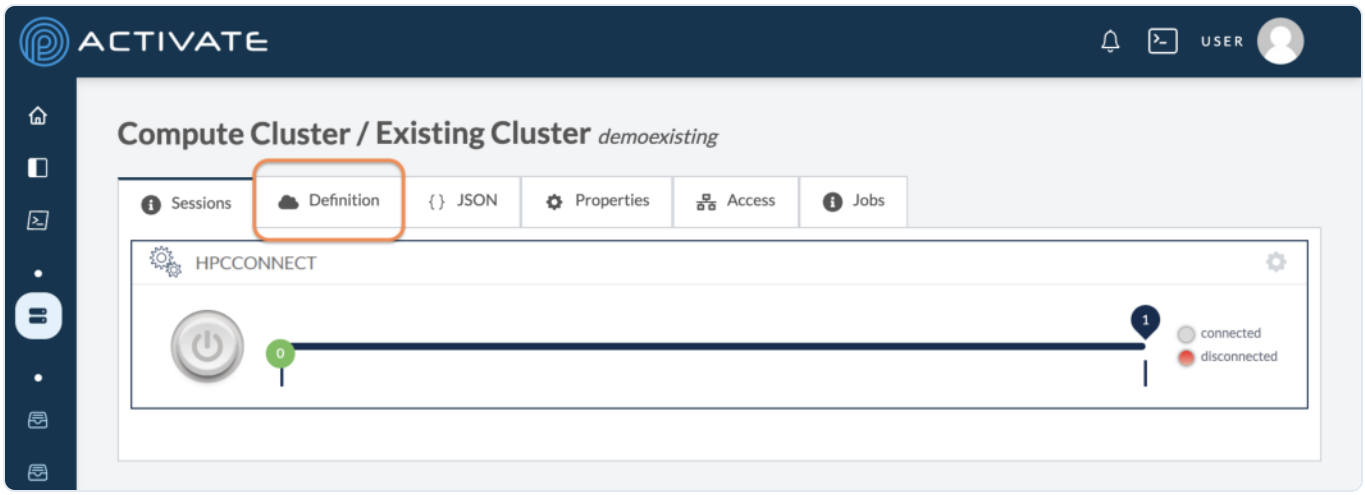


You must enter a **Resource Name**. The **Display Name**, **Short description**, and **Tags** are all optional.

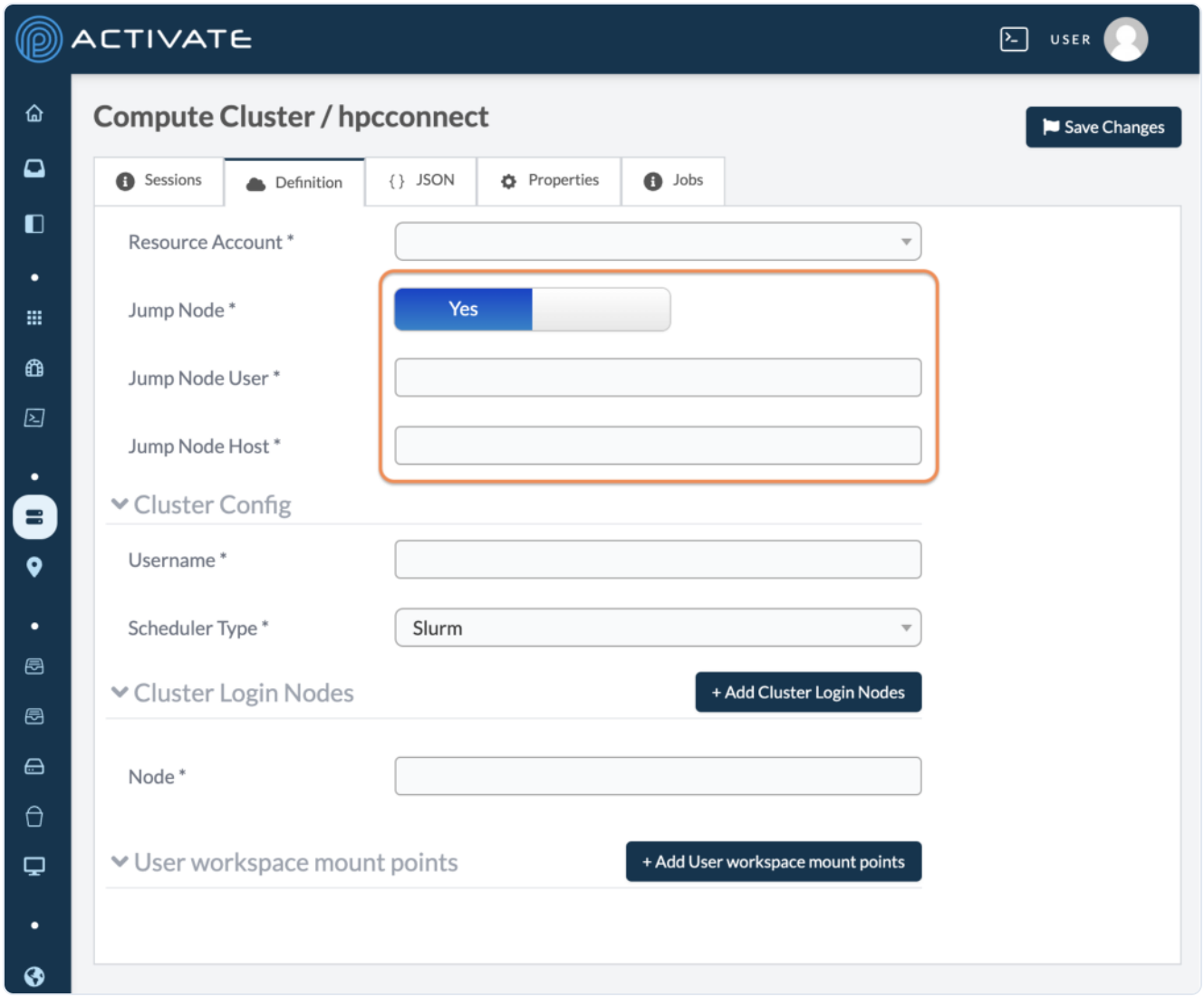
Click the **Add Resource** button.



On the next page, click the **Definition** tab.

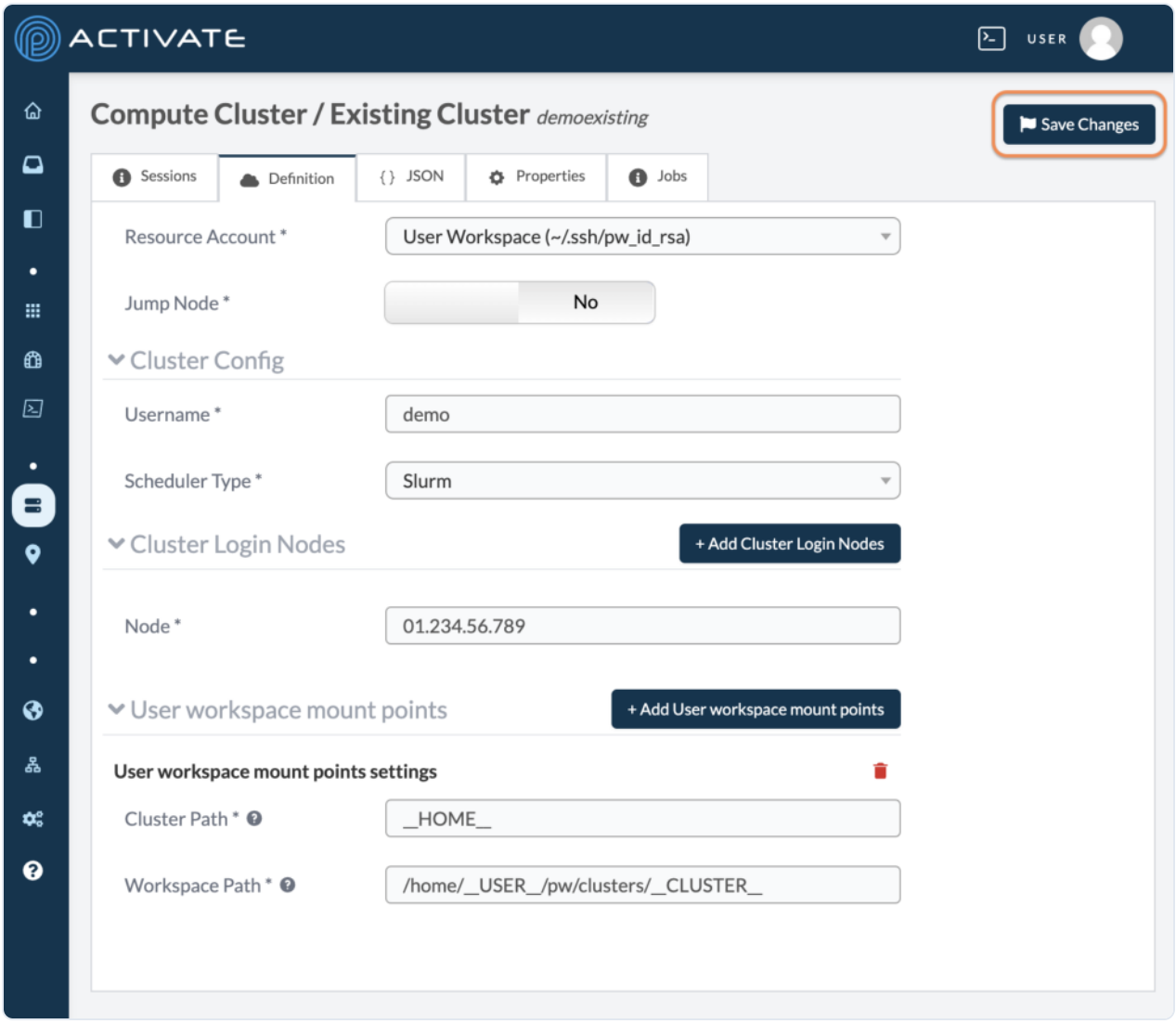


If the existing cluster you wish to connect to utilizes a jump host, select **Yes** for the **Jump Node** option. Enter the proper **Jump Node User** and **Jump Node Host** information to continue.



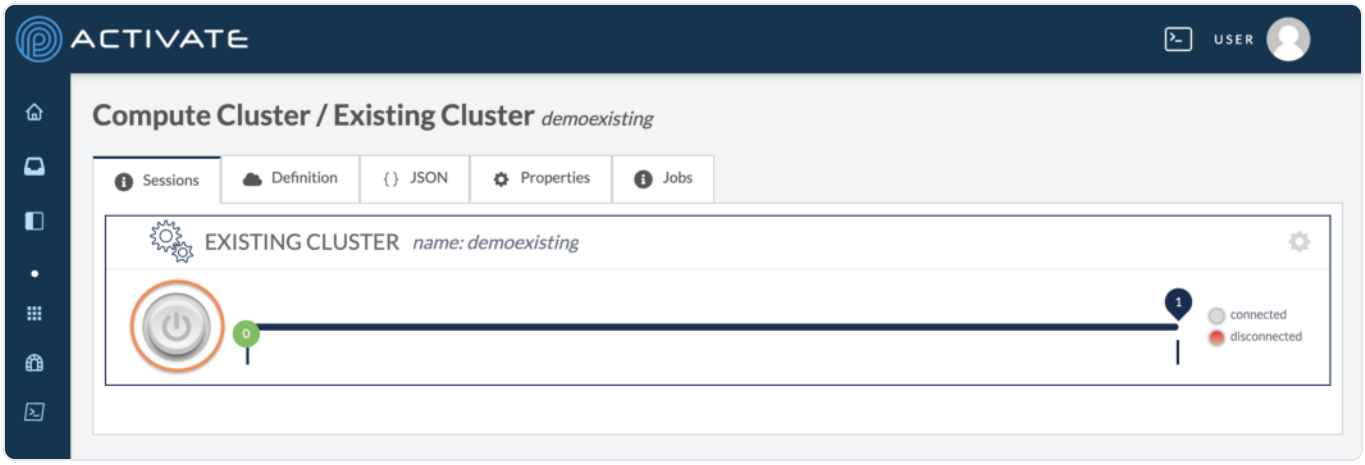
In the **Cluster Configuration** section, enter the **Username** and **Cluster Login Node**. Typically, the **Cluster Login Node** refers to the IP address of the cluster you wish to access. Your organization should have provided this information to you.

Click **Save Changes**.

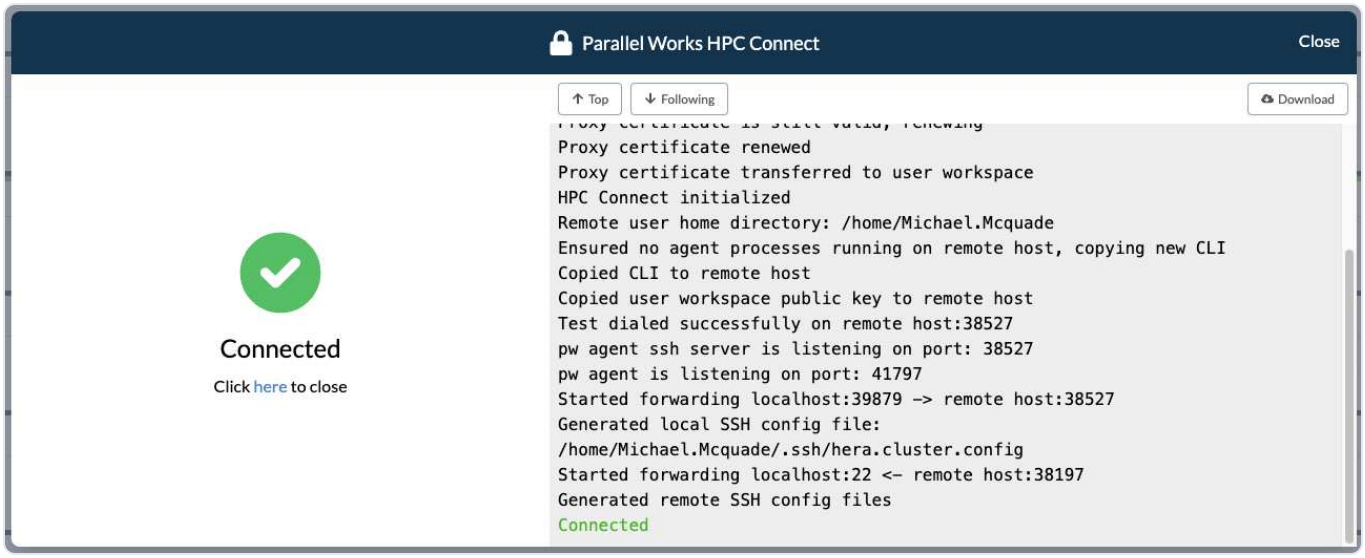


Connecting to the Cluster

After configuring the cluster, navigate to the **Sessions** tab and click the power button.



A dialog box will appear that will show HPC Connect authenticating your connection.



If your organization uses multi-factor authentication, you will be prompted to enter a token. This may be an RSA token, YubiKey, or other type of token that's specific to your organization.

This prompt will also appear if you use a passphrase protected key instead of the default workspace key.

If no token is required, you will be connected to the cluster immediately.

Once you're logged in to the cluster, you can run workflows as well as browse, edit, and upload files using the **Editor**.

Disconnecting

After you've finished utilizing the cluster, click the power button in the **Sessions** tab. A dialog box will appear. Click **Turn Off**.

A notification will appear in the bottom right corner of your screen with the message *Stopping [cluster name]*.

Configuring Settings

Typically when you create an existing cluster, you'll be connecting to an on-premises cluster associated with your organization. However, the instructions here for existing clusters can be used with any pre-created Slurm cluster. The settings that are specific to this type of cluster are outlined below. If you're unsure what to choose for these options, contact your organization's ACTIVATE administrator.

General Settings

Resource Account

Use this dropdown menu to select how ACTIVATE will connect to the existing cluster.

The `User Workspace (~/ssh/pw_id_rsa)` options means that ACTIVATE will try to SSH to the cluster by using only your account's SSH key, which is stored at `~/.ssh/pw_id_rsa`. For more information about your SSH key, see our documentation.

Jump Node

Use this toggle button if you're connecting to a cluster that has a jump node enabled. A jump node—also called a host node, bastion node, or login node—is a high-security server that allows a user to access a private machine or network.

If you enable this feature, two new fields will appear for **Jump Node User** and **Jump Node Host**. Your organization will have these credentials if you need them.

Cluster Configuration Settings

Username

Use this field to enter the username assigned to you for this cluster.

> **USERNAME SUBSTITUTION**
>
> On existing clusters, you can enter `__USER__` into any box and ACTIVATE will automatically substitute your username for that field. For example, if your username is `jdoe`, ACTIVATE will automatically substitute `__USER__` for `jdoe` in the **Working Directory** field.

Cluster Login Node

Use this field to enter the IP address or host name of the cluster.

Scheduler Type

Use this dropdown menu to select the type of job scheduler the cluster uses. Currently, the **Existing Cluster** resource type supports Slurm and PBS.

Starting and Stopping Clusters

This page explains how to start, stop, and destroy clusters. For more information about these actions, please see this page.

Starting Clusters

On your **Home** page, navigate to the **Compute** module. Click the power button of the cluster you'd like to start.

You'll see the message *Starting [Cluster Name]*. While the cluster starts, the power button will flash green, and the **requested** status bubble will turn yellow.

When your cluster is provisioned, both the power button and the **active** status bubble will turn green.

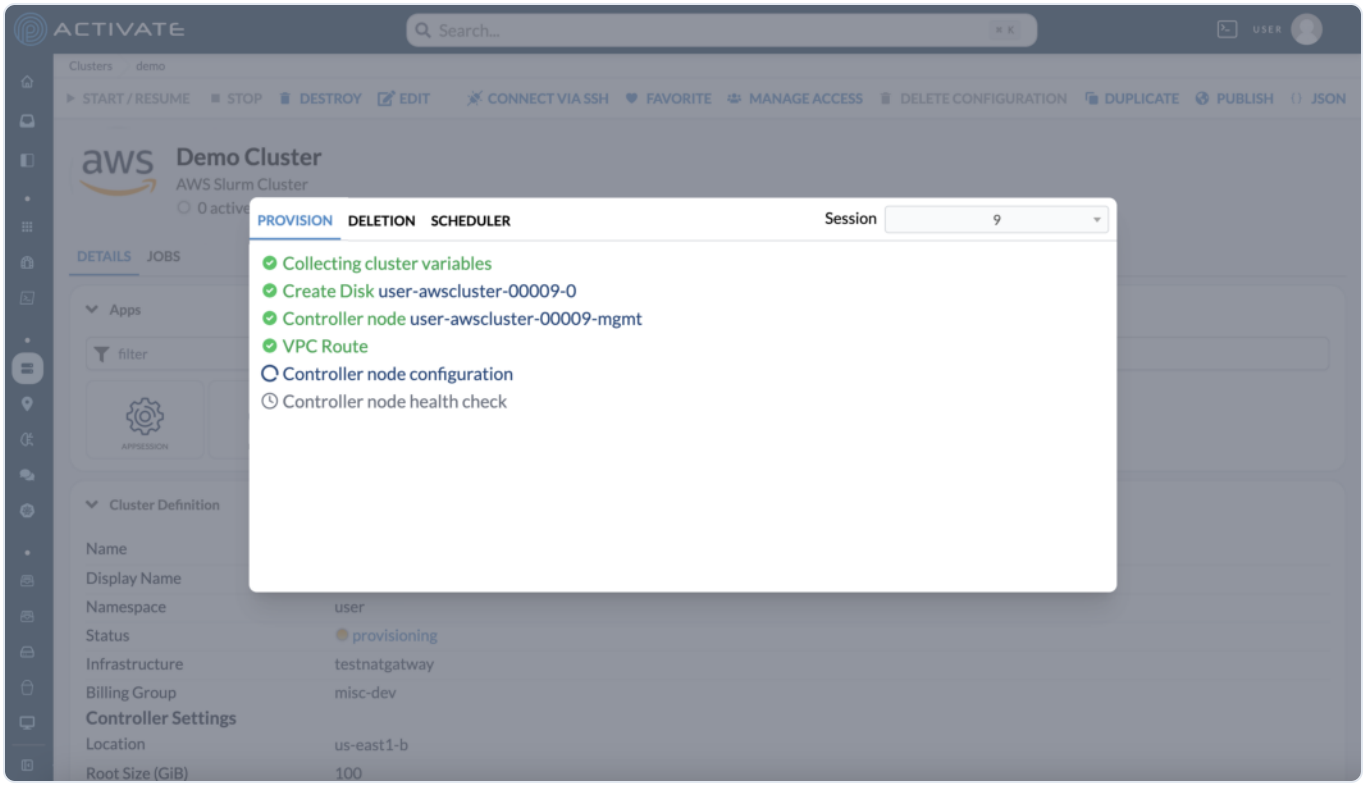
Alternatively, you can navigate to your cluster's page and click ** Start/Resume**. The cluster's **Status** will change to **provisioning**.

> **NOTE**
>
> It may take up to 5 minutes for a cluster to start.

Monitoring Status

While your cluster is starting, you can see its status in the provisioning logs. Go to **Clusters** > your cluster > **Details** and click the cluster's **Status**.

- Steps that have completed are green with a checkmark icon.
- Ongoing steps are blue with a loading icon.
- Upcoming steps are gray with a clock icon.
- If a cluster fails to start, the step at which it failed will be red with an error icon.



Stopping Clusters

When the cluster's controller is stopped, you can resume it anytime by pressing **Start / Resume**. Please note that you cannot stop a cluster if there are any active compute nodes.

To stop a cluster, navigate to its page and click ** Stop**.

Destroying Clusters

When you're ready to deprovision a cluster, navigate to the **Home** page and click the power button. A dialog box will appear with the message *Are you sure you want to turn off [Cluster Name]?*

Click **Turn Off** to stop the cluster. You'll see the message *Stopping [Cluster Name]*.

Alternatively, you can navigate to the cluster's page and click ** Destroy**.

When you destroy a cluster, any attached nodes and ephemeral storage resources will also be terminated. Only the data in persistent storage resources will remain. It’s important to copy any data you’d like to keep to persistent storage or to a remote location.

> **NOTE**
>
> It’s essential to destroy a cluster when you’ve finished your work. Clusters that run unmonitored continue to accrue additional charges, which will be subtracted from your organization’s allotment.

Connecting CoreWeave Slurm

This guide connects a CoreWeave Slurm cluster (SUNK - Slurm on Kubernetes) to ACTIVATE so your users can submit batch jobs, open desktop sessions, and run workflows on it.

There are two sides to it:

1. **Identity** - point the cluster's identity cache (`nsscache`) at ACTIVATE's SCIM API and add an SSH authorized-keys command, so ACTIVATE users, groups, and SSH keys resolve as real Linux accounts on the cluster.
2. **Connection** - register the cluster in ACTIVATE as an existing cluster.

> **CONNECTING COREWEAVE'S KUBERNETES API INSTEAD?**
>
> If you want to manage the cluster's Kubernetes workloads through ACTIVATE rather than submit Slurm jobs, see Connecting CoreWeave (Kubernetes).

Prerequisites

- **Organization admin permissions** in ACTIVATE.
- **SCIM provisioning enabled** for your organization, plus a **bearer token**. Follow SCIM Provisioning first and keep the token and endpoint URL handy.
- **`kubectl` access** to the cluster's `tenant-slurm` namespace (via the kubeconfig from CoreWeave).
- POSIX UIDs/GIDs and SSH public keys configured on your ACTIVATE users and groups - these are what get synchronized onto the cluster.

Point nsscache at ACTIVATE's SCIM API

CoreWeave's SUNK clusters resolve Linux identity through `nsscache` , which periodically syncs `passwd` , `group` , `shadow` , and `sshkey` maps from a source. We configure that source to be ACTIVATE's SCIM API, reading POSIX identity from the CoreWeave extension attributes.

Edit the `nsscache-conf` ConfigMap in the `tenant-slurm` namespace:

```
kubectl edit cm/nsscache-conf -n tenant-slurm
```

Update it to match the following. The two values you must set for your organization are **`scim_base_url`** (your SCIM endpoint) and **`scim_users_parameters`** (which requests the CoreWeave user extension):

```
apiVersion: v1
data:
  nsscache.conf: |
    [DEFAULT]
    cache=files
    files_cache_filename_suffix=cache
    files_dir=/etc/nsscache
    maps=passwd,shadow,group,sshkey
    scim_base_url=https://<platform-host>/api/organizations/<organization>/scim/v2
```

```
    scim_groups_endpoint=Groups    scim_groups_parameters=excludeInactiveUsers=true
scim_users_endpoint=Users
scim_users_parameters=attributes=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWea
veUser    source=scim    timestamp_dir=/var/lib/nsscache    [group]
scim_path_gid=sunkPosixGroupId    scim_path_groupname=sunkPosixGroupName
scim_path_username=members/sunkPosixUsername    [passwd]    scim_default_shell=/bin/bash
scim_override_home_directory=/mnt/home/%%u
scim_path_gid=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunkPosixGro
upId
scim_path_home_directory=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/s
unkPreferredHomeDirectory
scim_path_login_shell=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunk
LoginShell
scim_path_uid=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunkPosixUse
rId
scim_path_username=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunkPos
ixUsername    [shadow]
scim_path_username=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunkPos
ixUsername    [sshkey]
scim_path_ssh_keys=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunkSsh
Keys
scim_path_username=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser/sunkPos
ixUsername  nsswitch.conf: |    group: files cache    passwd: files cache
```

What the key settings do:

- **`scim_base_url`** - your organization's SCIM endpoint, shown on the SCIM Provisioning page (`https://<platform-host>/api/organizations/<organization>/scim/v2`).
- **`scim_users_parameters=attributes=...CoreWeaveUser`** - requests the CoreWeave user extension. ACTIVATE omits that block by default, so without this parameter the POSIX UID/GID, shell, home directory, and SSH keys would be missing.
- **`scim_groups_parameters=excludeInactiveUsers=true`** - drops disabled ACTIVATE accounts from group membership, so deactivated users stop resolving on the cluster.
- **`scim_override_home_directory=/mnt/home/%%u`** - forces home directories under `/mnt/home`. This overrides the `sunkPreferredHomeDirectory` value from SCIM; set it to wherever home directories are mounted on your cluster.

Provide the bearer token

The SCIM API requires a bearer token on every request. On a SUNK cluster, nsscache reads it from the `nsscache-scim-secret` Secret in the `tenant-slurm` namespace - not from the ConfigMap above. This Secret is provisioned with the cluster; update it with the token you minted in SCIM Provisioning:

```
kubectl edit secret nsscache-scim-secret -n tenant-slurm
```

Secret values are base64-encoded, so encode the token before pasting it into the Secret's data field:

```
printf '%s' '<your-scim-token>' | base64
```

Configure the authorized keys command

So that `sshd` can authorize logins using each user's ACTIVATE SSH keys, install an `AuthorizedKeysCommand` that fetches them through the `pw` CLI.

Save the following as `slurm-nsscache-authorized-keys-command.yaml`, **setting `PLATFORM_HOST`** to your platform URL:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: slurm-nsscache-authorized-keys-command
  namespace: tenant-slurm
data:
  # Filename kept as .py for drop-in compatibility with the existing
  # AuthorizedKeysCommand path in sshd_config. The shebang determines the
  # interpreter, so bash content here is fine.
  nsscache-authorized-keys-command.py: |
    #!/bin/bash
    # AuthorizedKeysCommand: fetch the user's SSH public keys via the pw CLI.
    # Installs pw on first invocation; subsequent calls reuse the cached binary.
    set -e

    PLATFORM_HOST="${PLATFORM_HOST:-https://<platform-host>}"
    PW_INSTALL_DIR="${PW_INSTALL_DIR:-/usr/local/bin}"
    PW_BIN="$PW_INSTALL_DIR/pw"
    # /tmp is always writable, even by `nobody`. The lock only needs to exist
    # during one install attempt, so ephemeral storage is fine.
    INSTALL_LOCK="${PW_INSTALL_LOCK:-/tmp/pw-install.lock}"

    locate_pw() {
        if [ -x "$PW_BIN" ]; then
            return
        fi
        local found
        found="$(command -v pw 2>/dev/null || true)"
        if [ -n "$found" ] && [ -x "$found" ]; then
            PW_BIN="$found"
        fi
    }

    locate_pw
    if [ ! -x "$PW_BIN" ]; then
        # flock prevents concurrent sshd invocations from racing the install.
        (
            flock -x 9
            if [ ! -x "/usr/local/bin/pw" ] && ! command -v pw >/dev/null 2>&1; then
                # Send install output to stderr so it doesn't end up in the
                # keys stream sshd reads from stdout.
                curl -fsSL https://activate.parallel.works/cli/install.sh \
                    | bash -s -- --to "$PW_INSTALL_DIR" 1>&2
            fi
        ) 9>"$INSTALL_LOCK"
        locate_pw
```

```
    fi    if [ ! -x "$PW_BIN" ]; then        echo "pw CLI not found and install failed" >&2
exit 1    fi    # Validate username contains only safe characters.    if [[ ! "$1" =~ ^[a-zA-Z0-
9._-]+$ ]]; then        exit 1    fi    exec "$PW_BIN" ssh-public-keys --platform-host
"$PLATFORM_HOST" "$1"
```

Apply it:

```
kubectl apply -f slurm-nsscache-authorized-keys-command.yaml
```

A running login pod won't pick up the new ConfigMap until it restarts. Delete the Slurm login pod so it's recreated with the updated command mounted (find it with `kubectl get pods -n tenant-slurm`):

```
kubectl delete pod -n tenant-slurm <login-pod>
```

On the first SSH login after the pod comes back, the script installs the `pw` CLI if it isn't already present, then calls `pw ssh-public-keys` to return the user's keys for `sshd` to authorize.

Confirm it works by execing into the recreated login pod and running the command with a username:

```
kubectl exec -it -n tenant-slurm <login-pod> -- \
  /usr/local/share/nsscache-authorized-keys-command.py <username>
```

When everything is wired up correctly, it prints that user's authorized SSH public keys - for example, `nsscache-authorized-keys-command.py mcquade` returns mcquade's keys.

Register the cluster in ACTIVATE

With identity resolving on the cluster, connect it like any other on-premises cluster:

1. Follow Configuring Existing Clusters to create the cluster definition.
2. Set the **Scheduler Type** to **Slurm**.
3. Enter the **Cluster Login Node** (the cluster's login/jump host) and your **Username**.

You can use the `__USER__` token in any field and ACTIVATE substitutes the logged-in user's username automatically.

Verify

First, confirm the identity cache is populating. nsscache writes each synced map into a `slurm-nsscache-<map>` Secret in `tenant-slurm`. Decode the `passwd` cache to check that your ACTIVATE users are landing on the cluster with the expected UID, GID, shell, and home directory:

```
kubectl get secret slurm-nsscache-passwd -n tenant-slurm -o yaml \
  | yq '.data."passwd.cache"' | base64 -d
```

The `group`, `shadow`, and `sshkey` maps populate the parallel `slurm-nsscache-group`, `slurm-nsscache-shadow`, and `slurm-nsscache-sshkey` Secrets (keyed `group.cache`, `shadow.cache`, and `sshkey.cache`).

An empty or stale cache usually means the SCIM URL, the bearer token, or the `attributes` parameter is wrong.

Then confirm the end-to-end connection:

1. From the **Sessions** tab, power on the cluster and confirm the connection succeeds.
2. Confirm your account resolves on the cluster (`id <username>` should show the POSIX UID/GID synced from ACTIVATE).
3. Submit a test job and confirm it runs. See Submitting Jobs via Slurm.

Logging In to the Controller

Clusters use one node, called a controller, to delegate tasks to compute nodes so they can carry out commands and complete jobs. You can complete many tasks on ACTIVATE after logging in to the controller, such as submitting jobs and transferring data.

There are multiple ways to log in to the controller.

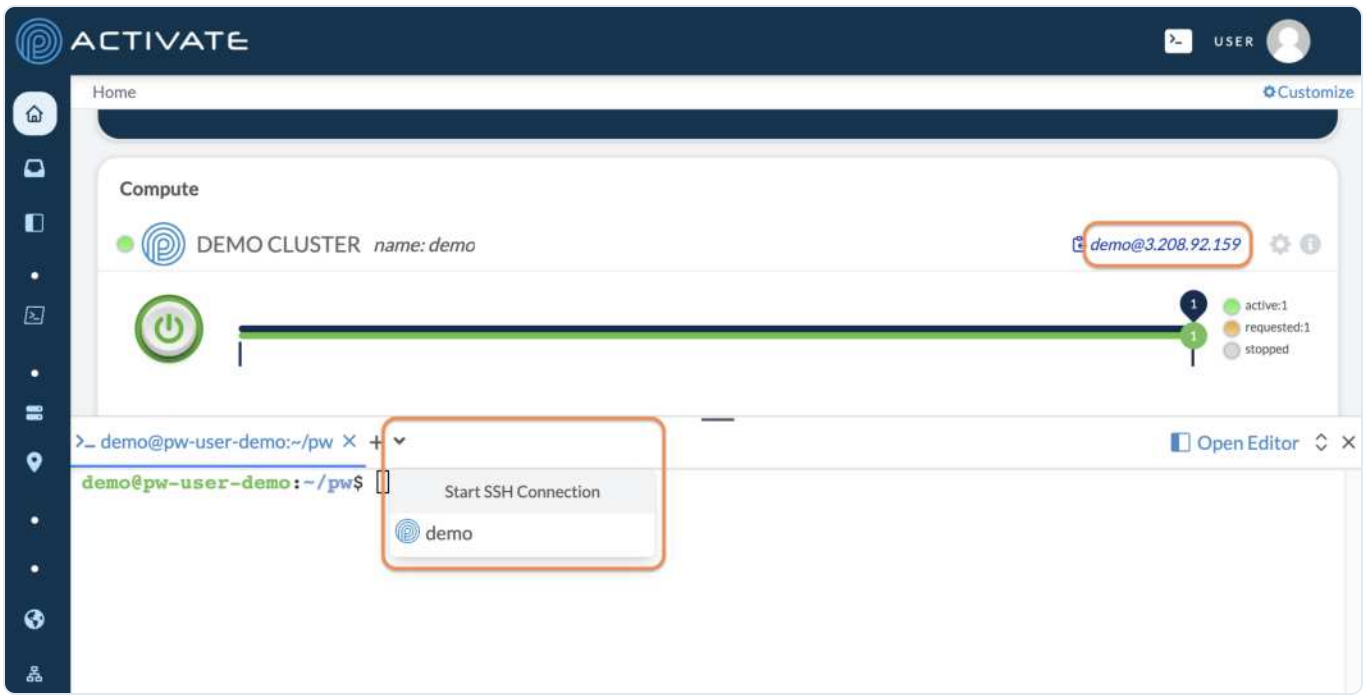
Within the Platform

One-Click SSH Connection

After you've started a cluster, navigate to the **Home** page.

In the **Compute** module, click on the cluster's IP address.

Alternatively, you can click the terminal icon, then click the arrow icon. A dropdown menu of your active clusters will appear. Click the one you'd like to SSH into.



A terminal will open on the bottom half of your screen, where you'll be automatically logged in to your cluster's controller node.

Manual SSH Connection

After you've started a cluster, navigate to the **Home** page.

In the **Compute** module, click on the clipboard next to the running cluster's IP address to copy it.

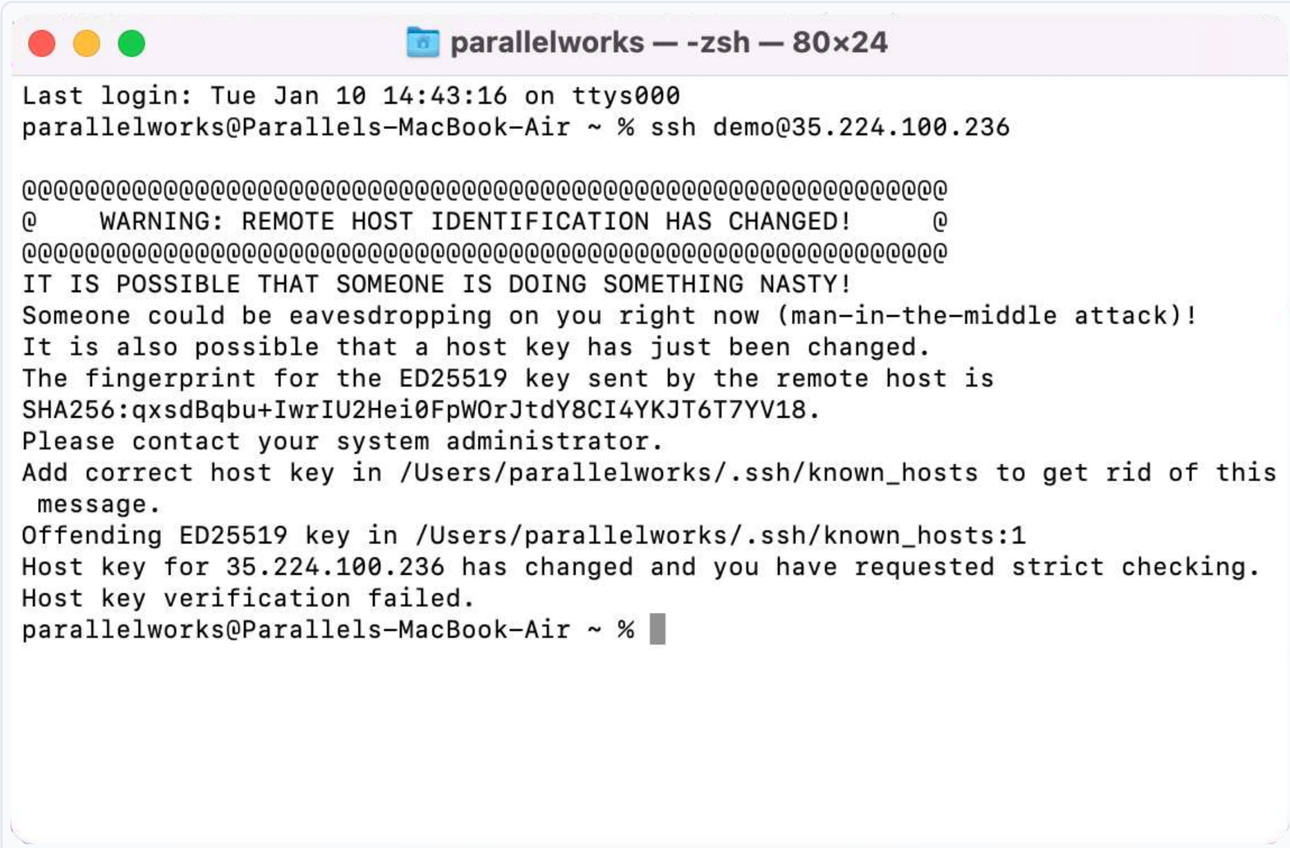
Open the terminal .

Enter the command `ssh username@IPaddress`. The terminal will display your last login and location.

NOTE

Each time you start a cluster, the controller will be assigned a random IP address from the cloud provider's available addresses. If you stop the cluster, this IP address is released automatically and will most likely not be the same when you start the cluster again.

Although it's unlikely that you'll encounter the same cluster IP address, it may occur if you start and stop a cluster frequently. You may receive the following message: `WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!`.



To resolve the issue, enter the command `ssh-keygen -R controllerIPaddress`. You should see the following message:

```
[demo@pw-user-demo ~]$ ~ ssh-keygen -R 35.224.100.236
# Host 35.224.100.236 found: line 137
/Users/demo/.ssh/known_hosts updated.
Original contents retained as /Users/demo/.ssh/known_hosts.old
[demo@pw-user-demo ~]$ ~ ssh canary.parallel.works
The authenticity of host '35.224.100.236' can't be established.
ED25519 key fingerprint is SHA256:KzR9SCW5QKmICeH2e5z7ZHPgAdoUOUuBNpkqj8UqSog.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Enter `yes`. You'll see a message that the IP address has been added to the list of known hosts.

```
Warning: Permanently added '35.224.100.236' (ED25519) to the list of known hosts.
Last login: Wed Jan 11 16:40:36 2023 from 104-60-190-63.lightspeed.hstntx.sbcglobal.net
[demo@pw-user-demo ~]$
```

Outside the Platform

You can add an SSH key to your ACTIVATE account, which will allow you to log in to any active cluster from any device with that SSH key on it. For more information, please see **Managing SSH Keys**.

FAQ

Why don't I have to use an SSH key from within the platform?

Each user's workspace has an SSH key preprovisioned, so logging in to the controller through the IDE doesn't require SSH key management. The public key is automatically propagated to the controller while the private key stays inside your workspace.

Whenever you start a new cluster, a new key is generated and propagated to facilitate SSH to compute nodes. This process works because the home directory is shared across all nodes in the cluster; adding your public workspace key to the `authorized_keys` file on the controller automatically allows you to SSH to the compute nodes too.

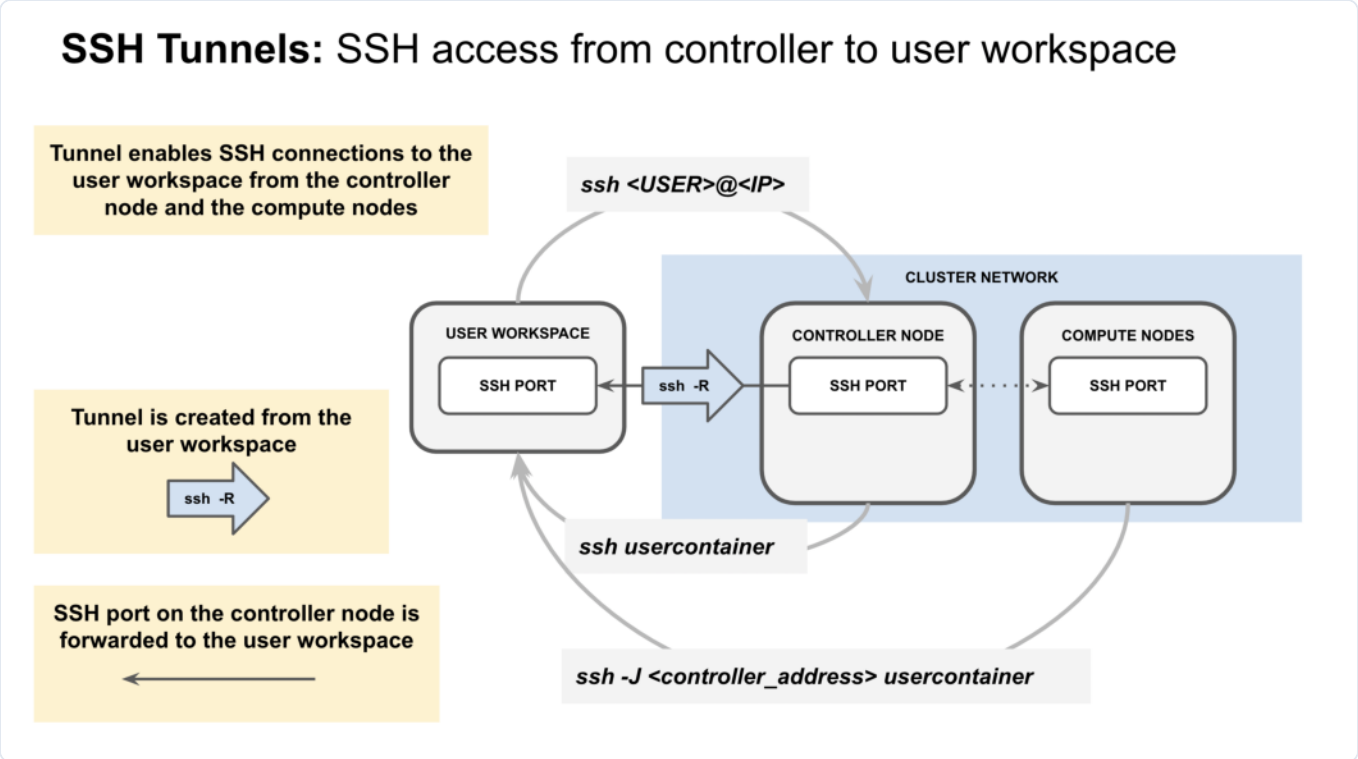
You can access your public workspace key at any time with the command `cat ~/.ssh/pw_id_rsa.pub` in an IDE terminal.

Why do I see `__ failed login attempts` the first time I log in to the controller?

These failed login attempts won't affect your cluster's performance: they're a result of how the ACTIVATE platform communicates with cloud service providers to make resources available.

During the provisioning process, the platform repeatedly tries to SSH to the controller. While the controller comes online, the platform attempts to establish a tunnel connection from the user workspace to the controller. Because the user workspace key isn't available on the system until partway through the bootstrap process, the system registers some of these connection attempts as failures.

Below is a diagram of this process.



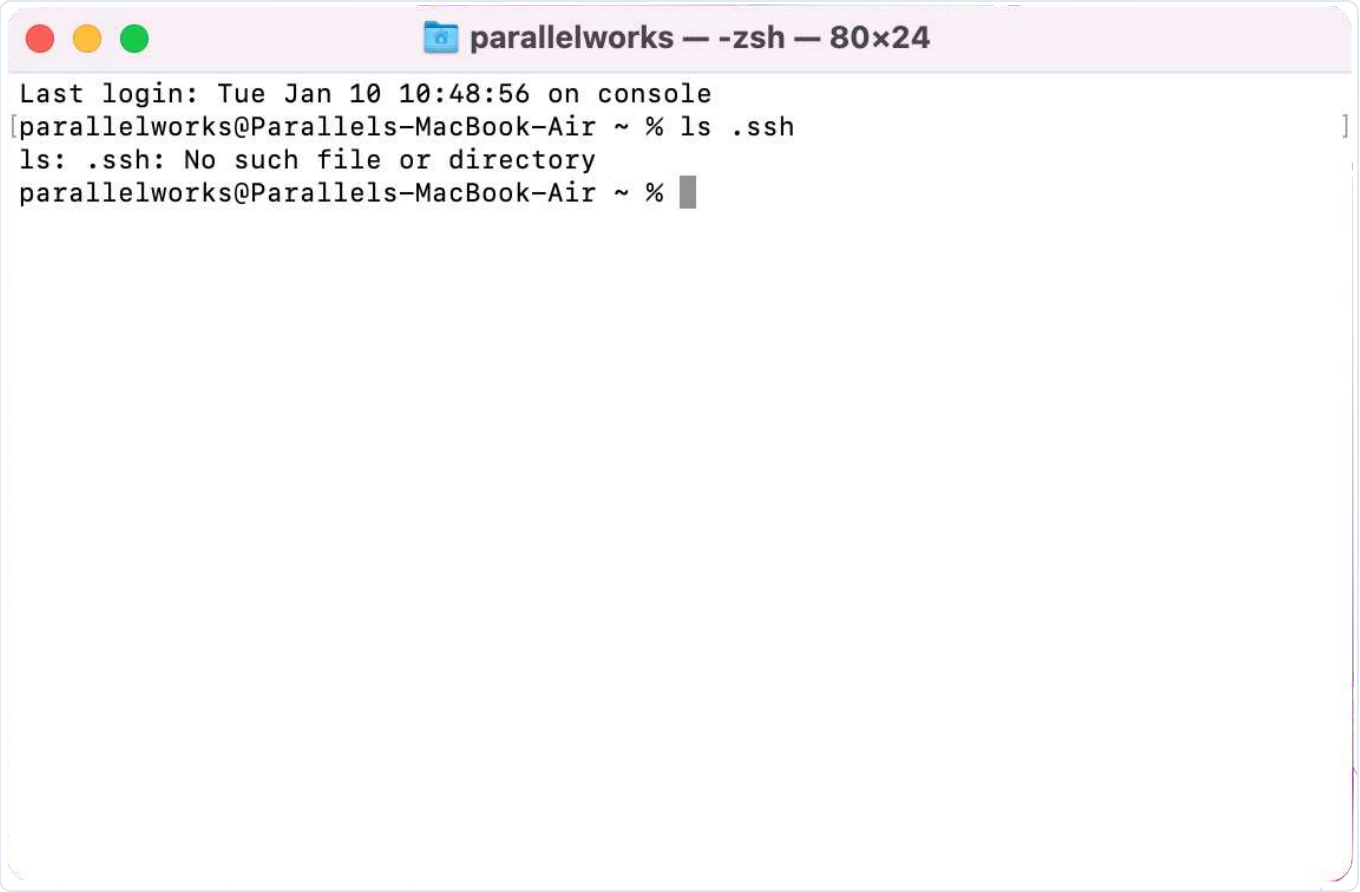
What if I don't have an SSH key?

The steps below will guide you through finding and creating SSH keys on your device for both macOS and Windows.

> **NOTE**
>
> The instructions in this section will work on Windows if you use a shell emulator like Git BASH or PowerShell.
>
> If you use PuTTY to manage your SSH keys on Windows, see **What if I use PuTTY?** below.

Check for SSH Keys

Open Terminal (macOS) or Command Prompt (Windows). Enter `ls .ssh` to check for SSH keys on your device. If you've never generated an SSH key, you'll see the message `ls: .ssh: No such file or directory`.

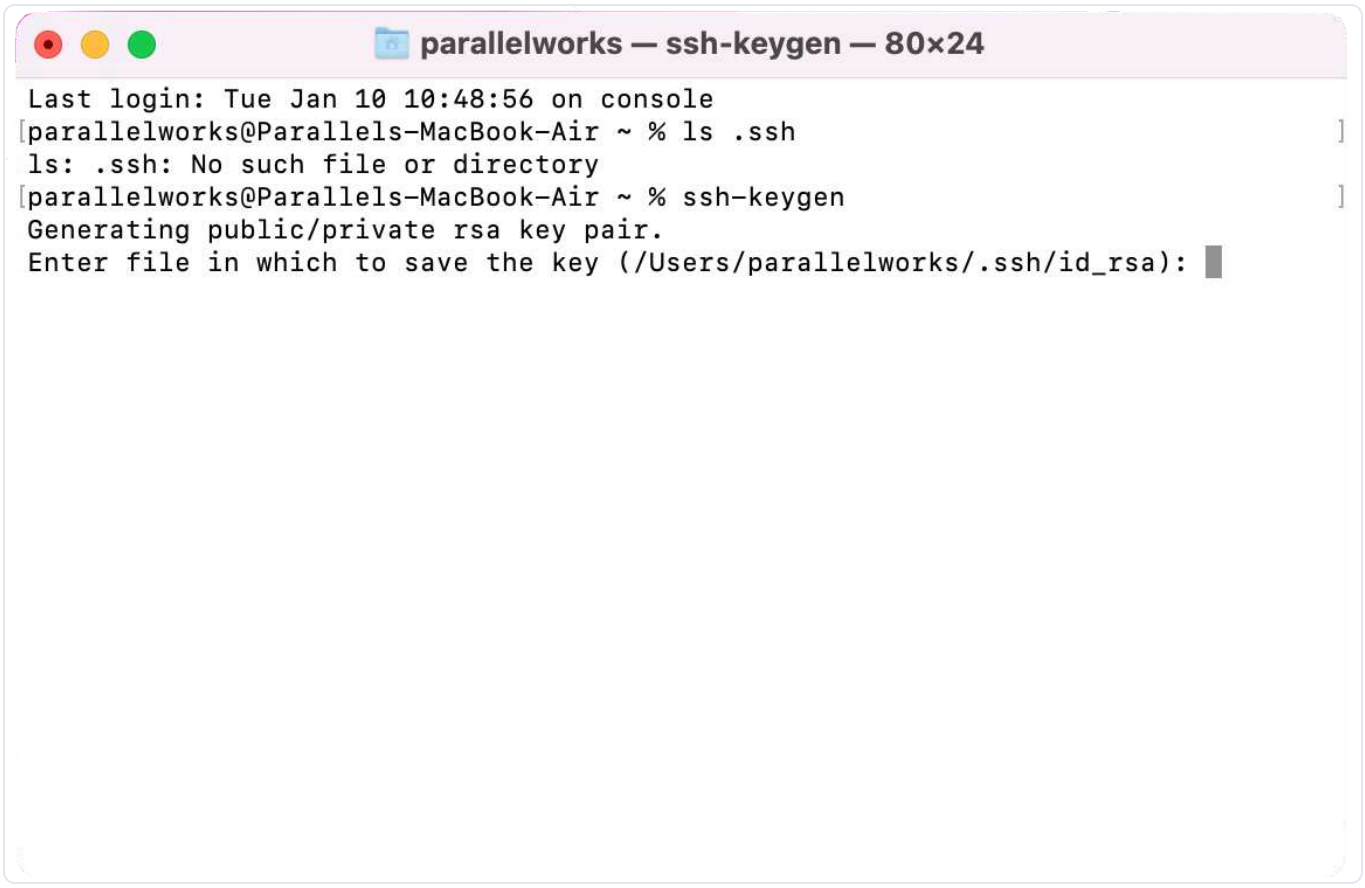


If you do have SSH keys on your device, they'll be listed. If you want to use one of your existing keys, see **Get Your Public Key** below.

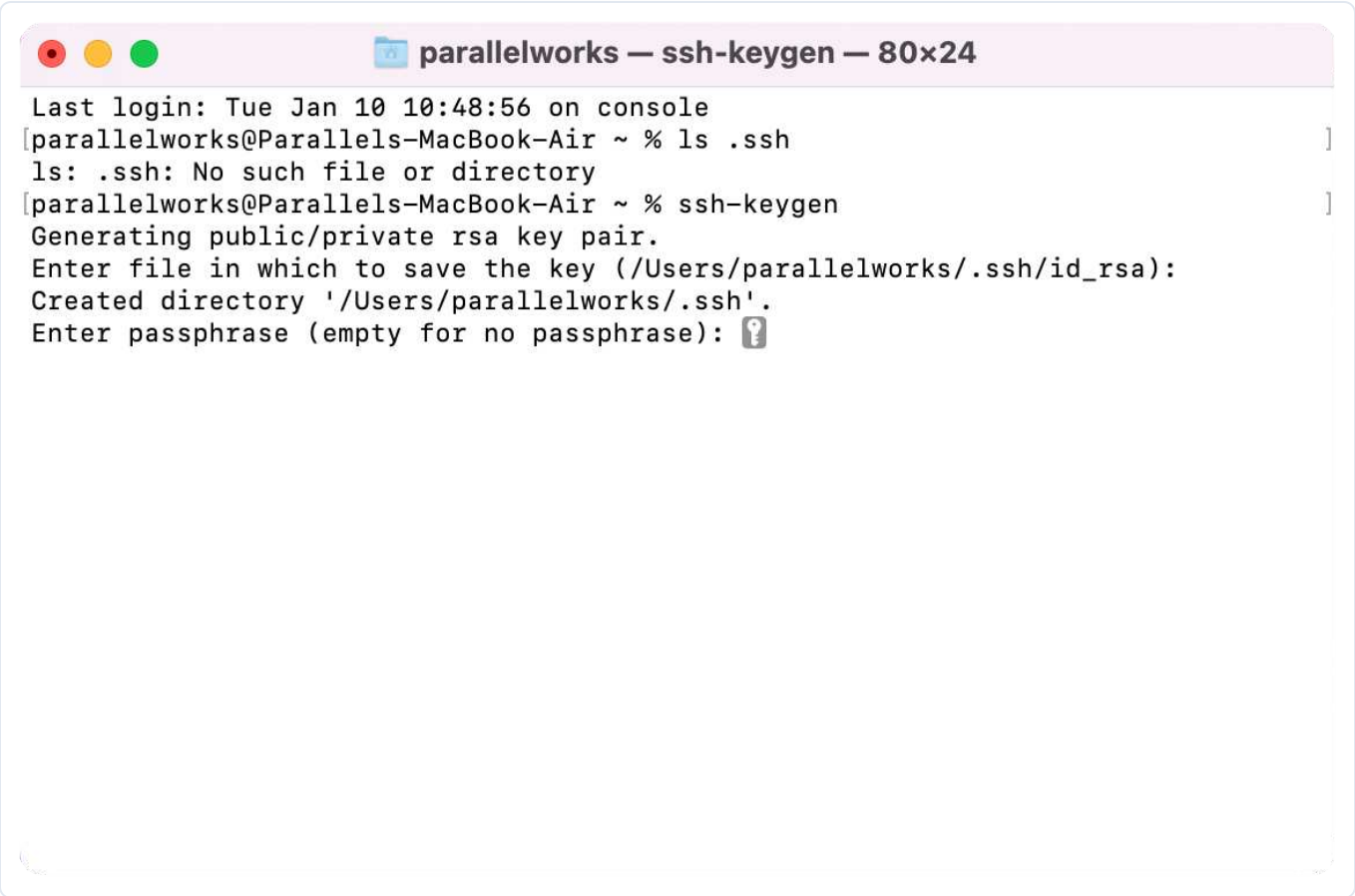
Create an SSH Key

Enter `ssh-keygen`. You'll see the following message:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/YourName/ .ssh/id_rsa):
```



Type the name of the new location, or press **return/enter** if you want to keep the default location. If this is the first time you've generated an SSH key and you use the default location, you'll see `Created directory '/Users/YourName/ .ssh'.`.

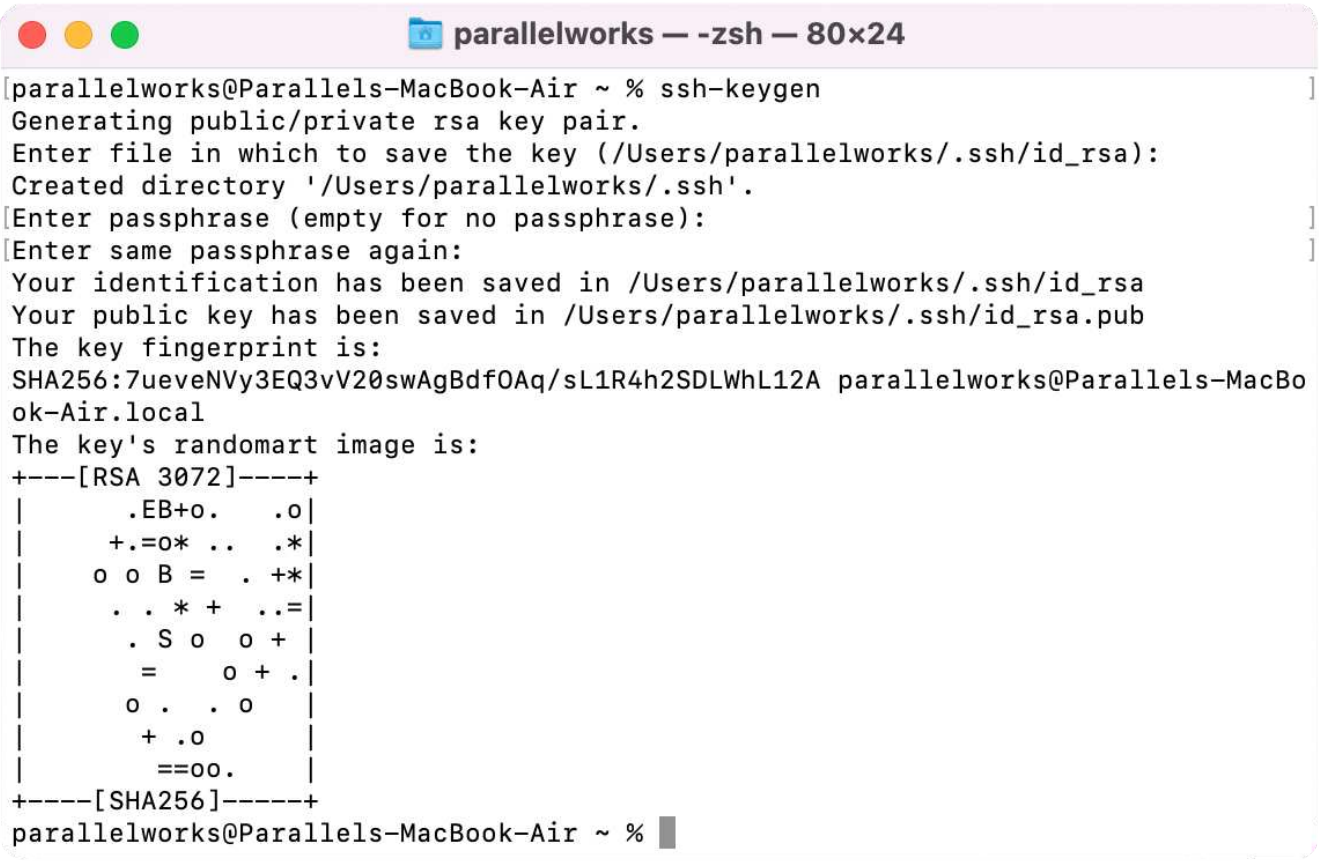


You'll be prompted to enter a passphrase for your SSH key with the following message:

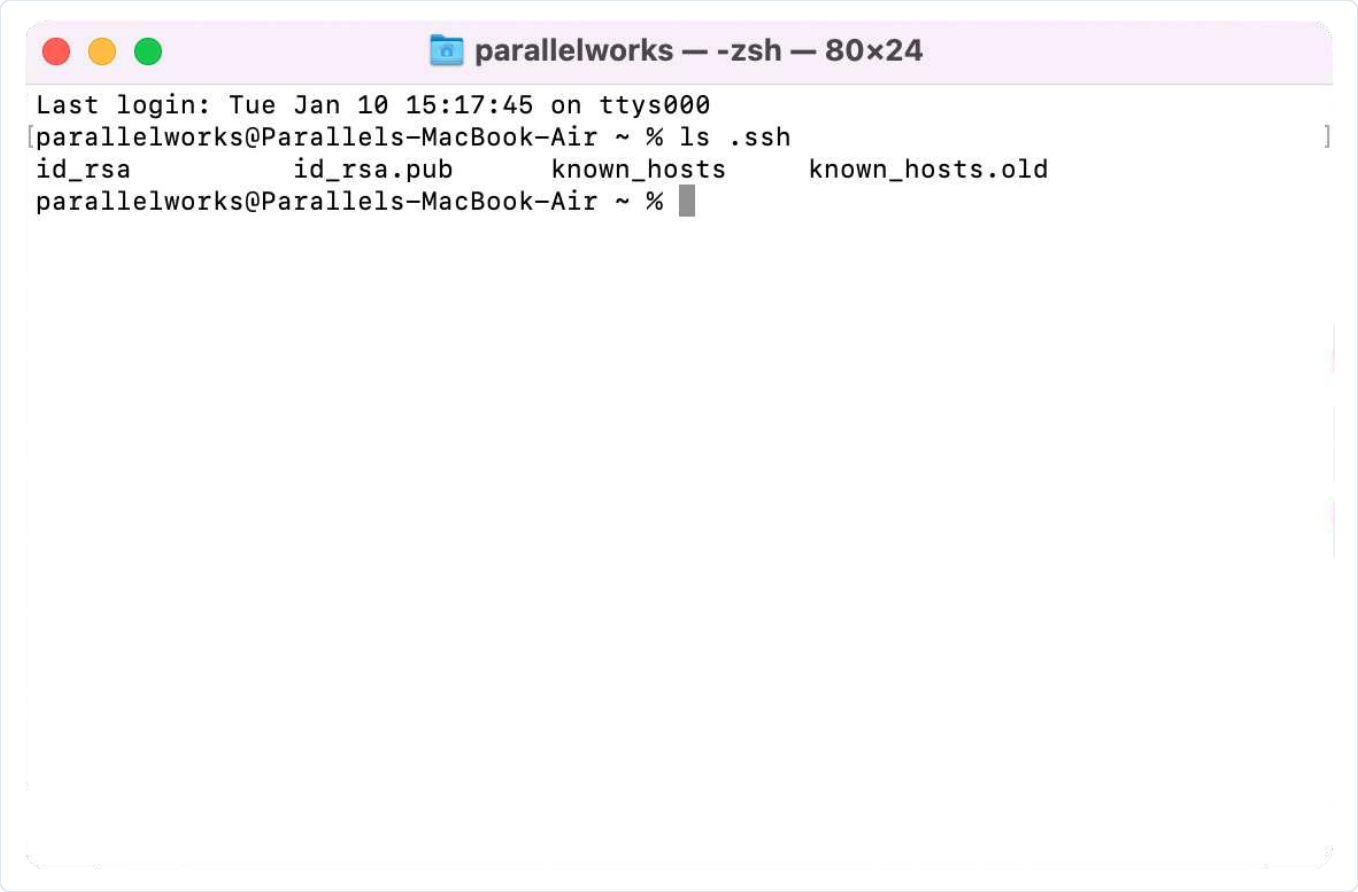
```
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
```

Enter a passphrase, or press **return/enter** for both lines if you don't want to use one. You'll see the following message, along with your key fingerprint and the key's randomart image:

```
Your identification has been saved in /Users/YourName/ .ssh/id_rsa
Your public key has been saved in /Users/YourName/ .ssh/id_rsa.pub
```



Enter `ls .ssh` again. Now you'll see your private and public SSH keys.



Get Your Public Key

Enter `cat .ssh/id_rsa.pub`. You'll see the full contents of your public SSH key.

Copy all of the text from the beginning of `ssh-rsa` to the end of `local`. This is what you'll paste into the **Access Public Key** text box in your cluster's configuration settings or the **Key** text box in **Account** > **Authentication** > **Key**.

> **IMPORTANT**
>
> For safety's sake, never store a private SSH key on a server. If someone gained access to your private key, they could use it to access any other device that key protects.

What if I use PuTTY?

Public SSH keys must be in OpenSSH format before you can use them on the ACTIVATE platform. If you use PuTTY, your SSH keys are likely saved in a PPK format like this:

```
PuTTY-User-Key-File-2: ssh-rsa
Encryption: none
Comment: rsa-key-20211005
Public-Lines: 6
AAAAB3NzaC1yc2EAAAABJQAAAQEAo7FYgire/SVeCEmN3CLxKgQhG5kCqt+eL4VU
X81/zO+lf1P8RjsjCNLrRd0o2zfquhaNFbYKA+DSo6Vpg18EYHiQA/HgEOgzUAF7
Tq4OaZl3yVyJKjzxQDXHnRhnJmCJ438PJD69crQh4apGtuPQuJOOKJe1PCpcp7Fy
P2+y2Hb0wM23K60tWsML9wf2P6gsY/vYxV+wNlohJa9LuY2dtk39kCs/5tmIOfHi
tOE+3ZjxzYTV0xlLNF+Gjxm1GV38YIa9R9fzdMXqm2HiHFbT5YbB6MaB/brDhVtO
dLha0y8OITQosmogA13mOcfYLlbgou65eHT+Nj9tALKEX3LgDw==
```

Before copying a PPK key, it must be converted to OpenSSH format. The easiest way to do this is with PuTTYgen, typically included with PuTTY installations. Below is a CLI example using PuTTYgen to convert a PPK key to OpenSSH format:

```
$ puttygen putty_key.ppk -L -o putty_key.pub
# Command explanation:
  * 'putty_key.ppk' -> the source key file
  * '-L' -> OpenSSH public key output type
  * '-o putty_key.pub' -> The output file name
```

The result is a single line public key file in OpenSSH format:

```
$ cat putty_key.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAo7FYgire/SVeCEmN3CLxKgQhG5kCqt+eL4VUX81/zO+lf1P8RjsjCNLrRd0o2zfquhaNFb
YKA+DSo6Vpg18EYHiQA/HgEOgzUAF7Tq4OaZl3yVyJKjzxQDXHnRhnJmCJ438PJD69crQh4apGtuPQuJOOKJe1PCpcp7FyP2+y
2Hb0wM23K60tWsML9wf2P6gsY/vYxV+wNlohJa9LuY2dtk39kCs/5tmIOfHitOE+3ZjxzYTV0xlLNF+Gjxm1GV38YIa9R9fzdM
Xqm2HiHFbT5YbB6MaB/brDhVtOdLha0y8OITQosmogA13mOcfYLlbgou65eHT+Nj9tALKEX3LgDw== rsa-key-20211005
```

Now you can add this key to your ACTIVATE account. For more details on that process, please see **Authentication**.

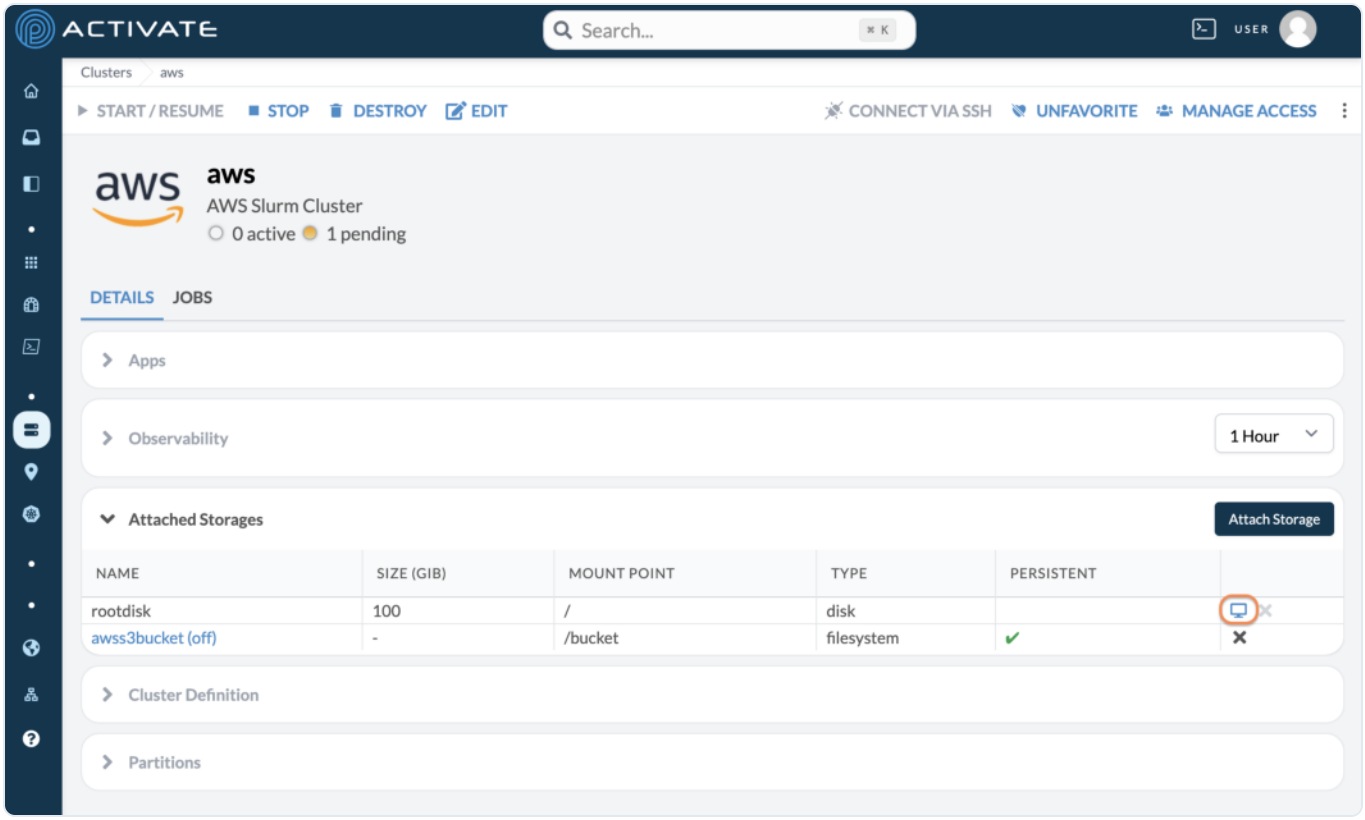
Creating OS Snapshots

After configuring a cluster, you may wish to save its data for later use on other clusters. You can create a snapshot of the cluster's root disk, which saves its operating system, data, and any applications you've installed. When you start a new cluster and select your snapshot as the image, the saved data will appear on the cluster.

OS snapshots are different from cloud snapshots in that cloud snapshots do not contain an operating system. This means that cloud snapshots cannot be used as your root disk.

Creating a Snapshot

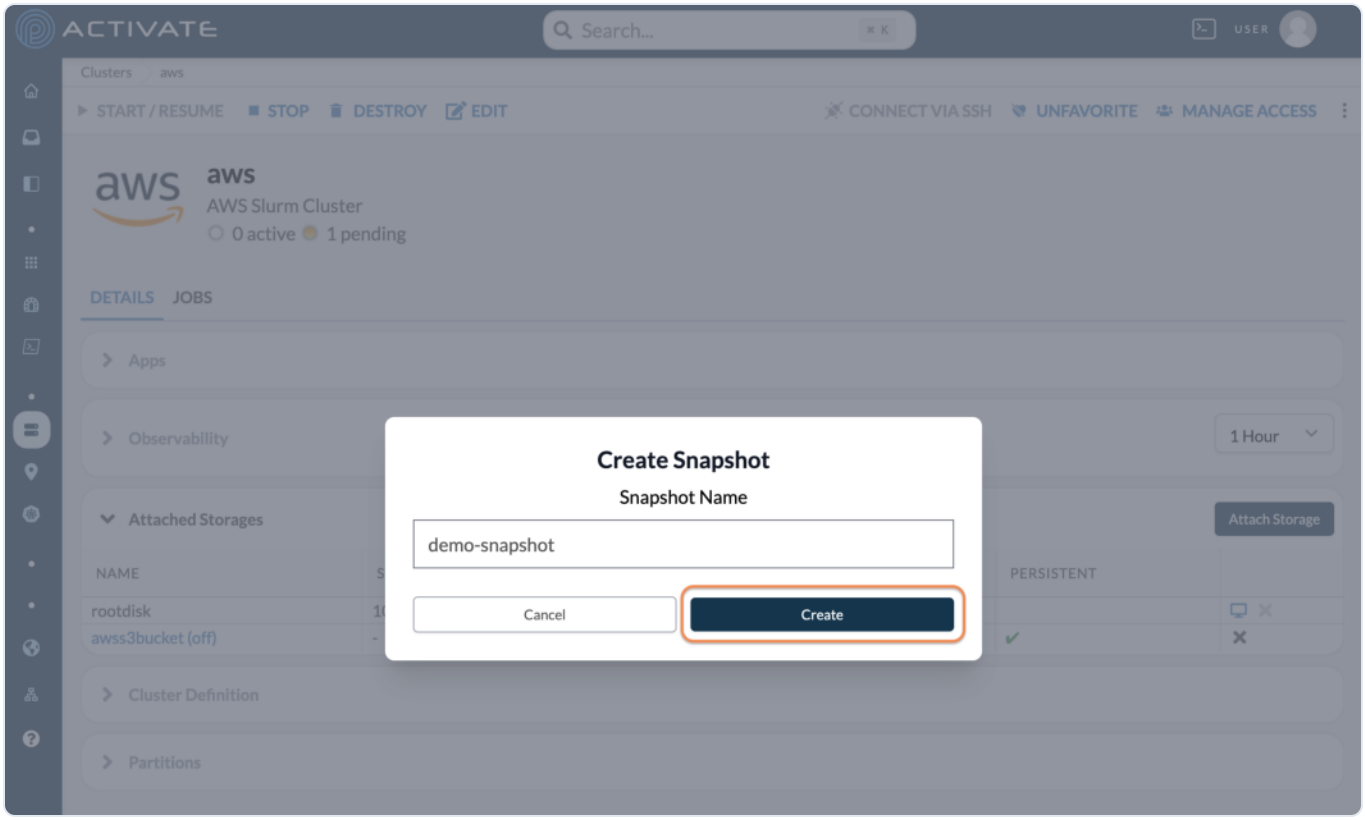
After starting a cluster, navigate to its **Sessions** page. In **Attached Storages**, click the snapshot icon.



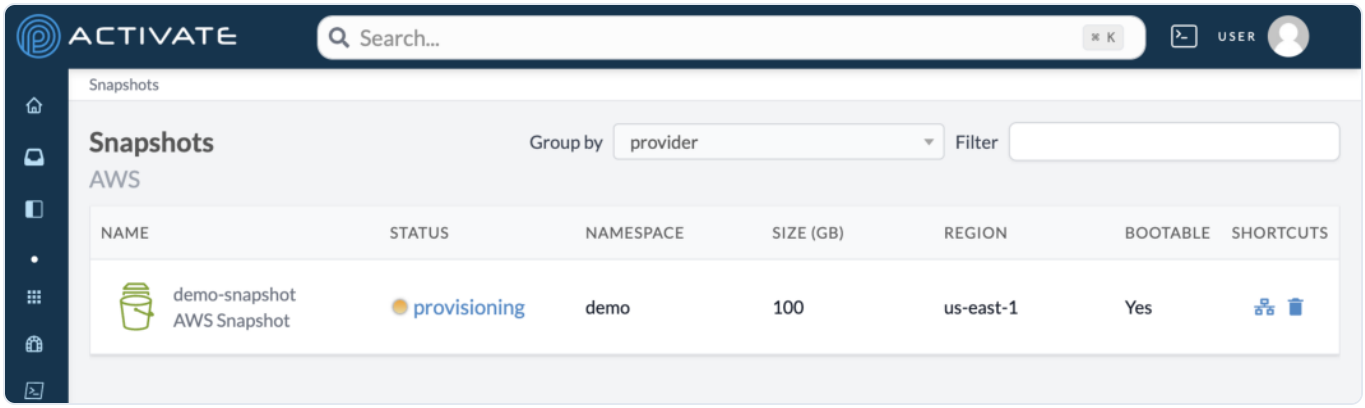
Name your snapshot. Please note that the name must:

- begin with a letter.
- be between 3 and 36 characters.
- use only numbers, hyphens, and lowercase letters. Spaces are not accepted.

Click **Create**.

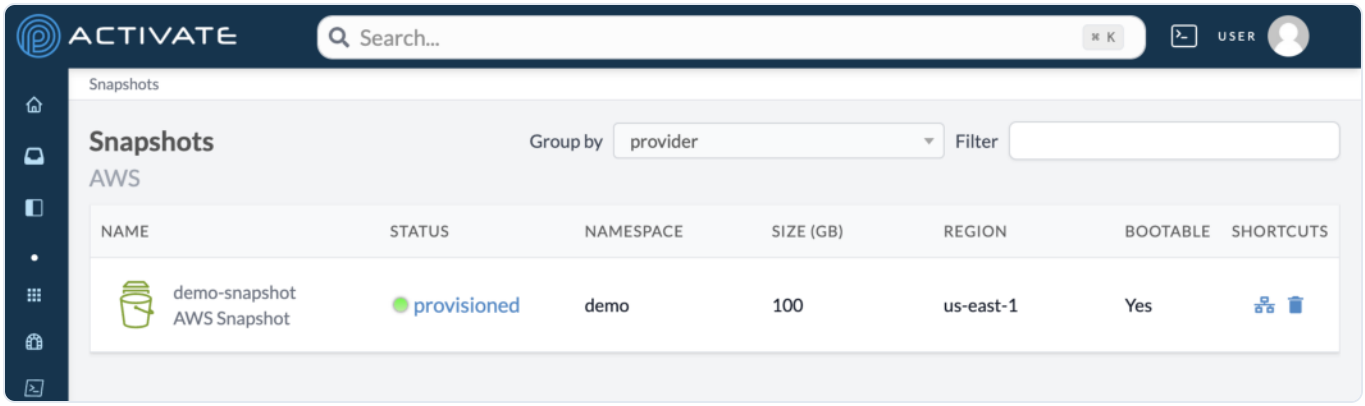


The snapshot will begin the provisioning process, and you'll be redirected to the **Snapshots** page.



Click **provisioning** if you'd like to see the steps of the process.

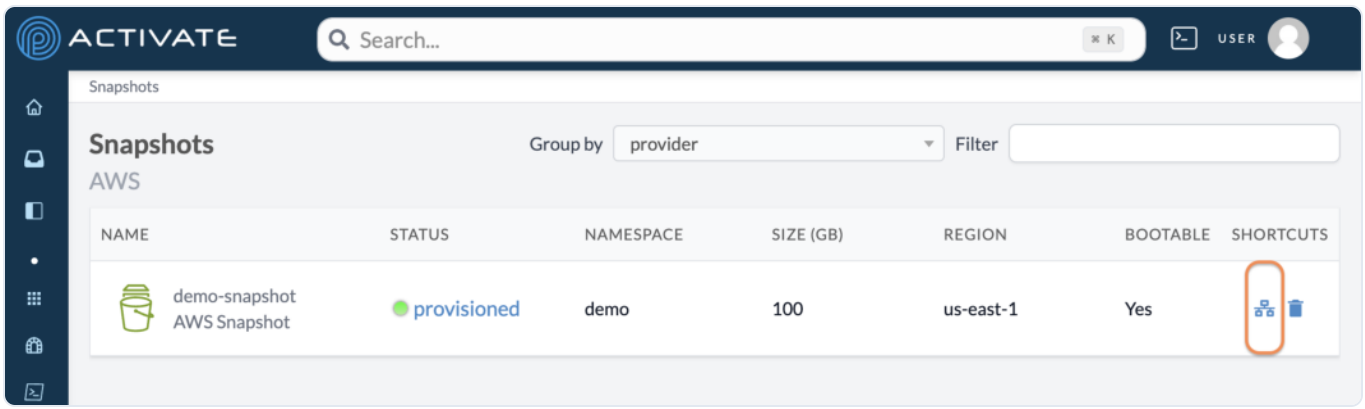
When the snapshot has been created, its status will change to **provisioned**.



Sharing a Snapshot

If you share a snapshot with a group in your organization, users in that group will be able to restore their own disks' settings from your snapshot. This is useful if, for example, you and another group's users are conducting similar types of work.

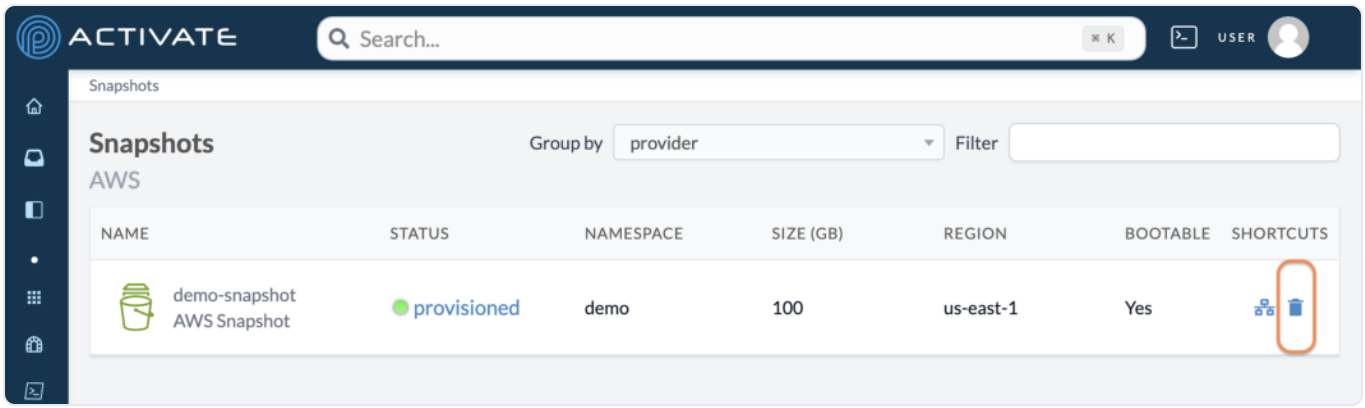
On the **Snapshots** page, click the share icon.



On the next page, select the group(s) you want to share the snapshot with. You'll see the message *Permission updated successfully*.

Deleting a Snapshot

On the **Snapshots** page, click the delete icon.



A dialog box will appear with the message *Are you sure you want to delete this snapshot?*

Click **Delete**.

A notification will appear with the message *Snapshot is waiting to be destroyed. This may take a few minutes.*

When this process is finished, the snapshot's **Status** will display as **deleted** for the next hour.

> **NOTE**
>
> Please note that deleting a snapshot does not stop the cluster it was created from. You must stop the cluster separately.

Managing IP Addresses

When creating a cluster, an ephemeral IP address is automatically chosen and assigned to the cluster. Alternatively, you can create a static, shareable IP address. This feature is useful if these two conditions are true for you:

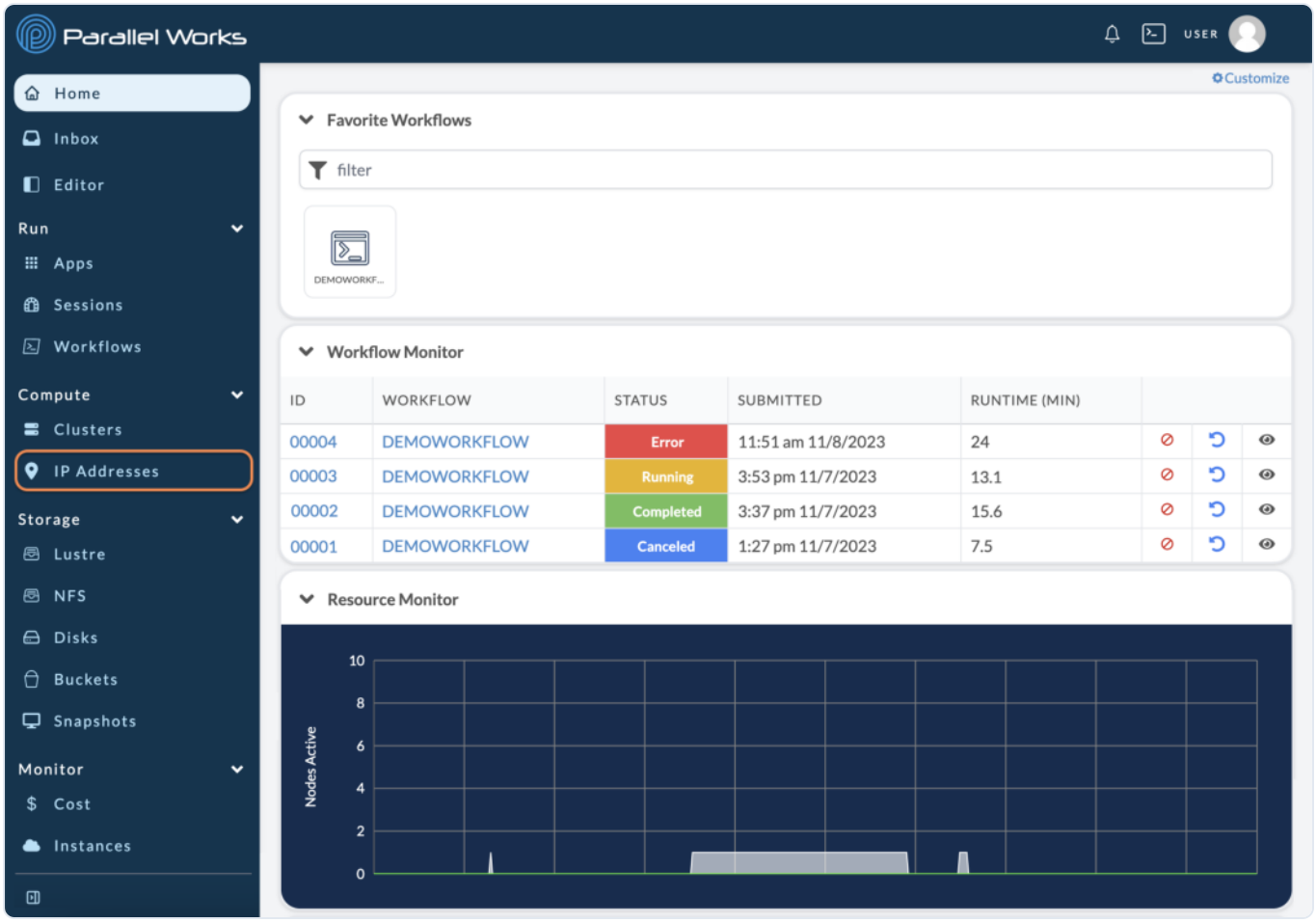
- Your application depends on having the same IP address each time you run it.
- You need to start and stop your cluster.

For example, you could be using a cluster to access a server that needs to whitelist your IP address before you can connect. If your IP address changes, the server firewall would need to be updated every time you access it. Instead, you could create an IP address to use each time and maintain your access.

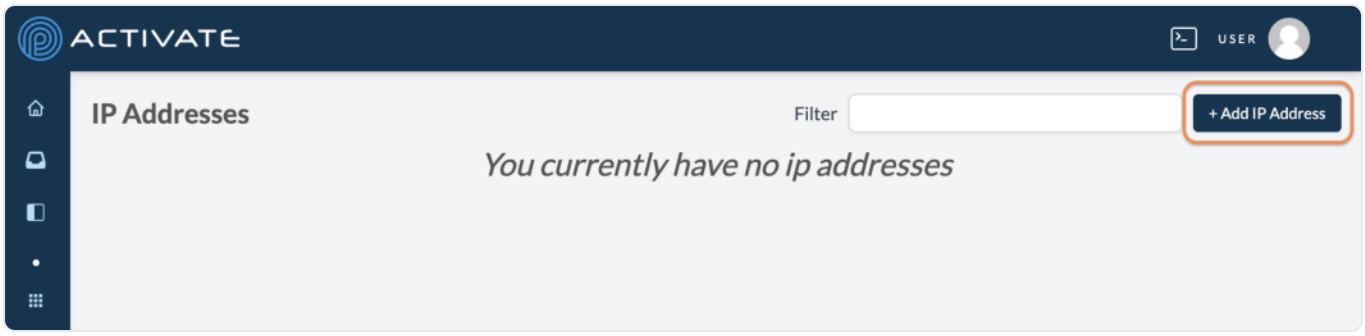
Another example is assigning a domain name to your cluster. You could create a reliable IP address to use in DNS.

Creating an IP Address

Navigate to **IP Addresses**.



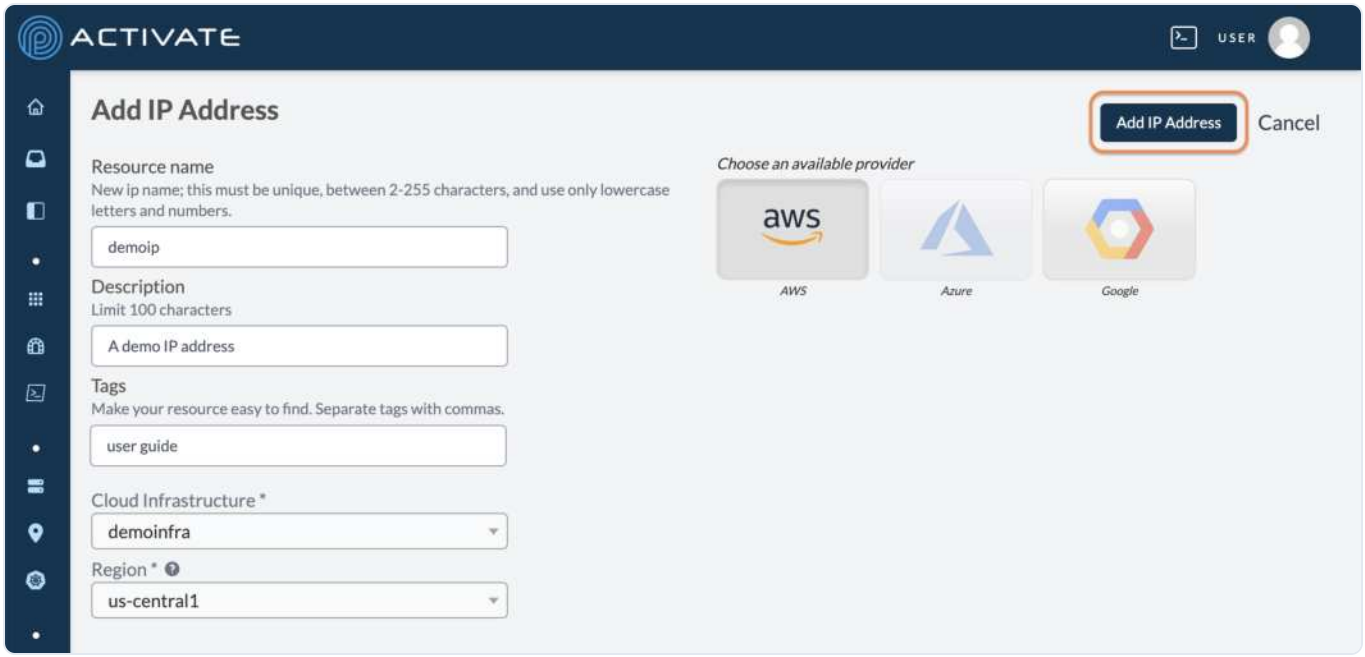
Click + **Add IP Address**.



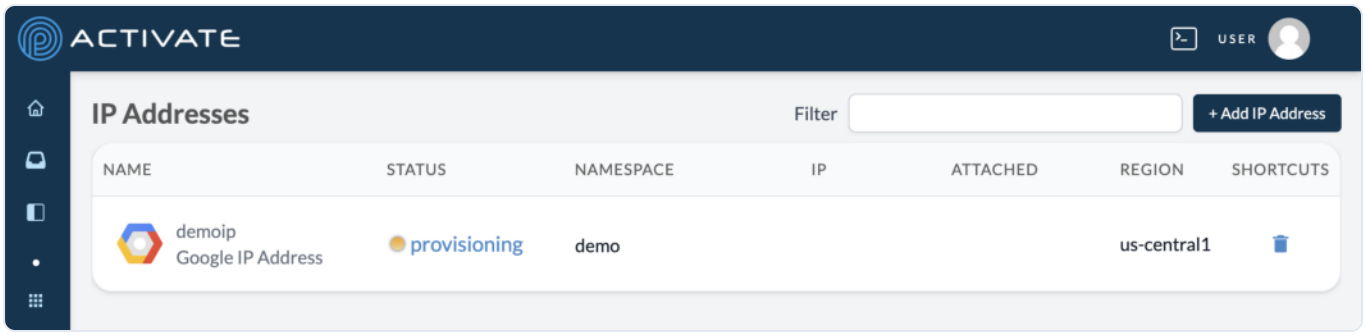
You must select a provider (AWS, Azure, or Google) before you can select a **Cloud Infrastructure** and **Region**.

A **Resource Name** is required. The **Description** and **Tags** are optional.

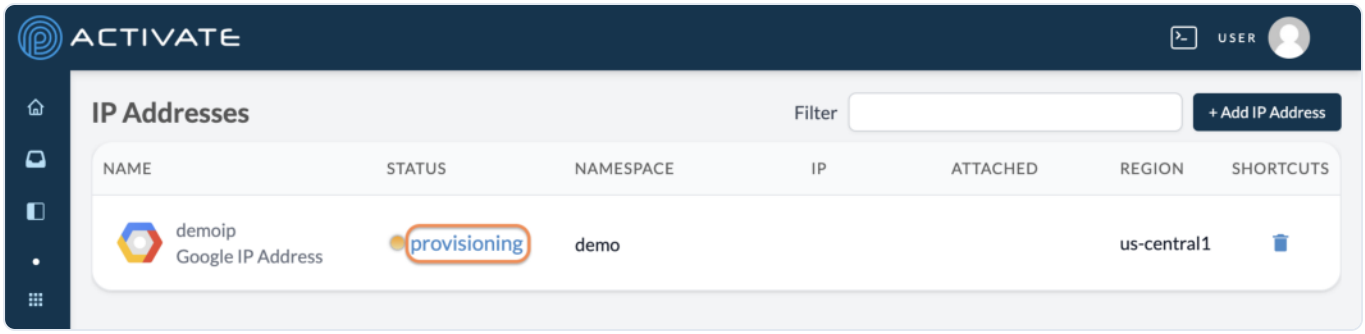
Click **Add IP Address**.

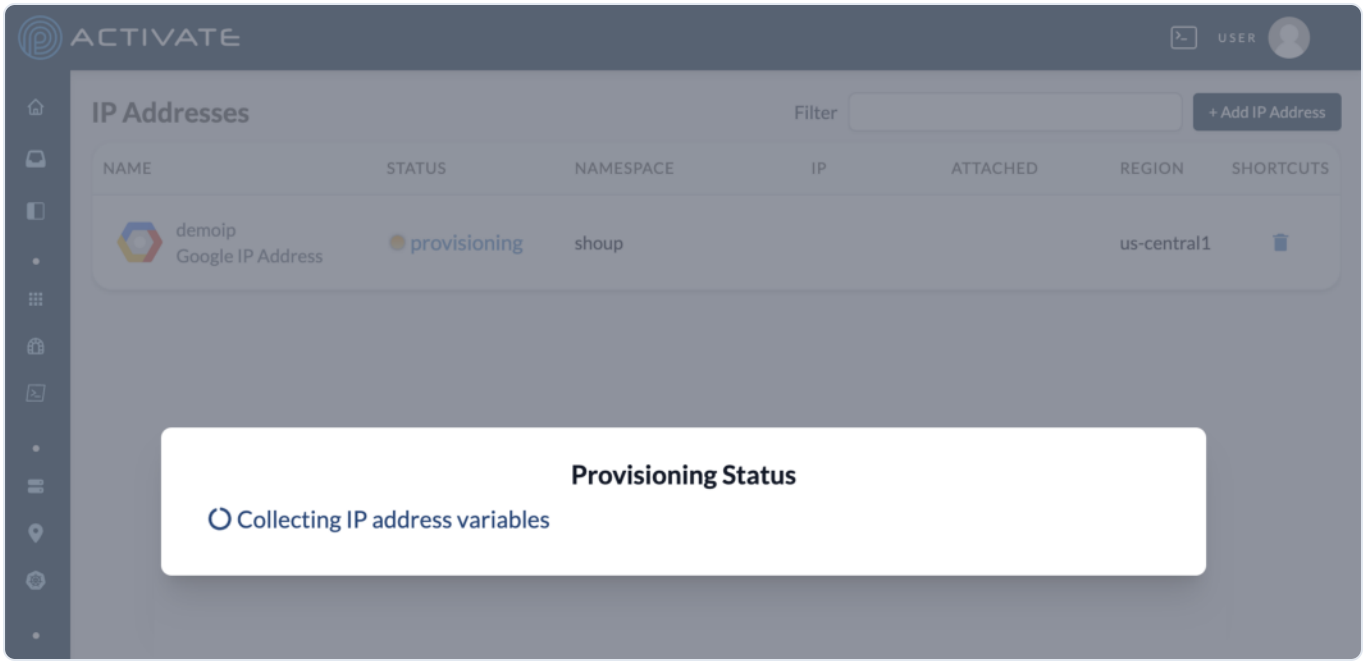


The message *IP created* will appear in the bottom right corner of your screen. Your IP address will begin the provisioning process.

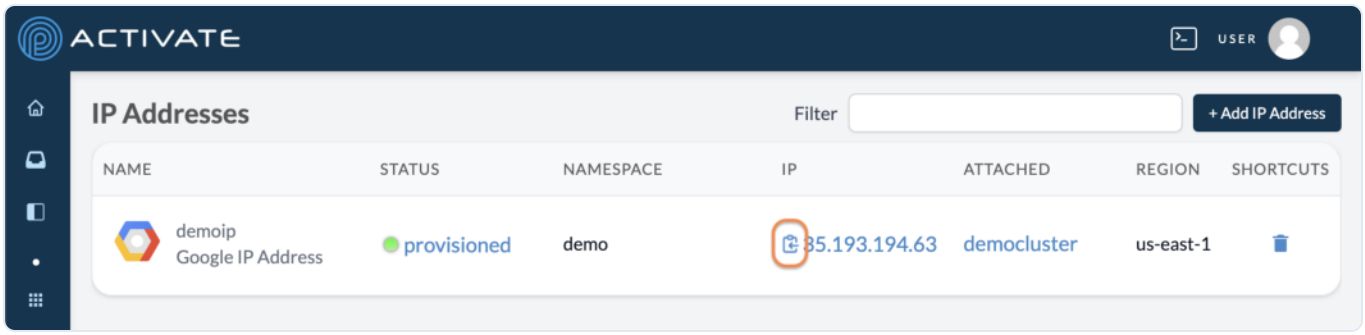


You can click the status to see a detailed provisioning log.



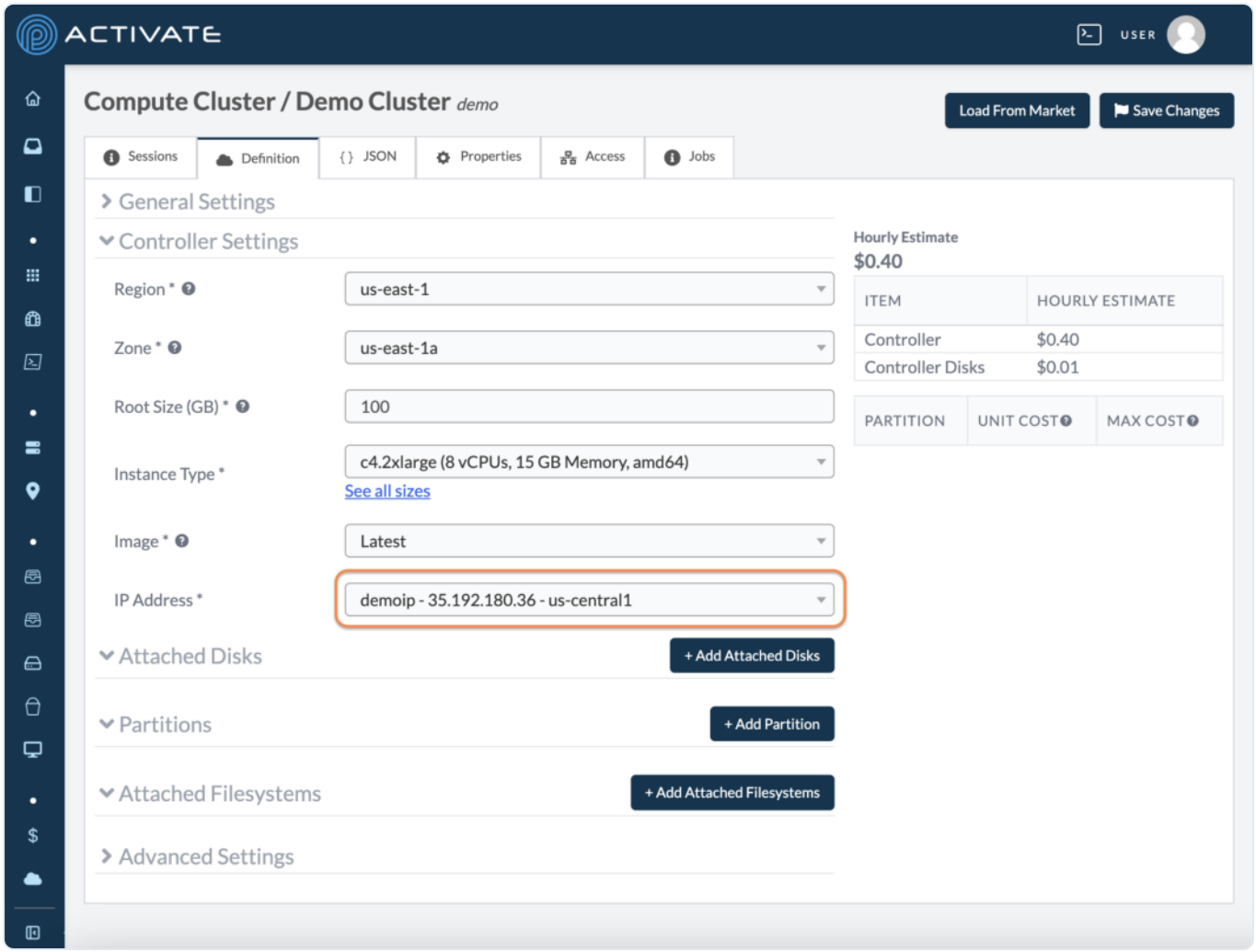


Once the IP address is provisioned, you can click the copy icon to share it.

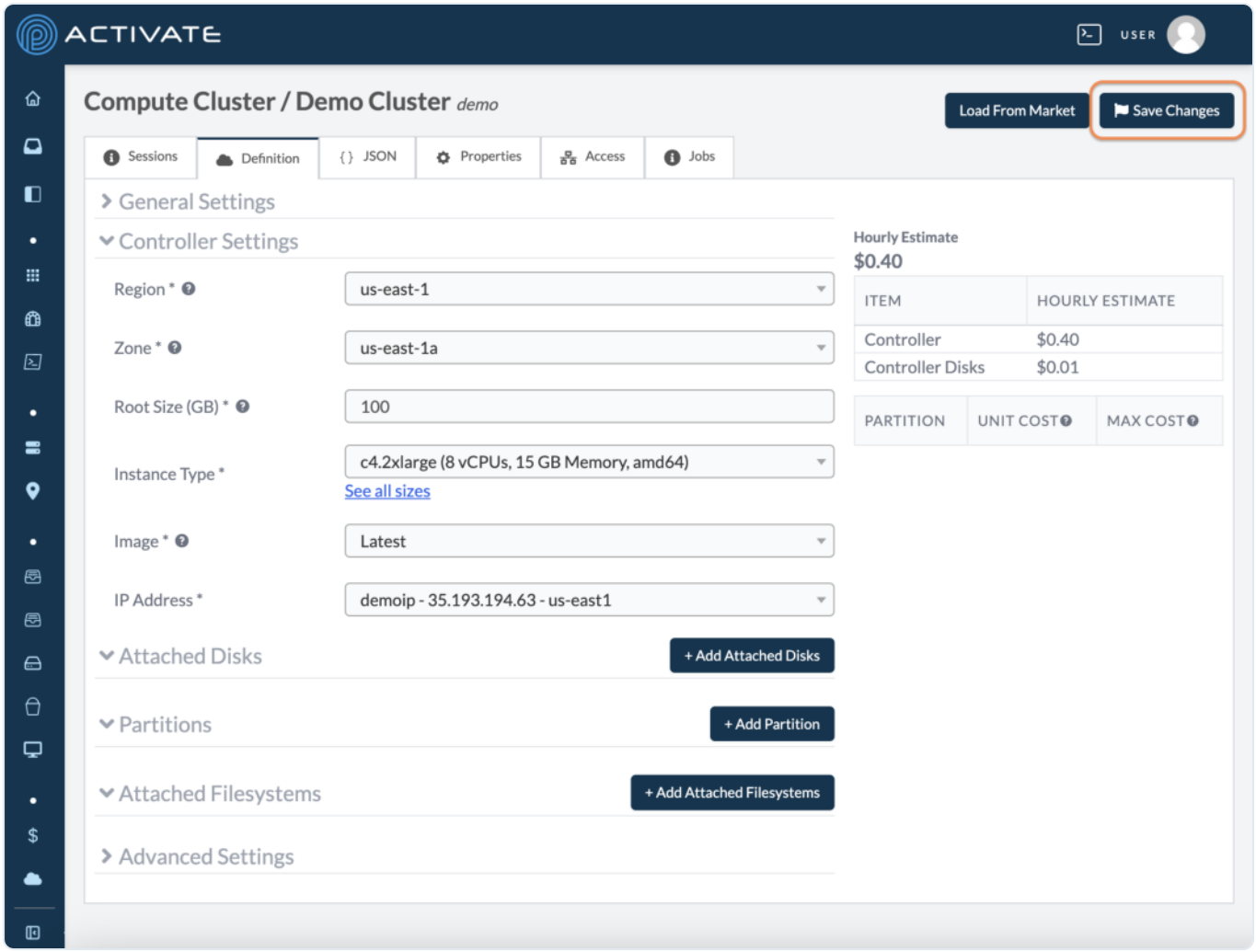


Attaching an IP Address

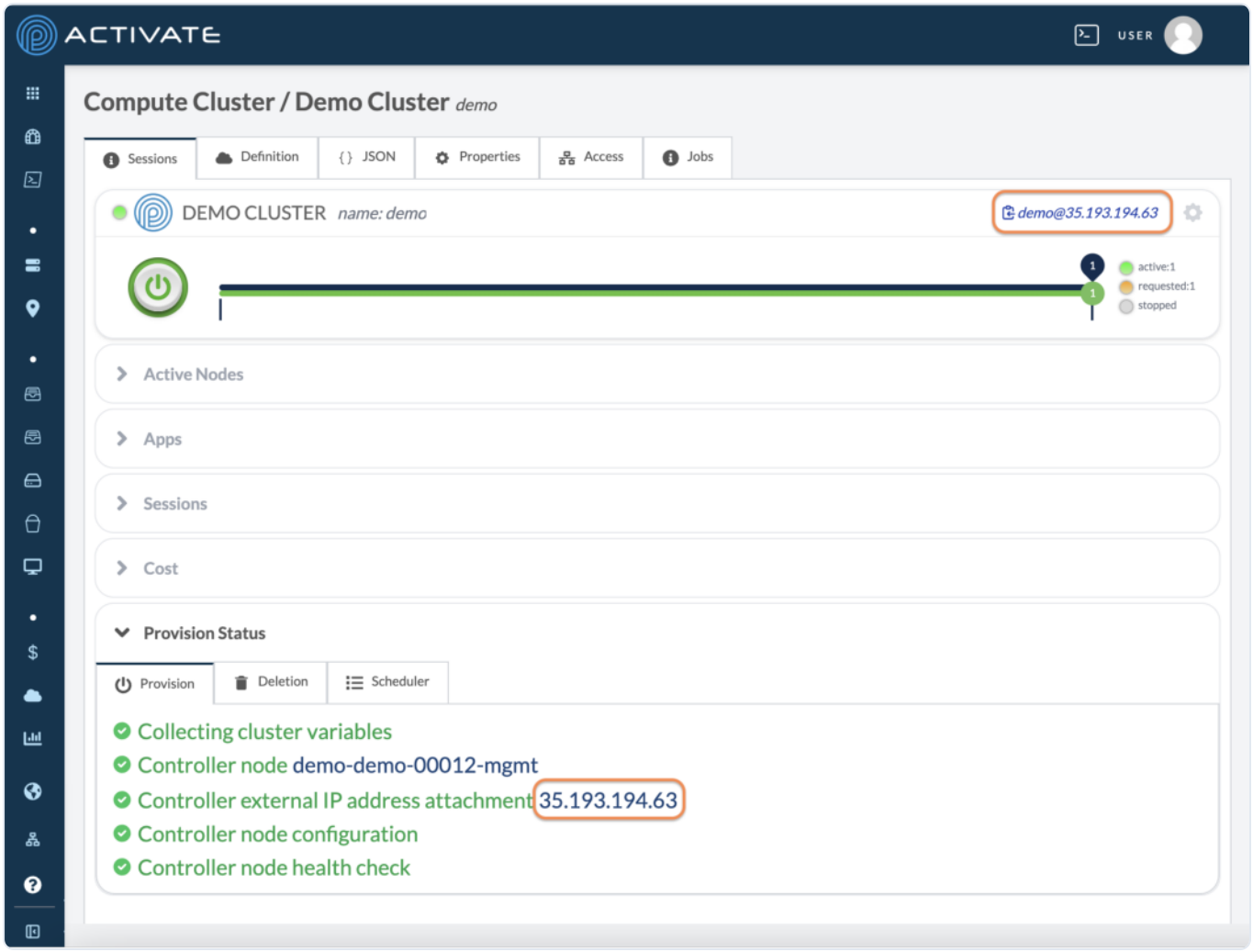
Navigate to the **Definition** tab of the cluster you'd like to attach the IP address to. In **Controller Settings**, use the **IP Address** dropdown menu to select one of your IP addresses.



Click **Save Changes**.

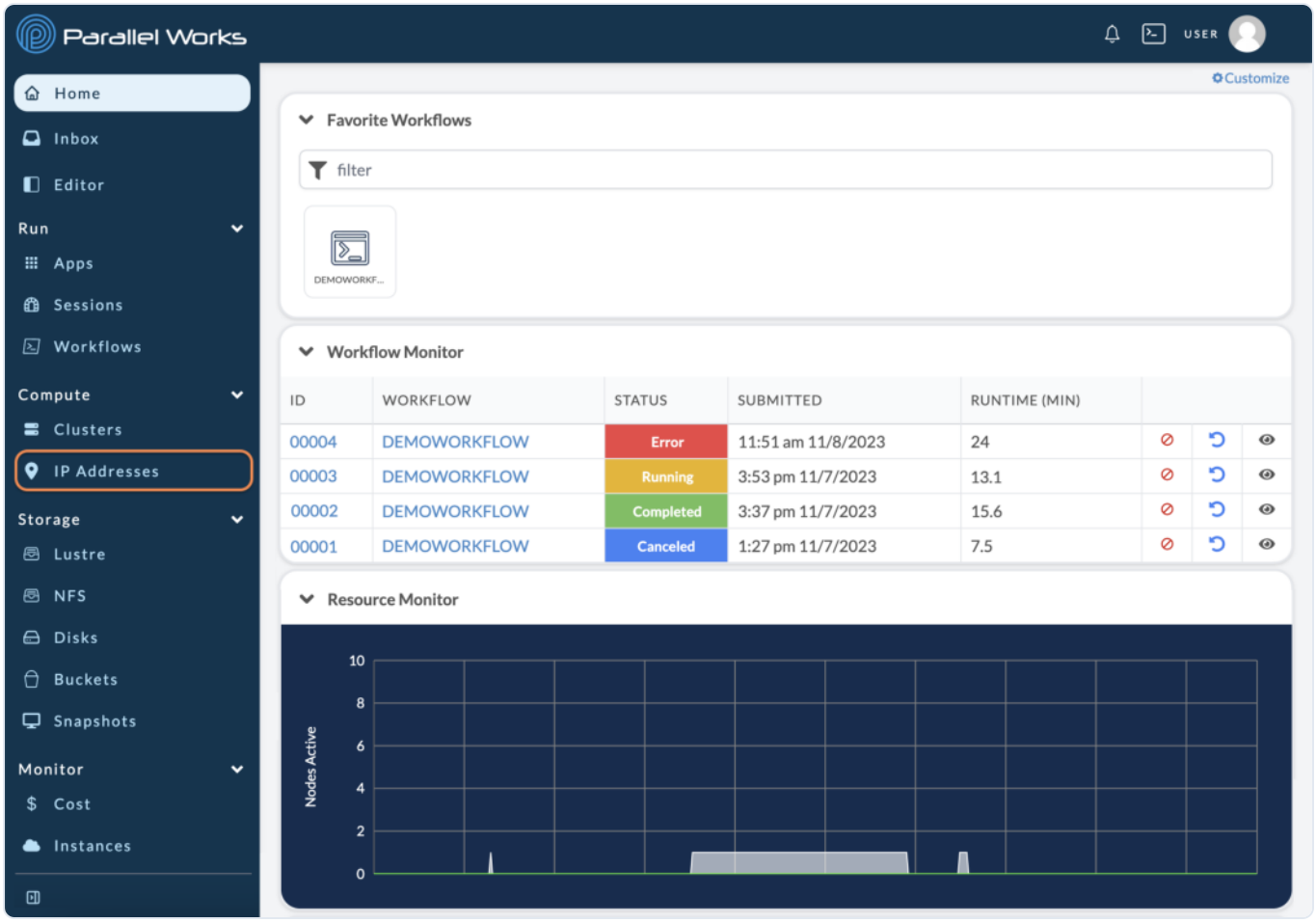


When your cluster has been provisioned, you'll see the attached IP address in the provisioning logs and at the top of the page.

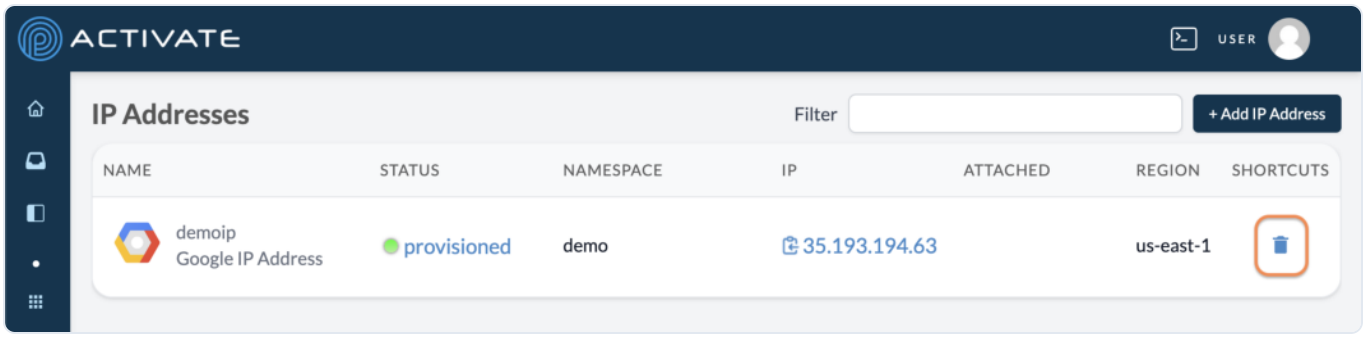


Deleting an IP Address

Navigate to **IP Addresses**.



Click the delete icon.



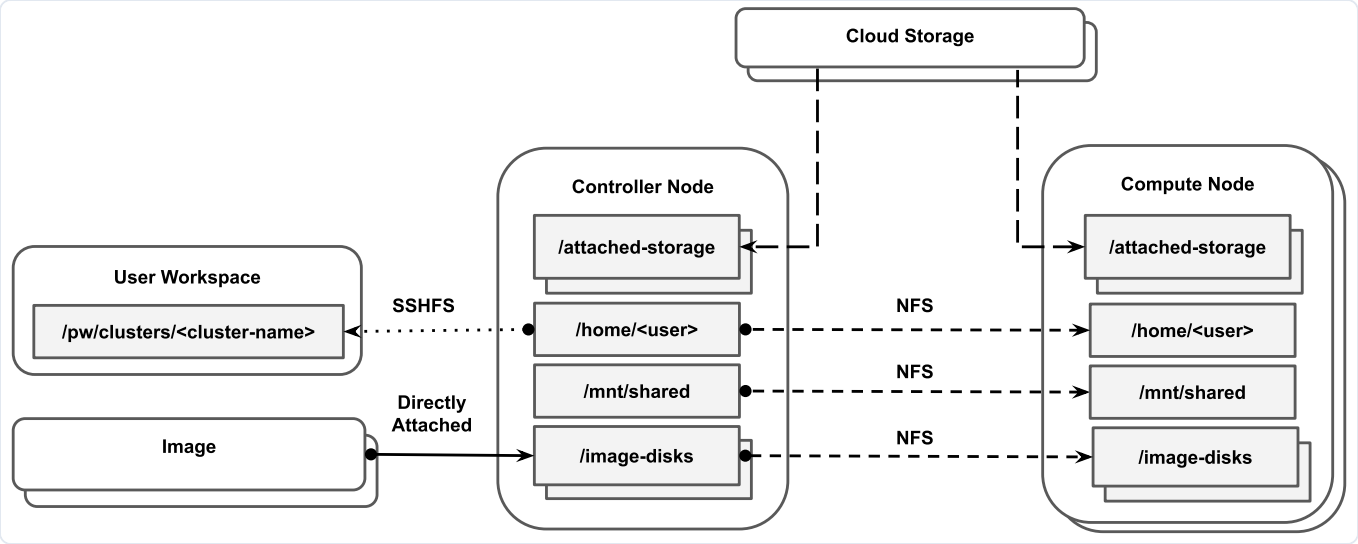
A dialog box will appear with the message *Are you sure you want to delete this IP address?*

Click **Delete**.

The message *IP address is waiting to be destroyed. This may take a few minutes.* will appear in the bottom right corner of your screen. Your IP address will begin the provisioning process.

Shared Directories

This section provides an overview of the directories shared between the compute nodes and the controller nodes in HPC cloud clusters. Understanding these shared directories is crucial for managing data and optimizing job performance in your cluster setup.



Home Directory

The home directory serves as a central storage location and plays a key role in your cluster's operation. It is automatically NFS exported from the controller node, and all compute nodes inside a cluster automatically mount `/home` over NFS.

Because the home directory is part of the controller's root disk, it gets deleted when a cluster is turned off. This directory is a no-cost alternative to cloud storage and can be used for job execution, storing configurations, and performing small file operations.

For ease of use, you can mount the cluster owner's home directory into their user workspace using SSHFS by configuring **User Workspace Mount Points** in the cluster configuration. We recommend leveraging this mount to conveniently drag and drop small files as well as open scripts and logs in the integrated development environment (IDE). However, it's important to note that using this directory for large file transfers between the user workspace and the controller is not recommended. Instead, we recommend using the commands `rsync` or `scp`. For more information, please see **Transferring Data**.

Cloud Storage

To meet your specific storage needs, you have the flexibility to create additional storage and filesystems that can be attached to your clusters. These options provide you with customizable storage solutions tailored to your workload. For more information on managing storage and attaching additional storage resources to your clusters, please see **Storage**.

Low Performance

Persistent Cloud Object Storage

Cloud storage presents an excellent option for persistent and cost-effective data storage. You can utilize AWS S3, Azure Blob Storage, Google Cloud Storage (GCS) objects that can be mounted to the controller and compute nodes using the Filesystem in Userspace (FUSE) technology. This type of storage allows you to store your results securely before turning off a cluster. To enhance performance, we recommend employing manual data copying with corresponding commands such as `gsutil`, `aws s3`, or `azcopy`.

Persistent Cloud NFS Storage

Alternatively to object storage, you have access to managed filesystem services such as Filestore on GCP or EFS on AWS. These services offer the ability to mount filesystems to the controller and compute nodes over NFS.

These filesystems, like object storage, are persistent, and can offer better performance than object storage. Please note that setting this up currently requires assistance from our support team.

> **PERSISTENT STORAGE**
>
> You can utilize persistent filesystems for installing custom user-specific software that needs to be readily available whenever a cluster is started. Moreover, it serves as a reliable location for storing your results securely before shutting down a cluster, allowing for seamless continuation of your work in subsequent sessions.

High Performance

Lustre

Lustre storage offers a powerful parallel distributed filesystem that can be mounted to the controller and compute nodes. You can configure Lustre storage as either persistent or ephemeral, depending on your requirements. However, it's important to be mindful of the cost implications associated with Lustre filesystems; in general, Lustre is more suitable for large-scale HPC workloads with demanding I/O operations.

Image Disks

Image disks are disks that are created from a cloud image, and that disk is directly attached to the controller. The controller automatically NFS exports image disks and compute nodes automatically mount those exports. Files written to these disks will not persist across sessions, as the disk is created from the image source each time a cluster is started. Leveraging image disks is a practical choice when you need to install organization-wide software, making it readily available to all users.

Submitting Jobs via Slurm

> **ABOUT JOBS**

> There are two ways you can submit jobs to a cluster: by using workflows or through any terminal or command-line interface. For the workflows option, please see **Running Workflows**.

After you've started a cluster, log in to the controller with your preferred method. The quickest way to submit a job is to transfer your file(s) to the cluster, then run the command `sbatch`.

In this example, we submitted the file `demo_test1.sbatch` with `sbatch`:

```
[demo@democluster-60 ~]$ ls
demo_test1.sbatch
[demo@democluster-60 ~]$ sbatch demo_test1.sbatch
Submitted batch job 2
```

After submitting a job, you can watch its progress with the command `watch squeue`, which will update every two seconds with the job's status in the `ST` column:

```
Every 2.0s: squeue

JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
    4 test.part     test     demo CF       0:08      2 demo-democluster-00060-1-[0001-0002]
```

You can also use `watch 'sinfo;echo;squeue'` if you want to see general cluster information in addition to your job's progress:

```
Every 2.0s: sinfo; echo; squeue

PARTITION        AVAIL  TIMELIMIT  NODES  STATE NODELIST
test.partition1*    up   infinite      2   mix# demo-democluster-00060-1-[0001-0002]
test.partition1*    up   infinite      3  idle~ demo-democluster-00060-1-[0003-0005]
test.partition2     up   infinite      5  idle~ demo-democluster-00060-2-[0001-0005]

         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             4 test.part     test     demo CF       0:26      2 demo-democluster-00060-1-[0001-
0002]
```

When using `watch squeue` or `watch 'sinfo;echo;squeue'`, the `ST` column will show `CF` while the node(s) configure. All of the rows beneath `JOBID` will clear when your job is finished:

```
Every 2.0s: sinfo; echo; squeue

PARTITION        AVAIL  TIMELIMIT  NODES  STATE NODELIST
test.partition1*    up   infinite      2  idle% demo-democluster-00060-1-[0001-0002]
test.partition1*    up   infinite      3  idle~ demo-democluster-00060-1-[0003-0005]
test.partition2     up   infinite      5  idle~ demo-democluster-00060-2-[0001-0005]

         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

Once the job is finished, you can check its output with `cat file_name`. Our file `demo_test1.sbatch` inlcuded instructions to send our completed job's data to an `std.out` file and any errors to an `std.err` file:

```
[demo@democluster-60 ~]$ ls
demo_test1.sbatch  std.err  std.out
[demo@democluster-60 ~]$ cat std.err
[demo@democluster-60 ~]$ cat std.out
demo-democluster-00060-1-0001
demo-democluster-00060-1-0001
demo-democluster-00060-1-0002
demo-democluster-00060-1-0002
```

Using `cat std.err` didn't return anything because the job executed without errors.

Common Slurm Commands

This section gives a quick overview of the commands you'll use most often when interacting with clusters. You can use any of these commands in any terminal after logging in to a controller node.

Because ACTIVATE uses Slurm to manage jobs, you can use any of their system commands. For an extensive list of those options, see Slurm's command guide. You can also enter `man` in front of any command (such as `man sacct`) to see its description and a list of other available commands in Slurm's virtual manual.

> **ABOUT JOB IDS**
>
> When we say "job ID" in this section, we mean the job ID that Slurm assigns to your work, which will appear when running many of these commands. ID numbers in the **Worflow Monitor** and the **jobs** folder on the ACTIVATE platform act as a separate identifier to help us track how many jobs we've ever run on the platform.
>
> Using any of the commands in this section will generate a new Slurm job ID.

> **ABOUT FAULT TOLERANCE**
>
> Fault tolerance is defined by how well an infrastructure remains functional or online even when there are service disruptions because of outages or natural disasters.
>
> On ACTIVATE, cluster deletions are queue-based for fault tolerance.
>
> The cluster startup process has no retries for fault tolerance, but the logs are visible so users can see any problems that occur.
>
> For compute node startup requests, fault tolerance is implemented with retries via Slurm (by default, there is a new startup attempt approximately every 20 minutes).

Job Management

salloc

`salloc` retrieves resources for your job without executing any tasks.

Using this command retrieves resources before you need them by signaling the system to reserve a specified number of nodes. For example, `salloc -N 2` will reserve two compute nodes, for a total of three nodes, including the controller.

`salloc` is useful if you're sharing a cluster with other users in your organization: using this command means that once a job is finished, the allocated nodes will remain on reserve for your use until you disconnect from the cluster (meaning that your wait times will be shorter because another user cannot take control of your allocated nodes, so you won't have to wait for more nodes to become available or wait for them to start once they're available).

sbatch

`sbatch` submits a job script that will execute later. You can also configure nodes with `sbatch` by adding these options:

- `--n-tasks-per-node` to specify the number of CPUs
- `-t` to specify the maximum amount of time you want these resources to run with the format of `0:0:0` for hours, minutes, and seconds

For example, `sbatch demo_test1.sbatch --n-tasks-per-node 5 -t 3:0:0` would run the file `demo_test1.sbatch` and request 5 CPUs for 3 hours of maximum run time.

srun

`srun` executes a job script. You can use the same options from `salloc` and `sbatch` with `srun`:

- `-N` to specify the number of nodes
- `--n-tasks-per-node` to specify the number of CPUs
- `-t` to specify the maximum amount of time you want these resources to run

For example, `srun -N 1 --pty bash` would request 1 compute node and open a pseudoterminal, creating an interactive command-line session.

scancel

`scancel` paired with a job ID ends a pending or running job or job step. For example:

```
[demo@democluster-60 ~]$ sbatch demo_test1.sbatch
Submitted batch job 6
[demo@democluster-60 ~]$ scancel
scancel: error: No job identification provided
[demo@democluster-60 ~]$ scancel 6
```

If you cancel a job, it will disappear from your queue.

Cluster Management

sinfo

`sinfo` shows information about the nodes and partitions you're using. By default, `sinfo` displays partition names, availability, time limit, the number of nodes, state, and the node's ID number (which is displayed as

`username-democluster-00019-1-[0001-0005]`).

- Please note that if you enter `sinfo` without setting up partitions, you'll receive the error message `slurm_load_partitions: Unable to contact slurm controller (connect failure)`.

squeue

`squeue` shows a list of running and pending jobs. By default, `squeue` shows job ID number, partition, username, job status, number of nodes, and node names for all queued and running jobs. You can also use these commands to adjust `squeue`'s output:

- `--user` to see only one user's jobs, such as `--user=yourPWusername`
- `--long` to show non-abbreviated information and add the field `timelimit`
- `--start` to estimate a job's start time

Notification Management

> **ABOUT NOTIFICATIONS**
>
> This section applies only to cloud clusters, not on-premises clusters.

By default, cloud clusters will send job start/finish notifications to ACTIVATE. You can change that setting or add it as an email notification by following the steps in **Managing Notifications**.

To enable additional job status notifications, you can also pair the flag `--mail-type` with the commands `salloc`, `sbatch`, or `srun`. For example, the command `sbatch --mail-type=FAIL exampleScript.sbatch` will send a notification if your job fails to start or complete.

You can add multiple notification events to the `--mail-type` flag at once and separate them with commas: `sbatch --mail-type=BEGIN,END exampleScript.sbatch`

Alternatively, you can add tags inside a Slurm batch file, as seen in this example:

```
#!/bin/bash

#SBATCH --mail-type=BEGIN,END

echo "Hello, World!"
```

> **WHICH METHOD SHOULD I USE?**
>
> Both methods above work equally well.
>
> The primary difference is that entering the flag and notification event(s) outside the file will override any settings inside of your batch script, but will not cause anything to be written into the file.

Notification Events

The table below lists the events currently supported by the `--mail-type` flag.

Type	Notification Event
`ALL`	equivalent to `BEGIN,END,FAIL,INVALID_DEPEND,REQUEUE,STAGE_OUT`
`NONE`	does not send notifications; this is the default
`BEGIN`	job start
`END`	job end
`FAIL`	job failure
`REQUEUE`	job is requeued
`INVALID_DEPEND`	a job's dependency cannot be satisfied, so the job will not run
`STAGE_OUT`	when a job has completed or been cancelled, but has not yet released its resources
`TIME_LIMIT_50`	when a job reaches 50% of its walltime* limit
`TIME_LIMIT_80`	when a job reaches 80% of its walltime* limit
`TIME_LIMIT_90`	when a job reaches 90% of its walltime* limit
`TIME_LIMIT`	when a job reaches its walltime* limit
`ARRAY_TASKS`	sends other option notifications for each array task instead of for the array as a whole; without this option, `BEGIN`, `END`, and `FAIL` notifications will only notify once for the full array instead of sending a notification for each individual array task

**The walltime limit is the user set limit for how long a job can run.*

Please note that walltime limits are infinite by default. A walltime limit can be added when starting a job.

Troubleshooting

sacct

`sacct` shows a summary of users as well as completed and running jobs. Using this command will display a table with a job's ID number, name, partition, status, exit code, whose account it's running on, and how many CPUs it's using.

For troubleshooting purposes, the `State` and `ExitCode` fields from running `sacct` are especially useful for determining whether a node has failed and, if so, why. If you reach out to us for help, one of our support engineers may ask you for the information you see after running `sacct`.

scontrol

`scontrol` can delegate commands to specific job IDs and nodes. Please note that many `scontrol` commands can only be executed as user root. You can use these commands with a job ID to adjust `scontrol`'s output:

- `suspend` to pause a job's processes
- `resume` to continue a job's processes

- `hold` to make a job a lower priority, putting it “on hold” so higher priority jobs will run first
- `release` to remove a job from the hold list
- `show job` to get detailed information about a job

Choosing Instance Types

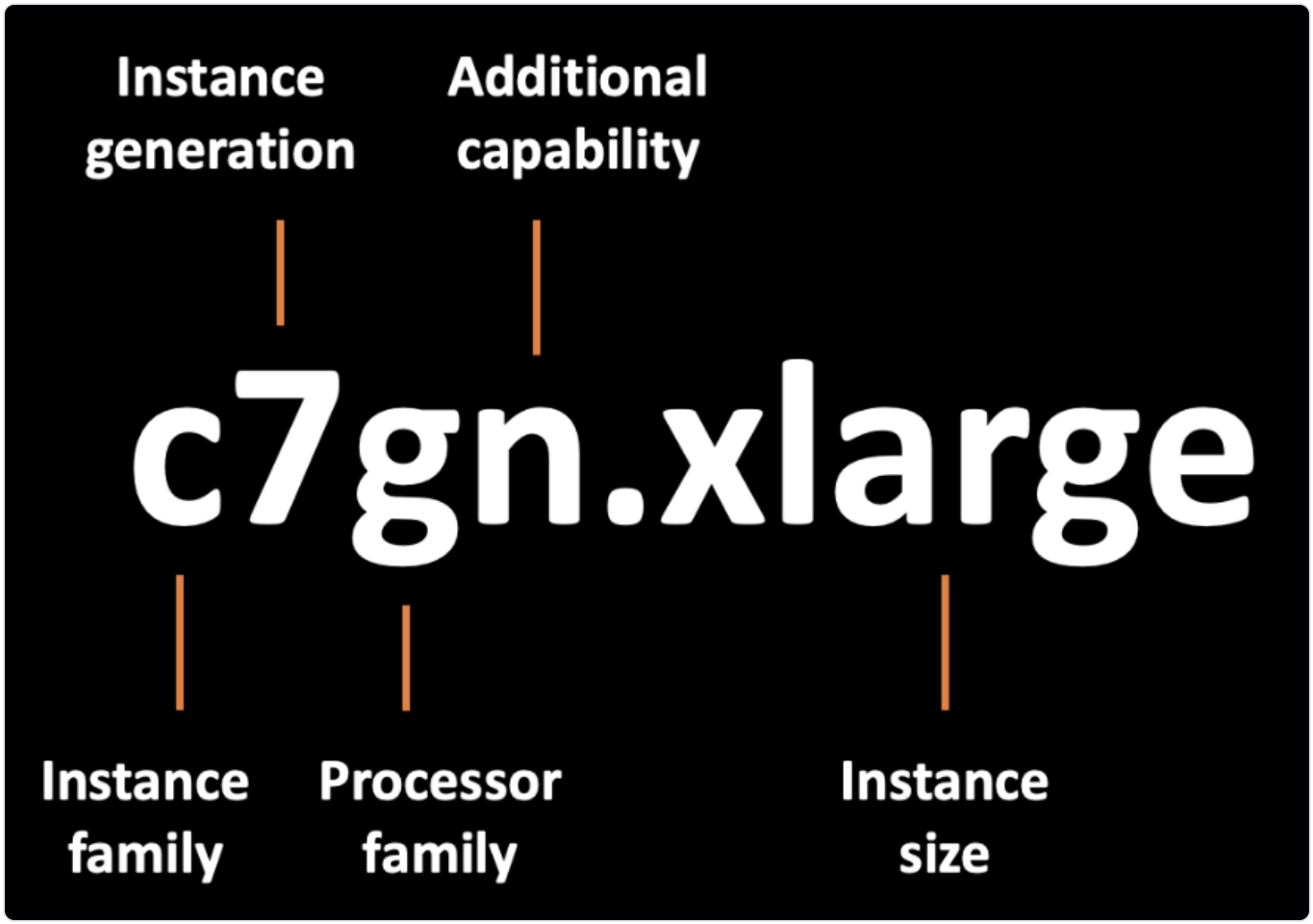
When configuring a cloud-based cluster, there are many instance types to choose from, and each cloud service provider (CSP) has their own naming conventions. Elements of instance type names correspond to elements of the physical machines that clusters are deployed from. This page explains these instance naming conventions for AWS, Azure, and Google clusters.

Please note that specific instance types are subject to change as CSPs add and remove hardware; however, you can always refer to this page to help identify nodes and any additional features that are included with them.

AWS

On AWS, instance types are named following a pattern based on:

- instance family
- generation family
- processor family
- additional capabilities
- instance size



An example of an AWS instance type from their documentation

Instance Families

Syntax	Description
c	Compute optimized
d	Dense storage
f	FPGA
g	Graphics intensive
hpc	High performance computing
inf	AWS Inferentia
m	General purpose
mac	macOS
p	GPU accelerated
r	Memory optimized

t	Burstable performance
trn	AWS Trainium
u	High memory
vt	Video transcoding
x	Memory intensive

Processor Families

Syntax	Description
a	AMD processors
g	AWS Graviton processors
i	Intel processor

Note for the `i` syntax: Many older Intel-based instance types do not include this code. It was likely added when AWS began offering more Graviton and AMD-based options.

Additional Capabilities

Syntax	Description
d	Instance store volumes
n	Network and EBS optimized
e	Extra storage or memory
z	High performance
flex	Flex instance

Instance Generations

Older instance generations are usually kept available for a set period of time, but it's suggested to use newer versions for optimal performance.

- documentation for the current generation of AWS instances
- documentation for the previous generation of AWS instances

Selection Guidelines

For CPU-based workloads, most ACTIVATE users will want to select **compute-optimized** instance types, which include the **c** and **hpc** instance families:

- **c5n.18xlarge**: One of ACTIVATE's default configuration instances. c5 instances are based on Intel Skylake processors. Note that the instance name is missing an `i` in the name because it predates other processors being included in the family.

- **c6in.24xlarge**: A newer generation of the `c` instance family. Note that this instance includes an `i` in the name to separate it from `c6a` (AMD) and `c6g` (Graviton) instances.
- **hpc6a.48xlarge**: AMD EPYC-based instances designed specifically for HPC workloads

For GPU-based workloads, look for instances in the **g** and **p** families:

- **g5.48xlarge**: g5 instances are equipped with NVIDIA A10G Tensor Core GPUs and AMD EPYC processors.
- **p3.16xlarge**: p3 instances include Intel Skylake processors and NVIDIA V100 Tensor Core GPUs.

> **NOTE**
>
> Instance options vary by zone and region. If you're trying to use a specific instance type and it's not visible in the dropdown, try changing to a different region first.

Further Reading

You can read more about AWS instances and naming conventions any time by visiting this page of their documentation.

Azure

Azure's naming structure follows this pattern:

[Family] + [Sub-family] + [# of vCPUs] + [Constrained vCPUs] + [Additive Features] + [Accelerator Type] + [Version]

On ACTIVATE, we also add an Azure instance's tier.

For example, the Azure instance `Standard_HC44rs` can be broken down into:

- **Tier**: Standard
- **Family**: H
- **Sub-family**: C
- **CPUs**: 44
- **Additive Features**: rs
  - **r**: RDMA capable
  - **s**: Premium Storage capable

Instance Families

Syntax	Description
A	Entry-level VMs for dev/test
Bs	Economical burstable VMs
D	General purpose compute
E	Optimized for in-memory applications

F	Compute optimized virtual machines
G	Memory and storage optimized virtual machines
H	High Performance Computing virtual machines
Ls	Storage optimized virtual machines
M	Memory optimized virtual machines
Mv2	Largest memory optimized virtual machines
N	GPU-enabled virtual machines

Instance Sub-families

Many Azure instance families include sub-families with different features. For example, H-Series instances come in two flavors:

- **HB**: Up to 120 AMD EPYC 7003-series CPU cores, 448 GB of RAM, and no hyperthreading
- **HC**: Up to 44 Intel Xeon Platinum 8168 processor cores, 8 GB of RAM per CPU core, no hyperthreading, and up to 4 Managed Disks

Additional Capabilities

Syntax	Description
a	AMD-based processor
b	Block Storage performance
d	diskful (that is, a local temp disk is present); this feature is for newer Azure VMs; see Ddv4 and Ddsv4-series
i	isolated size
l	low memory; a lower amount of memory than the memory intensive size
m	memory intensive; the most amount of memory in a particular size
p	ARM CPU
t	tiny memory; the smallest amount of memory in a particular size
s	Premium Storage capable, including possible use of Ultra SSD
C	confidential
NP	node packing
r	RDMA capable

Please note that this is not a complete list of additive features. Additionally, these identifiers are not used in all node types that may apply to them. For example, `Standard_HB60rs` instances have AMD EPYC processors, but don't have an `a` listed as an additional capability.

Instance Generations (version)

Like other cloud providers, Azure instances are routinely updated with newer generations. Azure has product pages for each instance series that describes their specifications, current generation, and additional features. Information for the H-Series nodes can be found on this page.

Selection Guidelines

For compute clusters, we suggest using Azure H-Series nodes as they are InfiniBand/RDMA enabled for high-speed networking. Our primary default cluster configuration uses `Standard_HC44rs` instances.

For Lustre, stick to instances that have `d` and `s` listed as additional features for their enhanced storage functionality.

Further Reading

You can read more about Azure naming conventions by visiting this page of their documentation.

You can read more about Azure instance types on this page of their documentation.

Google

Machine Families & Series

Google instances fall into one of four categories (called familes) and are further categorized by their series and generation.

- **General-purpose**: best price-performance ratio for a variety of workloads
  - **e2**
  - **n2, n2d, n1**
  - **c3**
  - **tau t2d, tau t2a**
- **Compute-optimized**: highest performance per core on Compute Engine and optimized for compute-intensive workloads
  - **h3**
  - **c2, c2d**
- **Memory-optimized**: ideal for memory-intensive workloads, offering more memory per core than other machine families, with up to 12 TB of memory
  - **m3, m2, m1**
- **Accelerator-optimized**: ideal for massively parallelized Compute Unified Device Architecture (CUDA) compute workloads, such as machine learning (ML) and high-performance computing (HPC); this family is the best option for workloads that require GPUs
  - **a2**
  - **g2**

Selection Guidelines

`h3-standard-88` is Google's newest node type that's suitable for HPC workloads. This type features 88 vCPUs (no hyperthreading), 352GB memory, and up to 200 Gbps network egress bandwidth.

`c2-standard-60 instances` are smaller than the **h3** nodes, but are also well suited for HPC applications.

For GPUs, try the **a2** series.

Further Reading

You can read more about Google instances and naming conventions any time by visiting this page of their documentation.

Run

About Workflows

At a high level, a workflow is a series of automated steps run on one or more compute resources.

More specifically, a workflow is a set of jobs and steps that run to completion whenever the workflow is executed. A workflow may have one or more jobs, and a job may have one or more steps. The jobs run in parallel, while the steps run sequentially. Workflows are defined using YAML. Below is an example of a simple workflow that prints the message `hello world` each time it is launched.

```yaml
jobs:
  main:
    steps:
      - name: Run
        run: echo hello world
```

For more information about building workflows, see Building Workflows.

About Sessions

A session is a secure, shareable connection between a cluster and a workflow.

When creating a session, the user chooses a port to expose. That port is made securely available through your ACTIVATE user workspace. The port is forwarded from the cluster back to the user workspace, allowing any user to log in to ACTIVATE and access that session.

Sessions add an authentication layer that only allows access through an SSH connection. Additionally, only users that are selected by the session creator have access.

Workflows

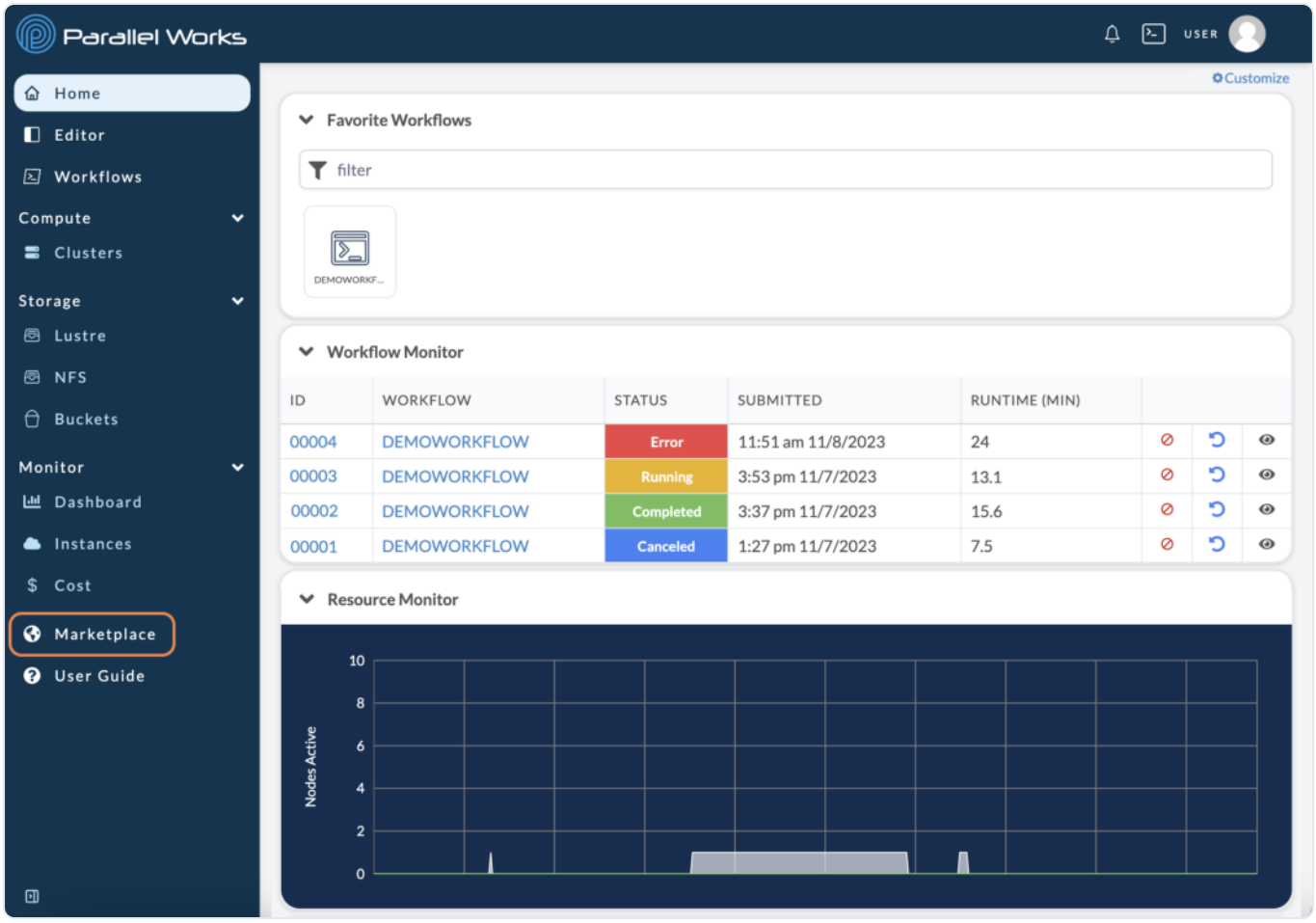
Adding Workflows

Adding Marketplace Workflows

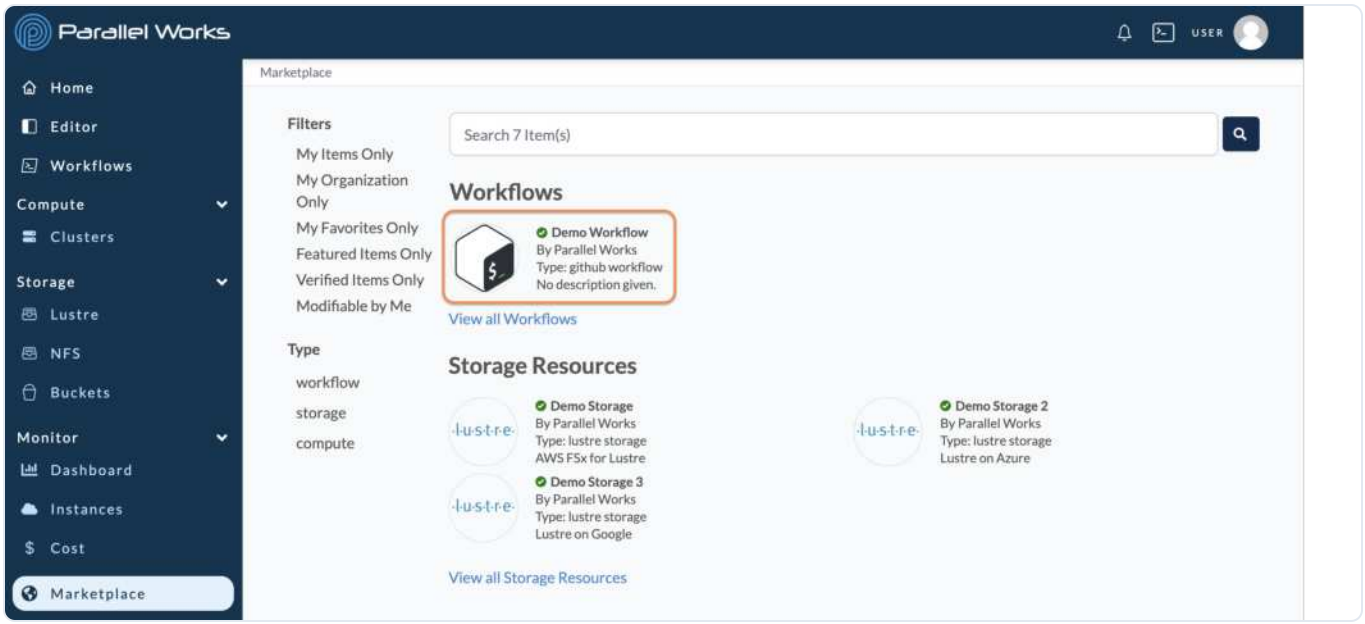
In the ACTIVATE Marketplace, we have pre-configured workflow tools and templates you can use to complete your work.

> **NOTE**
>
> This section is a quick-start guide for adding workflows. For more detailed information about the Marketplace, please see **Navigating the Marketplace**.

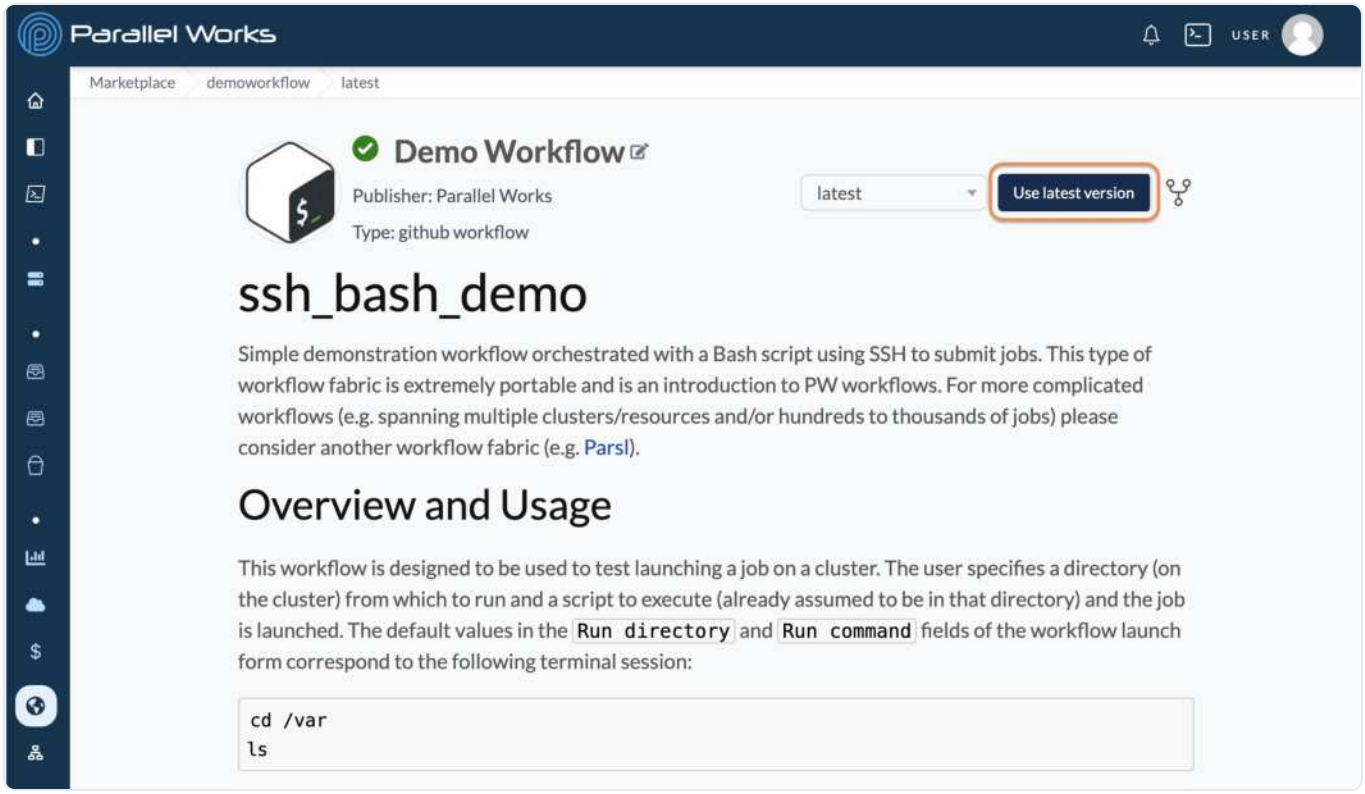
Click **Marketplace**.



Click the workflow you want to add.

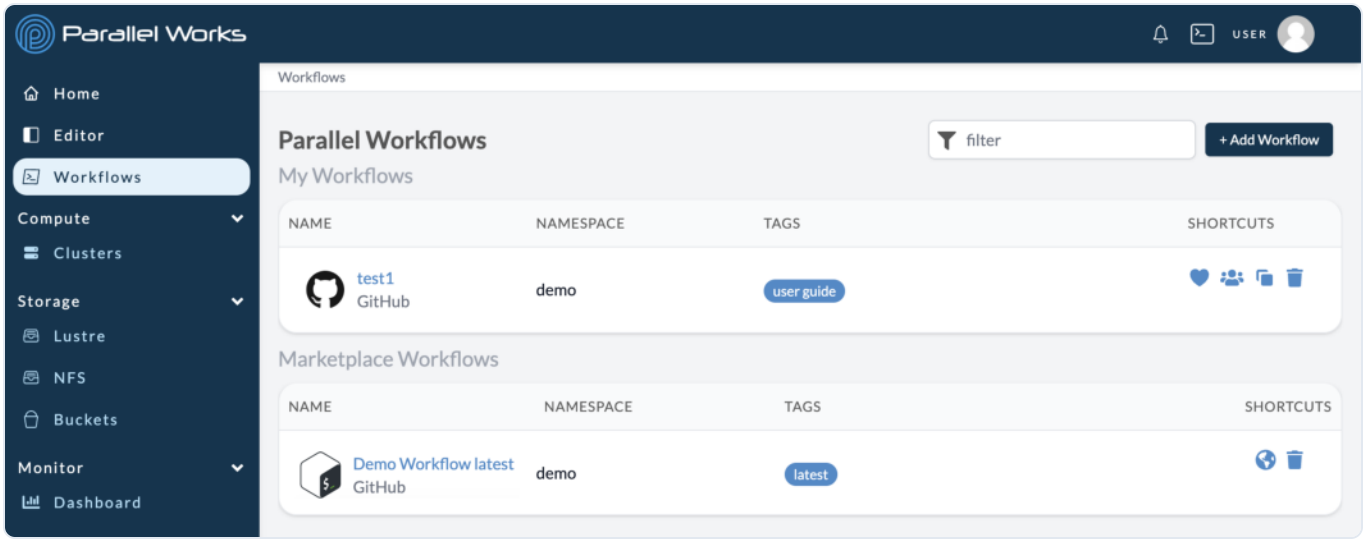


Click **Use latest version**.



The **Use latest version** button will turn green, and the message *Using version successfully* will appear in the bottom right corner of your screen.

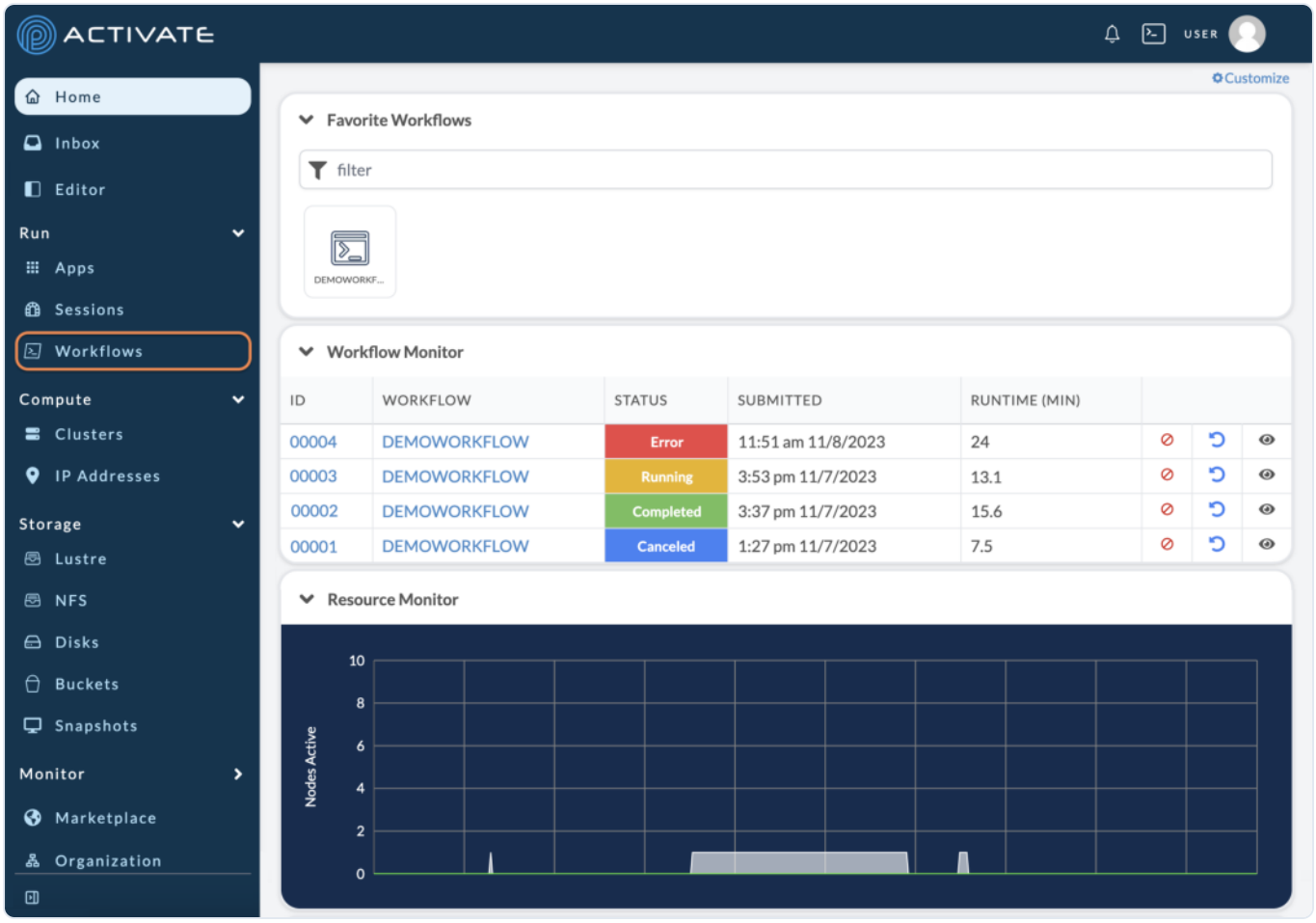
On the **Workflows** page, the workflow will appear in the **Marketplace Workflows** section. The workflow's version will be included in its **Name** and **Tags**.



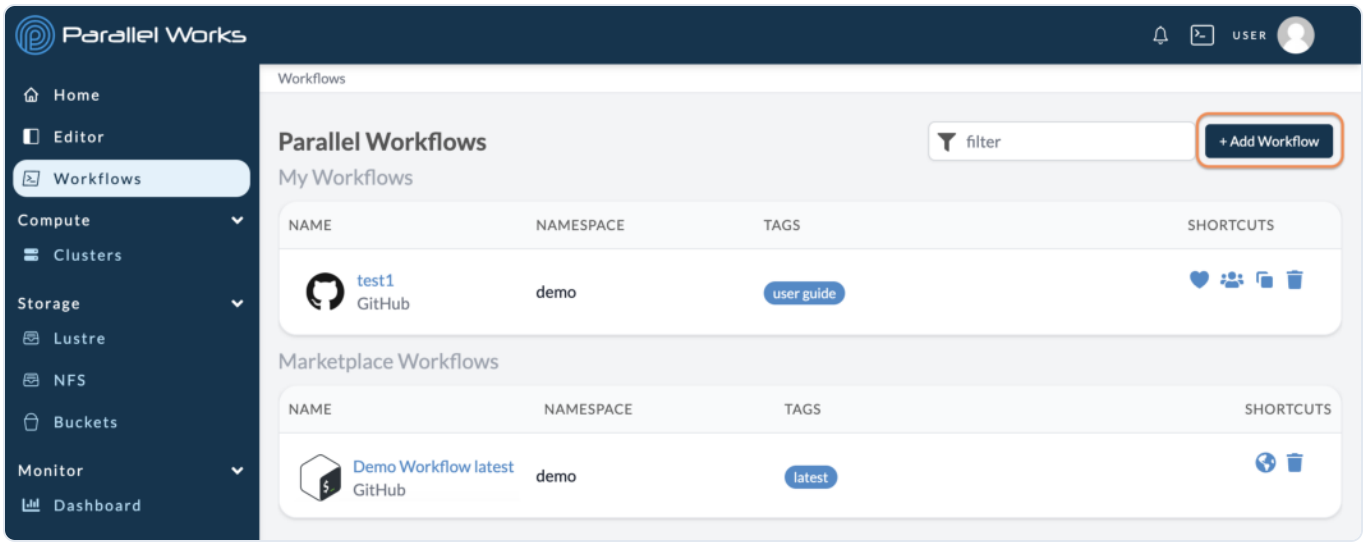
Creating Local Workflows

Local workflows have their YAML defined within the ACTIVATE platform.

Navigate to the **Workflows** page.



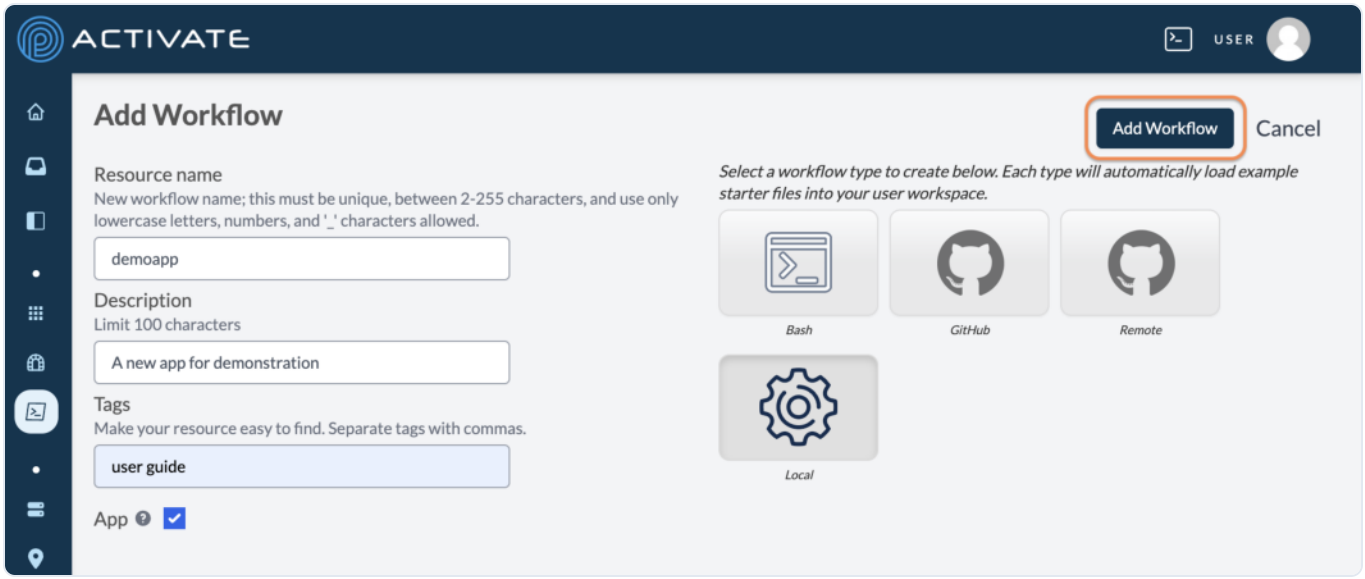
Click + **Add Workflow**.



On the next page, configure your workflow by selecting the **Local** workflow type.

Enter a **Workflow Name**. Optionally, enter a **Description** and **Tags**.

Click **Add Workflow**.

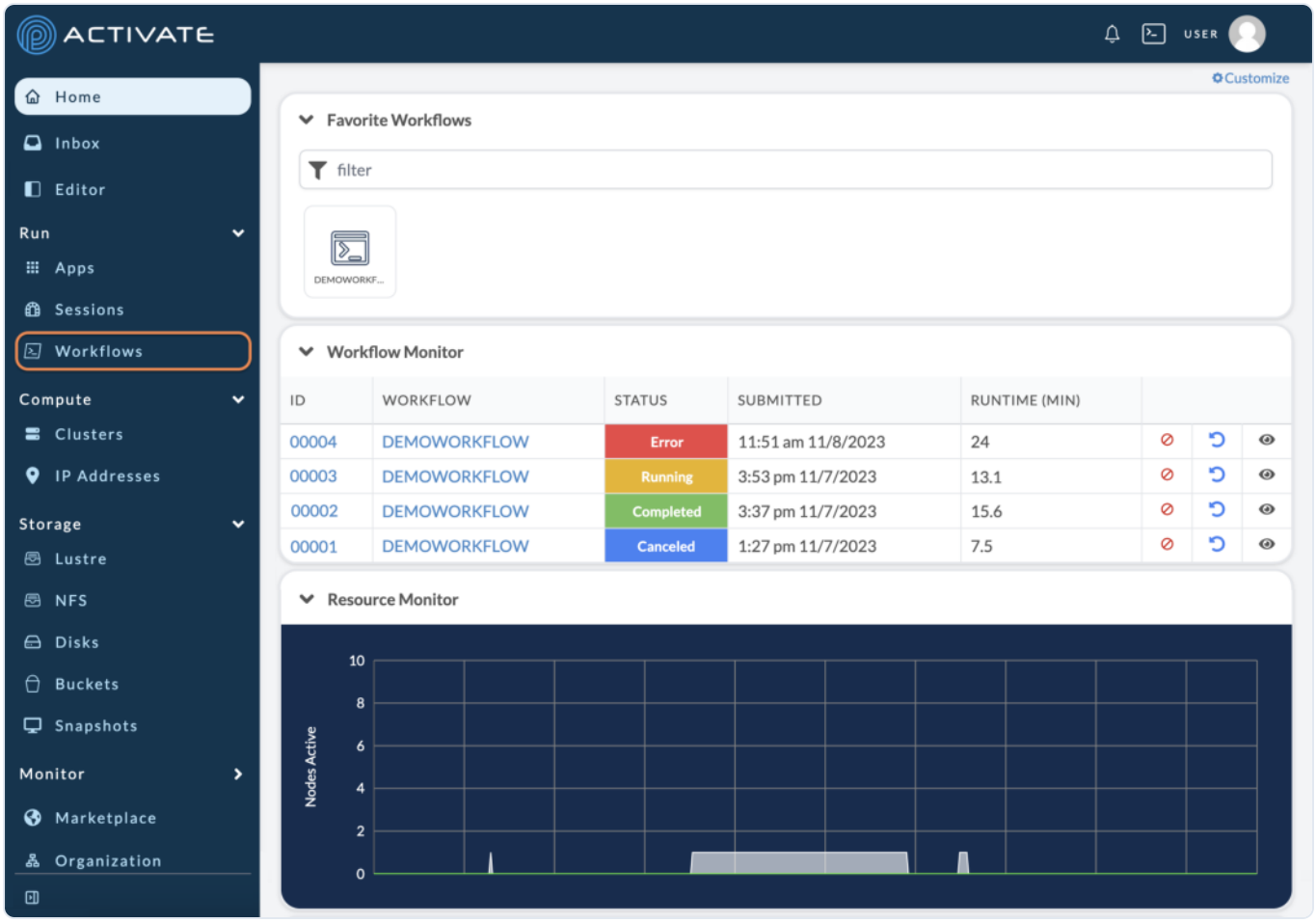


For the next steps, please see **Building Workflows**.

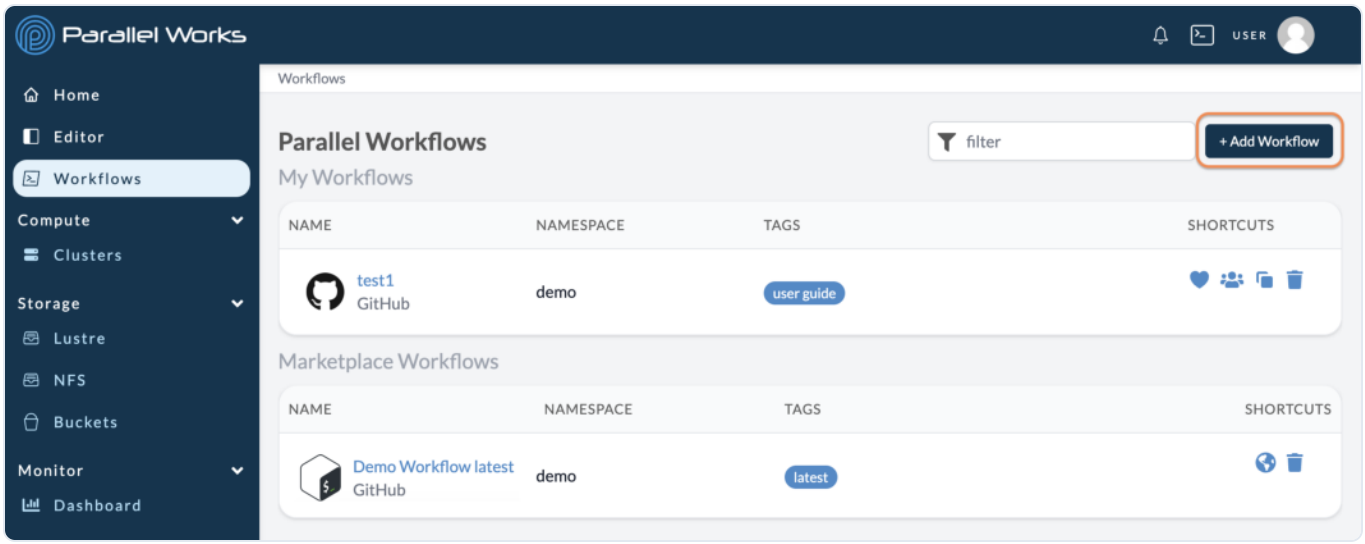
Creating Remote Workflows

Remote workflows have a remotely defined YAML file and import files from a remote repository every time the workflow is executed. With this approach, workflow files are updated automatically whenever changes are saved in the remote repository.

Navigate to the **Workflows** page.

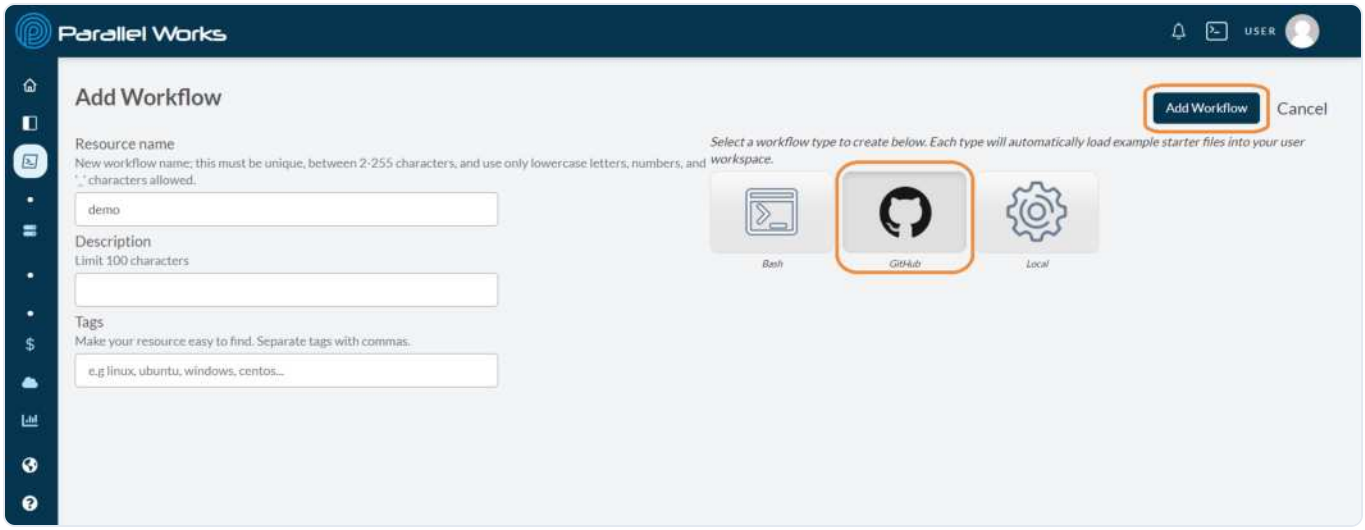


Click + **Add Workflow**.



On the next page, configure your workflow by entering a **Workflow Name** and selecting **Remote** as your workflow type. Optionally, enter a **Description** and **Tags**.

Click **Add Workflow**.

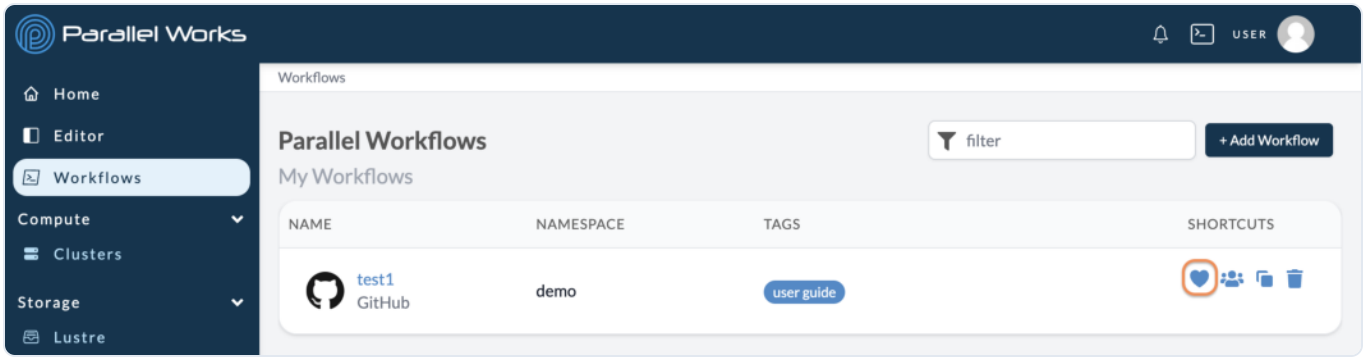


For the next steps, please see **Building Workflows**.

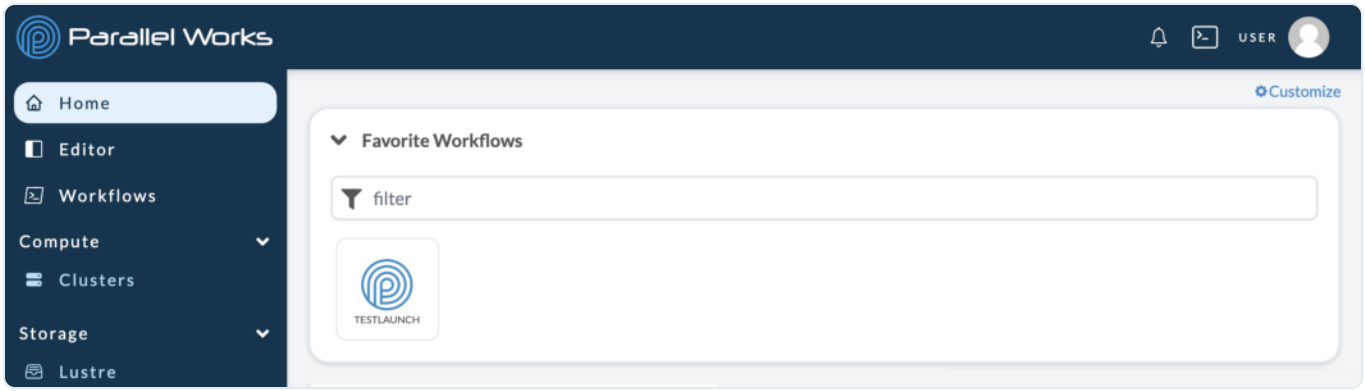
Favoriting Workflows

You can add frequently used workflows to your **Dashboard** for easy access.

On the **Workflows** page, click the slashed heart icon to favorite a workflow. The icon will change to a plain heart after you click it.



The workflow will now appear in **Favorite Workflows** on the **Dashboard**.

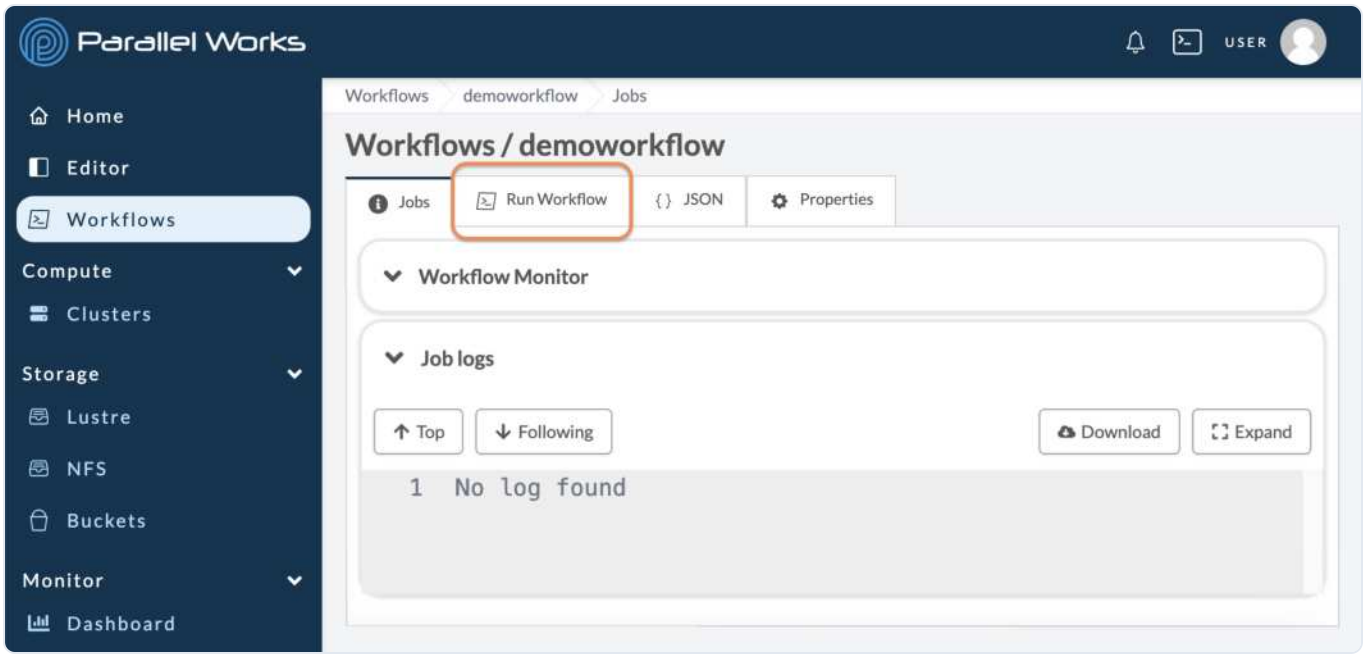


Running Workflows

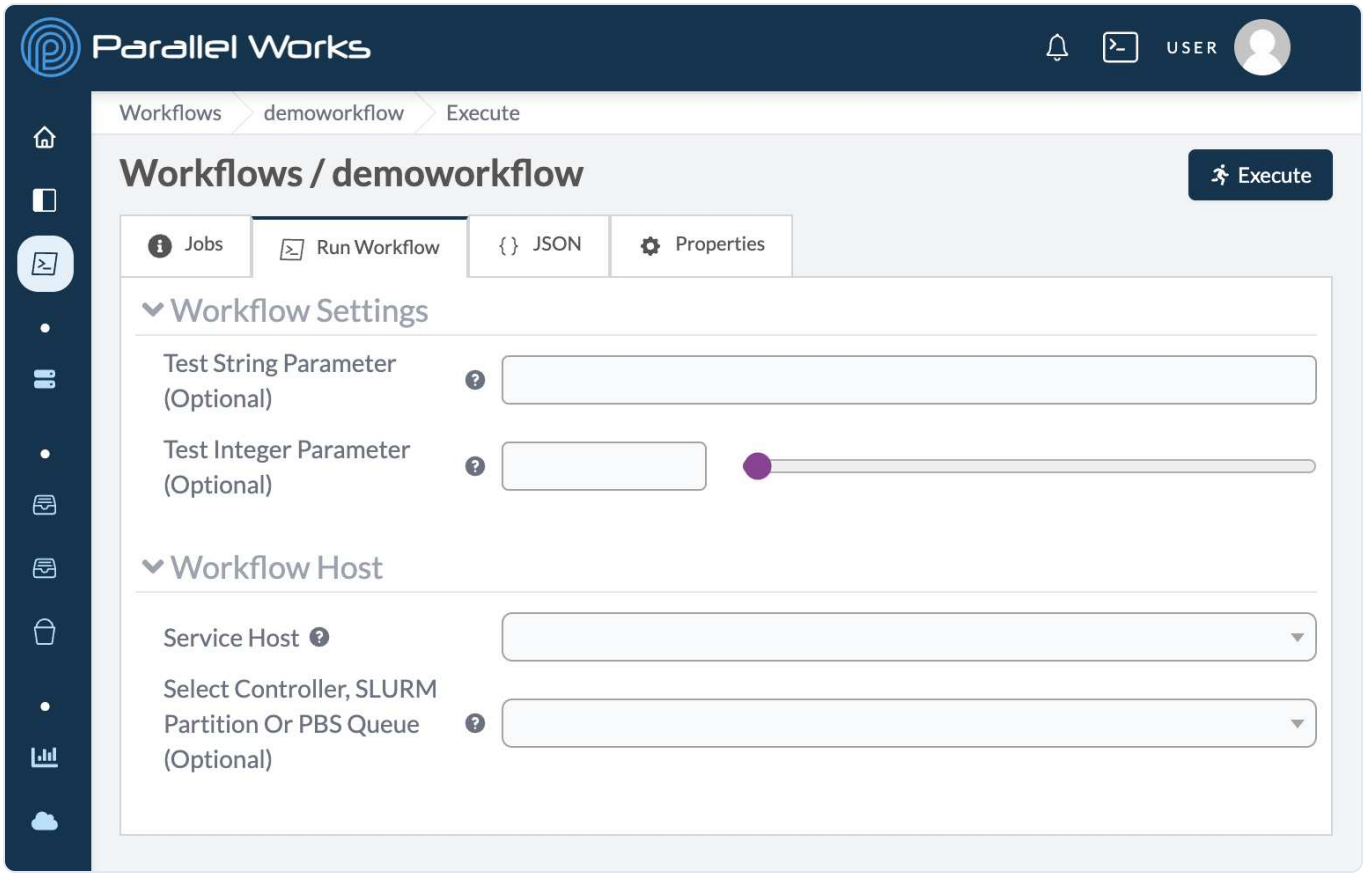
> **ABOUT JOBS**
>
> There are two ways you can submit jobs to a cluster: by using workflows or through any terminal or command-line interface. For the command-line interface option, please see **Submitting Jobs via Slurm**.

After you've started a cluster, navigate to the **Workflows** page and select the workflow you'd like to use. If you haven't added any workflows, you can select one from the Marketplace.

When you click the name of your workflow, you'll be taken to the workflow's **Jobs** page. Click the **Run Workflow** tab.



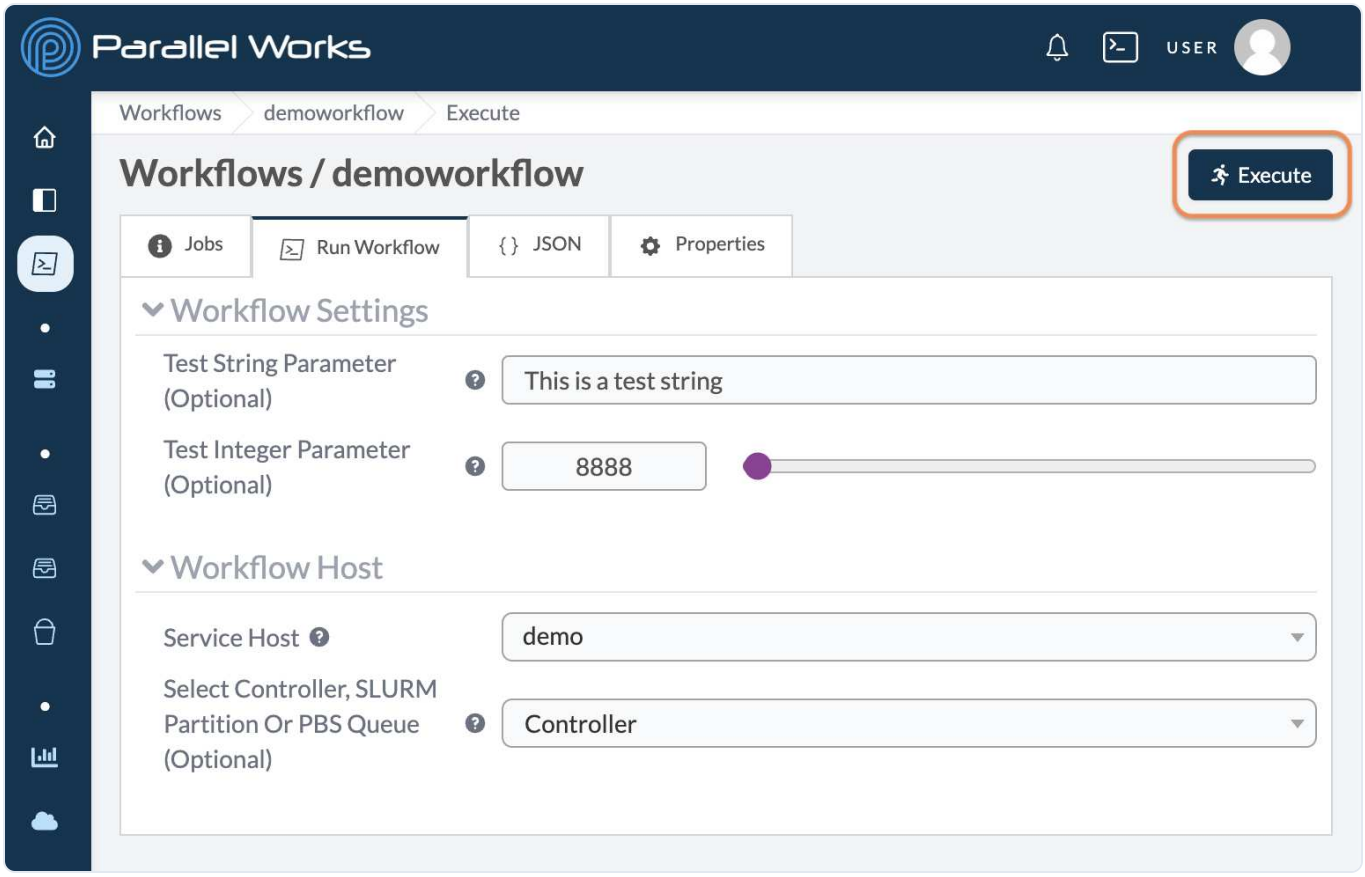
The **Run Workflow** tab is where you enter the inputs used in the workflow.



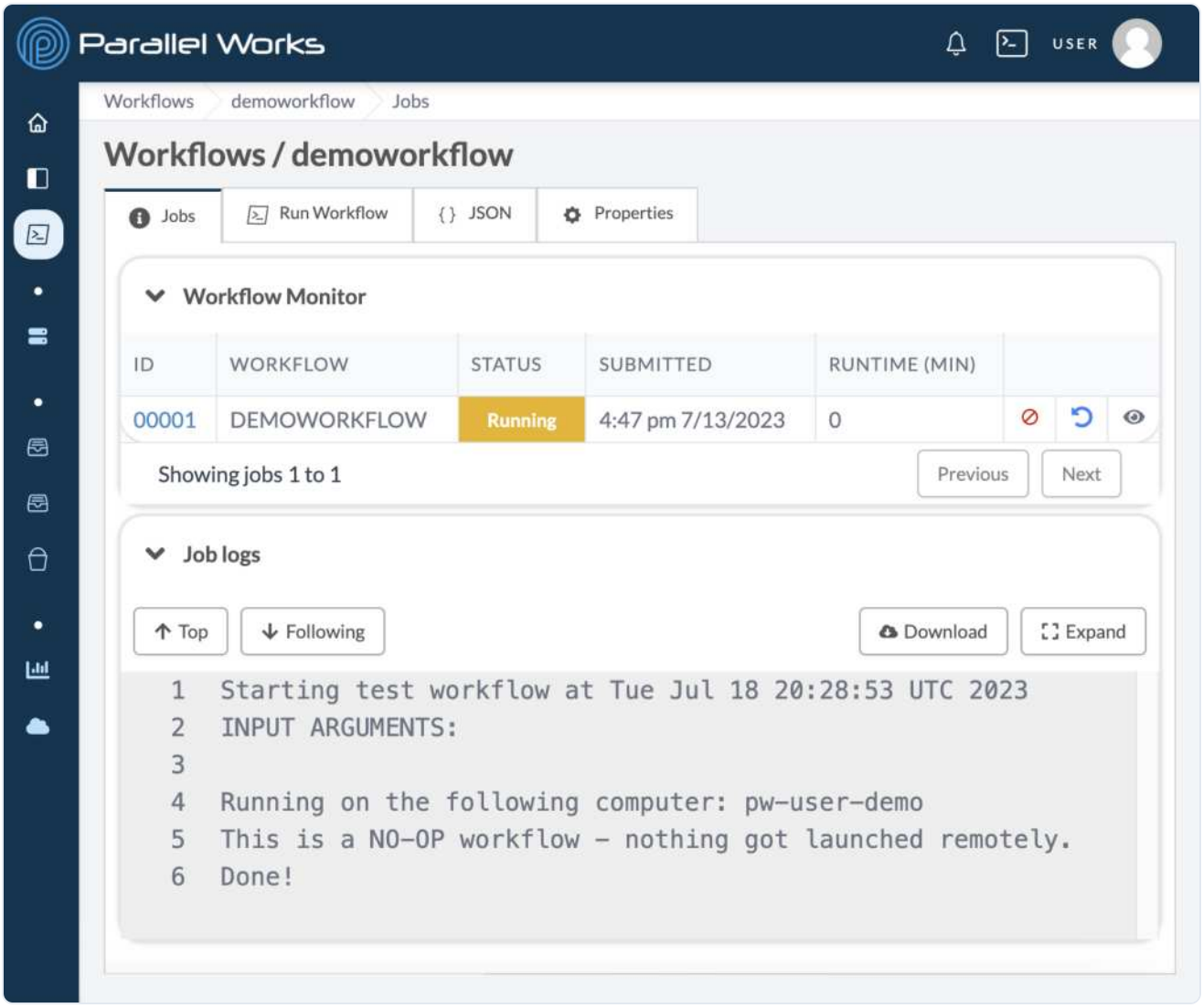
In the **Workflow Settings** section, enter any necessary parameters. The example workflow pictured has only two test parameters, but your selected workflow may have more or fewer options.

In the **Workflow Host** section, select your running resource and how the job will be submitted. In this example, we ran our workflow on the resource **Demo**. We submitted the job to Demo's **controller** because we used a small workflow. You might submit your job to a partition, Slurm, or PBS, depending on your resource's configuration and your workflow's needs.

Click **Execute**.



You'll be taken back to the workflow's **Jobs** tab, where you can monitor your workflow's progress.



The **Workflow Monitor** shows your workflow's status. In this example, the monitor shows **Completed** with a **Runtime** of 0 because the demo workflow ran in less than a second, bypassing the **Running** status. Workflows that create larger or more complex jobs will show **Running** for longer periods of time before showing **Completed**.

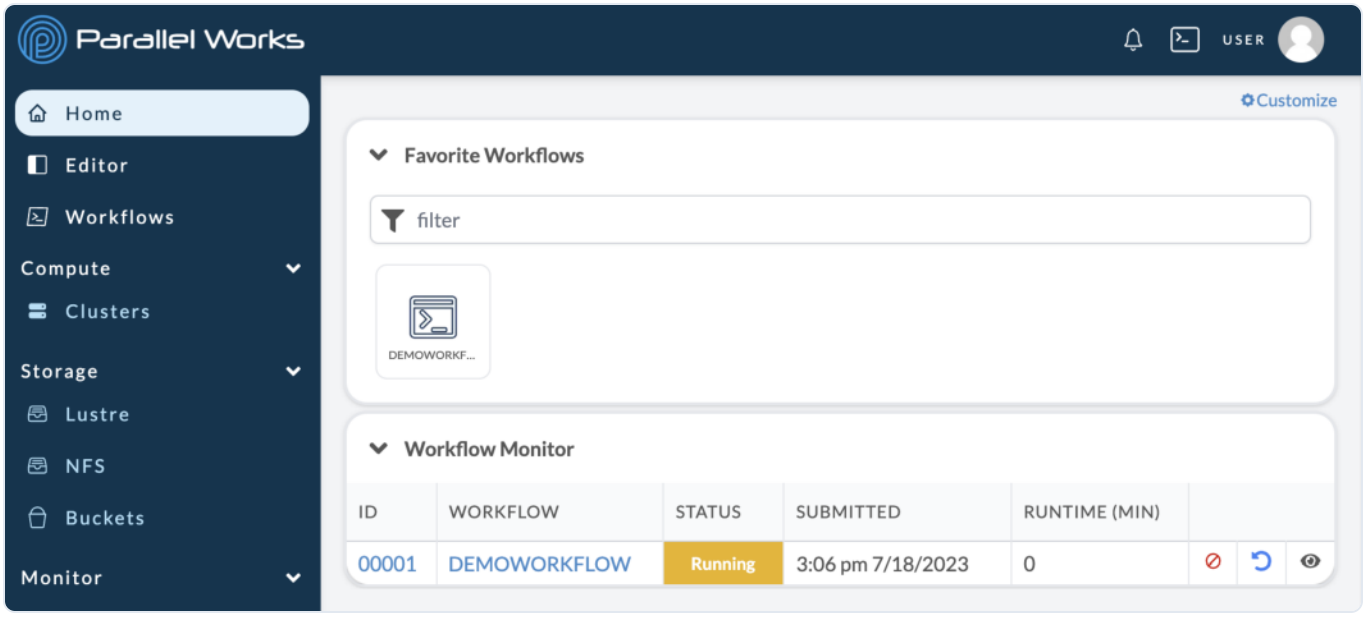
The **Job logs** module shows your job's output as well as details about your workflow's session.

In this example, we ran a simple `bash` script that returns the running resource's hostname, so **Job logs** displayed the following output:

```
Starting test workflow at Wed Jan 25 17:12:21 UTC 2023
INPUT ARGUMENTS:

Running on the following computer: pw-user-demo
This is a NO-OP workflow - nothing got launched remotely.
Done!
```

In addition to your workflow's output, logs also include start times and messages for job submissions, job completions, and job errors if any occur.

If you have the **Workflow Monitor** widget on your **Dashboard**, it will also reflect your job's status.

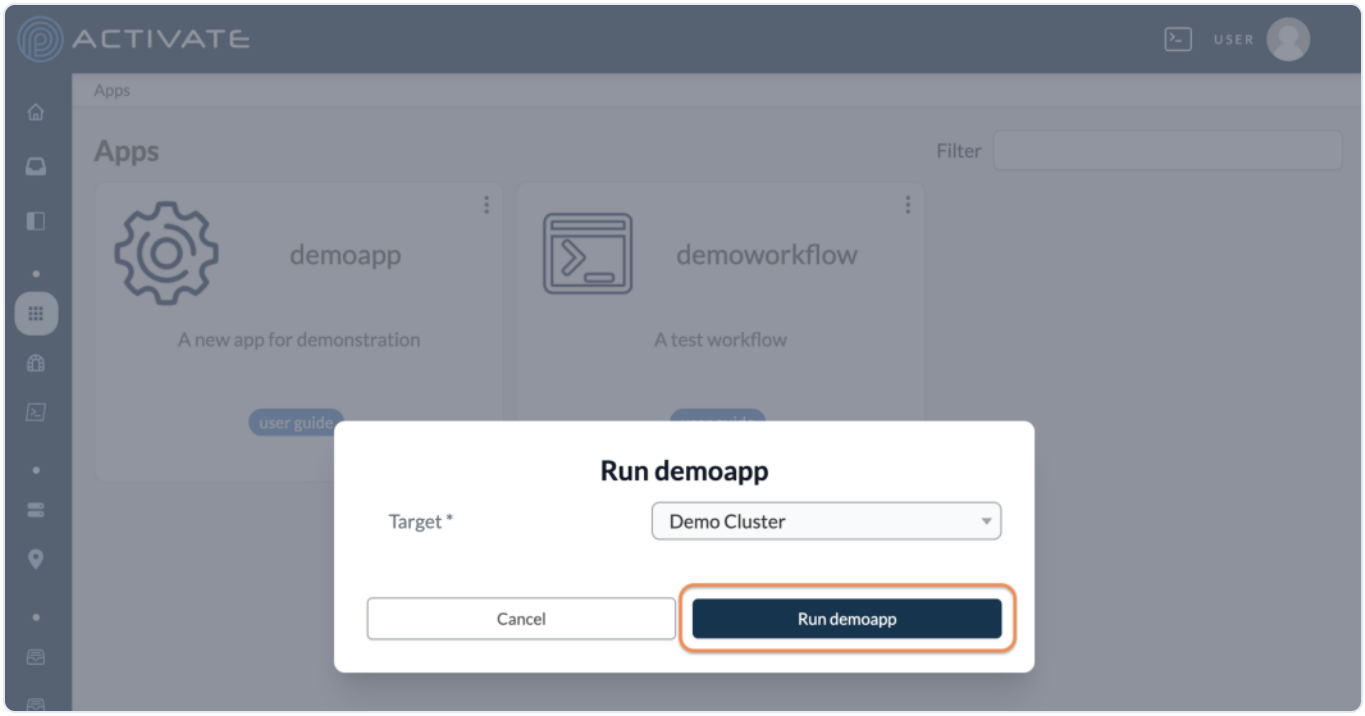


From the **Workflow Monitor**, you can click a workflow's **ID** number to return to its **Jobs** tab.

> **ABOUT WORKFLOW PARAMETERS**
>
> The example workflow used here has only two parameters meant for testing purposes. Many workflows on ACTIVATE have additional parameters, especially those that execute multiple or more complex tasks. If you need help with setting parameters, you can reach out to your admin or to us at any time.

Monitoring Workflow Runs

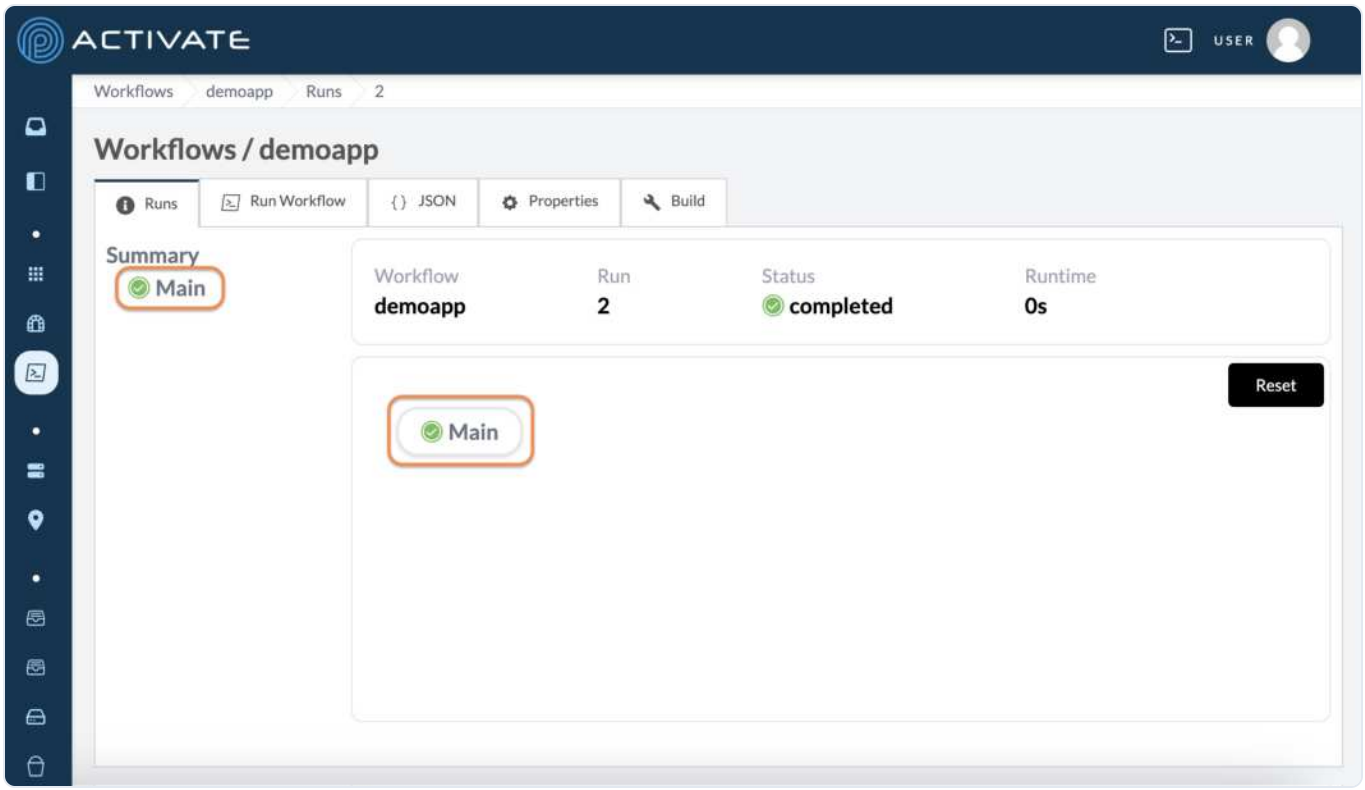
After starting a workflow, you can get more details about a run by clicking it.



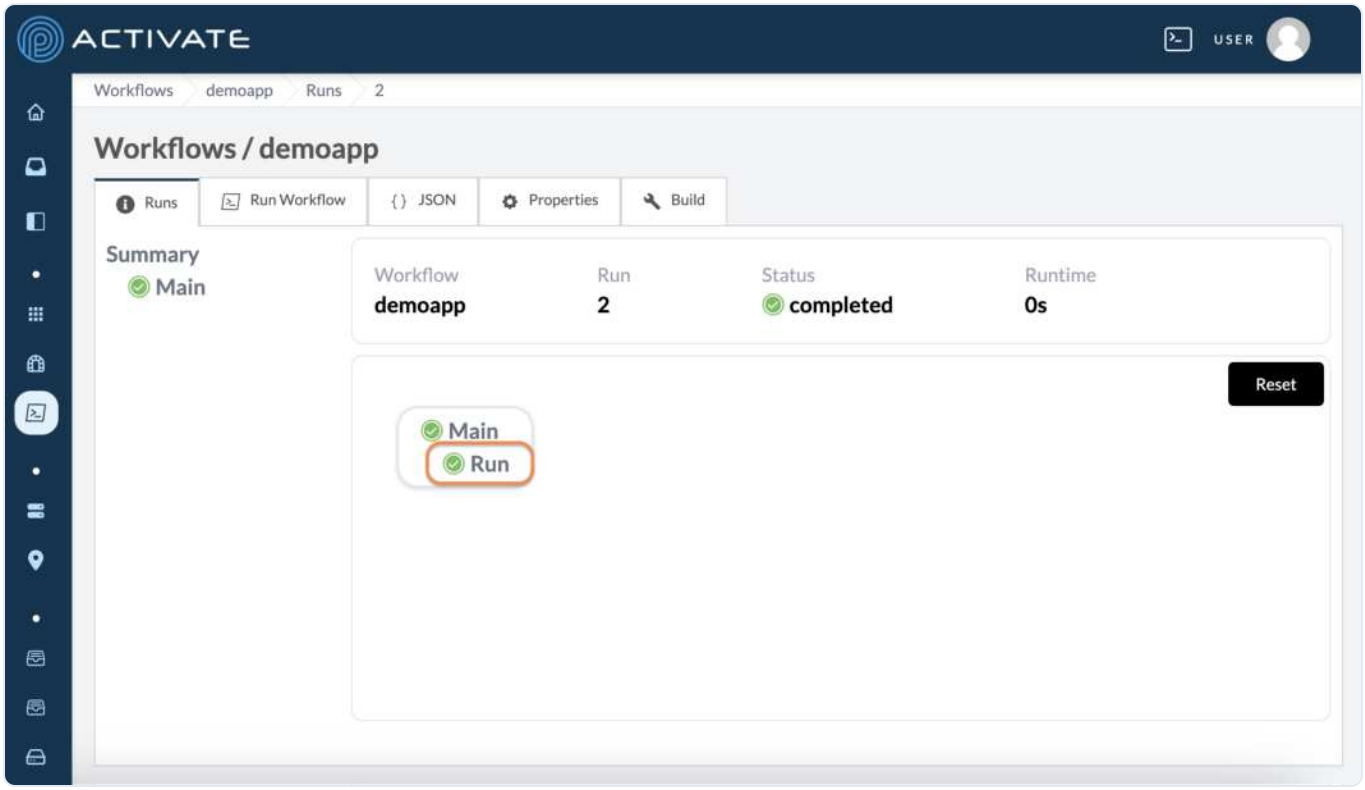
The next page displays a **Summary** of the workflow's steps, as well as the **Workflow** name, the **Run** number, its **Status**, and its **Runtime**.

You can click, drag, zoom in, and zoom out inside the mapping module. Click **Reset** to revert your changes.

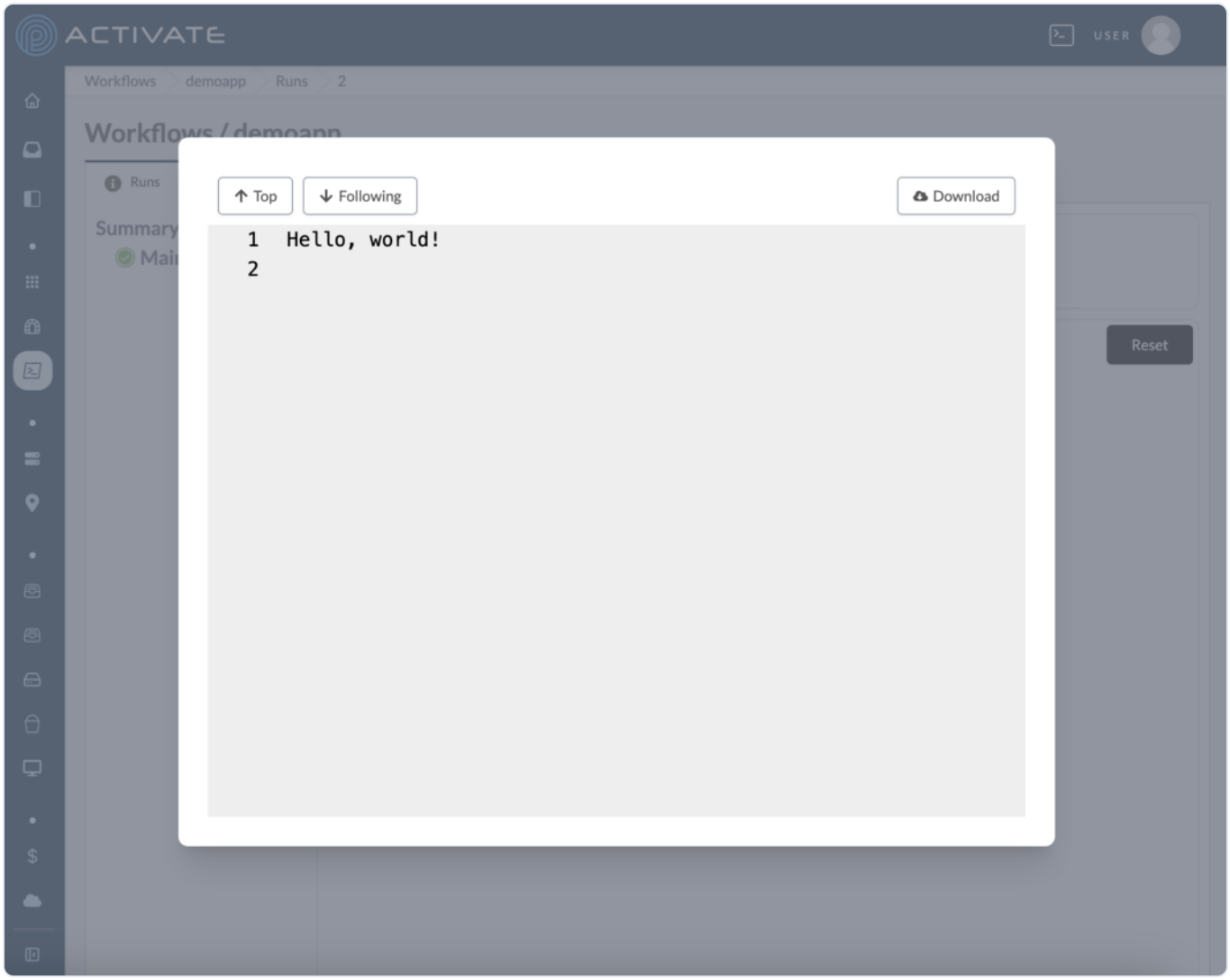
To expand a step, click it either in the sidebar or in the mapping module.



Click a sub-step to see more information.



A pop-up log module will appear.



By default, the module is set to **Following**. This means the module will display individual log lines at the top as they appear. Click **Following** to disable this setting.

Click **Download** to save a copy of the log data. The file will download as a document, which can be opened with any text or code editor.

Click anywhere outside the module to close it.

Building Workflows

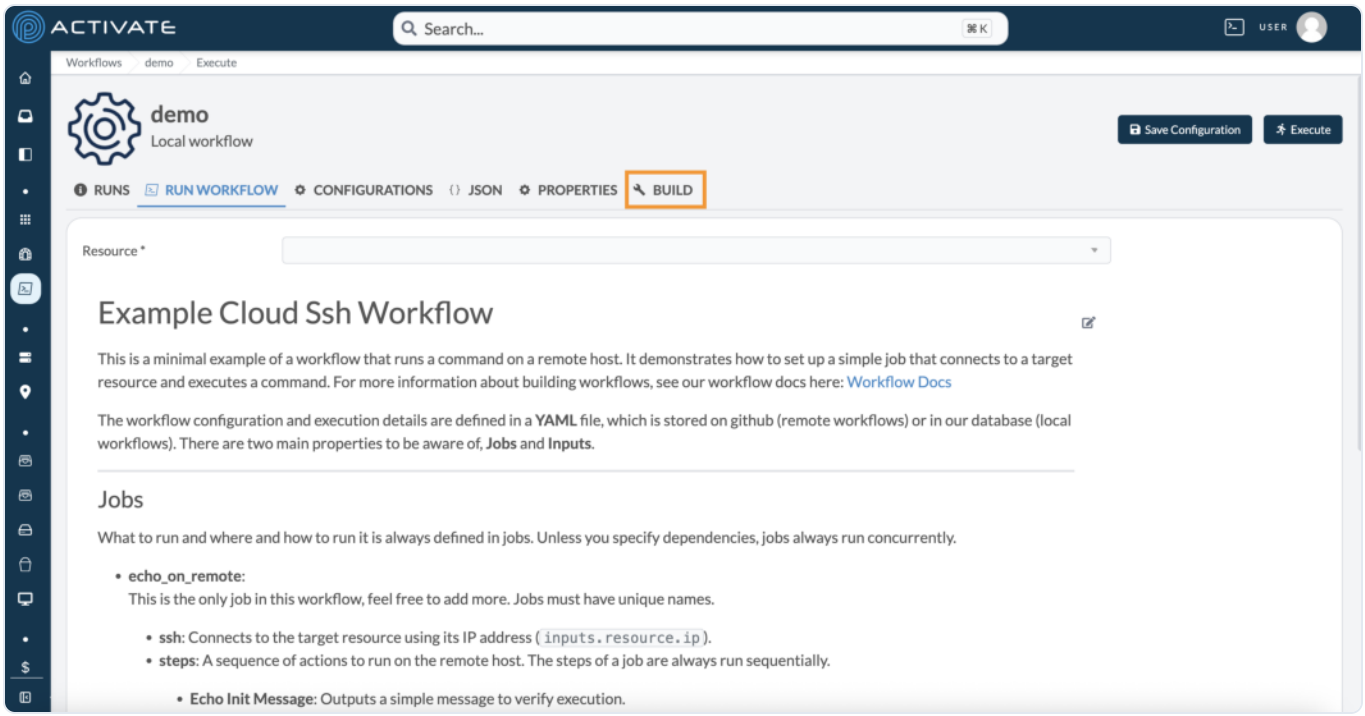
Building Workflows

The input parameters that appear in the **Run Workflow** tab are defined by a YAML (`.yaml`) file. The YAML for local workflows is stored by the platform, so you can view and edit the YAML directly by using the **Build** tab. From there, you can define the jobs and inputs for your workflow.

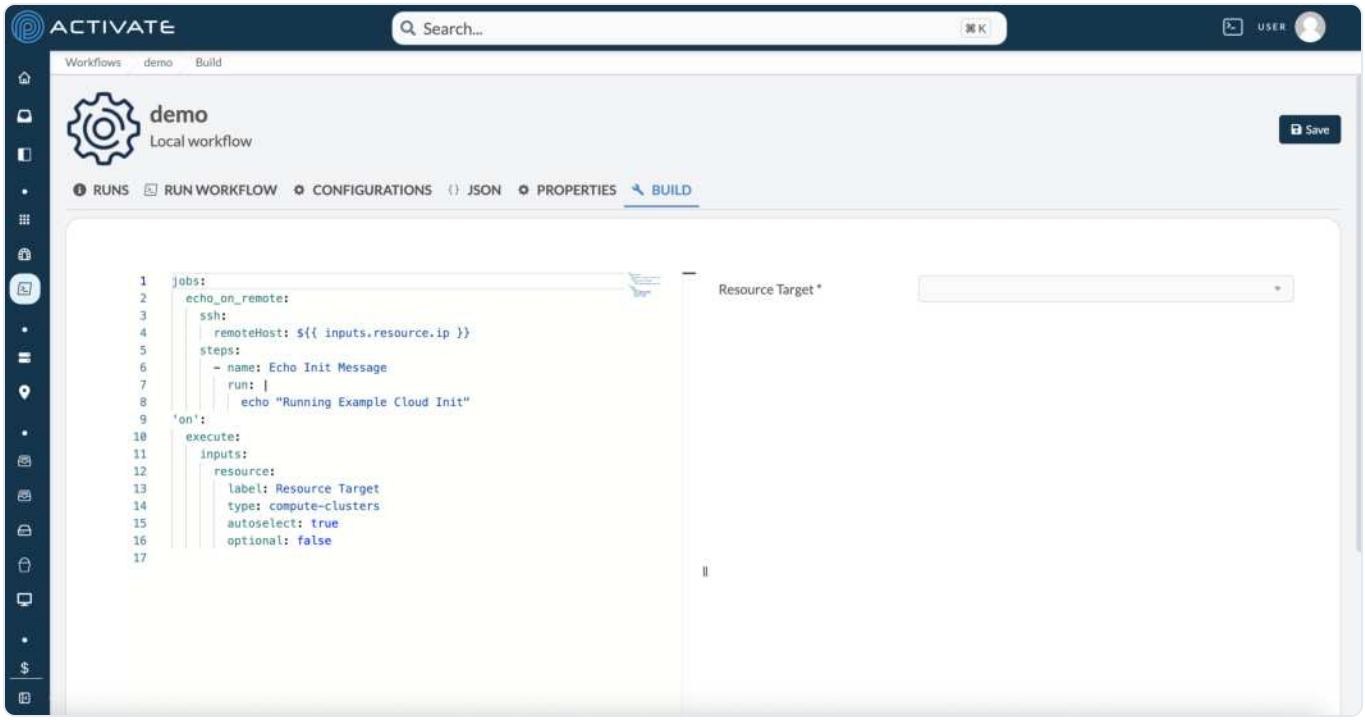
> **YAML SYNTAX**

> Since the workflows on ACTIVATE are written in YAML, it's worthwhile to get a grasp on YAML basics before diving into building workflows. For a quick overview, check out the Red Hat website.

Navigate to **Workflows** > your workflow > the **Build** tab.



Newly created local workflows are loaded with a default YAML file configuration, as seen in the screenshot below.



WORKFLOW BUILDER TIPS

You can get information on what possible fields are allowed in a certain spot with the shortcut `Ctrl+Space`. Many fields, especially within jobs, have a description which you can read by hovering your cursor over the name of the field.

Ctrl+Space	Cursor Hover
	

Yaml Fields

See the building workflows page if you have not already. If you are looking for the documentation for a specific field, find the name of the field from the sidebar on the right. This page documents all of the schema fields of the workflow yaml, with the exception of input fields (under `on.execute.inputs`) which can be found here.

`jobs`

The most essential part of the YAML configuration is `jobs`, which defines the jobs this workflow will run. Each job must define a list of steps to be executed using the `steps` field. By default, all jobs run in parallel. If a job needs to depend on the completion of another job, it can be specified using the `needs` field. You must have at least one job containing at least a single step for a workflow to be valid. Job names must be unique within a workflow.

`jobs` take the following shape in the YAML file:

```yaml
jobs:
  job-name:
    steps:
      - name: Step 1
        run: echo step 1
      - name: Step 2
        run: echo step 2
```

`jobs..needs`

`needs` defines a list of jobs that must complete before this job can start. If a job fails, all jobs that depend on it will not run.

In the following example, the job `first` will run, and only after it successfully completes will the job `second` run.

```
jobs:
  first:
    steps:
      - run: echo runs first
  second:
    needs:
      - first
    steps:
      - run: echo runs second
```

`jobs..steps`
A list of steps to be executed. These are executed in order, and if any step fails, the job will fail. Each step requires either a command to execute using the `run` field, or another workflow to execute by using the `uses` field. All other attributes are optional.

You can define `steps` with the attributes below.

`jobs..steps[*].name`
The name of the step.

Step names are useful when viewing a workflow's progress in the **Runs** tab. If a job fails at a certain step, you can quickly identify its name on the **Runs** graph.

If you don’t provide a name, the step's command will be shown instead.

`jobs..steps[*].run`
Defines the command to be executed, e.g. `echo hello world`. This cannot be used in conjunction with `uses`. It is possible to combine bash commands into a single step:

```
jobs:
  main:
    steps:
      - run: |
          echo part 1
          echo part 2
```

`jobs..steps[*].uses`
Runs another workflow as a step, e.g. `marketplace/<marketplace slug>`. This cannot be used in conjunction with `run`. You may also use `workflow/<workflow name>` to use one of your personal

workflows. Additionally, when using `marketplace/<marketplace slug>`, a version may be specified after another slash, like so: `marketplace/<marketplace slug>/<version>`, where version probably looks somethng like `v1.0.0`. If a version is not specificed, the latest version published on the marketplace is used.

`uses` calls an existing workflow either from another workflow or from the Marketplace. To reference another workflow, prefix the workflow name with `workflow/`. To reference a Marketplace workflow, prefix the marketplace workflow slug with `marketplace/`. To reference a workflow published in a GitHub repository, use `github/<owner>/<repo>@<ref>`, where `<ref>` is a branch, tag, or commit SHA.

If the workflow requires inputs to run, use `with` to provide those inputs. The fields defined inside `with` are the same names as defined in the workflow's `inputs` section.

There are also two rules that must be followed when running workflows within workflows:

- A workflow may never call itself, even indirectly through another workflow.
- There is a limit to how deep the workflow call stack can get. Only up to 5 layers of workflows calling workflows are supported.

Attempts to run the workflow will fail immediately if these rules are violated.

The `uses` field is also used to execute **actions**.

`jobs..steps[*].with`
Defines the inputs passed to the workflow defined by `uses`.

If you include `uses` in the YAML file to call a workflow, it may have inputs that must be defined in order to run. In that case, `with` is used to define those inputs. In the example below, we used the inputs from the default YAML configuration.

```
jobs:
  job-name:
    steps:
      - name: Sample Step
        uses: marketplace/default-local-workflow
        with:
          resource: sample-cluster
```

[`jobs..steps[*].ssh`](#jobsjobssh)
`jobs..steps[*].cleanup`
Defines a cleanup command to be run after step execution finishes, e.g. `rm -rf /tmp/myapp`.

This can be used to delete temporary files, terminate connections, remove credentials, or perform any any other clean up necessary when a job is shutting down. `cleanup` always runs at the end of a job in reverse order of definition. Clean up steps always run regardless of if a job was successful, failed, or was canceled. If a step did not run, it's cleanup step will not run.

For example, if you have a job with three steps that all have `cleanup` attributes, each `cleanup` will add a clean up step to the end of the job and the job will ultimately execute steps in this order:

- step 1 `run`
- step 2 `run`
- step 3 `run`
- step 3 `cleanup`
- step 2 `cleanup`
- step 1 `cleanup`

[`jobs..steps[*].if`](#jobsjobif)
`jobs..steps[*].retry`
If a step has a certain chance of failure but is necessary for your workflow, using the `retry` field is recommended. `retry` is an object with the following properties:

- `max-retries`: The number of retries before giving up. Defaults to 0 without a retry object, and 10 with a retry object but no max-retries field.
- `interval`: The amount of time to wait between retries. Defaults to 5s.
- `timeout`: The amount of time to wait before giving up on an attempt. Defaults to 30s.

The supported units for `interval` and `timeout` are:

- `n` (nanoseconds)
- `s` (seconds)
- `m` (minutes)
- `h` (hours)
- `d` (days)

[`jobs..steps[*].env`](#env)
`jobs..steps[*].ignore-errors`
If true, non-zero exit codes will not be counted as failures. By default, this attribute is set to `false`.

[`jobs..steps[*].working-directory`](#jobsjobworking-directory)
`jobs..steps[*].early-cancel`
Conditions for early cancellation of the step. Right now, only `early-cancel: any-job-failed` is supported, which cancels the step if any job fails before the step finishes running.

[`jobs..steps[*].timeout`](#timeout)
`jobs..ssh`
If you want the workflow to execute commands on a remote host, using the `ssh` field is recommended. `ssh` is an object with the following properties:

- `remoteHost` : The ip address of the remote host.
- `remoteUser` : The username to utilize when attempting to ssh to the remoteHost.
- `jumpNodeHost` : The ip address of the (optional) jump node.
- `jumpNodeUser` : The username to utilize when attempting to ssh to the jump node.

However, you generally will not need to use the `remoteUser` and `jumpNodeUser` fields, as they will be populated automatically. The step-level `ssh` overwrites the job-level `ssh`, and you may also pass `ssh: null` (or just `ssh:` with nothing else) at the step level to execute on the user workspace (the default behavior without ssh set).

Example:

```
jobs:
  echo_on_remote:
    ssh:
      remoteHost: ${{ inputs.resource1.ip }}
      jumpNodeHost: ${{ inputs.resource2.ip }}
    steps:
      - run: echo This is executed on the remote host!
      - run: echo This is executed on the jump node!
        ssh:
          remoteHost: ${{ inputs.resource2.ip }}
      - run: echo This is executed in the user workspace!
        ssh: null
on:
  execute:
    inputs:
      resource1:
        label: Resource 1
        type: compute-clusters
        optional: false
      resource2:
        label: Resource 2
        type: compute-clusters
        optional: false
```

`jobs..if`

`if` prevents a job/step from running unless a conditional evaluates to true.

Example:

```
jobs:
  main:
    steps:
      - run: echo hello world
      - run: echo This ran because of an input!
        if: ${{ inputs.should-run }}
  extra:
    if: ${{ inputs.should-run }}
    steps:
```

```
      - run: echo This ran because of an input!  on:  execute:    inputs:      should-run:
type: boolean        label: Run extra step + job?
```

In the above example, the second step and second job will only run if the user selects **Yes** on "Run extra step + job?".

The `if` field also accepts `${{ always }}`, which for jobs ensures that it runs even when one of its dependencies failed, and for steps ensures that it runs even when a previous step in the job failed. Unlike `cleanup`, steps with `if: ${{ always }}` can be cancelled and are run in order rather than in reverse order. So in this example:

```yaml
jobs:
  main:
    steps:
      - run: fail
        cleanup: echo fifth
      - run: echo first
        if: ${{ always }}
        cleanup: echo fourth
      - run: echo second
        if: ${{ always }}
        cleanup: echo third
```

The commands will be executed in this order:

- `fail`
- `echo first`
- `echo second`
- `echo third`
- `echo fourth`
- `echo fifth`

[`jobs..env`](#env)
`jobs..working-directory`
Defining this field changes the directory that `run` commands are run in from the default, which is the job directory (~/pw/jobs/workflow-name/job-number/). If the path does not exist before the command is run, it will be created. Step-level `working-directory` overwrites job-level.

[`jobs..timeout`](#timeout)
`jobs..strategy`
Runs a job multiple times across a set of variable combinations (a "matrix"). Each combination produces an independent run of the job that can be referenced by matrix keys in expressions.

`strategy.matrix` defines the variables. Each top-level key is a variable name mapped to an array of values. The job runs once for every combination of all variables.

```yaml
jobs:
  test:
    strategy:
      matrix:
        os: [ubuntu, centos]
        version: [1, 2, 3]
    steps:
      - run: echo ${{ matrix.os }} v${{ matrix.version }}
```

The example above runs the job six times: `ubuntu v1`, `ubuntu v2`, `ubuntu v3`, `centos v1`, `centos v2`, `centos v3`.

`strategy.matrix.include` adds extra combinations (or augments existing ones with extra fields):

```yaml
strategy:
  matrix:
    os: [ubuntu, centos]
    version: [1, 2]
    include:
      - os: ubuntu
        version: 2
        experimental: true
      - os: fedora
        version: 3
```

`strategy.matrix.exclude` removes combinations:

```yaml
strategy:
  matrix:
    os: [ubuntu, centos]
    version: [1, 2, 3]
    exclude:
      - os: centos
        version: 1
```

`strategy.fail-fast` (default `true`) cancels all still-running matrix jobs the moment one fails. Set to `false` to let every combination finish regardless.

`strategy.max-parallel` caps how many matrix jobs run concurrently. If not set, all combinations run in parallel.

```yaml
strategy:
  fail-fast: false
  max-parallel: 2
  matrix:
    region: [us-east-1, us-west-2, eu-west-1]
```

`sessions`
This field is used to define the sessions that will be used when running the workflow. For an example usage, see Workflow Sessions.

`sessions..type`
What type of session to create (tunnel by default). The only possible options are tunnel and link.

`sessions..openAI`
If true, will mark the session as providing an OpenAI API, and connect it to the built-in chat interface.

`sessions..prompt-for-name`
If `prompt-for-name` is `null`, the user will be prompted to name the session before workflow execution. If `prompt-for-name.default` is defined, the passed default will be used.

`sessions..redirect`
If true, the user will be redirected to this session once the workflow is executed. Only one session is allowed to have this set to true.

`sessions..useTLS`
If true, will use HTTPS to connect to the session. This should only be enabled if the application requires it.

`sessions..useCustomDomain`
If true, will use a custom domain to connect to the session. This should only be enabled if the application requires it.

`sessions..detach`
If true, the session will persist after the workflow run reaches a terminal status (completed, error, canceled) and will only be removed when the user manually deletes it. Defaults to `false`.

`permissions`
Workflow runs can be granted additional access via `permissions`. Adding the `*` permission will allow the workflow to do anything a user would be able to. Without the `*` permission the workflow will only be able to update any sessions it creates.

```
permissions: ["*"]
jobs:
  main:
    steps:
```

```
      - name: Print buckets        run: pw buckets ls
```

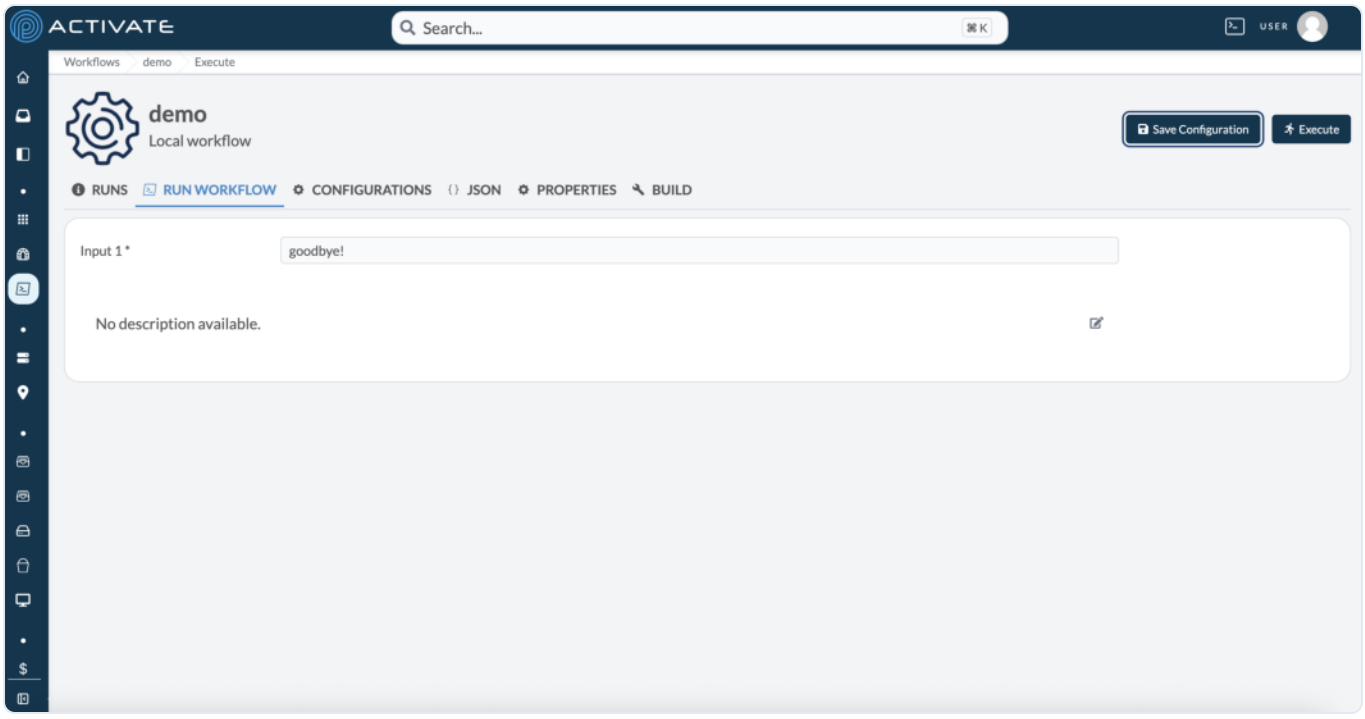
Without the permission, `pw buckets ls` would not allow the workflow to see all of the buckets.

`configurations`

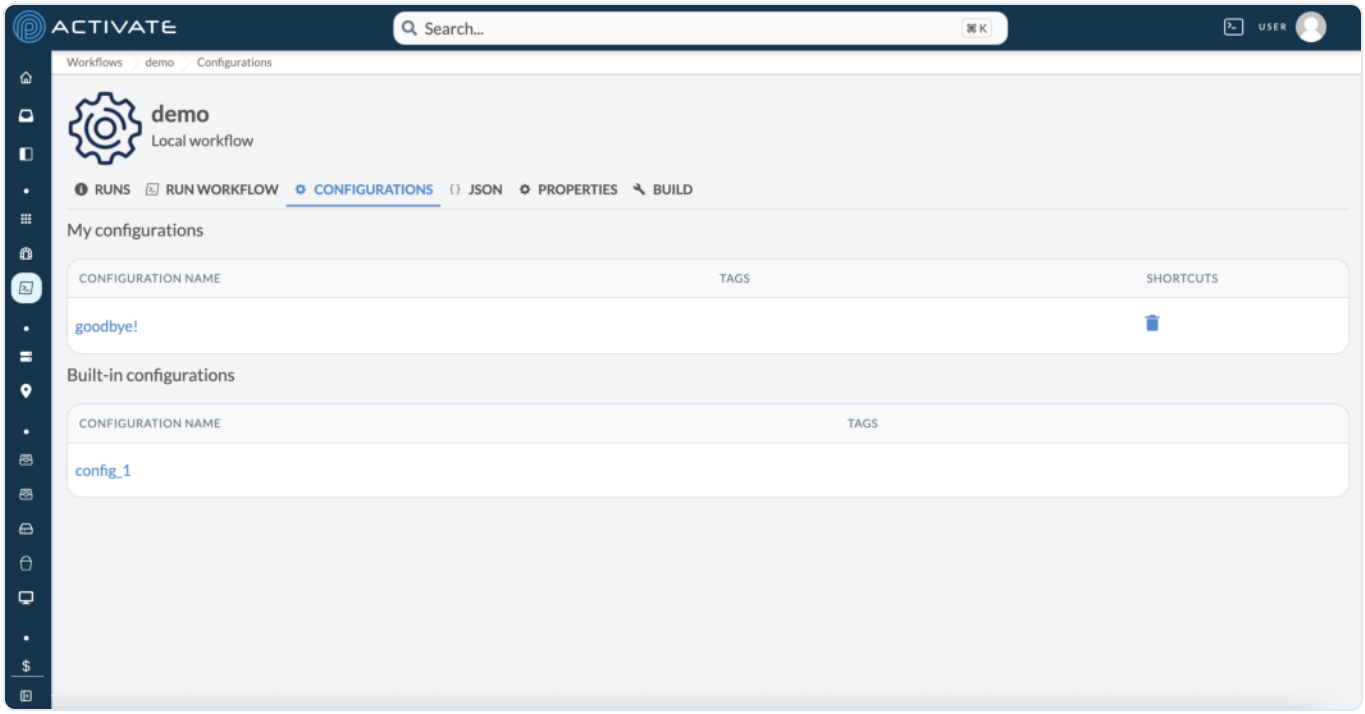
Configurations are saved inputs that can be used when running a workflow, as opposed to manually filling out the form, to save time and ensure consistency. Configurations can be saved by users of a workflow, but can also be defined in the workflow yaml file. Here is a basic example of a definition in yaml:

```yaml
configurations:
  config_1:
    inputs:
      input_1: hello!
jobs:
  echo:
    steps:
      - run: echo ${{ inputs.input_1 }}
'on':
  execute:
    inputs:
      input_1:
        type: string
```

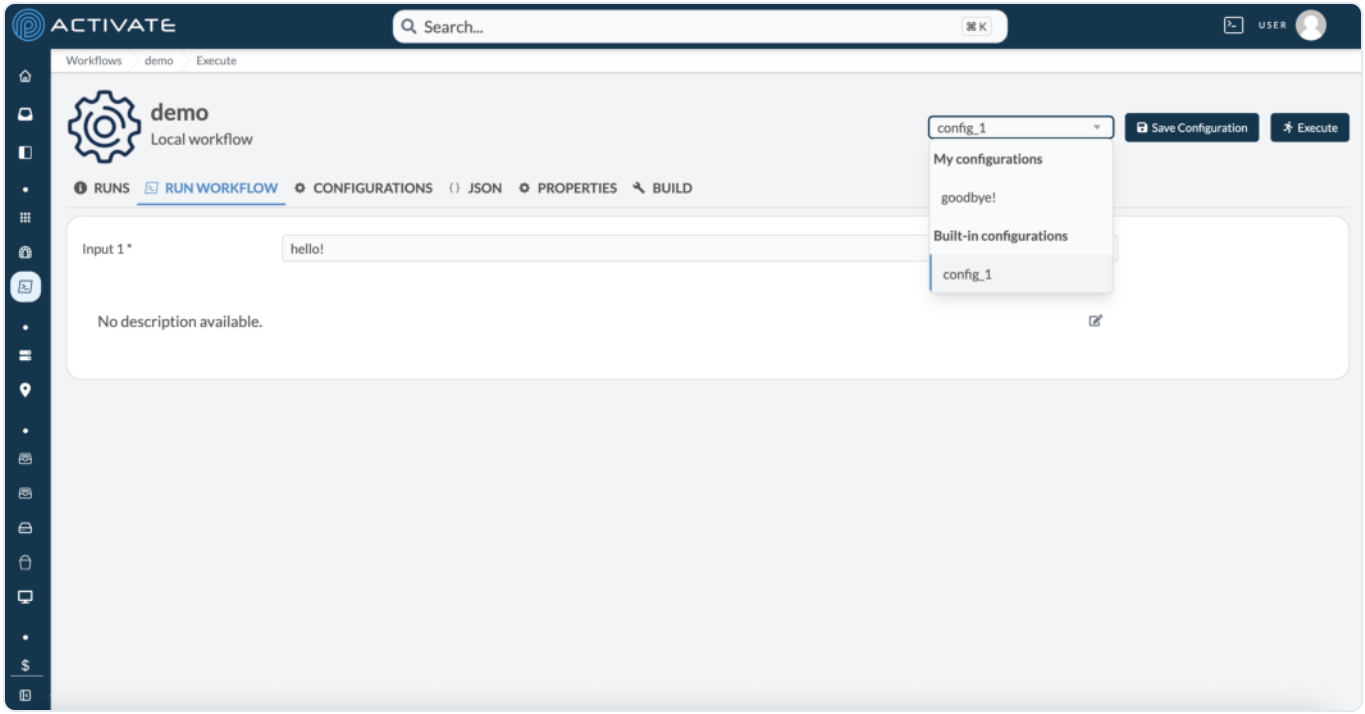
You can save a config for personal use by pressing the **Save Config** button after filling out the workflow form with the inputs you wish to save. In the example below, we save with `input_1 = "goodbye!"`:



Both built-in and personal configurations can be viewed in the configuration tab:



If you wish to execute using a configuration, you can select a configuration from the dropdown in the top right on the **Run Workflow** tab, which will fill the inputs with the saved configuration values, then press **Execute** as usual.



`env`
Defines environment variables to be set. Note that variables set at the step level overwrite job level env variables, which overwrite global env variables.

Example:

```
env:
  foo: a
jobs:
  job-name:
    env:
      foo: b
    steps:
      - name: Print an environment variable
        run: echo $foo
        env:
          foo: c
```

The above example will print `c`.

`timeout`
Defines a maximum amount of time for a workflow/job/step to run.

Supported units are:

- `n` (nanoseconds)
- `s` (seconds)
- `m` (minutes)
- `h` (hours)
- `d` (days)

Example:

```
timeout: 1d
jobs:
  main:
    timeout: 1m
    steps:
      - name: Sample Step
        run: sleep 30
        timeout: 10s
```

In the example above, the sleep command finishes after 1 minute. We set the step `timeout` attribute to 10 seconds. Since the step has not completed in 10 seconds, the job will fail. Without a `timeout` value, the step will run until it finishes. Note that timeouts at all levels are applied rather than overwritten, so if the job timeout was `5s`, the command would be cancelled after 5 seconds instead of 10.

`needs`
Declares workflow-level requirements that must be satisfied before the workflow can run.

`needs.organizationVariables` is an array of organization variable names that must be set by an org admin. If any are missing, the workflow will refuse to execute and the user will see which variables still need to be configured.

```
needs:
  organizationVariables:
    - AWS_REGION
    - LICENSE_KEY
```

`on`
Use the `on` field to define the event that triggers the workflow. In a future release, the `on` field will support additional events. At this time, workflows only support the `execute` event, which is triggered when the workflow is manually executed via the UI or the REST API. When the workflow is manually triggered, the `inputs` context is populated with values from the input form.

[`on.execute.inputs`](/docs/run/workflows/building-workflows/inputs-and-expressions#inputs)

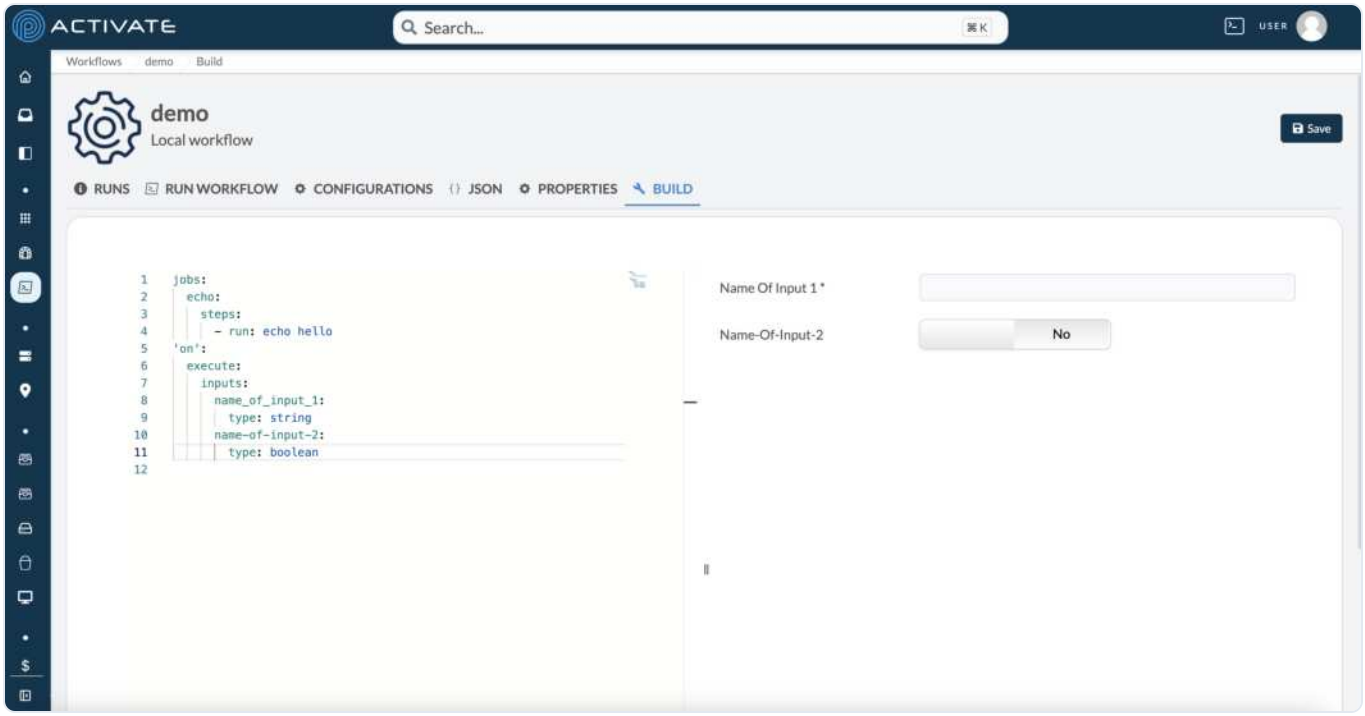
Inputs & Expressions

See the building workflows page if you have not already. If you are looking for the documentation for a specific input field, find the name of the field from the sidebar on the right. Expression documentation can be found here.

Inputs
In addition to defining what your workflow does with jobs, you can define the workflow form users will fill out when running the workflow. This form is what's displayed in the **Run Workflow** tab, and the values of the inputs can be accessed from inside jobs using expressions. Inputs are defined in the following structure:

```
on:
  execute:
    inputs:
      name_of_input_1:
        type: type_of_input_1
        ... other fields (potentially type specific) ...
      name-of-input-2:
        type: type_of_input_2
        ... other fields (potentially type specific) ...
```

The name of an input must be composed of alphanumeric characters, dashes, and underscores. When defining the YAML file in the **Build tab**, the form is previewed on the right side of the screen:



Input Types

Below is a list of each `type` and its modifying attributes. Unless otherwise noted, each `type` can be modified by any of the universal attributes.

boolean

A checkbox or toggle switch representing a boolean value (true/false).

Useful for optional settings, enabling/disabling features, or making binary choices (such as "Enable logging?").

bucket

A dropdown selector for cloud storage buckets (S3, GCS, Azure Blob). Displays the user's own buckets and any buckets shared with them. Only provisioned buckets can be selected.

Attributes:

- `csp` : Filters the list to a specific cloud service provider. Options are `aws` , `azure` , `google` , and `openstack` . When omitted, buckets from all providers are shown.
- `generateCredentials` : When set to `true` , the platform generates temporary cloud credentials for the selected bucket and includes them in the input value at runtime. See Bucket credentials below for details.

Allows users to select a bucket to use during workflow execution, such as for reading input data or writing results.

Bucket value at runtime {#bucket-runtime-value}

When a user selects a bucket, the input value is an object (not a plain string). The following fields are always present:

Field	Description	Example
`type`	Always `"bucket"`.	`"bucket"`
`csp`	Cloud service provider of the bucket.	`"aws"`, `"google"`, `"azure"`, `"openstack"`
`id`	Internal ID of the bucket resource.	`"66342baec24f12..."`
`name`	Name of the bucket resource on the platform.	`"my-bucket"`
`user`	Username of the bucket owner.	`"jdoe"`
`uri`	Platform URI for the bucket.	`"pw://jdoe/my-bucket"`

You can access individual fields using dot notation in expressions. For example, `${{ inputs.mybucket.uri }}` returns the platform URI and `${{ inputs.mybucket.csp }}` returns the cloud provider.

Bucket credentials {#bucket-credentials}

When `generateCredentials: true` is set on a bucket input, the platform fetches temporary credentials from the vault for the selected bucket and adds them to the input value before the workflow runs. The following additional fields are added:

Field	Description
`bucketName`	The actual cloud bucket name (e.g. the S3 bucket name).
`region`	The cloud region of the bucket.
`credentials`	An object containing CSP-specific credentials.

The shape of the `credentials` object depends on the cloud provider:

AWS (`csp: "aws"`):

Credential Field	Description
`credentials.accessKeyId`	AWS access key ID.
`credentials.secretAccessKey`	AWS secret access key.
`credentials.sessionToken`	AWS session token (temporary).

Azure (`csp: "azure"`):

Credential Field	Description
`credentials.subscriptionId`	Azure subscription ID.
`credentials.tenantId`	Azure tenant ID.
`credentials.clientId`	Azure client/application ID.
`credentials.clientSecret`	Azure client secret.

Google Cloud (`csp: "google"`):

Credential Field	Description
`credentials.projectId`	Google Cloud project ID.
`credentials.oauth2Token`	OAuth2 access token.

Example — defining a bucket input with credentials and using them in a job:

```
on:
  execute:
    inputs:
      data_bucket:
        type: bucket
        label: Data Bucket
        csp: aws
        generateCredentials: true

jobs:
  sync:
    steps:
      - name: Sync data from bucket
        run: |

          aws s3 sync s3://${{ inputs.data_bucket.bucketName }}/data ./local-data --region ${{
inputs.data_bucket.region }}
```

> **BUCKET ACCESS**
>
> The user running the workflow must own the selected bucket or have shared access to it. If the user does not have access, the workflow run will fail with an error.

color-picker

An input field that allows users to select a color.

Useful for customizing the appearance of visual elements, such as selecting colors for charts, backgrounds, or UI components.

compute-clusters

Represents a selection of compute clusters.

Attributes:

- `csp` : Specifies the CSP (cloud service provider). Options are `aws` , `azure` , `google` , and `openstack` .
- `include-workspace` : Indicates whether to include workspace clusters.
- `provider` : Specifies the compute resource provider.

Allows users to select the specific compute cluster where their job will run, essential for distributing workloads across available resources.

dropdown

A dropdown menu allowing the user to select from a list of predefined options.

Attributes:

- `options` : Either a flat array of options (each with `label` and `value`), or — for *dynamic dropdowns* — an object whose keys are selectable sets of options keyed by another input's value.
- `option-key` : Only used with the dynamic form. An expression (e.g. `${{ inputs.scheduler }}`) whose resolved value selects which entry from the `options` object to render. When the referenced input changes, the dropdown's choices re-render to match.

Dynamic example: the second dropdown's options depend on which scheduler was picked above it.

```yaml
scheduler:
  type: dropdown
  options:
    - label: SLURM
      value: slurm
    - label: PBS
      value: pbs
partition:
  type: dropdown
  option-key: ${{ inputs.scheduler }}
  options:
    slurm:
      - label: gpu
        value: gpu
      - label: cpu
        value: cpu
    pbs:
      - label: workq
        value: workq
```

Ideal for selecting from multiple predefined options, such as job schedulers, input methods, or configuration settings.

editor

A code editor input field with syntax highlighting.

Attributes:

- `language` : Specifies the language for syntax highlighting (such as yaml, json, etc.).

Useful for entering and editing code snippets, configuration files, or scripts directly within the workflow.

group

A collapsible group of related input fields.

Attributes:

- `items` : A list of inputs that belong to this group.

- `collapsed` : Boolean to indicate if the group should be initially collapsed.

Organizes related inputs together. Especially useful for advanced settings or additional information that can be optionally expanded by the user.

header

Creates a header in the input form.

Attributes:

- `text` : The header text.
- `bold` : Indicates if the text should be bold.
- `size` : Font size of the header.

Adds titles or section headers in the input form for better organization and readability.

instance-type

Represents a selection of instance types for a specific CSP (cloud service provider) and region.

Attributes:

- `csp` : Specifies the CSP (`aws` , `azure` , `google` , or `openstack`).
- `region` : The region associated with the instance types.

Allows users to choose specific instance types for their compute tasks. Important for matching job requirements with appropriate resources.

kubernetes-clusters

Represents a selection of Kubernetes clusters.

Attributes:

- `autoselect` : Controls automatic selection of a cluster.

Allows users to choose a Kubernetes cluster for subsequent resource specifications. Useful when the workflow can operate on any available cluster or when the cluster is determined at runtime.

kubernetes-configmaps

Represents a selection of Kubernetes ConfigMaps.

Attributes:

- `clusterName` : The name of the Kubernetes cluster containing the ConfigMap.
- `namespace` : The namespace that holds the ConfigMap.
- `autoselect` : Controls automatic selection of a ConfigMap.

Used to inject non-secret configuration data into pods. Enables dynamic configuration without rebuilding container images.

kubernetes-deployments

Represents a selection of Kubernetes Deployments.

Attributes:

- `clusterName` : The name of the Kubernetes cluster where the Deployment is defined.
- `namespace` : The namespace that contains the Deployment.
- `autoselect` : Controls automatic selection of a Deployment.

Enables selection of declarative workload specifications for scaling, rolling updates, and version control.

kubernetes-namespaces

Represents a selection of Kubernetes namespaces within a specific cluster.

Attributes:

- `clusterName` : The name of the Kubernetes cluster that contains the namespace.
- `autoselect` : Controls automatic selection of a namespace.

Enables users to pick a namespace after a cluster has been chosen. Essential for scoping resources such as pods, services, or config maps to a particular namespace.

kubernetes-pods

Represents a selection of Kubernetes pods.

Attributes:

- `clusterName` : The name of the Kubernetes cluster where the pod resides.
- `namespace` : The namespace that contains the pod.
- `autoselect` : Controls automatic selection of a pod.

Allows users to reference a specific pod for actions like log retrieval, exec, or monitoring. Critical for targeting workloads at the pod level.

kubernetes-pvc

Represents a selection of Kubernetes PersistentVolumeClaims (PVCs).

Attributes:

- `clusterName` : The name of the Kubernetes cluster containing the PVC.
- `namespace` : The namespace that holds the PVC.
- `autoselect` : Controls automatic selection of a PVC.

Facilitates binding storage resources to workloads. Important for managing data persistence in Kubernetes environments.

kubernetes-secrets

Represents a selection of Kubernetes Secrets.

Attributes:

- `clusterName` : The name of the Kubernetes cluster where the secret is stored.
- `namespace` : The namespace that contains the secret.
- `autoselect` : Controls automatic selection of a secret.

Provides a way to reference sensitive configuration data (e.g., passwords, tokens) for use in pods or other resources.

kubernetes-services

Represents a selection of Kubernetes Services.

Attributes:

- `clusterName` : The name of the Kubernetes cluster where the Service exists.
- `namespace` : The namespace that holds the Service.
- `autoselect` : Controls automatic selection of a Service.

Used to expose pods internally or externally, providing stable networking endpoints for applications.

kubernetes-statefulsets

Represents a selection of Kubernetes StatefulSets.

Attributes:

- `clusterName` : The name of the Kubernetes cluster where the StatefulSet resides.
- `namespace` : The namespace that contains the StatefulSet.
- `autoselect` : Controls automatic selection of a StatefulSet.

Allows users to target stateful workloads that manage stable network identities and persistent storage.

kubernetes-workloads

Represents a generic selection of Kubernetes workloads (any resource type that runs pods).

Attributes:

- `clusterName` : The name of the Kubernetes cluster containing the workload.
- `namespace` : The namespace that contains the workload.
- `autoselect` : Controls automatic selection of a workload.

Provides a flexible way to reference any pod-based workload (Deployments, StatefulSets, DaemonSets, etc.) when the exact type is not important for the operation.

list

A list of items, typically used for repeating input patterns.

Attributes:

- `template` : The template defining the type of items in the list.

- `items-collapsible`: Whether each item should be collapsible.

Useful for inputs where multiple entries of the same type are needed, such as a list of regions, parameters, or configuration items.

multi-dropdown

A dropdown menu allowing the user to select multiple options.

Attributes:

- `options`: A list of options with `label` and `value`.

Ideal for selecting multiple items from a list, such as tags, categories, or resource groups.

number

An input field for entering numerical values.

Attributes:

- `min`: The minimum allowable value.
- `max`: The maximum allowable value.
- `step`: The increment step for the number input.
- `slider`: Boolean to indicate if the input should be displayed as a slider.

Useful for specifying numerical parameters, such as counts, thresholds, limits, or any other numeric settings.

organization-groups

Represents a selection of organization groups.

Attributes:

- `csp`: Specifies the CSP (cloud service provider). Options are `aws`, `azure`, `google`, and `openstack`.

Allows users to select organizational groups for permissions, access control, or grouping related resources.

password

Allows users to input a password, conventionally obfuscated.

region

Represents a geographic region associated with a CSP.

Attributes:

- `csp`: Specifies the CSP (cloud service provider). Options are `aws`, `azure`, `google`, and `openstack`.

Allows users to select a specific geographic region for their resources, essential for optimizing latency, compliance, and availability.

slurm-accounts

Represents a selection of SLURM accounts.

Attributes:

- `resource`: The compute resource associated with the SLURM accounts.

Allows users to select SLURM accounts for job submissions. Essential for managing resource allocation and accounting in SLURM environments.

slurm-partitions

Represents a selection of SLURM partitions.

Attributes:

- `resource`: The compute resource associated with the SLURM partitions.
- `account`: Specifies the SLURM account associated with the partitions.

Allows users to select SLURM partitions for job submissions. Essential for organizing and managing job queues in SLURM environments.

slurm-qos

Represents a selection of SLURM QoS (Quality of Service) settings.

Attributes:

- `account`: Specifies the SLURM account associated with the QoS settings.
- `partition`: Specifies the SLURM partition associated with the QoS settings.
- `resource`: The compute resource associated with the SLURM QoS.

Allows users to select SLURM QoS settings for job submissions. Essential for prioritizing and managing job performance in SLURM environments.

string

A single-line text input or a multi-line text area for string values.

Attributes:

- `placeholder`: Placeholder text for the input field.
- `textarea`: Indicates if the input should be a text area.
- `prefillDefault`: If `true`, the `default` value is pre-filled into the form field. If `false` (the default for string inputs), the form field starts empty and `default` only applies when the workflow runs without a value.

Collects textual input from users. Suitable for names, descriptions, commands, or any other text-based configuration.

zone

Represents a selection of zones within a region for a specific CSP (cloud service provider).

Attributes:

- `csp` : Specifies the CSP. Options are `aws` , `azure` , `google` , and `openstack` .
- `region` : Specifies the region associated with the zone.

Allows users to select specific zones for their resources. Important for optimizing resource placement, redundancy, and compliance.

Universal Input Attributes

Most input types above can be modified by a set of common attributes, which help in configuring the basic behavior and presentation of the input fields. The only `type` without these universal attributes is `header` , which has its own unique set of attributes focused on display rather than interaction.

`default`

Sets the default value for the input field.

Usage: Provides an initial value if the user does not input one.

Example:

```
default: 'Enter description here'
```

`description`

Descriptive text shown beneath the field's label. Unlike `tooltip` , the description is always visible rather than only appearing on hover — use it for guidance the user should read before filling in the field.

Example:

```
description: The GPU type to allocate. Changing this restarts the instance.
```

`disabled`

Disables the input field, making it read-only and uneditable by the user.

Usage: Use this when you want to display information without allowing the user to change it.

Example:

```
disabled: true
```

`hidden`

A condition to hide or show the input field based on other inputs or logic.

Usage: Creates a dynamic form where inputs can be conditionally displayed.

Example:

```
hidden: ${{ inputs.show_advanced == false }}
```

`ignore`

When set to `true`, the input field's value is ignored and not sent to the backend.

Usage: Useful for fields that are needed for UI logic but should not be part of the final payload.

Example:

```
ignore: true
```

`label`

A human-readable name for the input field that will be displayed in the UI.

Usage: Helps users understand what data is expected.

Example:

```
label: Job Description
```

`optional`

Indicates whether the input field is optional or required.

Usage: Determines whether the user must provide a value.

Example:

```
optional: true
```

`tooltip`

A tooltip text providing additional information about the input field.

Usage: Assists users by offering more context or instructions when they hover over the input field.

Example:

```
tooltip: 'Provide a detailed description of the job'
```

Wizard Mode

Wizard mode turns the input form into a multi-step flow. Useful for long forms where grouping related fields into sequential steps reduces cognitive load.

Enable it by setting `$meta.wizard.mode` to `wizard` at the top of your `on.execute.inputs` object, then define each step using input fields of `type: step`.

```
on:
  execute:
    inputs:
      $meta:
```

```
        wizard:          mode: wizard      step1:        type: step        title: Basic Info
options:          name:            type: string            label: Name      step2:        type:
step        title: Resources        description: Pick where this will run        options:
resource:            type: compute-clusters            label: Compute Cluster
```

Wizard step attributes (`type: step`)

- `title` (required) — Shown as the step heading.
- `description` — Optional subheading.
- `nextLabel` — Custom label for the Next button. Default: `Next →`.
- `prevLabel` — Custom label for the Previous button. Default: `← Previous`.
- `options` — The input fields rendered in this step. Uses the same input schema as top-level inputs.

Wizard `$meta.wizard` attributes

- `mode` (required) — Must be `wizard` to enable wizard mode.
- `submitLabel` — Label for the final submit button. Default: `Execute`.
- `flatten` — If `true` (default), step fields are merged into the top-level `inputs` context so you can reference them as `${{ inputs.name }}` rather than `${{ inputs.step1.name }}`. Set to `false` to keep the nesting.
- `navigation.showSteps` — Show the step indicator. Default: `true`.
- `navigation.allowJump` — Allow jumping to previously visited steps. Default: `false`.
- `navigation.hideStepNumbers` — Hide numbers in the step indicator. Default: `false`.

`$meta.labelPosition`

Independent of wizard mode, setting `$meta.labelPosition` to `left` or `top` controls where the input label appears relative to the field. Default is `top`.

Expressions

Expressions allow workflow authors to programatically control the input form based on other inputs and access contexts inside of jobs.

In order for expressions to be evaluated, they must be wrapped in `${{ }}`.

Operators

The following operators are available for use in expressions, and generally correspond to their javascript counterparts. Example expressions and what they evaluate to are included.

- `&&` : Logical AND
  - `${{ true && 'truthy value' }}` = 'truthy value'
  - `${{ 'truthy value' && true }}` = true
  - `${{ 0 && true }}` = 0
  - `${{ 'truthy value' && false }}` = false
- `||` : Logical OR:

    - `${{ false || 'any value' }}` = 'any value'
    - `${{ 'truthy value' || 'any value' }}` = 'truthy value'
- `==` : Equal to
    - `${{ 1 == true }}` = true
- `!=` : Not equal to
    - `${{ 1 != true }}` = false
- `===` : Strict equal to
    - `${{ 1 === true }}` = false
- `!==` : Strict not equal to
    - `${{ 1 !== true }}` = true
- `>` : Greater than
    - `${{ 'a' > 'b' }}` = false
- `<` : Less than
    - `${{ 1.5 < 1 }}` = false
- `>=` : Greater than or equal to
    - `${{ 1.0 >= 1 }}` = true
- `<=` : Less than or equal to
    - `${{ false <= true }}` = true
- `+` : Addition/Concatenation
    - `${{ 'a' + 'b' }}` = 'ab'
    - `${{ 1 + 1.5 }}` = 2.5
- `-` : Subtraction
    - `${{ 1 - 1.5 }}` = -0.5
- `*` : Multiplication
    - `${{ 4 * 4 }}` = 16
- `/` : Division
    - `${{ 9 / 4 }}` = 2.25
- `**` : Exponentiation
    - `${{ 2 ** 4 }}` = 16
- `//` : Floor division
    - `${{ 4 // 3 }}` = 1
- `??` : Nullish coalescing
    - `${{ 'hello' ?? 'default' }}` = 'hello'

  - `${{ undefined ?? 'default' }}` = 'default'
- `get` : Get index/property:
  - `${{ ["a", "b"] get 1 }}` = "b"
  - `${{ {"a":"c", "b":"d"} get a }}` = "c"
  - `${{ 'ab' get 0 }}` = 'a'
- `in` : Left in right:
  - `${{ "a" in ["a", "b"] }}` = true
  - `${{ "b" in {"a":"c", "b":"d"} }}` = true
  - `${{ "d" in {"a":"c", "b":"d"} }}` = false

Special operators (not binary):

- `!` : Logical NOT:
  - `${{ !false }}` = true
  - `${{ "hello" !in ["hello", "goodbye"] }}` = false
- `( )` : Logical grouping:
  - `${{ true && (true || true) && false }}` = false
  - `${{ 10 / (2 + 8) }}` = 1
- `? :` : Conditional (ternary):
  - `${{ true && false ? 'a' : 'b' }}` = 'b'
  - `${{ 'truthy value' ? 'a' : 'b' }}` = 'a'

Parser idiosyncracies to take note of:

- Expressions are very much whitespace sensitive, relying on spaces to separate operators and operands. Thus `${{ 1 + 1 }}` will evaluate to 2 but `${{ 1+1 }}` will evalutate to the string `1+1`. Logical not `!` is the only operator that should not be separated with a space.
- If an expression has even a minor mistake, the parser will never throw an error but may return any value, screwing up your workflow in strange ways. So double check your expressions.

Contexts

The following contexts are available for use in expressions:

- `inputs` : Used to access the values and schema of the input fields.
- `needs` : For now only `needs.<job>.outputs.<name>` is supported, which allows access to outputs from the current job or its dependency jobs.
- `sessions` : Contains a field for each session described in the workflow. For an example, see Workflow Sessions.
- `org` : Contains organization-specific variables that are set by your administrator. Use `org.<variable name>` in an expression to access these.
- `.` and `../` : Used to access schema values.

The workflow form can only access the values and schema of the inputs field through the `inputs` context and `.` self referencing, alongside your organization variables through `org`. Information from job dependencies accessed through the `needs` context and sessions accessed through the `sessions` context are only available during execution, so they should only ever be accessed inside of `jobs`.

Outputs

Accessing outputs

Outputs from a job can be accessed using the `needs` context. The following example demonstrates how to access the outputs from within the current job:

```
jobs:
  job1:
    steps:
      - name: fake output
        run: echo PORT=3001 >> $OUTPUTS
      - name: Echo output from self
        run: echo ${{ needs.job1.outputs.PORT }} # Prints 3001
  job2:
    needs:
      - job1
    steps:
      - name: Echo output from job 1
        run: echo ${{ needs.job1.outputs.PORT }} # Prints 3001
```

Accessing outputs from named steps

Steps can be given an `id` field to create named steps. Outputs from named steps can be accessed using `needs.<job>.steps.<step_id>.outputs.<output_name>`. When using named steps, outputs are not automatically published to the job outputs and must be explicitly mapped to job outputs using the job outputs field.

```
jobs:
  main:
    steps:
      - name: Create step output
        id: test
        run: echo FOO=BAR >> $OUTPUTS
      - name: Access output from named step
        run: echo ${{ needs.main.steps.test.outputs.FOO }} # Prints BAR
```

Writing outputs

To create outputs that can be accessed by other jobs or steps, you must write key-value pairs to a special output file. The path to this file is provided through the `$OUTPUTS` environment variable.

Output format: Each output must be written as `key=value` on a separate line.

Basic example:

```yaml
jobs:
  example:
    steps:
      - name: Create outputs
        run: |
          echo PORT=3001 >> $OUTPUTS
          echo DATABASE_URL=postgres://localhost:5432/mydb >> $OUTPUTS
          echo STATUS=success >> $OUTPUTS
```

Important considerations:

- **Job dependencies**: Jobs that consume outputs from other jobs should list those jobs as dependencies using the `needs` field. This ensures the producer job completes before the consumer job starts, making the outputs reliable.
- **Named step outputs**: When using steps with `id` fields, outputs are scoped to that specific step and are not automatically published to the job level.
- **Output persistence**: Outputs are preserved throughout the workflow execution and can be accessed by any subsequent job or step that has the proper dependencies.
- **Deprecation notice**: Automatic publishing of step outputs to job level is planned to be removed in a future version. Using the explicit `outputs` map (described below) is the recommended approach for forward compatibility.

Publishing outputs using the job-level outputs map

To access job outputs from another job, jobs can explicitly define which outputs to publish using an `outputs` map.

Example

```yaml
jobs:
  build:
    outputs:
      version: ${{ needs.build.steps.get_version.outputs.VERSION }}
      artifact_url: ${{ needs.build.steps.upload.outputs.URL }}
    steps:
      - name: Get version
        id: get_version
        run: echo VERSION=1.2.3 >> $OUTPUTS
      - name: Upload artifact
        id: upload
        run: echo URL=https://example.com/artifact.zip >> $OUTPUTS
```

Inputs and Self Referencing

How and when to use the `inputs` keyword and/or the self referencing `.` can be confusing; hopefully this section can make things a bit clearer.

Let us consider the following example of a workflow form defined in the yaml. There are two string inputs:

```
on:
  execute:
    inputs:
      username:
        type: string
      workdir:
        type: string
```

`username` is the user's username and `string_field_2` is a directory I want to create based on the username by default. To access the **value** of an input, we use the `inputs` keyword:

```
on:
  execute:
    inputs:
      username:
        type: string
      workdir:
        type: string
        default: ${{ "~/path/to/directory/" + inputs.username }}
```

This ensures that every time the user updates the username, the default workdir is updated. Doing it like this allows the user to override the workdir with their own custom directory path in the workflow form if they would like to.

Now let us consider another example:

```
on:
  execute:
    inputs:
      ...
      option_1_input:
        type: string
        hidden: ${{ some long and complicated boolean expression }}
      option_2_input:
        type: string
```

We want the option 2 input to appear when the option 1 input is hidden, and be hidden when the option 1 input appears. Rather than rewriting the long and complicated boolean expression, we can access the **schema** of the input by using `../` :

```
on:
  execute:
    inputs:
      ...
      option_1_input:
        type: string
        hidden: ${{ some long and complicated boolean expression }}
      option_2_input:
        type: string
        hidden: ${{ !../option_1_input.hidden }}
```

We use the `../` because it represents going backwards in our path to the level of the object containing the property with the expression; if we wanted to access a property inside option_2_input from the `hidden` property instead, such as `type`, we could simply do `${{ .type }}`, which would be equivalent to `${{ ../option_2_input.type }}`. We can also access the **schema** by using the `inputs` keyword and ensuring the last property is wrapped in brackets; so an equivalent expression for `${{ !../option_1_input.hidden }}` in the example above would be `${{ !inputs.option_1_input[hidden] }}`.

In summary, here are all the different possibilities:

```
on:
  execute:
    inputs:
      field_1:
        type: number
      field_2:
        type: string
        default: ${{ inputs.field_1 }} # The value of field 1, which is whatever the user sets it
to on the form
      field_3:
        type: string
        default: ${{ inputs[field_1] }} # This would actually be the field_1 schema object which
is {"type": "number"}. It would probably not be rendered as a string properly.
      field_4:
        type: string
        default: ${{ inputs.field_1[type] }} # This would be just the string "number"
      field_5:
        type: string
        default: ${{ ../field_1 }} # This would be the field_1 schema object which is {"type":
"number"}.
      field_6:
        type: string
        default: ${{ ../field_1.type }} # This would be just the string "number"
      field_7:
        type: string
        default: ${{ .type }} # This would be just the string "string" since field_7 has type:
string
```

Conditional Visibility

Inputs can be dynamically hidden based on the value or attributes of other inputs. This is useful for creating more dynamic and user-friendly forms where advanced options are only shown when needed.

```
on:
  execute:
    inputs:
      header:
        type: header
        text: Starter Local Workflow
        size: 20
      show_advanced:
        type: boolean
        label: Show Advanced Settings
        default: false
```

```
      advanced_setting:        type: string        label: Advanced Setting        hidden: ${{
!inputs.show_advanced }}      advanced_setting2:        type: string        label: Advanced
Setting 2        hidden: ${{ inputs.advanced_setting[hidden] }}
```

In this example, `advanced_setting` and `advanced_setting2` are hidden fields until the user selects **Yes** from the `show_advanced` toggle switch. `advanced_setting2` is hidden based on the `hidden` attribute of `advanced_setting`, whereas `advanced_setting` is hidden based on the value of `show_advanced`.

Dynamic Default Values

Default values can be calculated based on other input values.

```yaml
on:
  execute:
    inputs:
      header:
        type: header
        text: Starter Local Workflow
        size: 20
      base_value:
        type: number
        label: Base Value
        default: 10
      calculated_value:
        type: number
        label: Calculated Value
        default: ${{ inputs.base_value * 2 }}
        hidden: true
```

In this example, the `calculated_value` input field has a default value that is twice the value of the `base_value` input field.

Actions

See the building workflows page if you have not already. If you are looking for the documentation for a specific action, find the name of the action from the sidebar on the right.

Actions

Certain operations that are used regularly in workflows have been abstracted into **actions**, and can be executed in a workflow using the following syntax:

```yaml
jobs:
  job-name:
    steps:
      - uses: parallelworks/<name of action>
        with:
          action_input_2: value_1
          action_input_2: value_2
```

Note that valid action inputs passed to the action through the `with` property depend on the action. Below is a list of actions and their input fields.

`checkout`
Checks out a git repo. The example below is the simplest possible usage:

```
jobs:
  job-name:
    steps:
      - uses: parallelworks/checkout
        with:
          repo: https://github.com/parallelworks/interactive_session.git
          branch: main
```

If you want to only check out part of a repo, you can define `sparse_checkout`:

```
jobs:
  job-name:
    steps:
      - uses: parallelworks/checkout
        with:
          repo: https://github.com/parallelworks/interactive_session.git
          branch: main
          sparse_checkout:
            - utils
            - platforms
```

To clone the repo to the file system of a cluster, use the `ssh` field at the job or step level to specify the cluster.

```
jobs:
  job-name:
    steps:
      - uses: parallelworks/checkout
        with:
          repo: https://github.com/parallelworks/interactive_session.git
          branch: main
        ssh:
          remoteHost: ${{ inputs.cluster.ip }}
on:
  execute:
    inputs:
      cluster:
        type: compute-clusters
        optional: false
```

Inputs:

- `repo` *string*: Specifies the git repo to be cloned. Required.
- `branch` *string*: Specifies the branch to be cloned. Required.
- `sparse_checkout` *string*[]: Only clone files that match the patterns defined in the array.

- `path` *string*: Where to clone the git repo to. Defaults to the workflow run directory.

`cancel-jobs`

Cancels a workflow run, or a set of jobs in a workflow run. Here is an example:

```yaml
permissions:
  - '*'
jobs:
  main:
    steps:
      - run: sleep 2
      - uses: parallelworks/cancel-jobs
        with:
          jobs:
            - j1
  j1:
    steps:
      - run: sleep 200
        early-cancel: any-job-failed
  j2:
    steps:
      - run: sleep 200
        early-cancel: any-job-failed
```

Note that the running workflow does not have to be the same as the workflow to be canceled.

Inputs:

- `jobs` *string*[]: Specifies the jobs to be canceled. Defaults to all jobs.
- `workflow` *string*: Specifies the workflow to be canceled. Defaults to the current workflow. Should be used only if you want to cancel a run of another workflow (not the workflow in which the cancel action is defined).
- `run` *int*: Specifies the run to be canceled. Defaults to current if workflow is undefined, required if workflow is defined.
- `slug` *string*: Specifies the slug of the run to be canceled; an alternative to workflow and run. Defaults to the current slug. Should be used only if you want to cancel a different run than the one running the cancel action.

Note that there is slightly different behavior if neither `slug` nor `workflow` is passed. This is deliberate, to ensure support for subworkflows. If you want to cancel jobs in a workflow that is to be used as a subworkflow in another workflow, make sure that you do not pass `slug` or `workflow` as an argument using `with`. This will ensure that the action correctly cancels the job or jobs in the subworkflow rather than attempting to cancel the jobs in the superworkflow. In general, the `slug` or `workflow` field should only be used when the desired value is not the default.

`update-session`

Updates a session with information. If you want to use sessions, this action is always necessary because sessions have no initial information on what to display. See Workflow Sessions for example usage.

Shared Inputs:

- `name` *string*: Name of the session to update. Must be present in `sessions` property of YAML. Required.
- `type` *string*: Type of session, must be either 'link' or 'tunnel'. Default is tunnel.
- `target` *string*: Id of the target compute cluster (get using an expression like `${{inputs.cluster_input_name.id}}`). Defaults to user workspace.
- `status` *string*: Status of the session. For tunnel sessions, defaults to 'creating' (or 'running' if the target is user workspace). For link sessions, defaults to 'running'. Can be set to 'running' to indicate the session is ready immediately, but this is optional as the platform will detect readiness automatically.

Tunnel Inputs:

- `remotePort` *int*: Port to read on the target. Required for tunnel sessions.
- `remoteHost` *string*: Host of the target. Default is localhost.
- `slug` *string*: Added at the end of session url. Defaults to empty string.
- `localPort` *int*: Port on the local machine. Defaults to a random open port.
- `openAI` *boolean*: Whether the session is an OpenAI session. Default is false.

Link Inputs:

- `url` *string*: The url of the link for the session.

`scheduler-agent`

You can use the `scheduler-agent` action to provision a node and start an ssh server so that you can run commands on it through ssh. Note that you must use a cluster with a partition for this to properly work. See Workflow Sessions for example usage.

Inputs:

- `wait` *boolean*: Whether to wait for the node to provision. Defaults to true.
- `scheduler-type` *string*: Use this scheduler. Defaults to slurm, options are slurm and pbs.
- `scheduler-flags` *object*: Use these flags for the sbatch command. Defaults to none. You can find a list of flags here under job submission for slurm and here for pbs.
- `script-headers` *string*: Additional lines to add to the top of the bash script being submitted via sbatch or qsub (depending on your scheduler). Useful if you want to allow users to add their own `#SBATCH/PBS` comment flags at will.

`wait-for-agent`

This action should only be used if you already used scheduler-agent with `wait: false` passed as an input to the `with` property, which might be a good idea if you want the workflow job to perform other operations while the scheduler-agent node is configuring.

```
permissions:
  - '*'
sessions:
  session:
jobs:
  main:
    ssh:
      remoteHost: ${{ inputs.resource.ip }}
    steps:
      - uses: parallelworks/scheduler-agent
        id: slurmstep
        with:
          wait: false
      - run: echo "Perform some other operations here, like a checkout action or second scheduler-
agent"
      - uses: parallelworks/wait-for-agent
        id: waitstep
        with:
          agentId: ${{ needs.main.steps.slurmstep.outputs.agentId }}
      - name: Get open port
        ssh:
          jumpNodeHost: ${{ inputs.resource.ip }}
          remoteHost: ${{ needs.main.steps.waitstep.outputs.remoteHost }}:${{
needs.main.steps.waitstep.outputs.sshPort }}
        run: |
          echo sessionPort="$(pw agent open-port)" >> $OUTPUTS
          cat $OUTPUTS
      - uses: parallelworks/update-session
        with:
          status: running
          name: ${{ sessions.session }}
          target: ${{ inputs.resource.id }}
          remoteHost: ${{ needs.main.steps.waitstep.outputs.remoteHost }}
          remotePort: ${{ needs.main.outputs.sessionPort }}
      - name: Serve port
        run: |
          cat << EOF > myServer.go
          package main

              "fmt"
              "net/http"
          )
          func hello(w http.ResponseWriter, req *http.Request) {
              fmt.Fprintf(w, "Hello from slurm agent server!\n")
          }

          func main() {
              http.HandleFunc("/", hello)
              http.ListenAndServe(":${{ needs.main.outputs.sessionPort }}", nil)
          }
          EOF
          go run myServer.go
        ssh:
          jumpNodeHost: ${{ inputs.resource.ip }}
          remoteHost: ${{ needs.main.steps.waitstep.outputs.remoteHost }}:${{
```

```
needs.main.steps.waitstep.outputs.sshPort }}
'on':  execute:    inputs:      resource:        label: Resource Target        type: compute-
clusters        autoselect: true        optional: false
```

Inputs:

- `agentId` *string*: The id of the agent to wait for. The output from the scheduler-agent action includes this field (see usage in example above).
- `schedulerJobId` *string*: Optional id of the scheduler job that is provisioning the agent. When provided, the action monitors the scheduler job's health while waiting and fails early if the job enters a terminal state (such as cancelled or failed) before the agent becomes ready.

Workflow Sessions

A workflow can expose an application as a session by including a top-level `sessions` definition in its YAML. The session is created when the workflow runs and removed when it stops.

Similar to the top-level definition `jobs`, the `sessions` definition can describe multiple sessions with unique names. You can then use the `session` context to call a session by name inside `steps`.

In the example below, there is one session defined in `sessions`, named `expose`. That session is then referenced inside `steps` inside `jobs` with the context `${{ sessions.expose }}`. The `expose` property `redirect` designates whether the user will be redirected to a specific session, especially in the event that you define multiple sessions. The property `prompt-for-name` determines whether a user will be prompted to name their session. If `prompt-for-name` is excluded, a name will automatically be generated based on the run numbers.

```
sessions:
  expose:
    redirect: true
    prompt-for-name:
      default: 'mysession'
jobs:
  main:
    steps:
      - name: Expose port
        uses: parallelworks/update-session
        with:
          remotePort: '3001'
          name: ${{ sessions.expose }}
          status: running
          target: ${{ inputs.target.id }}
      - name: Run forever
        run: ssh ${{ inputs.target.ip }} tail -f /dev/null
'on':
  execute:
    inputs:
      target:
        type: compute-clusters
```

To target your user workspace, you can set the `target` input to `user-workspace`. This following example will start a small http server on a specified port in your user workspace, then expose that port as a session.

```
sessions:
  expose:
    redirect: true
jobs:
  expose:
    steps:
      - name: Wait for server
        run: |
          while ! nc -z localhost ${{ inputs.port }}; do
            sleep 1
          done
      - name: Expose port
        uses: parallelworks/update-session
        with:
          remotePort: ${{ inputs.port }}
          name: ${{ sessions.expose }}
          status: 'running'
          target: 'user-workspace'
      - name: Keep running
        run: sleep 500
  main:
    steps:
      - name: Create simple http server
        run: |
          cat <<EOF > myServer.go
          package main

            "fmt"
            "net/http"
          )
          func hello(w http.ResponseWriter, req *http.Request) {
            fmt.Fprintf(w, "Hello from GO server!\n")
          }

          func main() {
            http.HandleFunc("/", hello)
            http.ListenAndServe(":${{ inputs.port }}", nil)
          }
          EOF
      - name: Run Server
        run: go run myServer.go
'on':
  execute:
    inputs:
      port:
        type: number
        min: 4000
        max: 65000
        label: Port to run server on
```

You can also use the `scheduler-agent` action to provision a node and start an ssh server so that you can run commands on it through ssh. Note that you must use a cluster with a partition for this to properly work.

```
permissions:
  - '*'
sessions:
  session:
jobs:
  main:
    ssh:
      remoteHost: ${{ inputs.resource.ip }}
    steps:
      - uses: parallelworks/scheduler-agent
        id: slurmstep
      - name: Get open port
        ssh:
          jumpNodeHost: ${{ inputs.resource.ip }}
          remoteHost: ${{ needs.main.steps.slurmstep.outputs.remoteHost }}:${{
needs.main.steps.slurmstep.outputs.sshPort }}
        run: |
          echo sessionPort="$(pw agent open-port)" >> $OUTPUTS
          cat $OUTPUTS
      - uses: parallelworks/update-session
        with:
          status: running
          name: ${{ sessions.session }}
          target: ${{ inputs.resource.id }}
          remoteHost: ${{ needs.main.steps.slurmstep.outputs.remoteHost }}
          remotePort: ${{ needs.main.outputs.sessionPort }}
      - name: Serve port
        run: |
          cat << EOF > myServer.go
          package main

              "fmt"
              "net/http"
          )
          func hello(w http.ResponseWriter, req *http.Request) {
              fmt.Fprintf(w, "Hello from slurm agent server!\n")
          }

          func main() {
              http.HandleFunc("/", hello)
              http.ListenAndServe(":${{ needs.main.outputs.sessionPort }}", nil)
          }
          EOF
          go run myServer.go
        ssh:
          jumpNodeHost: ${{ inputs.resource.ip }}
          remoteHost: ${{ needs.main.steps.slurmstep.outputs.remoteHost }}:${{
needs.main.steps.slurmstep.outputs.sshPort }}
'on':
  execute:
    inputs:
      resource:
```

```
        label: Resource Target        type: compute-clusters        autoselect: true
optional: false
```

Sessions

This page includes information about running, sharing, and deleting sessions.

What is a Session?

A session is a way to access an application or environment running on a remote resource through the platform. ACTIVATE supports several kinds of sessions:

- **VS Code** sessions give you a code editor on a resource — in your browser or connected to your local VS Code. See **VS Code Sessions**.
- **Desktop** sessions give you a graphical desktop on a resource, streamed to your browser. See **Desktop Sessions**.
- **Tunnel** sessions expose an application running on a port of a remote compute resource or user workspace.
- **Link** sessions point to an application hosted outside of ACTIVATE.

The rest of this page focuses on **tunnel** and **link** sessions.

Tunnel Sessions

A `tunnel` session is a secure tunnel that exposes an application running on a remote compute resource or user workspace to the platform. You can then access it through ACTIVATE. Sessions are created when you run a workflow that includes a `sessions` definition in its YAML file, but you can also create sessions manually.

For example, if you have a simple web application running on port 3000 of a remote compute cluster, you can create a session that exposes the application to the platform. Once the session is created, you can access the application in your web browser through ACTIVATE.

Link Sessions

A **link** session is a simple URL that points to an application running outside of ACTIVATE. You can create one manually by providing a name and a URL. This is useful when the application is hosted externally but you want to link to it from ACTIVATE.

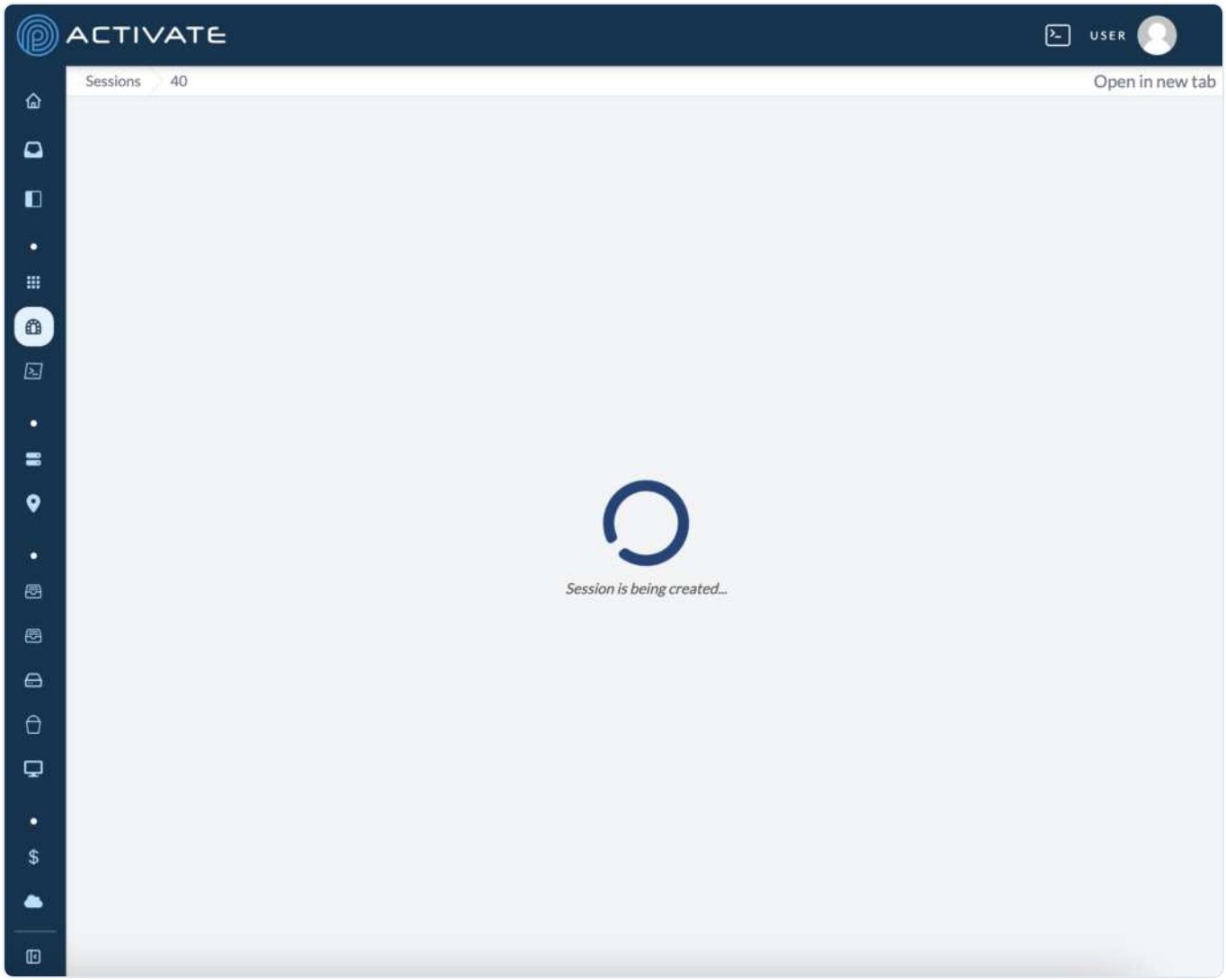
Running a Workflow Session

Any workflow can generate a session as long as its YAML file contains a `sessions` definition. Below is an example of a YAML file with a `sessions` definition, with a session named `desktop`:

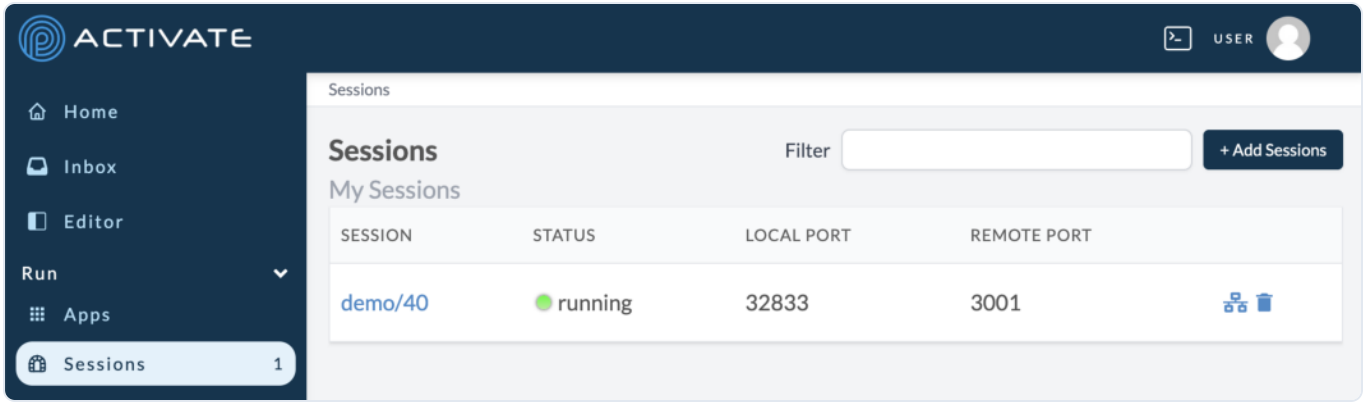
```
sessions:
  desktop:
```

You can check for `sessions` in the workflow's **Build** tab. For more information, please see **Workflow Sessions**.

Start your workflow. A notification will appear with the message *Executing workflow [name]*. You'll also be redirected to the **Sessions** page.



Once the session has been created, the page will display the session, status, local port, and remote port.



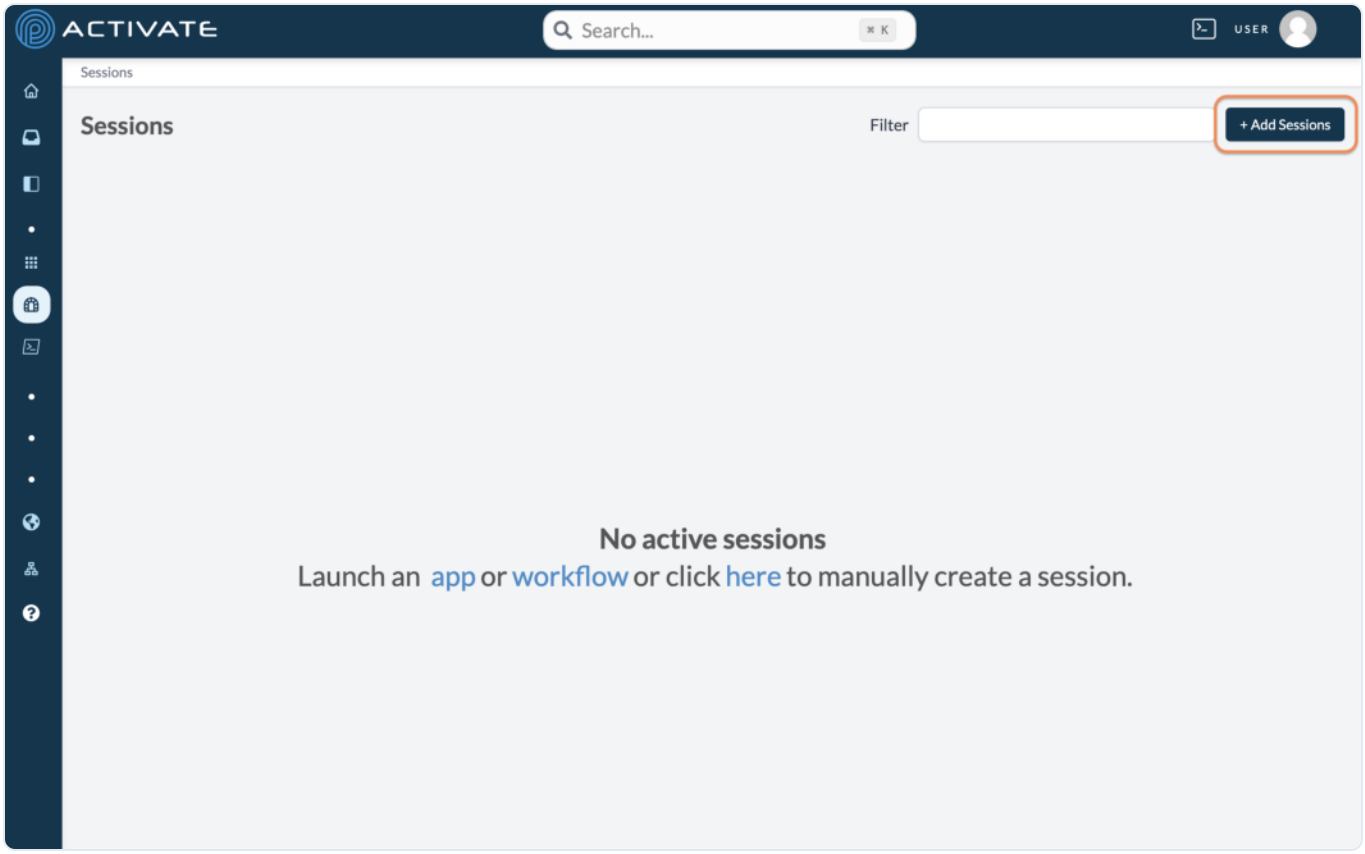
When your navigation bar is expanded, your number of running sessions will appear next to **Sessions**.

Click the session name to see view the exposed application.

Running a Manual Session

You can create any type of session manually, which does not require a running workflow. Manually created sessions are the same as sessions created by workflows, except they are not tied to the lifecycle of a workflow and need to be manually removed when you're done with them.

To create a session manually, navigate to **Sessions**. Click **Add Sessions**.



On the next page, select **Tunnel** or **Link**.

Link sessions only require a **Name** and a **URL**. After setting these parameters, you can click **Add Session**.

Tunnel sessions require a **Name**, **Remote Port**, and **Target Type**.

After selecting a **Target Type**, you must select a running resource to launch the session.

- Compute clusters only require one field, **Target**, which refers to the running cluster.
- Kubernetes clusters require **Kubernetes Cluster** (the running cluster), the **Resource Type** (pods, deployments, or services), and a **Resource Name**.

Advanced Options

All parameters in this section are optional.

The **Workspace Port** refers to the port in your workspace that the session will be forwarded to. You can leave this field blank or enter `0` to choose a random port.

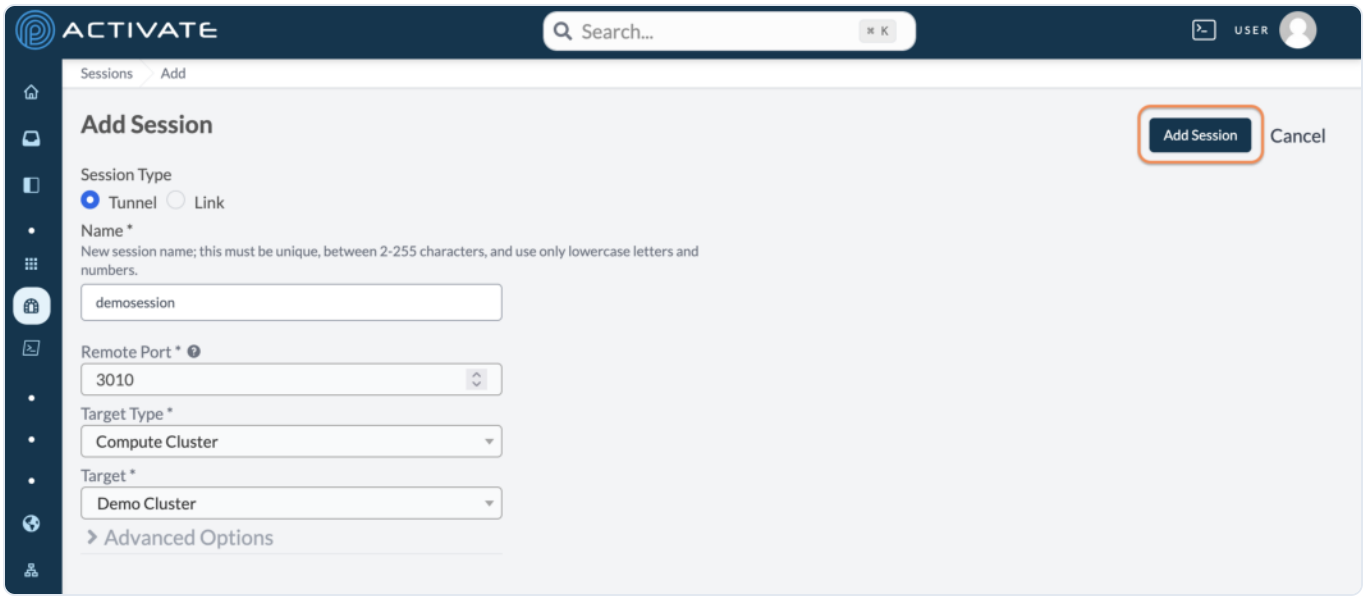
The **Slug** will be appended to the URL when accessing the session. This parameter is useful for applications whose basePath is not `/`.

The **Remote Host** is the hostname of the application's running location. This parameter is set to `localhost` by default.

Application Uses HTTPS? determines whether the session will use HTTPS to access the application running on the forwarded port. Please note that you should only select use HTTPS if the application you are running is using HTTPS. This does not affect encryption between your browser and ACTIVATE.

Use Custom Domain determines whether the session's URL will use a custom domain to access the application running on the forwarded port. If you're using the SaaS version of our platform, the URL will take the form of `*.activate.pw` where `*` is the session domain. For standalone deployments of ACTIVATE, the URL will vary.

After configuring these settings, click **Add Session**.



You will be redirected to the **Sessions** page as the session starts. When the session's status displays **running**, click its name to access the session.

Sharing a Session

On the **Sessions** page, click the share icon.

On the next page, select the group(s) you want to share the session with. You'll see the message *Permission updated successfully*.

ABOUT SHARING

> Only share a session with groups you trust. Sharing is the same as allowing users in the group to run the workflow as you.

Deleting a Session

On the **Sessions** page, click the delete icon.

A dialog box will appear with the message *Are you sure you want to delete this session?*

Click **Delete**.

A notification will appear with the message *Session deleted*.

> **NOTE**
>
> Please note that deleting a session does not cancel the workflow that created the session. You must cancel the workflow separately.

VS Code Sessions

There are two ways to use VS Code on a compute resource. Both run on the resource itself, so your code, data, and processes stay on the cluster:

Method	Where it runs	Where you use it	Best for
VS Code session	VS Code server on the resource	Your web browser	Editing and running code with zero local setup
`pw vscode`	Your *local* VS Code app	Your computer	Working in your own VS Code — with all your extensions and settings — connected over SSH

Looking for a graphical desktop instead? See Desktop Sessions.

> **THE RESOURCE MUST BE RUNNING**
>
> A resource (a compute cluster or your user workspace) must be **running** before you can launch a VS Code session on it.

VS Code in the browser

A **VS Code session** runs a VS Code server on the resource and opens it in your browser, so there's nothing to install on your computer.

Launch from a cluster (Quick Launch)

1. Open your cluster and find the **Quick Launch** panel (*Open this cluster in your browser*).
2. On the **VS Code** card, click **Launch**.
3. *(Optional)* Set a **Directory** to open — for example `~/projects`. Leave it blank to open your home directory.

4. Click **Launch**.

Launch from the Sessions page

1. Go to **Sessions** and click **New Session**.
2. Choose **VS Code** as the session type.
3. Pick the **Target** resource (a compute cluster or your user workspace).
4. *(Optional)* Set a **Directory** and a **Name**.
5. Click **Launch**.

ACTIVATE installs and starts the VS Code server on the resource — you'll see the status move through *installing → starting → running*. Once it's running, VS Code opens in an embedded view on the session page. Click **Open in new tab** for a full-window editor.

> **TIP**
> VS Code sessions work on both compute clusters and your user workspace. For lightweight editing, the user workspace is a quick option that doesn't require starting a cluster.

Connect your local VS Code with `pw vscode`

If you'd rather work in the VS Code app already installed on your computer — with your own keybindings, themes, and extensions — use the `pw vscode` command. It connects your local VS Code to the resource using VS Code's **Remote-SSH** extension, with the connection routed securely through ACTIVATE.

Prerequisites

- The **PW CLI** installed and authenticated. See About the PW CLI.
- **VS Code** installed locally with the `code` command on your PATH. In VS Code, open the command palette (`Cmd/Ctrl+Shift+P`) and run **Shell Command: Install 'code' command in PATH**.
- The **Remote-SSH** extension installed in VS Code.

Usage

```
# Open VS Code on a resource
pw vscode my-cluster

# Open a specific directory
pw vscode my-cluster /home/user/project

# Open VS Code on another user's resource
pw vscode pw://otheruser/their-cluster
```

When you run the command, the PW CLI:

1. Registers your SSH key with the platform (if it isn't already).
2. Writes an SSH host entry to `~/.ssh/pw_config` (included from your `~/.ssh/config`) that proxies the connection through ACTIVATE.

3. Launches VS Code and connects it to the resource via Remote-SSH, opening the directory you specified — or your home directory by default.

See the `pw vscode` command reference for the full list of arguments.

Which should I use?

- Use a **VS Code session** when you want zero local setup — everything runs in the browser.
- Use `pw vscode` when you want your own local VS Code, extensions, and settings, connected to the resource.

Desktop Sessions

A **Desktop session** runs a graphical (VNC) desktop on a compute resource and streams it to your browser — useful for GUI applications, visualization tools, and file management. The desktop runs on the resource itself, so your data and processes stay on the cluster.

Need a code editor instead? See VS Code Sessions.

> **THE RESOURCE MUST BE RUNNING**
>
> A resource (a compute cluster or your user workspace) must be **running** before you can launch a desktop session on it.

Launch a desktop

From a cluster (Quick Launch)

1. Open your cluster and find the **Quick Launch** panel (*Open this cluster in your browser*).
2. On the **Remote Desktop** card, click **Launch**.
3. Click **Launch** to confirm.

From the Sessions page

1. Go to **Sessions** and click **New Session**.
2. Choose **Desktop** as the session type.
3. Pick the **Target** resource (a compute cluster or your user workspace).
4. Click **Launch**.

Once the session is running, the desktop appears in an embedded viewer on the session page. Use **Open in new tab** for a full-screen desktop.

Desktop sessions vs. interactive sessions

There are two ways to get a graphical desktop on a resource, and which one to use depends on **where** the desktop should run:

- **On a controller (head node) or user workspace → use a Desktop session.** Desktop sessions launch directly through the ACTIVATE agent, which is more reliable and more general than the workflow-based approach. This is the preferred way to get a desktop on a controller.

- **On a compute node → use an interactive session.** Interactive sessions submit a job through the scheduler to allocate a compute node, and remain the right tool for running a desktop — or an application like MATLAB or RStudio — on compute nodes.

In short: prefer **Desktop sessions** on the controller or workspace, and use **interactive sessions** when you need the desktop on a compute node.

> NOTE
>
> Running a Desktop session directly on a **compute node** isn't available yet — it's planned for a future release. Until then, use an interactive session for compute-node desktops.

About Interactive Sessions

Parallel Works publishes a set of workflows called interactive sessions which makes it easy to get started with many interactive applications on ACTIVATE. These workflows are available on the ACTIVATE Marketplace and their source code is publicly available on GitHub.

> JUST NEED A DESKTOP ON A CONTROLLER?
>
> If you only need a graphical desktop on a cluster **controller** (head node) or your user workspace, use a **Desktop session** instead. Desktop sessions launch directly through the ACTIVATE agent, which is more reliable and general than the workflow-based approach. Interactive sessions remain the best way to launch a desktop — or apps like MATLAB and RStudio — on **compute nodes**, where the job must go through the scheduler.

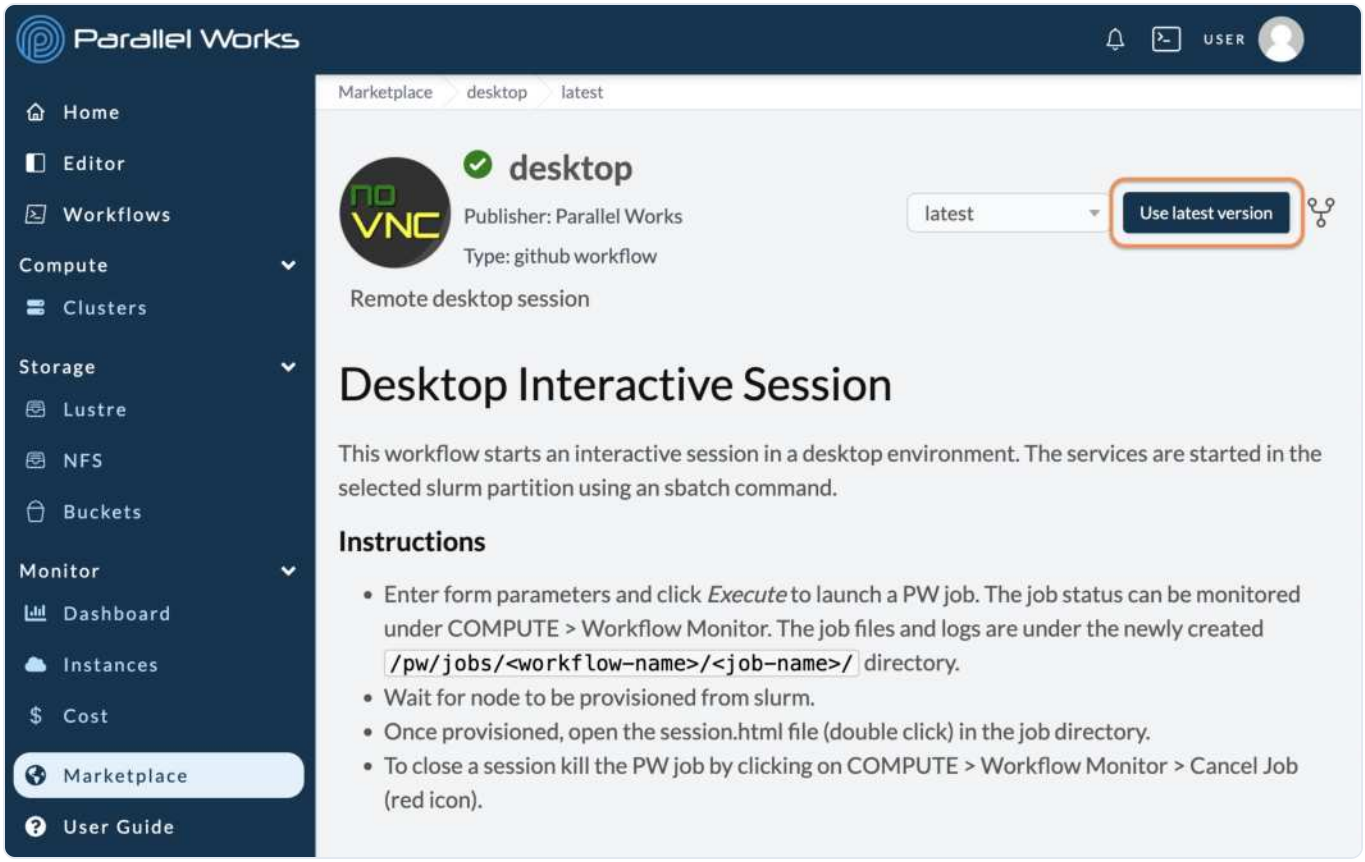
Running Interactive Sessions

You can run interactive sessions either inside your user workspace on ACTIVATE, or on any cluster's controller or compute nodes. Running an interactive session on a controller or compute node requires a running resource. If you plan to run an interactive session in your ACTIVATE user workspace, you do not have to have a resource running. In general, it's best to only run lightweight interactive session applications in the user workspace, such as small tests or running a Jupyter Notebook that launches jobs on other resources.

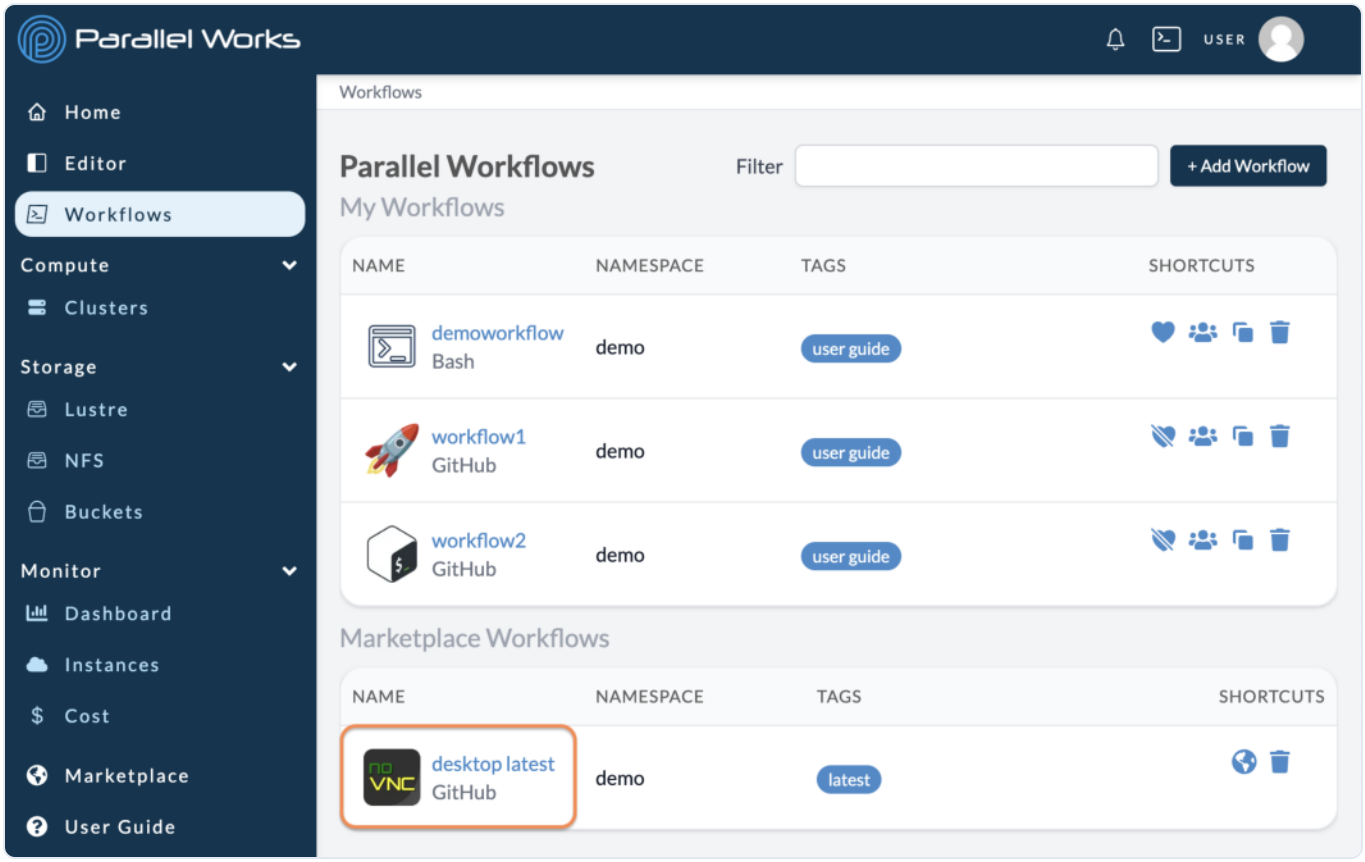
If you want to practice with a sample interactive session, you can follow our Jupyter Notebook instructions or explore other interactive sessions in the Parallel Works Marketplace, especially JupyterLab, VSCode, NGINX, VNC Server, and RStudio.

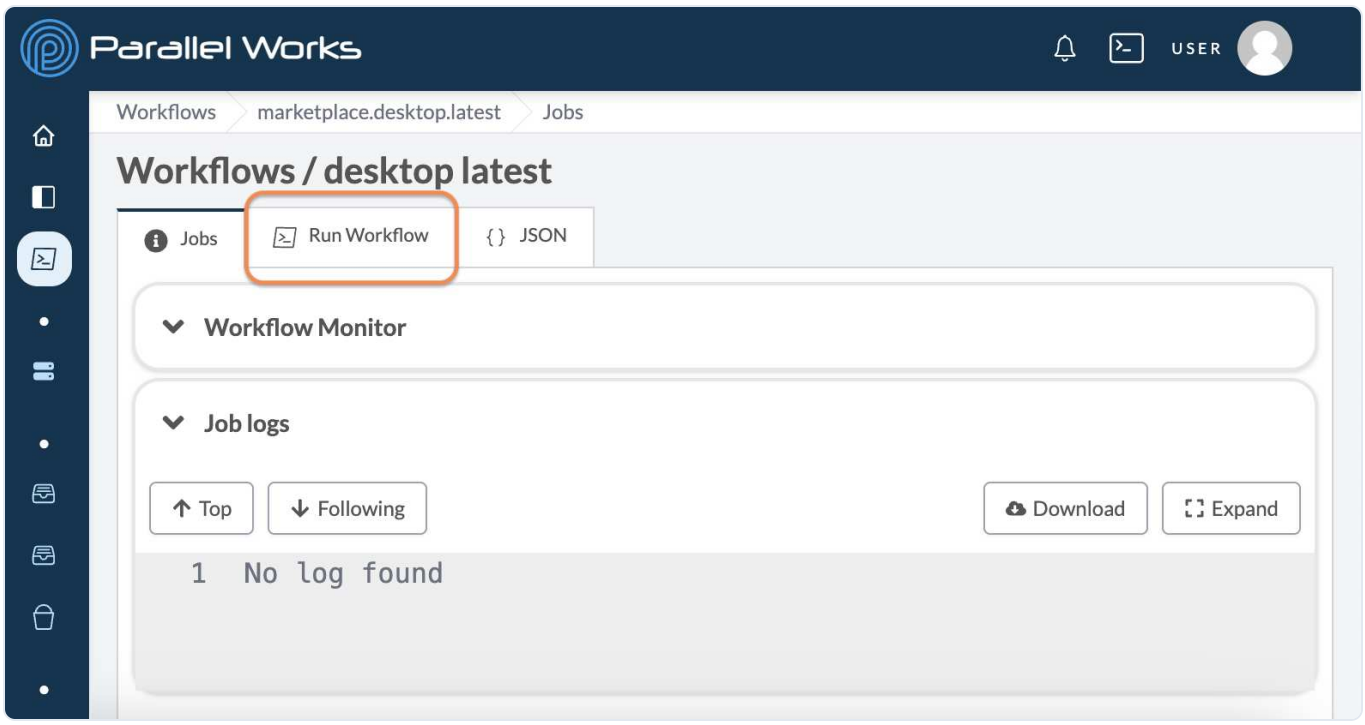
To run any interactive session:

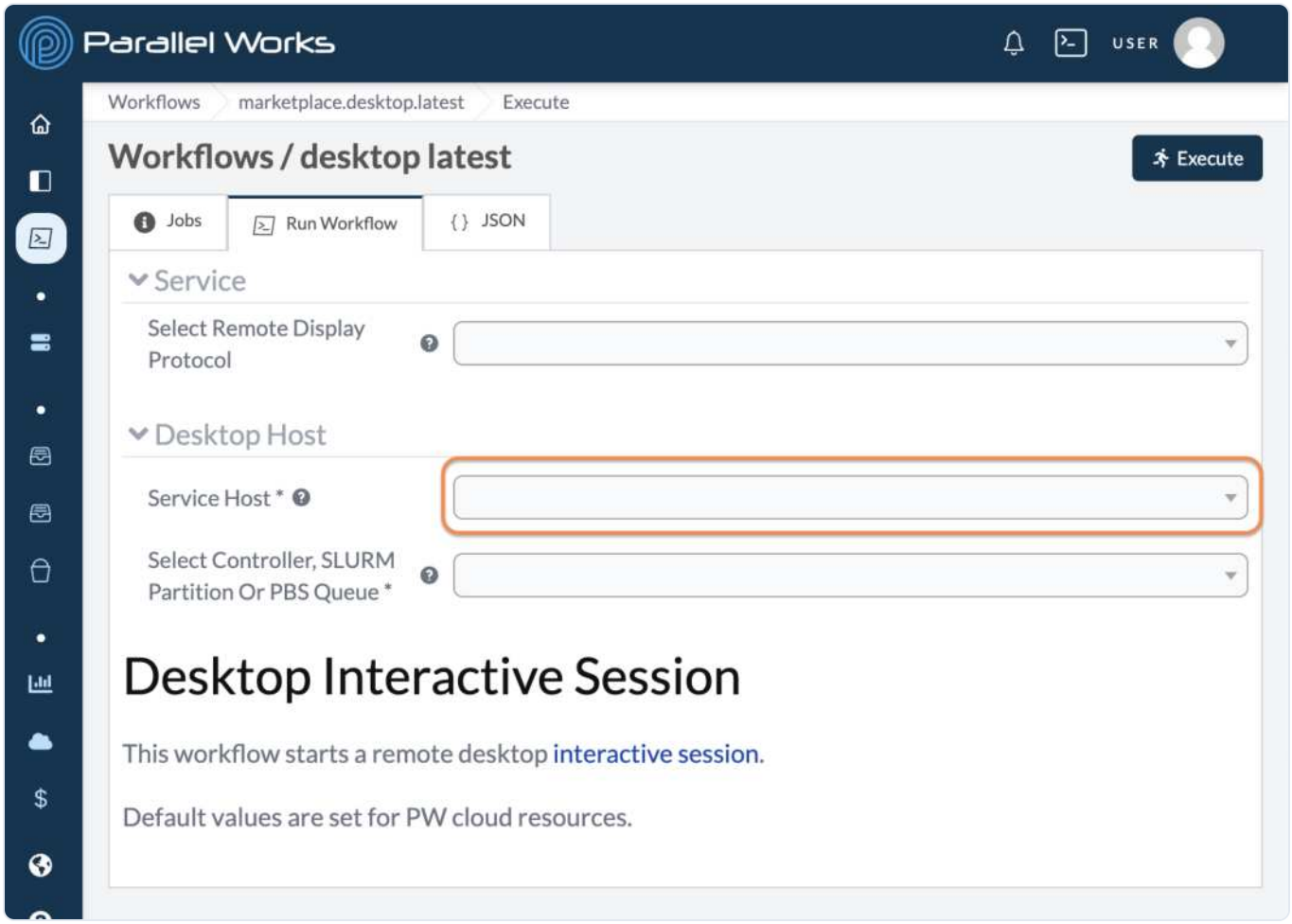
1. Add the interactive session workflow you want from the ACTIVATE Marketplace. If you're importing files from GitHub instead, we suggest building your own workflow.



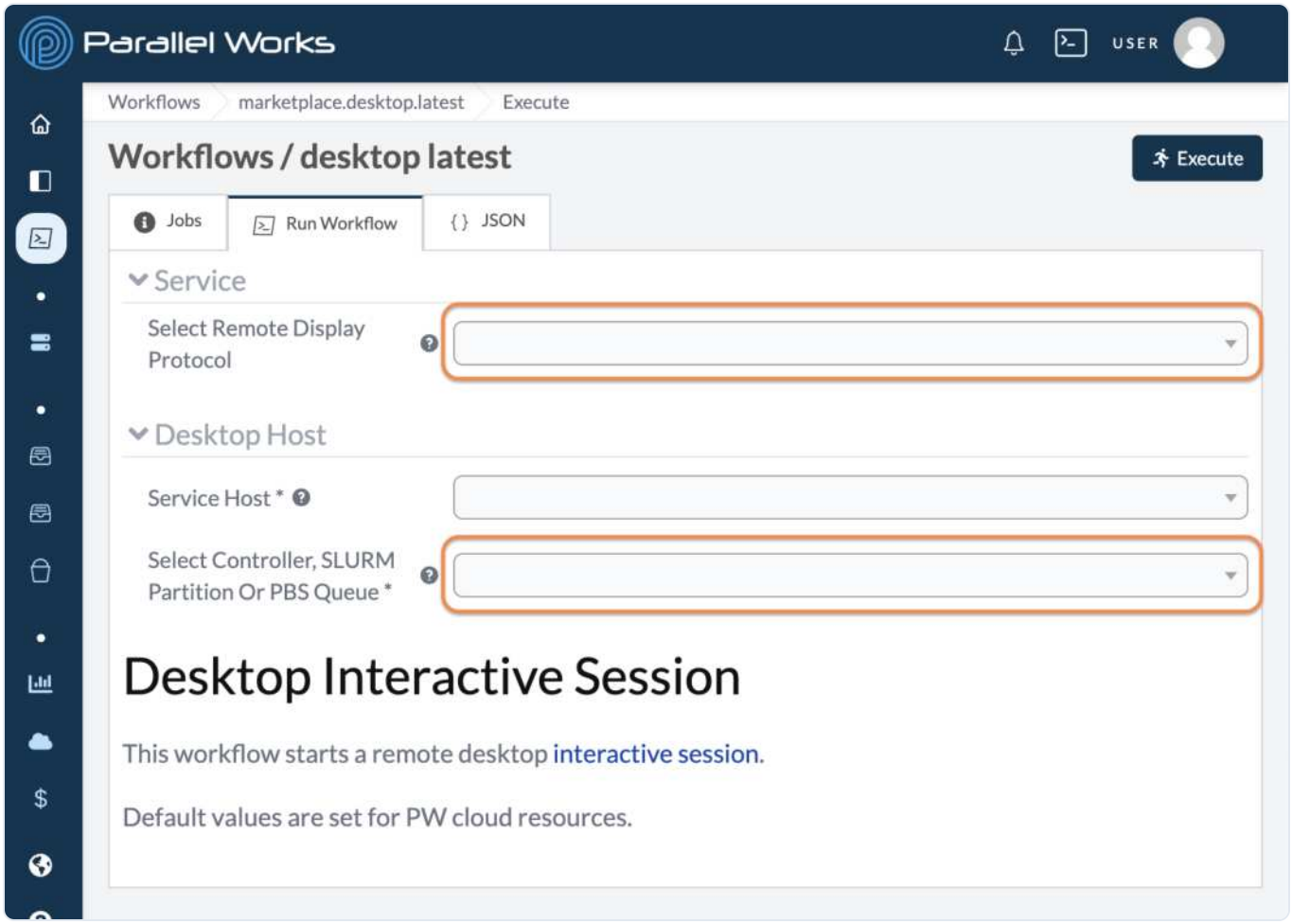
2. Start your resource if you plan to run the interactive session on a controller or compute node.
3. Go to **Workflows** > **Your Workflow** > **Run Workflow** and choose the resource where you want to start the server.


Parallel Works
USER
Home
Editor
Workflows
Compute
Clusters
Storage
Lustre
NFS
Buckets
Monitor
Dashboard
Instances
Cost
Marketplace
User Guide
Workflows
Parallel Workflows
Filter
+ Add Workflow
My Workflows
NAME
NAMESPACE
TAGS
SHORTCUTS
demoworkflow
Bash
demo
user guide
workflow1
GitHub
demo
user guide
workflow2
GitHub
demo
user guide
Marketplace Workflows
NAME
NAMESPACE
TAGS
SHORTCUTS
desktop latest
GitHub
demo
latest

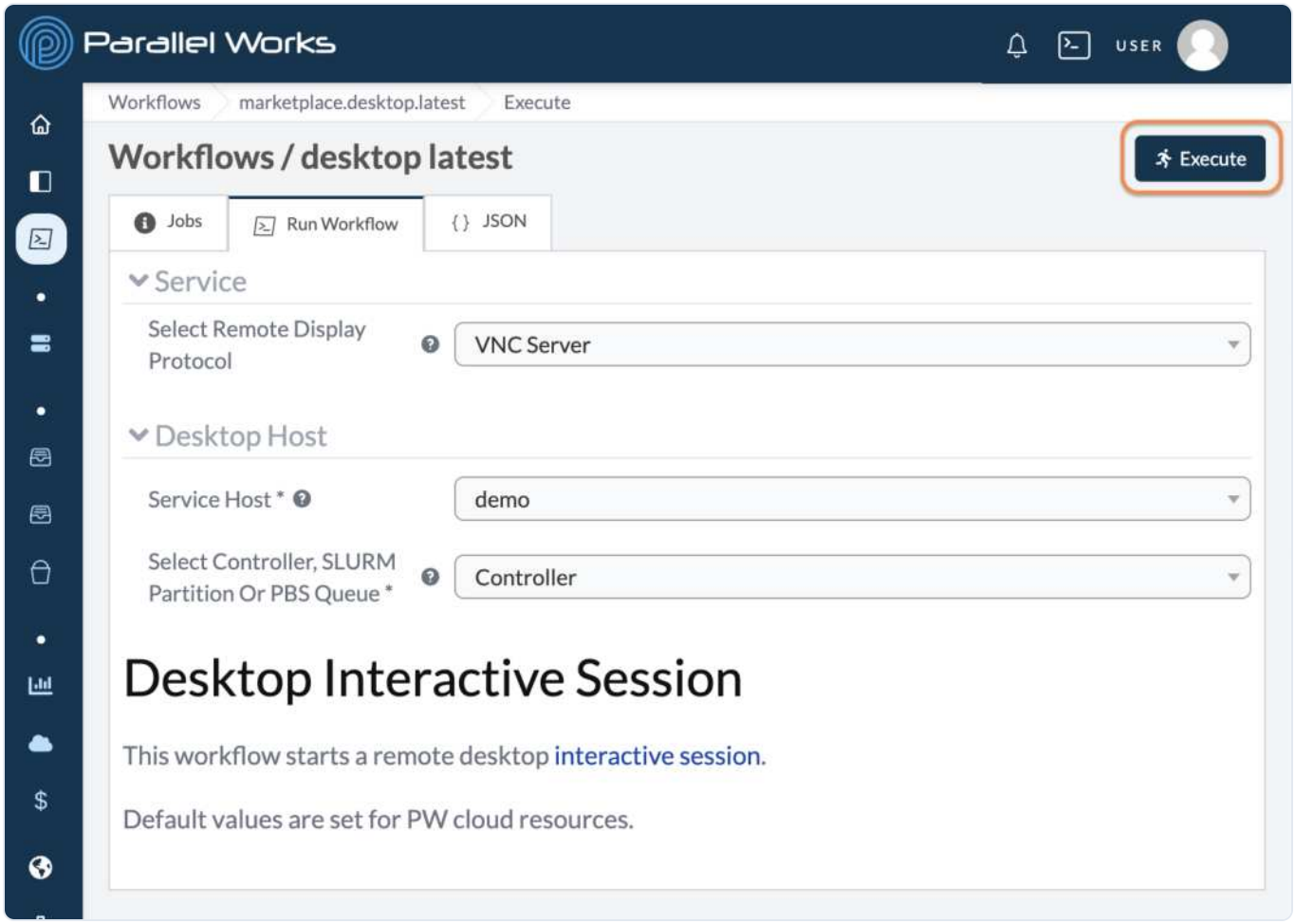

Parallel Works
USER
Workflows
marketplace.desktop.latest
Jobs
Workflows / desktop latest
Jobs
Run Workflow
JSON
Workflow Monitor
Job logs
Top
Following
Download
Expand
1 No log found



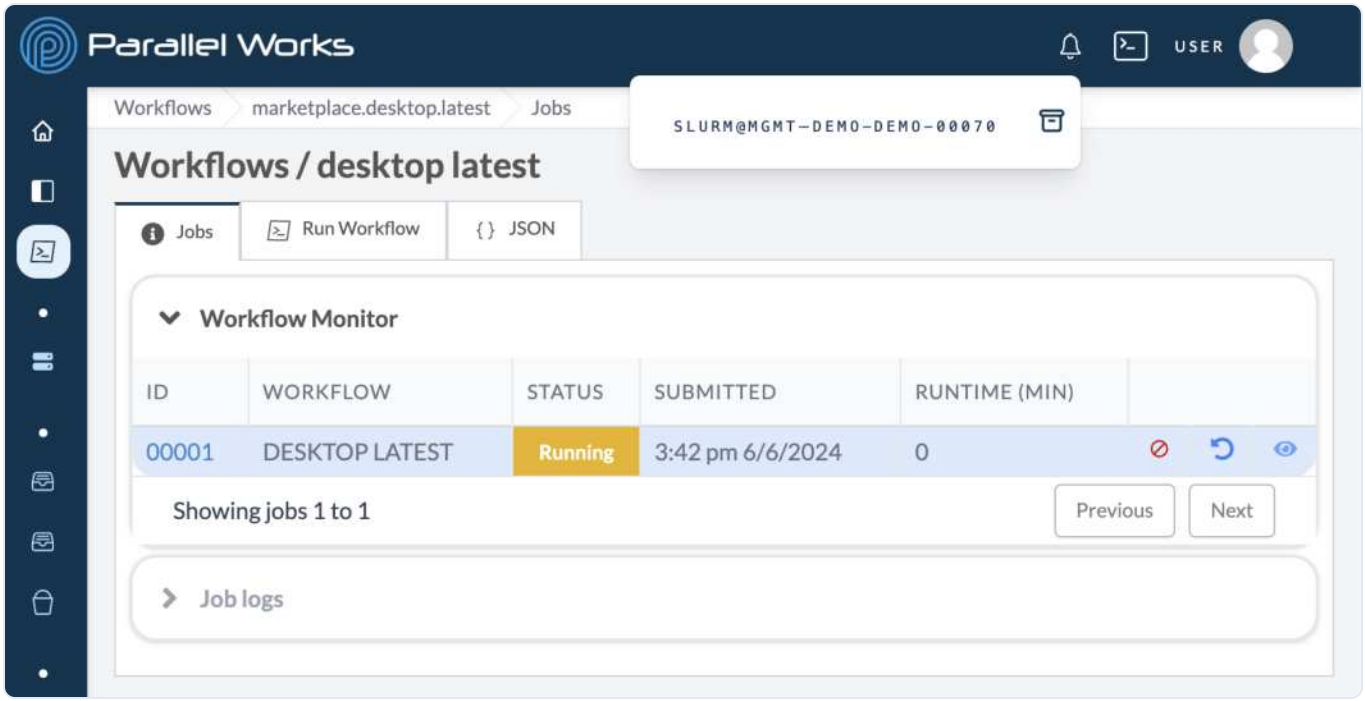
4. Enter or review the other input parameters in the **Run Workflow** tab. You can find detailed descriptions of each parameter by hovering over the help tip icons next to their labels.



5. Click the **Execute** button to launch the job.

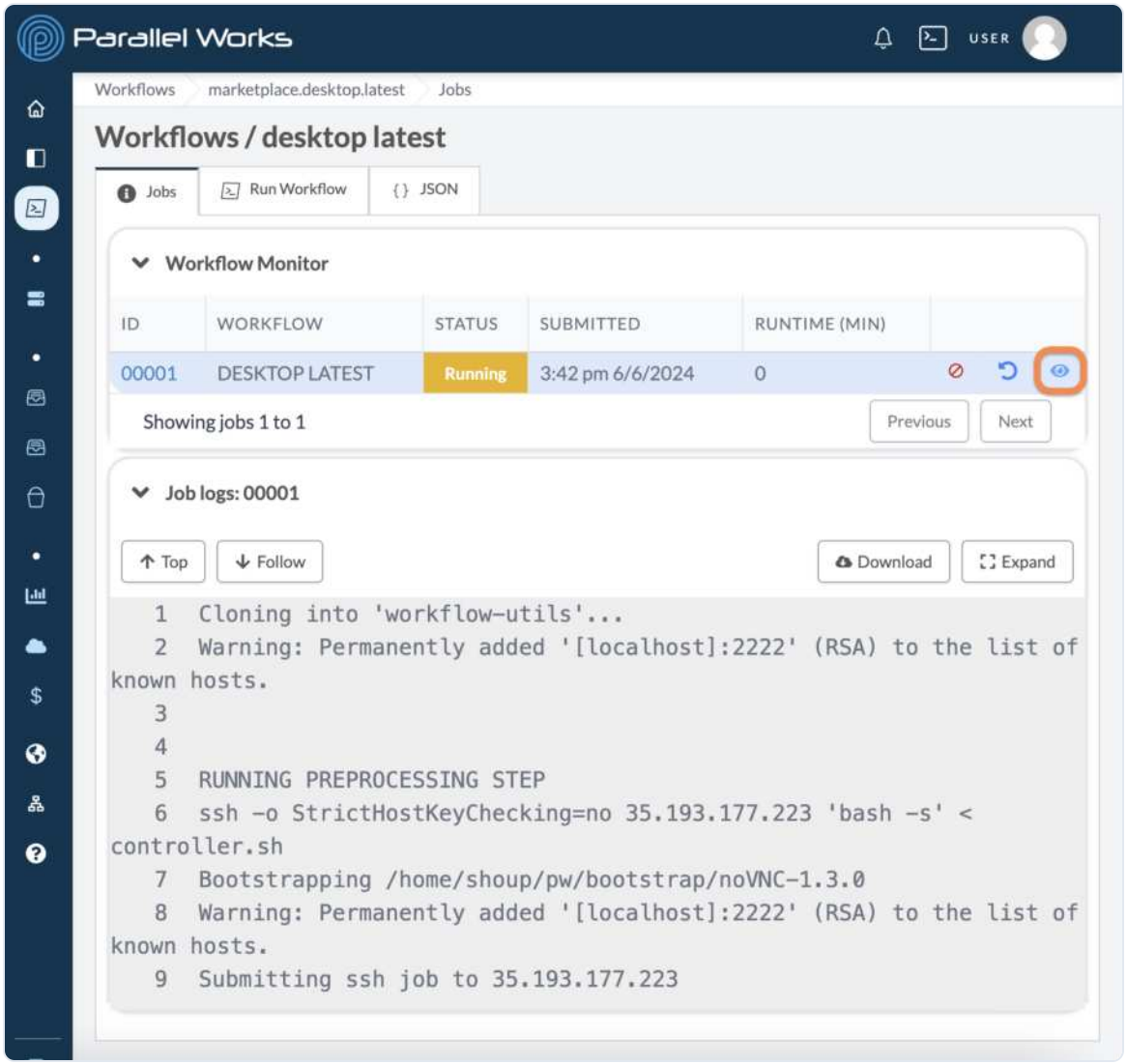


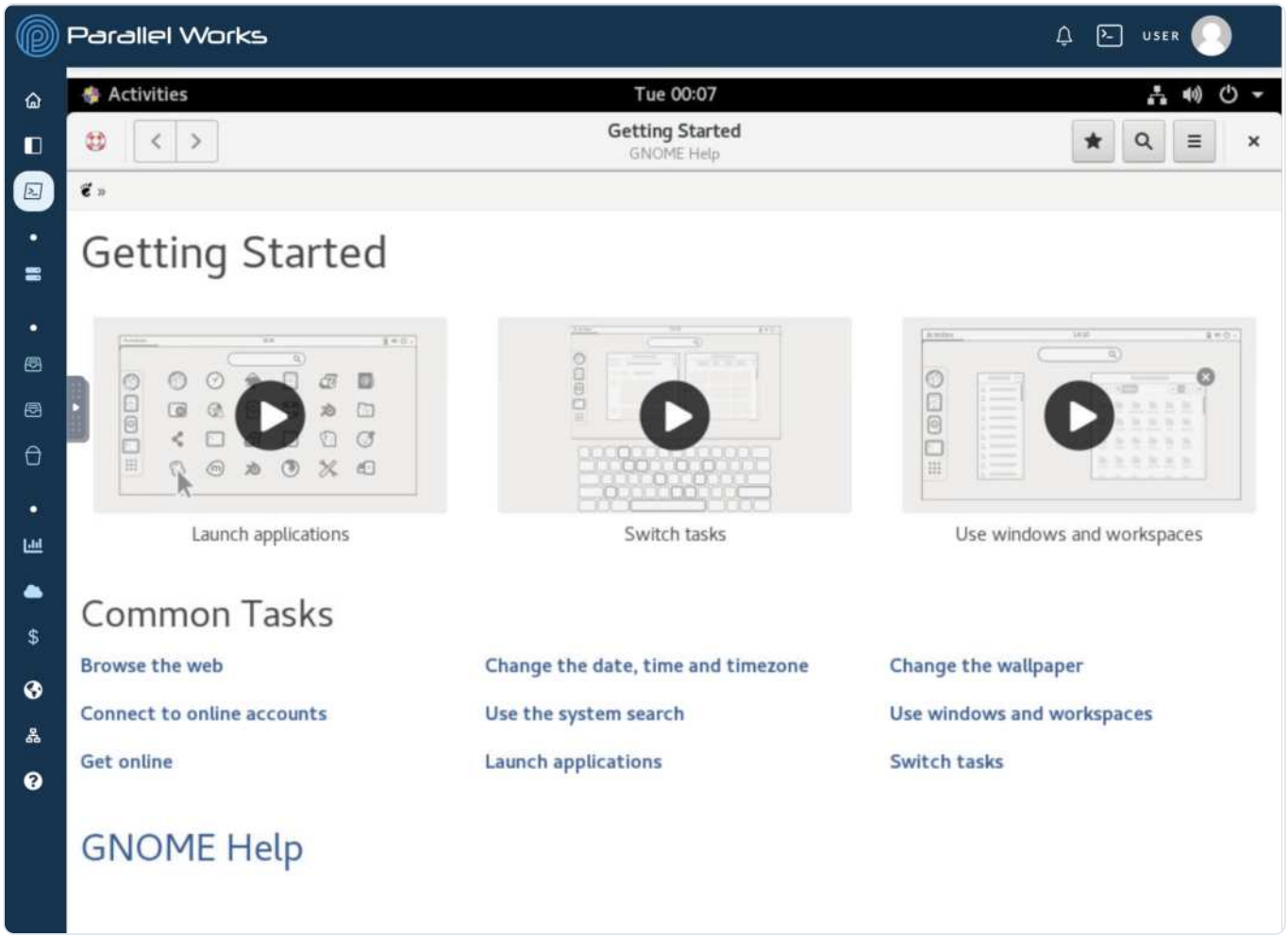
6. When a connection has been established and your workflow is ready to use, a notification will appear in the bell icon in the navigation bar. Please note that the connection is established only after the server is running. There may be a wait time if the job is in a queue, if compute nodes are starting, or if the job is installing required software. For more information on the job's status, you can check the **Logs** module.



The notification is shown in the form of [Partition Type]@MGMT-[Username]-[Resource Name]-[Partition Number].

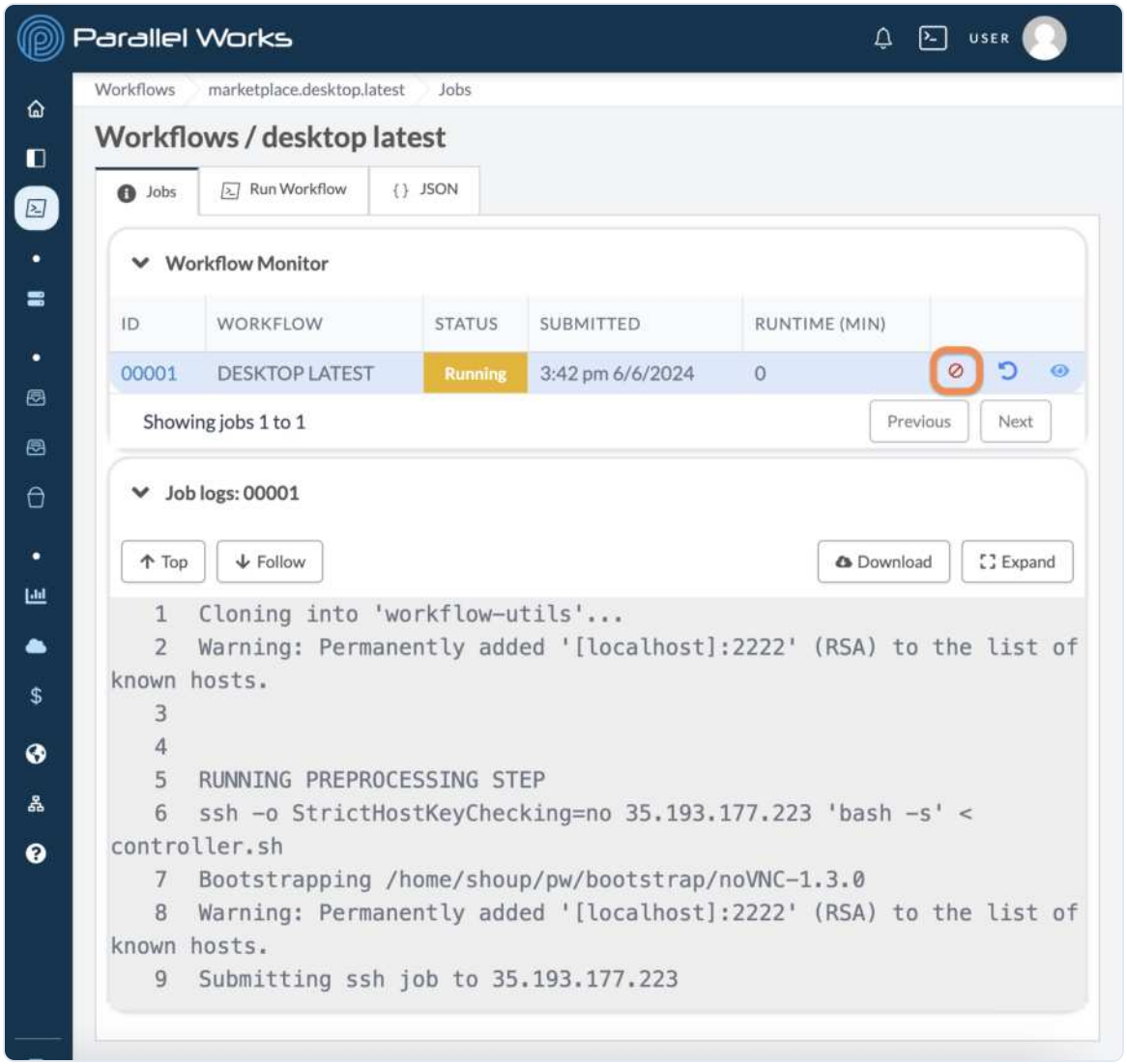
To access the server, click the **View** icon in the **Workflow Monitor** module.


Parallel Works
USER
Workflows
marketplace.desktop.latest
Jobs
Workflows / desktop latest
Jobs
Run Workflow
{} JSON
Workflow Monitor
ID
WORKFLOW
STATUS
SUBMITTED
RUNTIME (MIN)
00001
DESKTOP LATEST
Running
3:42 pm 6/6/2024
0
Showing jobs 1 to 1
Previous
Next
Job logs: 00001
Top
Follow
Download
Expand
1 Cloning into 'workflow-utils'...
2 Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
3
4
5 RUNNING PREPROCESSING STEP
6 ssh -o StrictHostKeyChecking=no 35.193.177.223 'bash -s' < controller.sh
7 Bootstrapping /home/shoup/pw/bootstrap/noVNC-1.3.0
8 Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
9 Submitting ssh job to 35.193.177.223



Pictured: the Desktop workflow running as an interactive session. The default desktop Getting Started window can be removed with the close box in the upper right corner to reveal a standard desktop environment. This environment is the desktop on the resource specified at the launch of the workflow (i.e. a cluster head node or worker node).

7. When you're done, click the **Cancel** icon to stop the job.



Developing Interactive Sessions

The rest of this page details the technical foundation for how interactive sessions work if you want to customize existing interactive sessions or develop new ones. As such, most users will be interested in the section above, while the rest of this page is most relevant to developers.

Required Files

Main Script

As we defined in **Building Workflows**, interactive sessions require a main script. This script is used to start and/or connect the server and start the SSH tunnel (if the server is running in the resource).

Cancel Script

Interactive sessions also require a cancel script. This script is used for clean-up tasks when the job is canceled on ACTIVATE, such as stopping the server and removing the SSH tunnel.

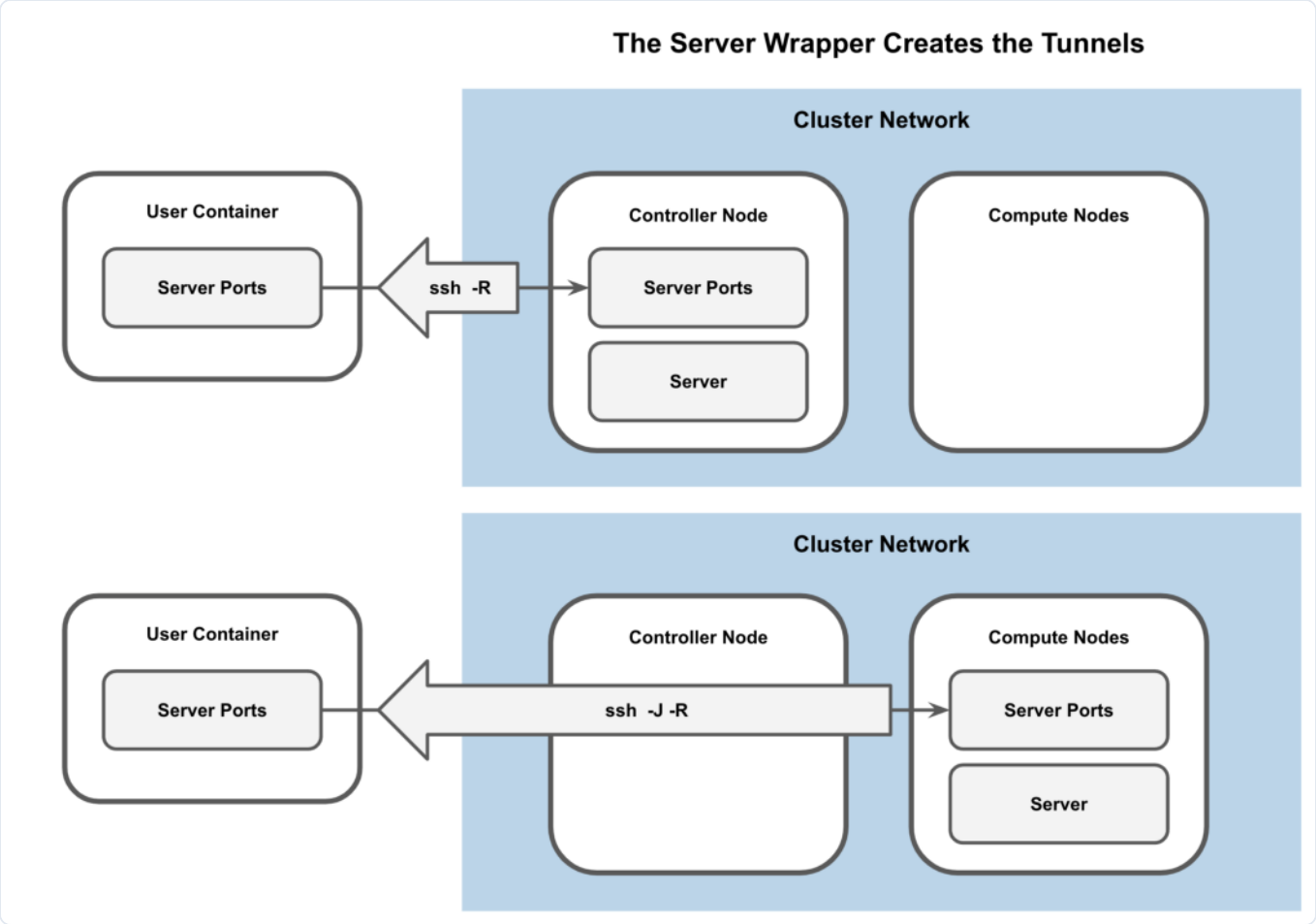
`service.json`

The `service.json` file contains the required information for ACTIVATE to establish a connection with the server using this URL: `https://activate.parallel.works/<URL>/<PORT>/<SLUG>`. Access to this URL is restricted to users who are logged in to ACTIVATE. It's important to highlight that the `service.json` file can be edited at any time and is typically updated during runtime.

This is an example of a sample `service.json` file for a Jupyter Notebook server:

```
{
  "URL": "/me",
  "PORT": "51458",
  "SLUG": "tree",
  "JOB_STATUS": " CANCELED ",
  "ERROR_MESSAGE": ""
}
```

- `PORT` represents the server's port number that ACTIVATE needs to reach. If the server is within the user's workspace, this is the server's port. Otherwise, it requires establishing a reverse SSH tunnel between the remote node and the user workspace, as depicted in the accompanying image. It is recommended to dynamically select an available port.
- `JOB_STATUS` sets the status for a workflow. If the job status is `running` or `r`, ACTIVATE connects to the server using this URL: `https://activate.parallel.works/<URL>/<PORT>/<SLUG>`. In the event of a `failed` status, ACTIVATE displays the associated error message. For all other job status values, ACTIVATE shows a loading page with the corresponding status. Uppercase or lowercase letters can be used for this parameter.
- `URL` represents either `/me` or `/sme` based on whether the server uses the HTTP or HTTPS protocols, respectively.
- `SLUG` represents the end of the URL. If the server lacks a `SLUG`, use `\"` instead.
- `ERROR_MESSAGE` is a custom error message displayed on ACTIVATE if the server encounters a failure. Note that the error message is only displayed if the job status is `failed`.



Customization

Workflow users can contact us to further customize existing workflows. For example, we can expose or enforce any Slurm or PBS directives, connect to existing services, change the service configuration, or start the GUI for a given software in the desktop.

Users can also develop their own interactive sessions. You can follow our developer's guide on GitHub to add a new service or customize any existing service.

When you add a new service, please note that interactive sessions typically follow these steps:

- Generate a submit script to start the server in the user workspace and the controller and compute nodes of the cluster with the corresponding Slurm or PBS directives
- Find available ports
- Submit job to the queue
- Monitor job in the queue
- Cancel job in the cluster when the job is canceled on ACTIVATE
- Create and destroy SSH tunnel

Any of the following folders on GitHub can serve as starting points for workflow customization:

- Jupyter Notebook Server: can be started directly in the user workspace, a controller, or a compute node; uses Docker or Singularity containers in the controller or compute nodes
- Jupyter Lab: functions similarly to Jupyter Notebook Server
- VSCode
- NGINX
- MATLAB
- RStudio
- Remote desktop services:
  - VNC Server
  - Scyld Cloud Workstation
  - NiceDCV

Supported Services

Several services can be connected to ACTIVATE using the steps above. Any directory in our interactive session repository with a `start_template.sh` file is its own service.

Each of these services can be customized for specific scenarios. For example, one NiceDCV workflow could start a server and another could connect to an existing server. The code for each scenario would be the same, but the inputs for each job would change.

We also offer services that start remote desktop servers and launch a specific application, such as MATLAB or RStudio.

Dependencies

The interactive sessions described below have specific dependencies for their successful execution. For JupyterLab and Jupyter Notebook (version 7 and above), a Singularity container with an unprivileged NGINX is essential for proxying. Typically, this and other optional Singularity containers are located in a shared directory within the cluster. Additionally, a desktop environment and VNC Server are required to establish a basic desktop session.

Jupyter

JupyterLab Host

Starts a JupyterLab server in the specified host.

Dependencies:

- JupyterLab, which can be loaded as a module or installed in a Conda environment. The workflow expects a command to load the Python environment with JupyterLab. If no command is provided, the workflow installs Conda, creates a Conda environment, and installs JupyterLab with the following commands:
  - `conda install -c conda-forge jupyterlab -y`
  - `conda install nb_conda_kernels -y`
  - `conda install -c anaconda jinja2 -y`

- Docker or an unprivileged NGINX Singularity container is required to proxy the Jupyter Notebook server. For more information, see this Singularity definition file.

Jupyter Notebook Host

Starts a Jupyter Notebook server in the specified host.

Dependencies:

- Jupyter Notebook, which can be loaded as a module or installed in a Conda environment. The workflow expects a command to load the Python environment with Jupyter Notebook. If no command is provided the workflow installs Conda, creates a Conda environment and installs Jupyter Notebook with the following commands:
  - `conda install -c anaconda jupyter -y`
  - `conda install nb_conda_kernels -y`
  - `conda install -c anaconda jinja2 -y`
- If the version of Jupyter Notebook is 7 or higher, Docker or an unprivileged NGINX Singularity container is required to proxy the Jupyter Notebook server. For more information, see this Singularity definition file.

Jupyter Notebook Singularity

Starts a Jupyter Notebook server inside a user specified Singularity container.

Dependencies:

- Singularity
- Singularity container with Jupyter. For example, see this TensorFlow Singularity definition file.
- If the version of Jupyter Notebook is 7 or higher an unprivileged NGINX Singularity container is also required to proxy the Jupyter Notebook server, see this Singularity definition file.

Jupyter Notebook Docker

Starts a Jupyter Notebook server inside a user specified Docker container.

Dependencies:

- Docker

Remote Desktop

Remote desktop sessions require a desktop environment and a remote display protocol (VNC Server, NICE DCV, SCW, etc.). Additionally, if other software tools are installed in the host, such as RStudio or MATLAB, the interactive session can be configured to open these GUI tools at start time.

VNC Server

Starts a remote desktop server using VNC Server.

Dependencies:

- VNC Server. For example:

- `dnf install tigervnc-server -y`
- `dnf install python3 -y`

- Desktop environment. For example:
  - `dnf groupinstall "Server with GUI" -y`

NICE DCV Server

Starts a remote desktop server using NICE DCV.

Dependencies:

- NICE DCV Server. Follow this link for installation instructions.
- Desktop environment. For example:
  - `dnf groupinstall "Server with GUI" -y`

Scyld Cloud Workstation Server

Starts a remote desktop server using Scyld Cloud Workstation.

Dependencies:

- Scyld Cloud Workstation. Follow this link for installation instructions.
- Desktop environment. For example:
  - `dnf groupinstall "Server with GUI" -y`

Other

MATLAB Docker

Starts a MATLAB server inside a Docker container, see this link.

Dependencies:

- Docker

This session could be converted to Singularity in which case it needs:

- Singularity container with MATLAB
- An unprivileged NGINX Singularity container, see this Singularity definition file.

Open Visual Studio Code

Starts a VS Code server.

Dependencies:

- VS Code. For example, see this link.

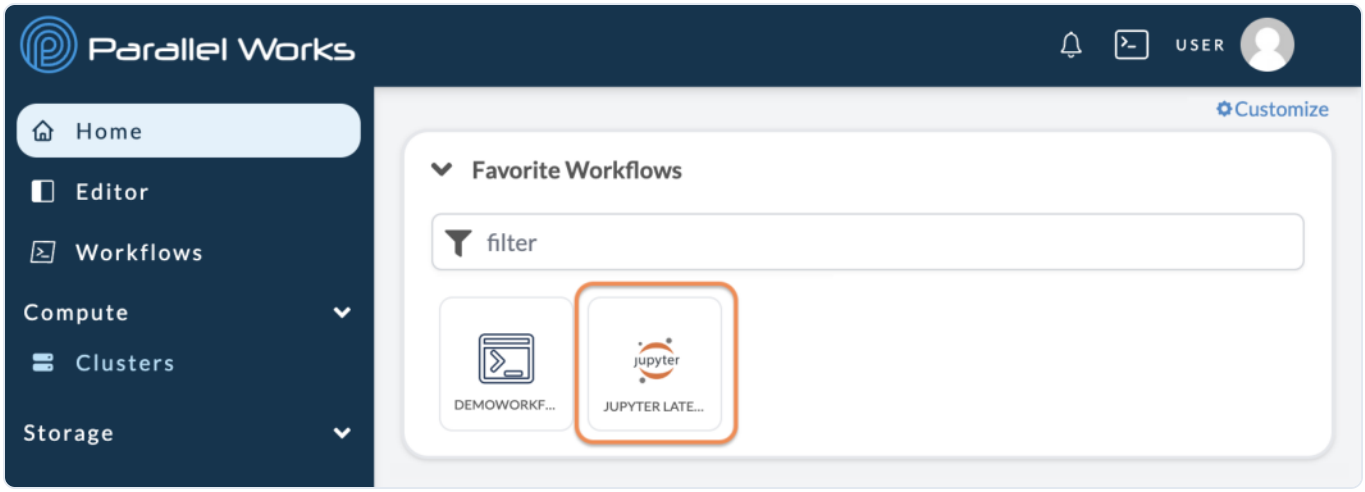
Webshell

Starts a terminal. It does not require admin installation.

Jupyter Notebook

This page focuses on the Jupyter workflow, which enables you to run and manage Jupyter Notebooks.

After adding the Jupyter Notebook Server workflow, navigate to the **Workflows** page and click its name. If you've favorited it, you can also click its card from the **Favorite Workflows** widget on your **Dashboard**.



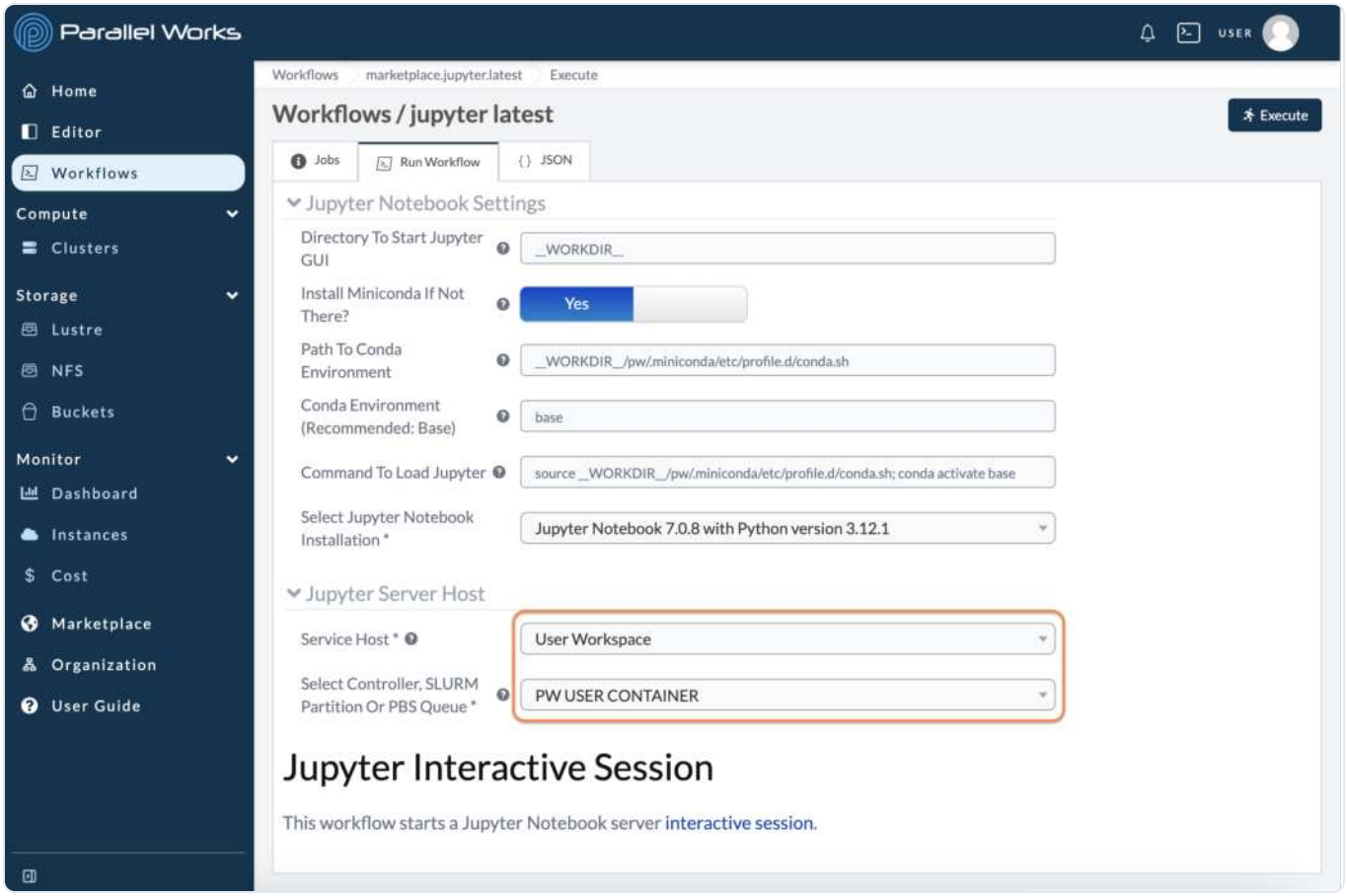
You'll be taken to the workflow's configuration page.

Navigate to the **Jupyter Server Host** section. The settings here depend on where you're running the workflow: in your user container, on a controller node, or on a compute node.

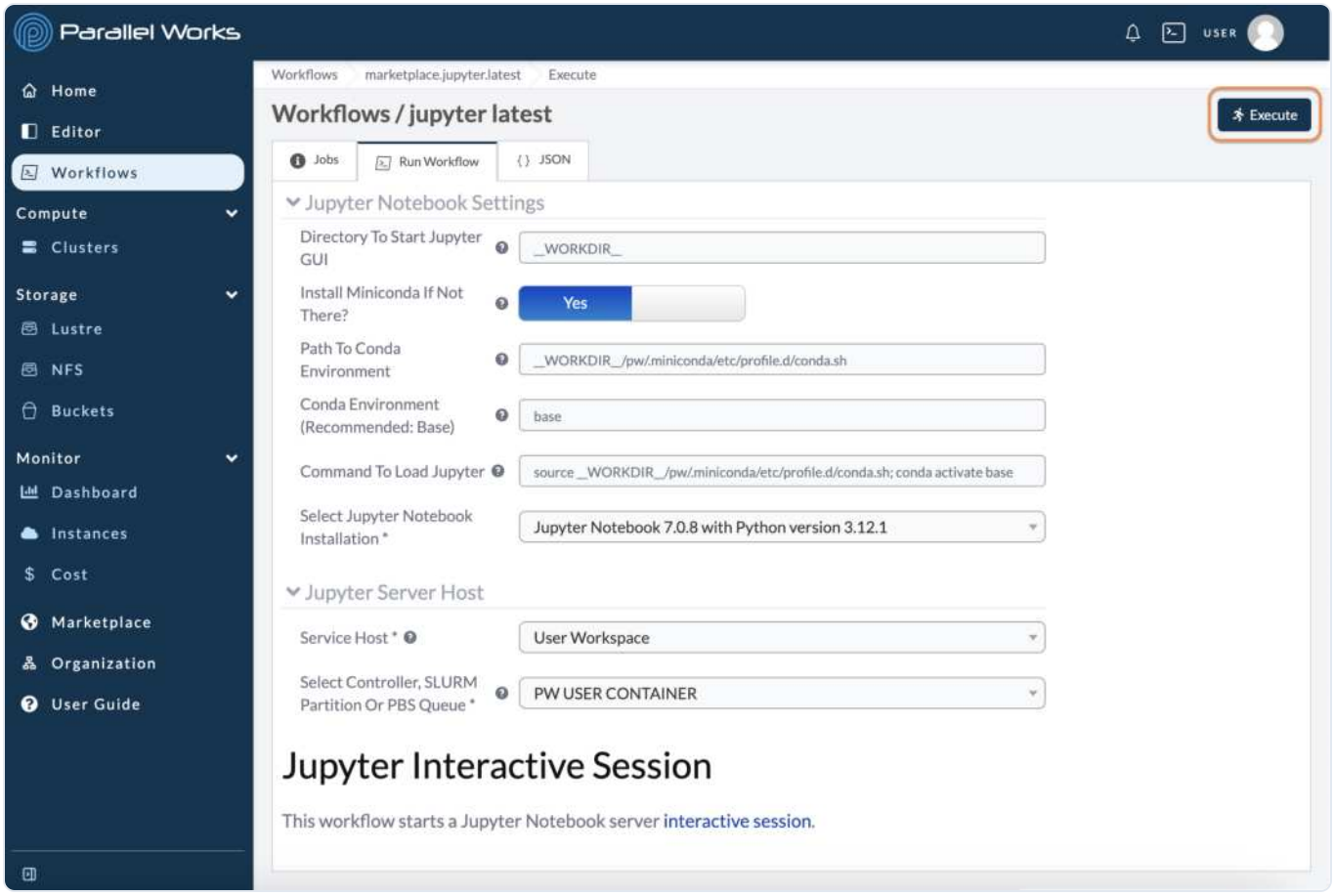
> **ABOUT RESOURCES**
>
> Running an interactive session on a controller or compute node requires a running resource. If you plan to run an interactive session in your ACTIVATE user workspace, you do not have to have a resource running.

Use the **Service Host** dropdown menu to choose where the workflow will be hosted. Select `User Workspace`.

Use the **Select Controller...** dropdown menu to choose how the job will be submitted. Select `PW USER CONTAINER`.

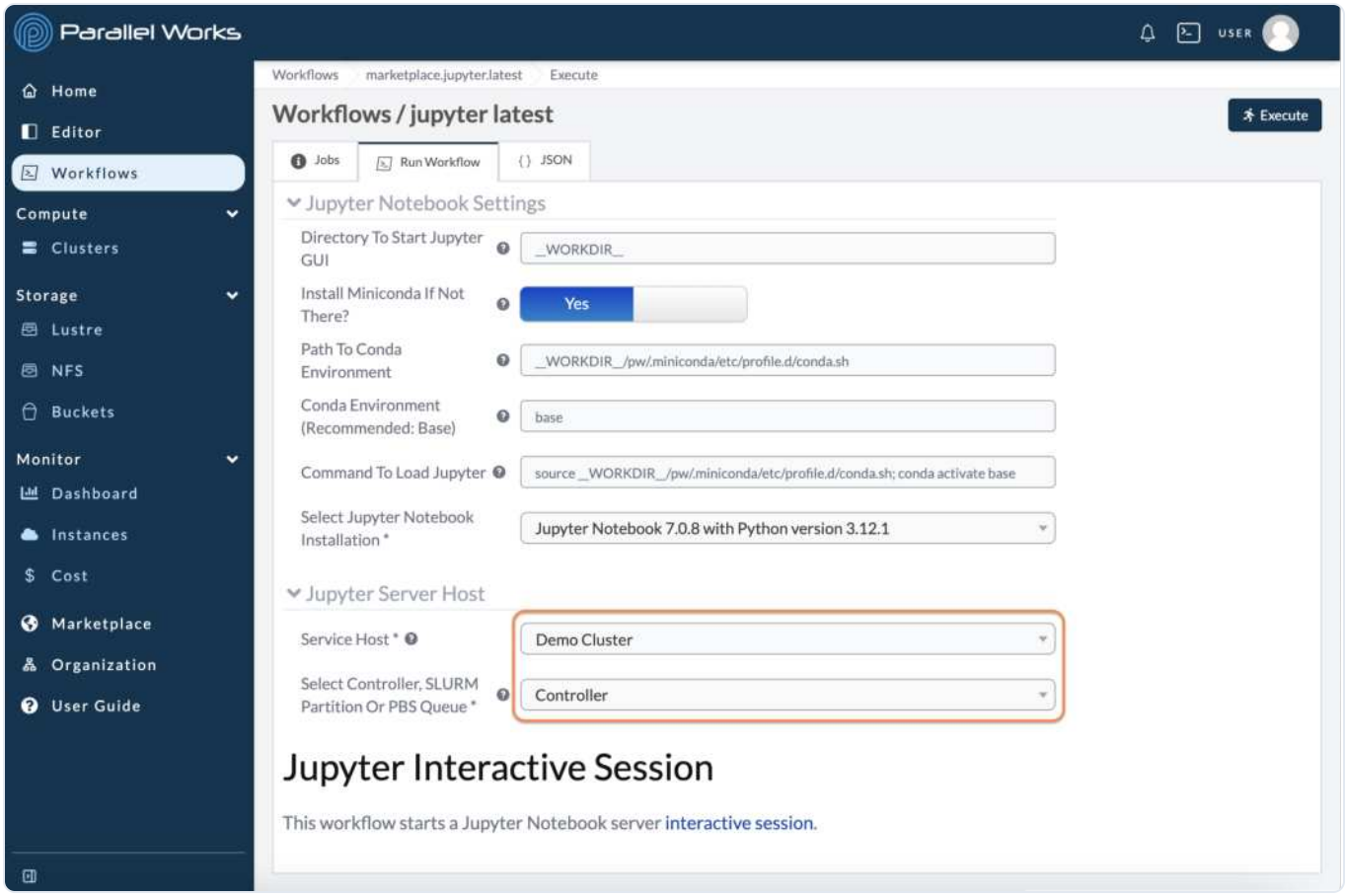


Click **Execute**.

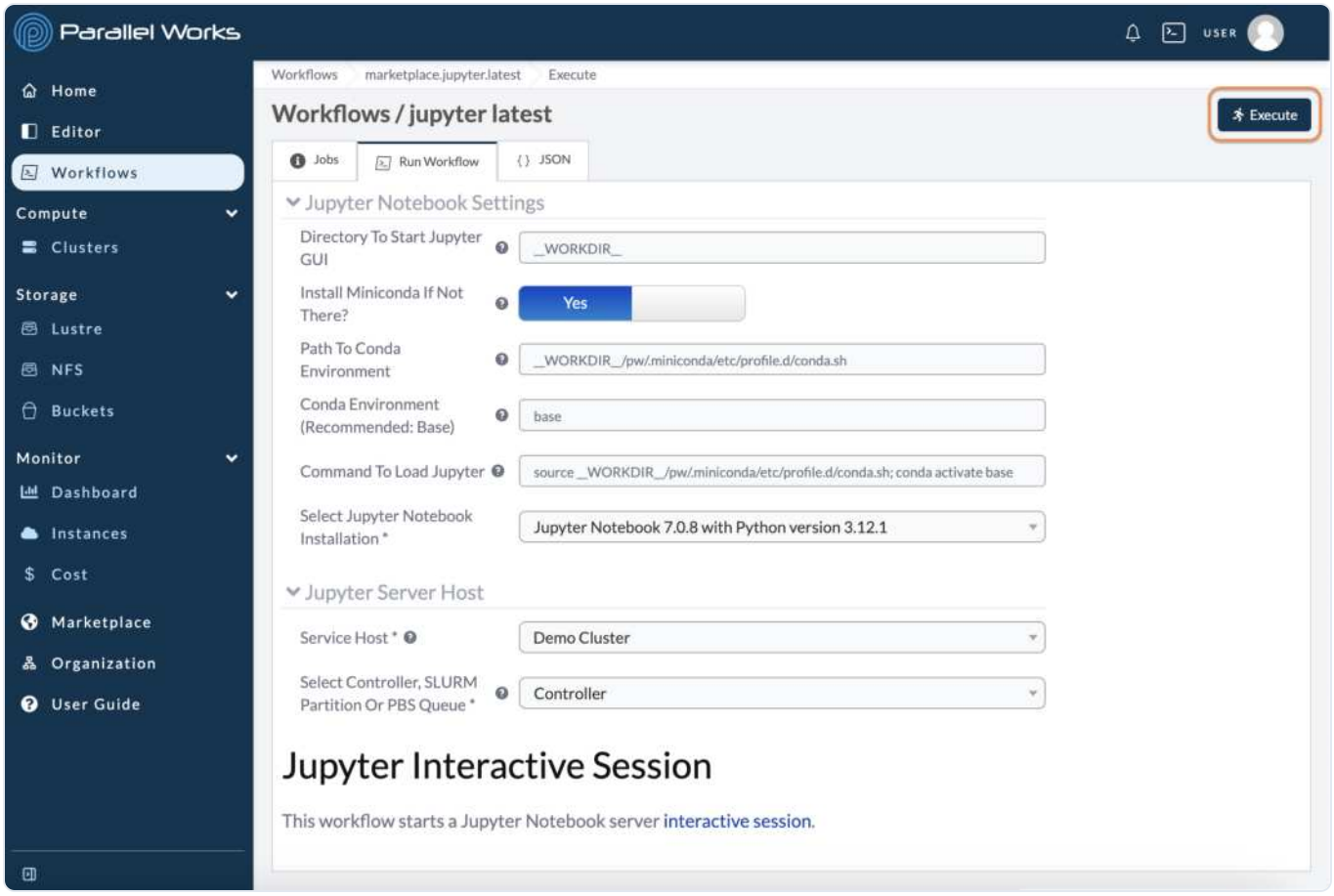


Use the **Service Host** dropdown menu to choose which cluster will host the workflow. We selected `Demo Cluster`.

Use the **Select...** dropdown menu to choose how the job will be submitted. Select `Controller`.



Click **Execute**.

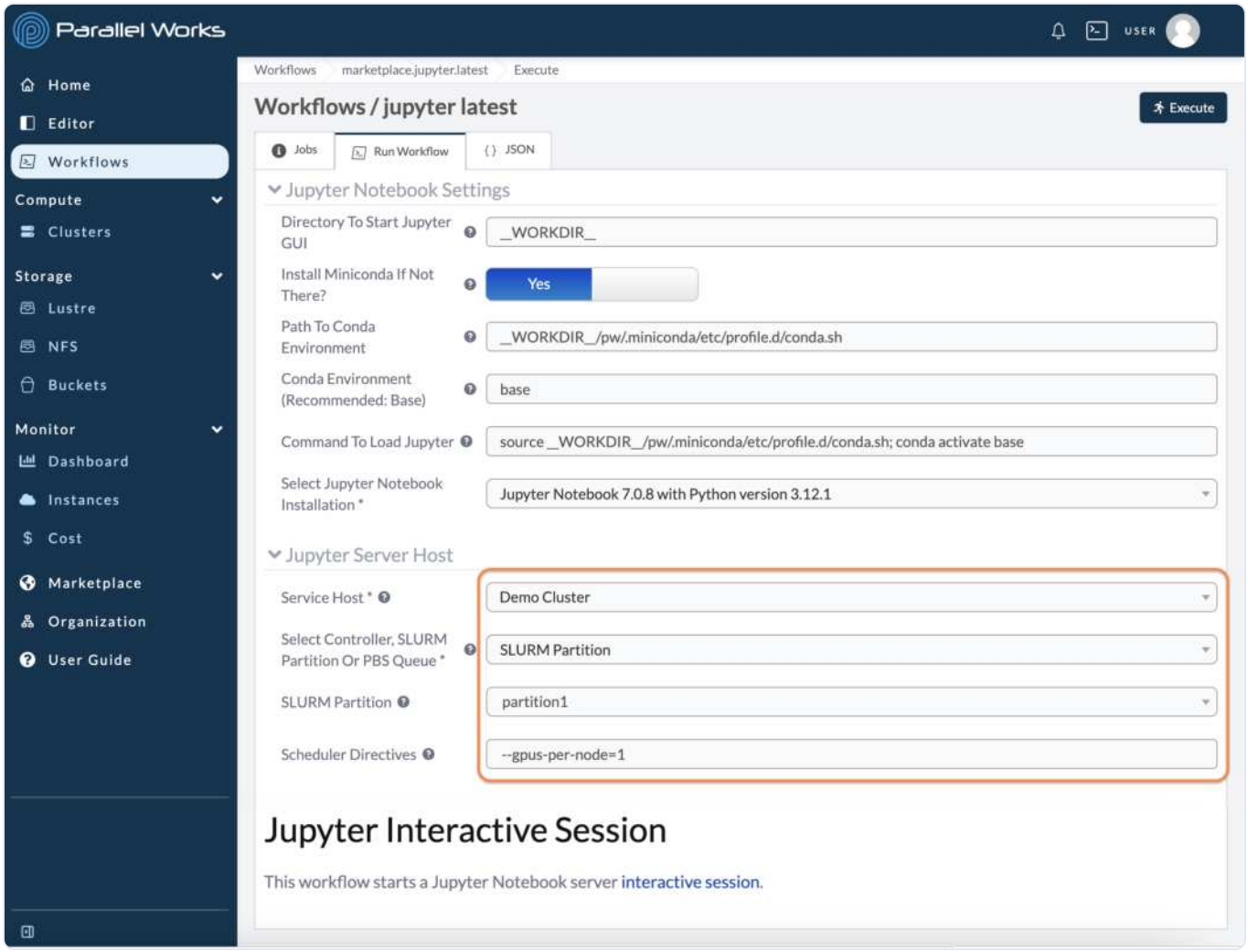


Use the **Service Host** dropdown menu to choose which cluster will host the workflow. We selected `Demo Cluster`.

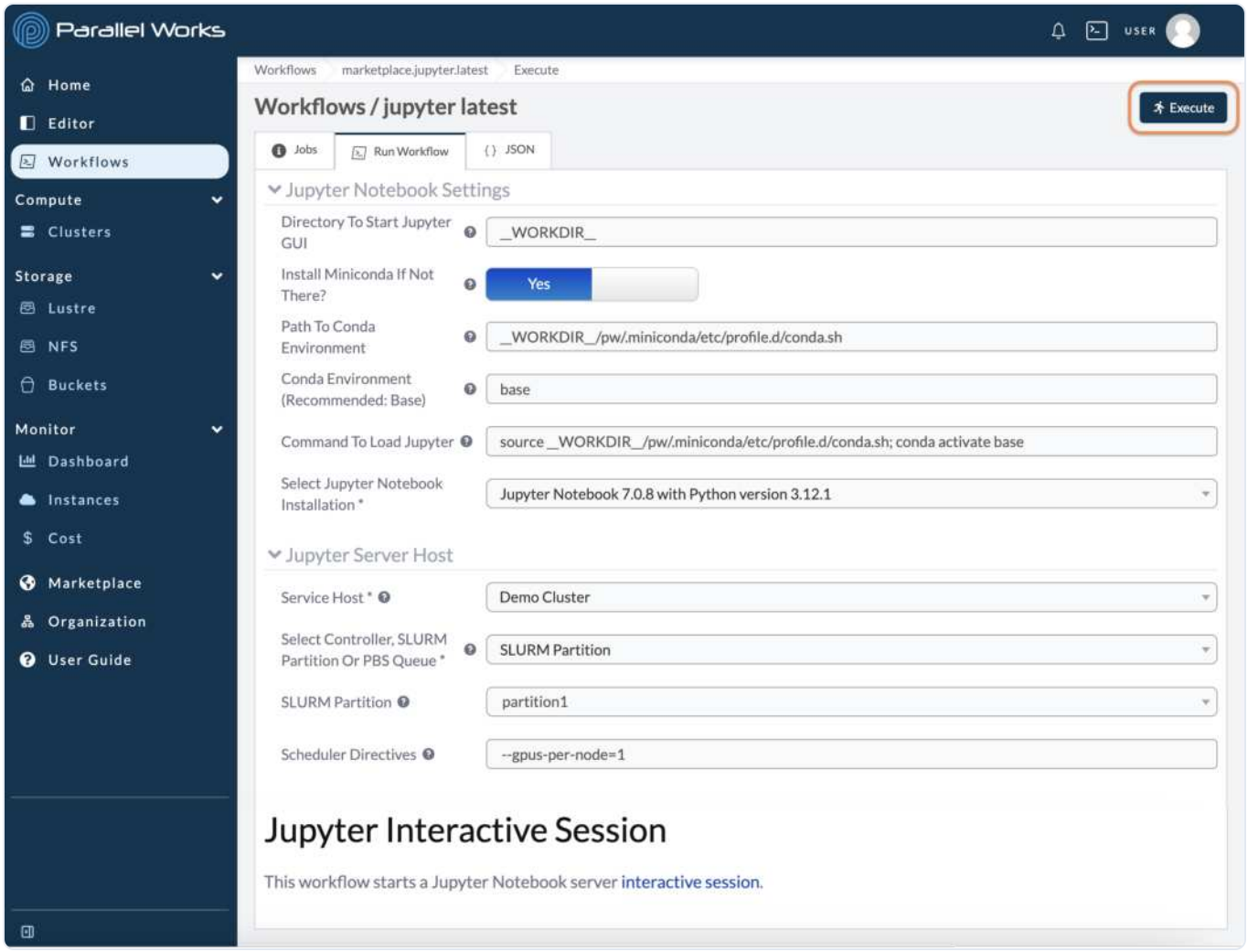
Use the **Select...** dropdown menu to choose how the job will be submitted. We selected `SLURM Partition`.

Optionally, you can use the next field to name a specific partition to run the workflow. This field will be named either **SLURM Partiton** or **PBS Queue**. We used `partition1`.

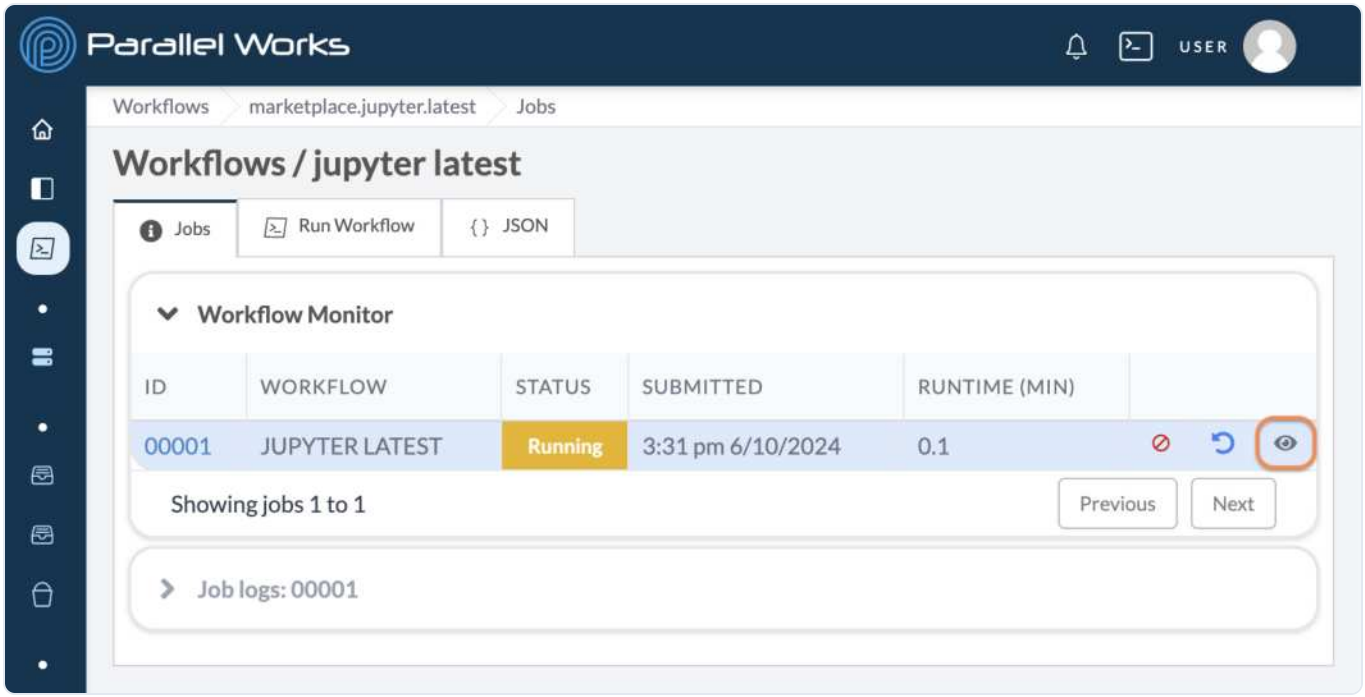
Optionally, use the **Schedular Directives** field to enter additional parameters for your partition. We entered `--gpus-per-node=1`.



Click **Execute**.

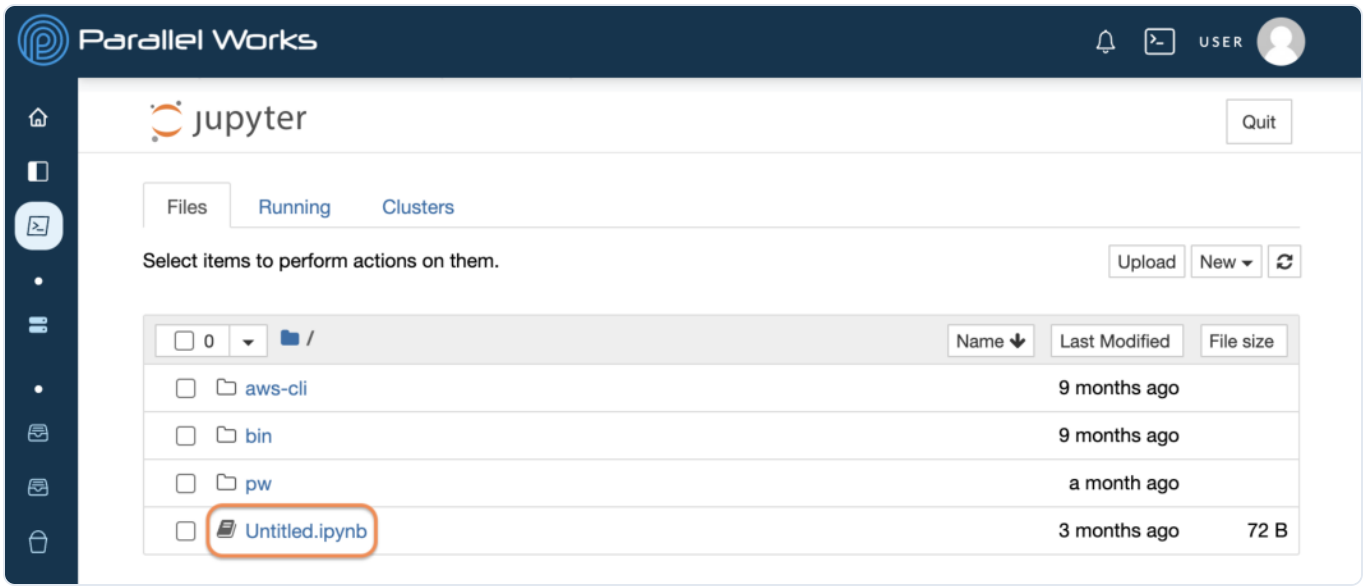


You'll be taken to your workflow's configuration page. In the **Workflow Monitor** module, click the **View** icon.

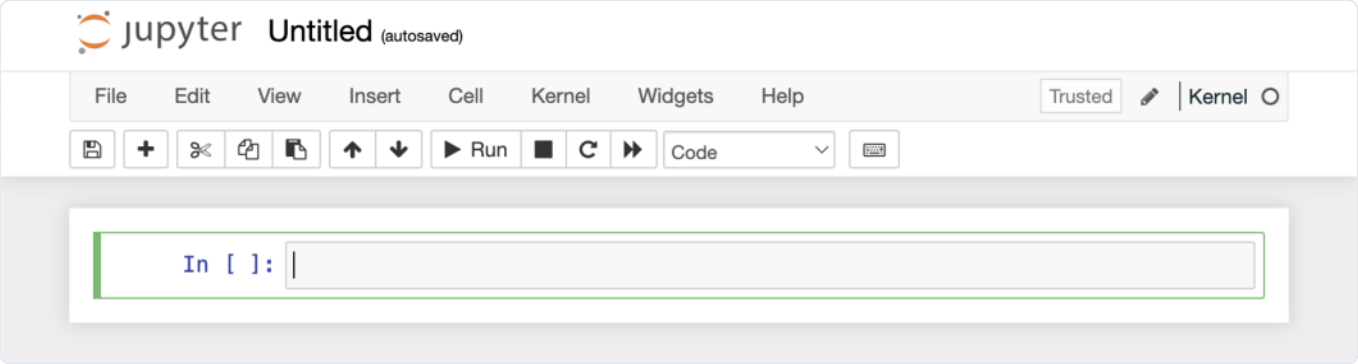


You may see a page with the message *Validating URL* before the page loads. If you see the message for more than 30 seconds, you can refresh the page.

On the Jupyter page, click the `untitled.ipynb` file.

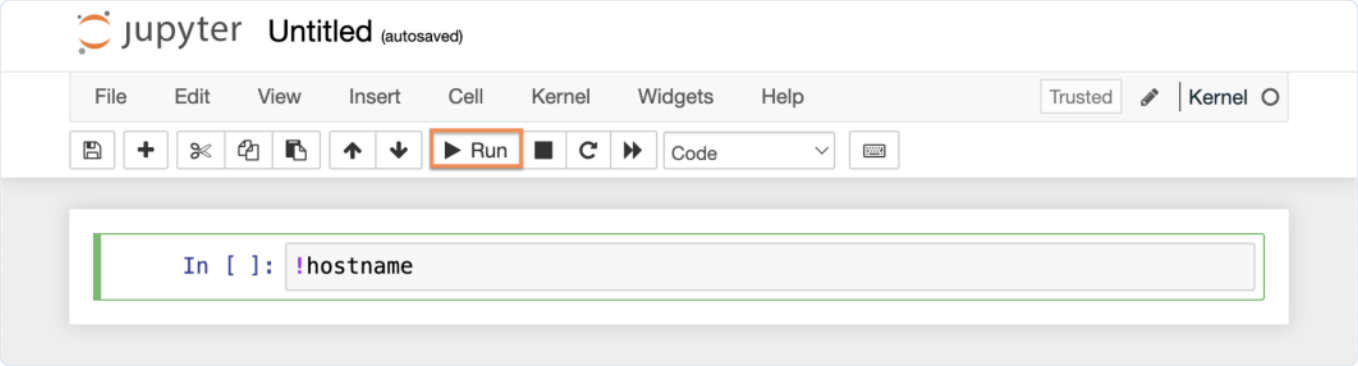


Your Jupyter Notebook will open in a new tab. You can browse files, create a new Notebook, or double-click on an existing Notebook within your `/pw` file system.

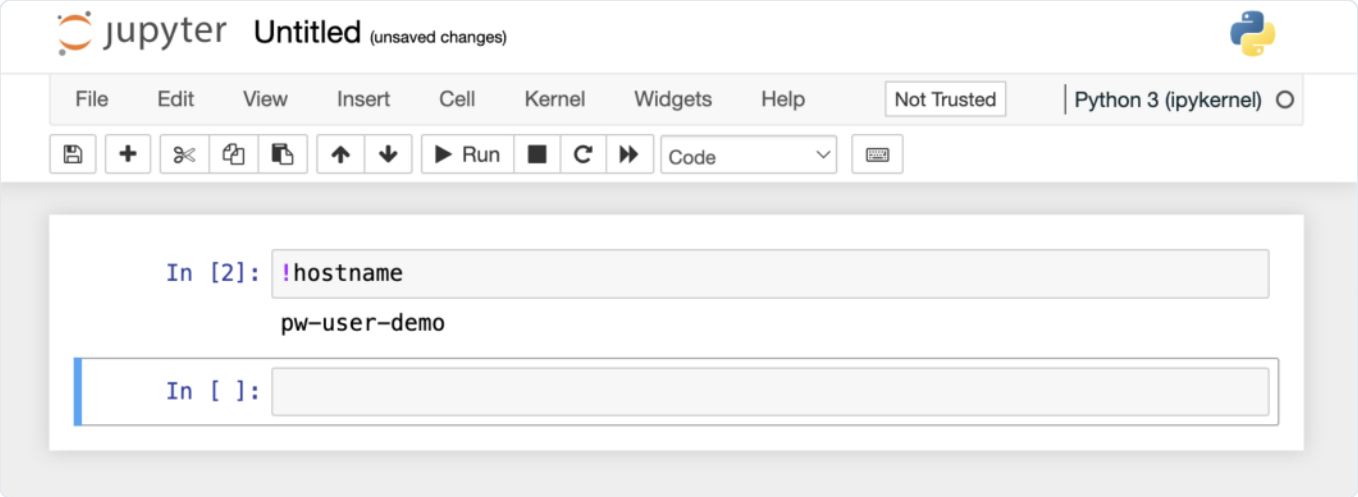


Enter `!hostname` to confirm where the Notebook is running. The `!` tells Jupyter to execute the `hostname` command in the shell of the underlying operating system.

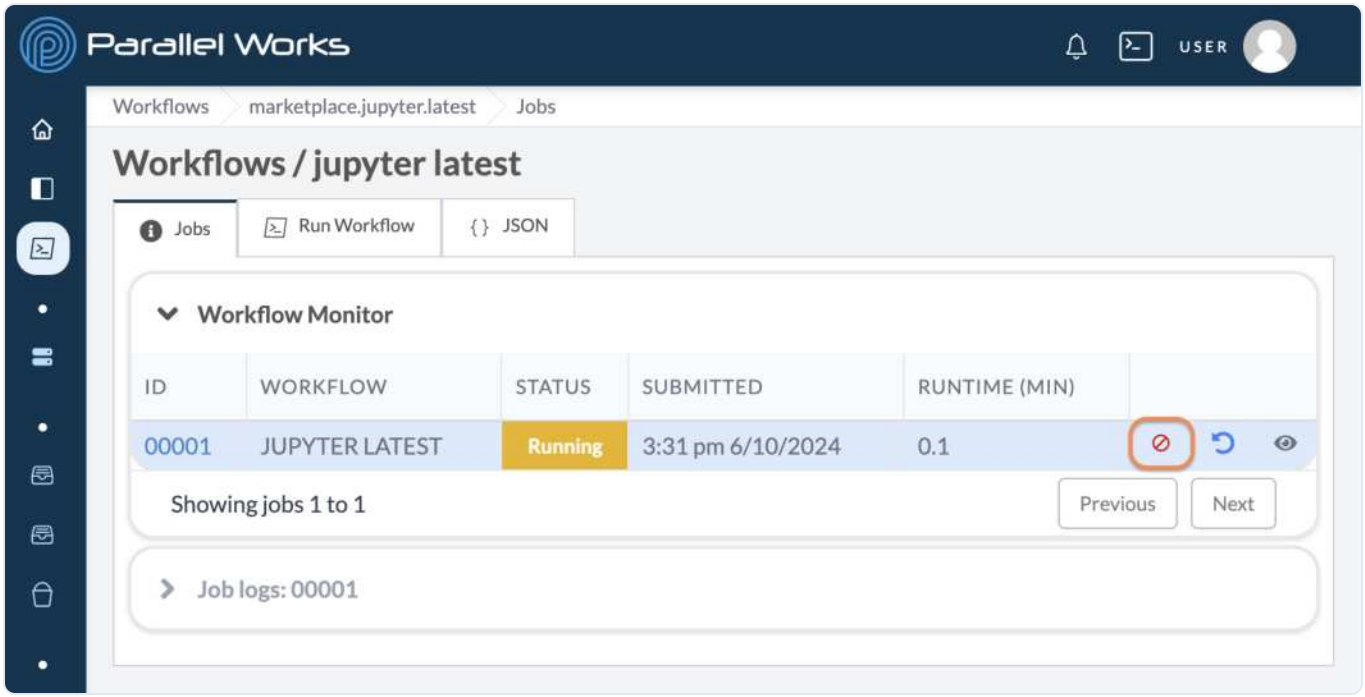
Click **Run**.



The Notebook will generate a new line with the output `pw_user_<your_user_id>`, which is the same hostname displayed in your ACTIVATE IDE terminal.



Now you can use the Jupyter Notebook for anything. Please be sure to end the workflow when you're finished by navigating to the **Workflows** page and clicking the **Cancel** icon.



ABOUT KERNELS

You can connect a running Jupyter Notebook to a kernel (i.e., execution engine) in a different Conda environment. To see your available kernels, click **Kernel > Change kernel** in the Jupyter banner.

Storage

This page explains how storage works with your resources and ACTIVATE.

> **NOTE**
>
> We make a distinction between the storage you create on the ACTIVATE platform and the storage you create outside the ACTIVATE platform. The information on this page and our pages for creating, configuring, attaching, and sharing storage apply to storage resources made on the platform.
>
> If you would rather create storage outside ACTIVATE, or if you have existing storage resources that you'd like to work with, please see the following:
>
> - Amazon S3 documentation
> - Azure Blob Storage documentation
> - Google bucket documentation
>
> Our section **Transferring Data** contains instructions for AWS, Azure, and Google.

Storage Types

When creating storage resources, you'll choose whether your storage is ephemeral or persistent.

Ephemeral storage is created and destroyed with a cluster. This type of storage is provisioned when you start the cluster it's attached to, and it's destroyed when you stop the cluster. Ephemeral storage resources do not have a power button, and they won't appear in the **Storage Resources** module on the the **Compute** page.

Persistent storage is created and destroyed independently. If you want to destroy a persistent storage resource, you can do so from the **Storage** page by clicking the power button.

Persistent storage can be attached to multiple clusters.

At this time, only Lustre storage options can be ephemeral.

Storage Provisioning Logs

For persistent storage resources, you can see the creation and deletion logs in this tab. All logs for current and previous sessions will be displayed. If a persistent storage resource is active, you'll see all the clusters that the storage is attached to. For more information, please see **Sessions** in **Configuring Storage**.

Ephemeral storage resources do not have a **Sessions** page. Instead, you can see their creation logs on the cluster they're attached to by clicking the cluster's status when it's provisioning.

Buckets

Typically, users would have to use a cloud service provider's CLI to upload or retrieve data from a bucket. With our new filesystem feature, buckets are instead mounted as a filesystem; this way, you can interact with a bucket as if it's a directory rather than using CLI commands.

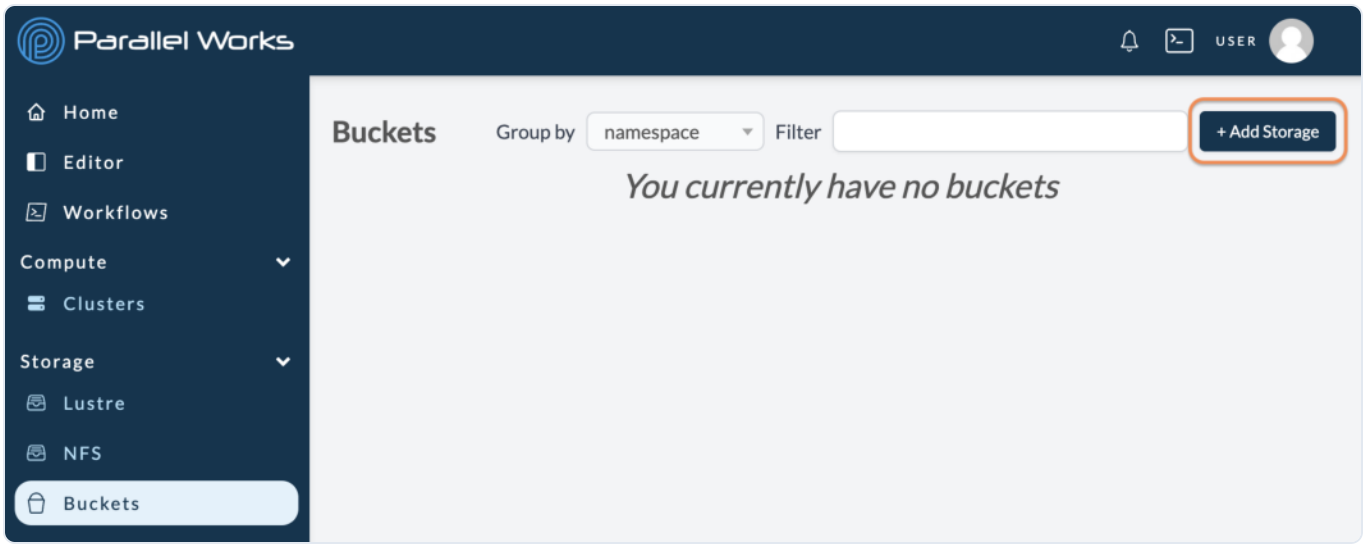
Creating Storage

Creating Storage

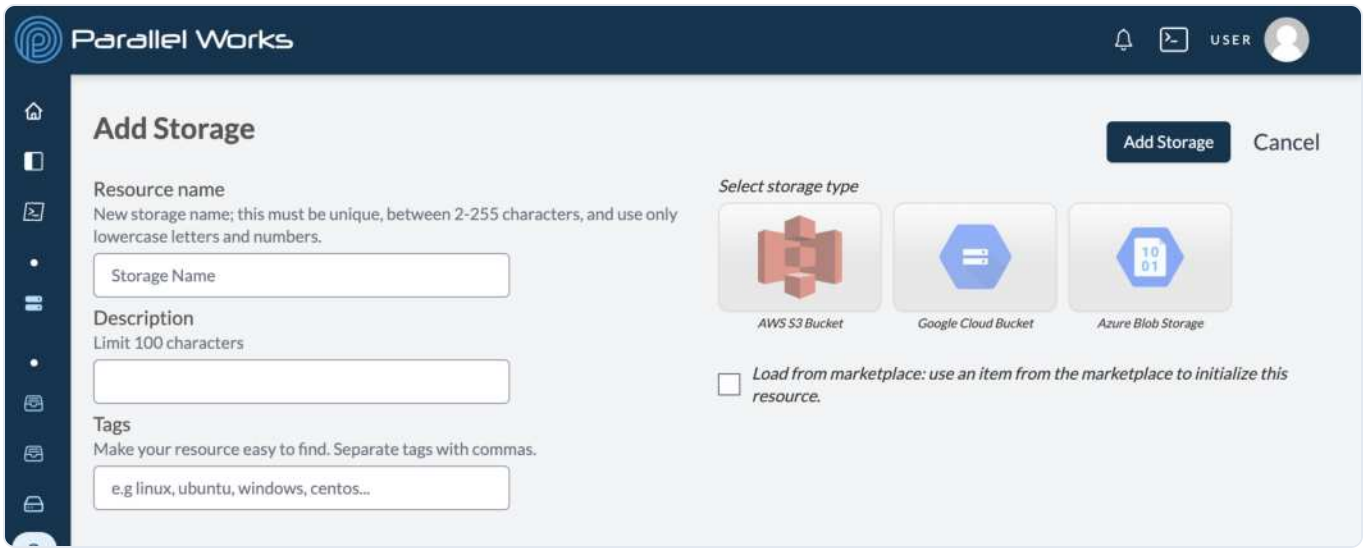
> **NOTE**
>
> This page is a quick-start guide for creating a storage resource from scratch. You can also add pre-configured storage resources from the ACTIVATE Marketplace. For more information, please see **Navigating the Marketplace**.

Navigate to one of the **Storage** pages: **Lustre**, **NFS**, or **Buckets**.

Click + **Add Storage**.



On the next page, you can choose which type of storage to create.

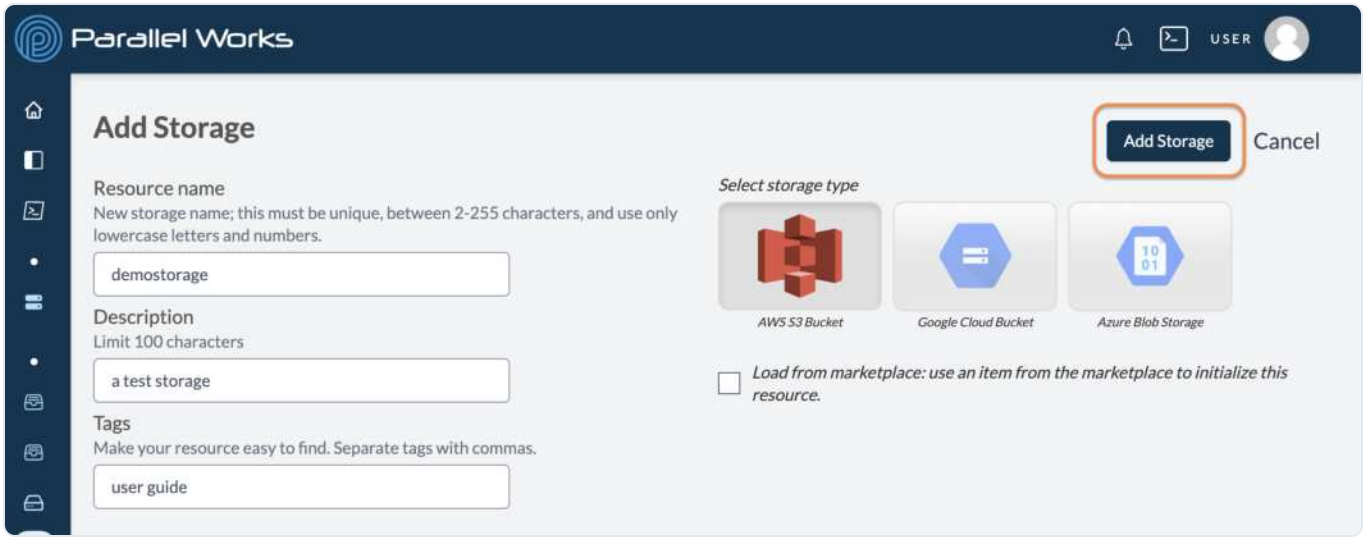


Select a **Storage Type**. The visible options depend on which storage category you're creating (Lustre, NFS, or a bucket).

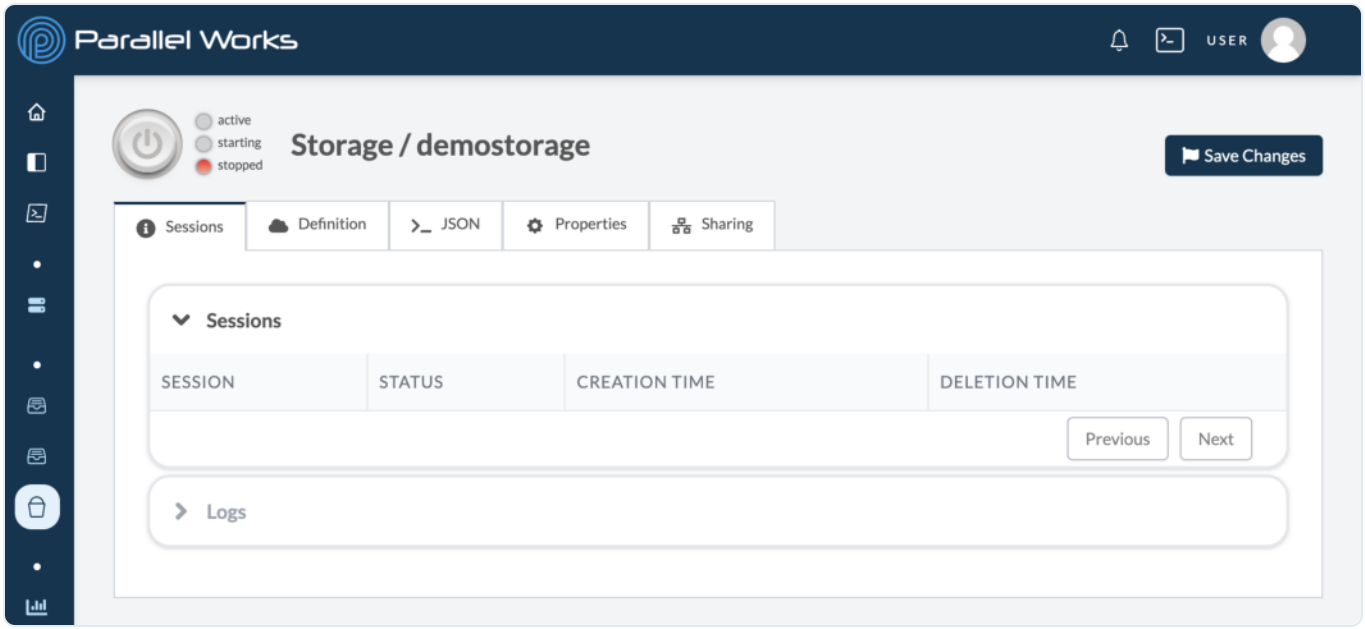
Enter a **Storage Name**. The **Description** and **Tags** are optional.

If you're configuring a Lustre storage, select whether it will be **Ephemeral** or **Persistent**. Please see **Storage Types** for more information.

Click **Add Storage**.



Next, you'll see the default settings page for the new storage. Please see **Configuring Storage** for the next steps.



NOTE

After you create a persistent storage resource, it will appear in the **Storage Resources** module on the **Home** page. You can hide storage resources from the **Home** page by clicking the eye icon on the **Storage** page.

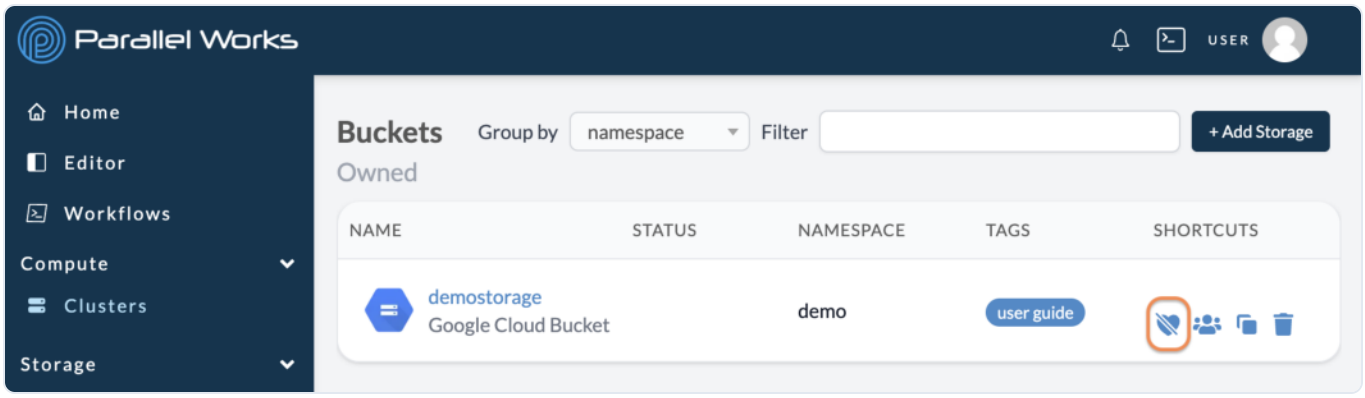
However, ephemeral storage resources will not appear on the **Home** page because they cannot be started independently; they must be attached to a cluster, where the storage will be created, started, and deleted along with the cluster.

For more information, please see **Storage Types**.

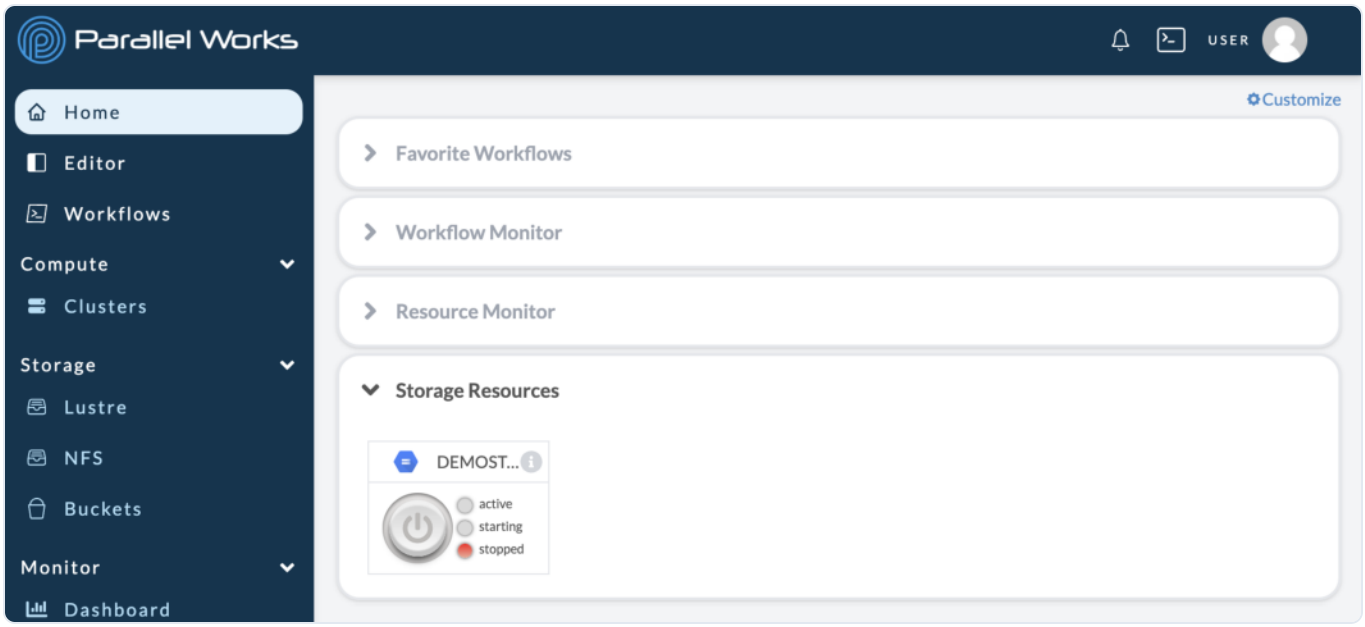
Favoriting Storage

When you create a persistent storage resource, it will automatically be added to the **Storage Resources** module on your **Home** page.

You can also add frequently used storage resources to your **Dashboard** by clicking the slashed heart icon. The icon will change to a plain heart after you click it.



The storage resource will now appear in **Storage Resources** on the **Dashboard**.



> **NOTE**
> You cannot favorite ephemeral storage resources.

Which type of storage should I use?

Lustre is a high input/output filesystem, so we recommend using Lustre storage resources only when you're completing compute work that needs that type of performance.

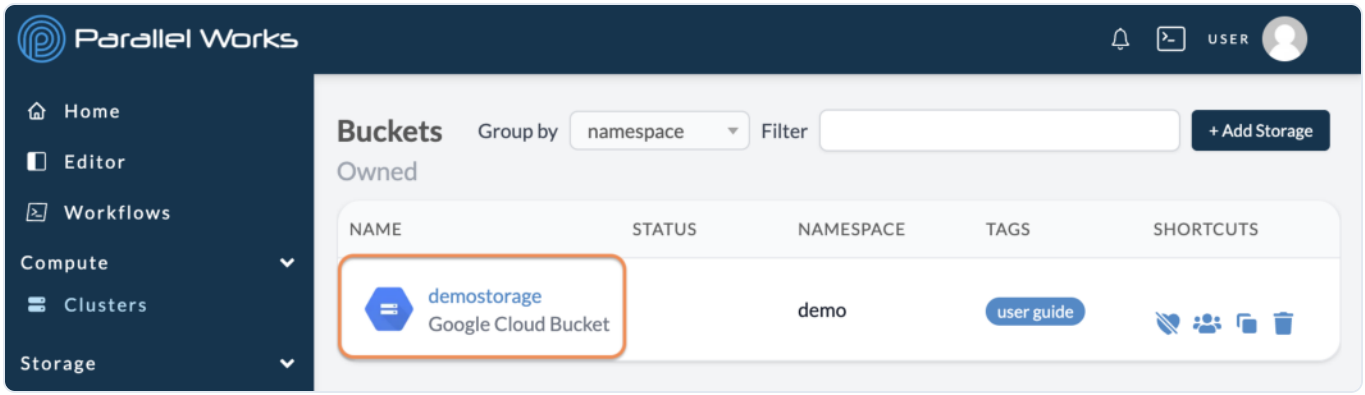
The other storage resources work well for general use.

Configuring Storage

Accessing Storage Configuration Settings

You'll see the storage configuration page immediately after creating any storage resource.

You can also access configuration settings from the **Lustre**, **NFS**, **Disks**, or **Buckets** page. Click the name of any existing storage resource to configure it.

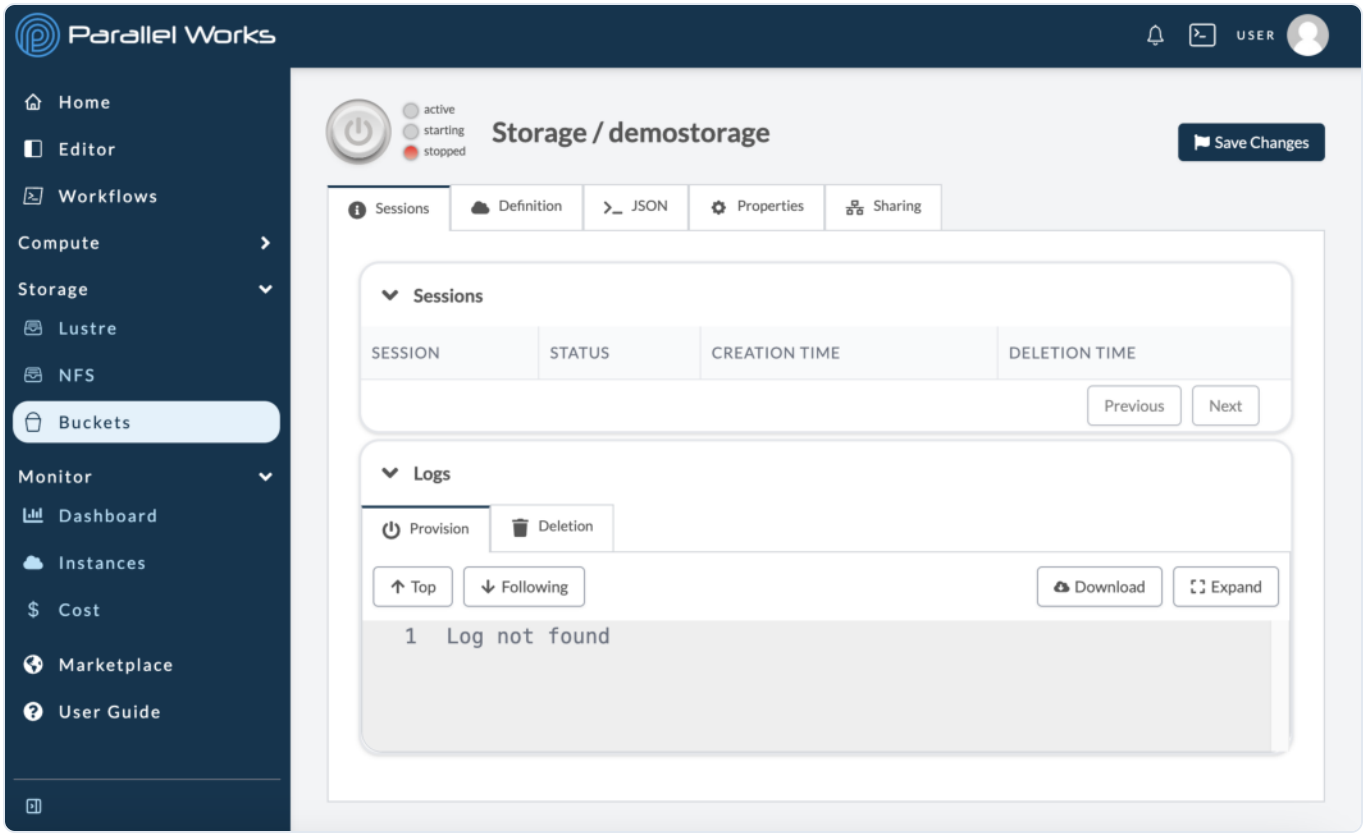


About the Storage Configuration Page

When you navigate to the storage configuration page, there are five tabs for customization.

Sessions

By default, you'll see the **Sessions** tab when you navigate to storage settings. This tab shows your previous sessions for using the storage as well as provision and deletion logs.



ABOUT STORAGE PROVISIONING LOGS

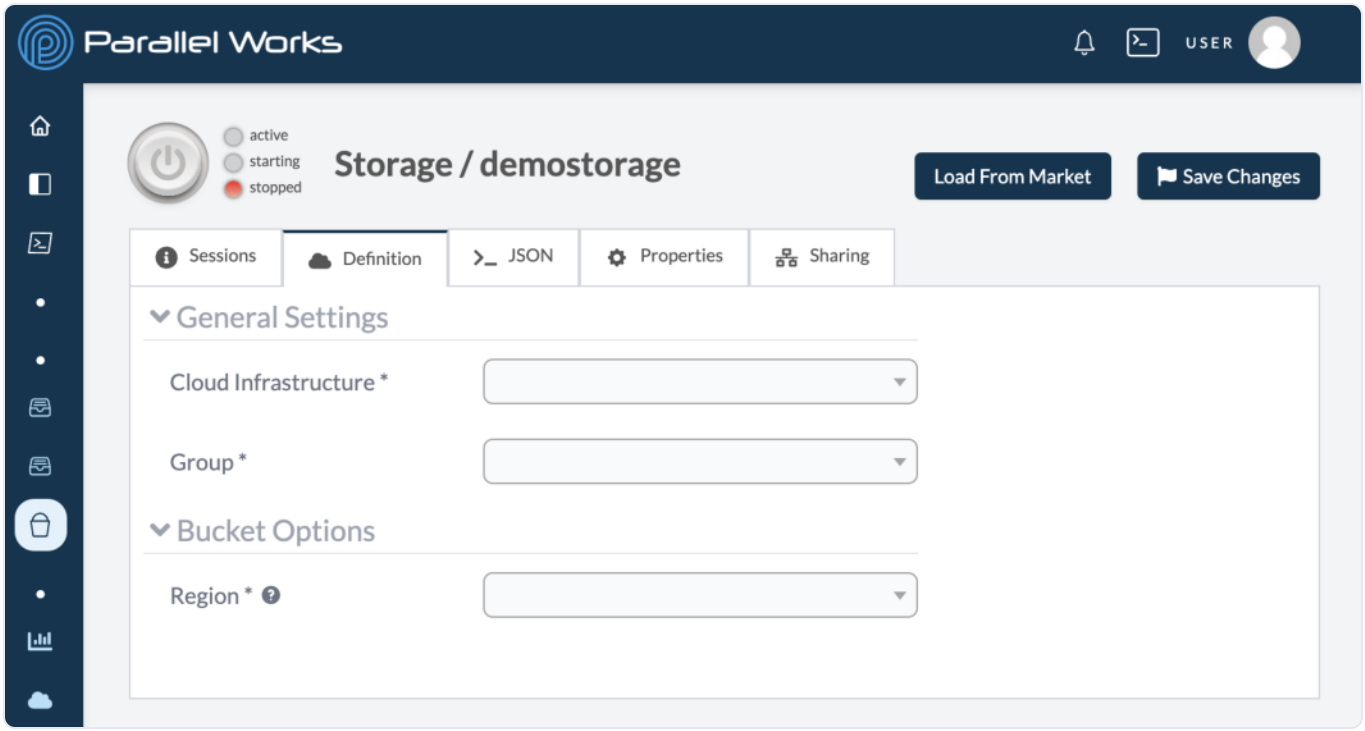
For persistent storage resources, you can see the creation and deletion logs in this tab. All logs for current and previous sessions will be displayed. If a persistent storage resource is active, you'll see all the clusters that the storage is attached to.

Ephemeral storage resources do not have a **Sessions** page. Instead, you can see their creation logs by navigating to **Resources** > **ResourceName** > **Sessions** > **Logs** > **Storages**.

For more information about persistent and ephemeral storage, please see **Storage Types**.

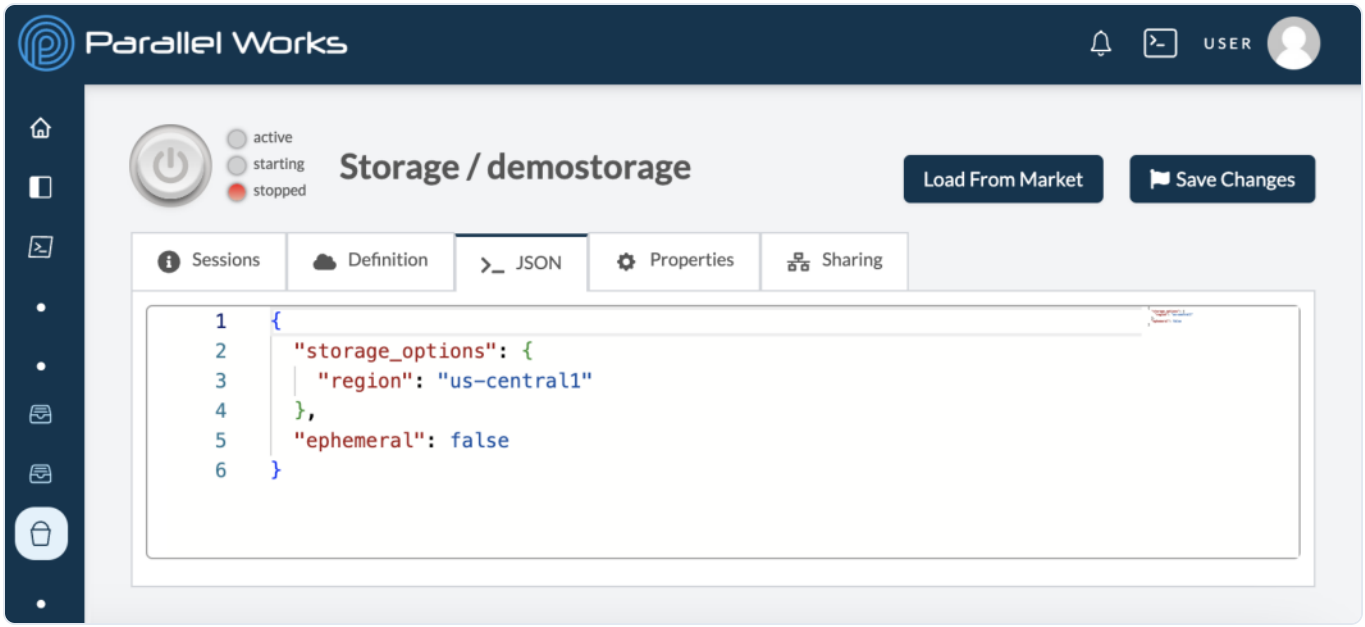
Definition

Here, you can adjust the configuration parameters of your storage. For more information, see **General Settings** below.



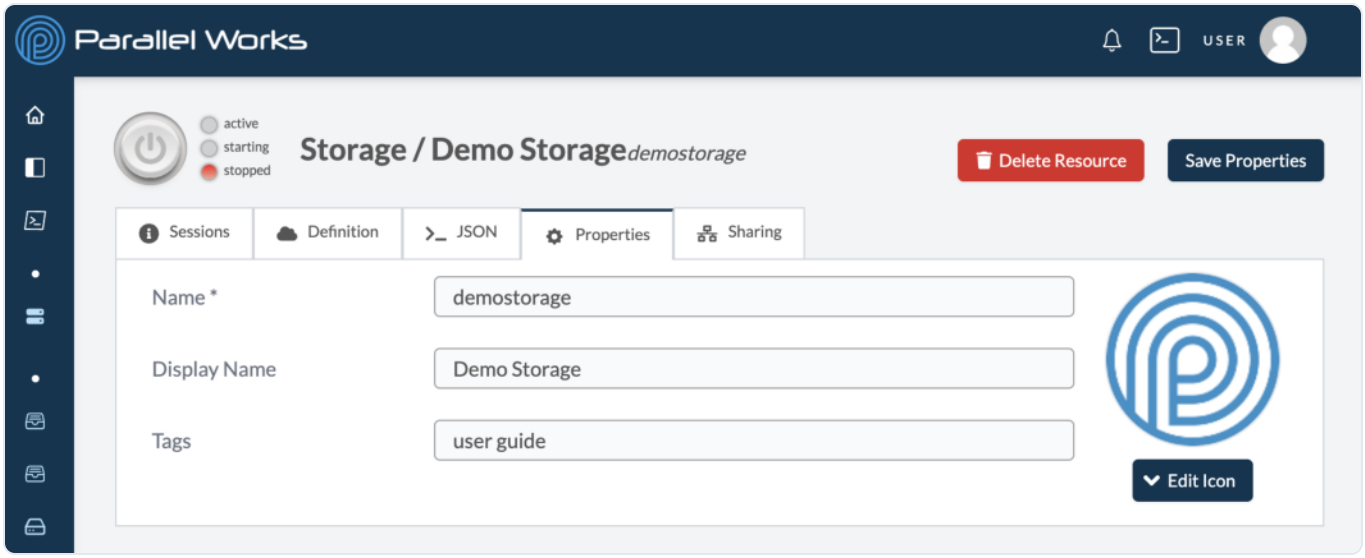
JSON

This tab shows the code version of your storage configuration settings. Here, you can manually adjust the parameters seen in the **Definition** tab.



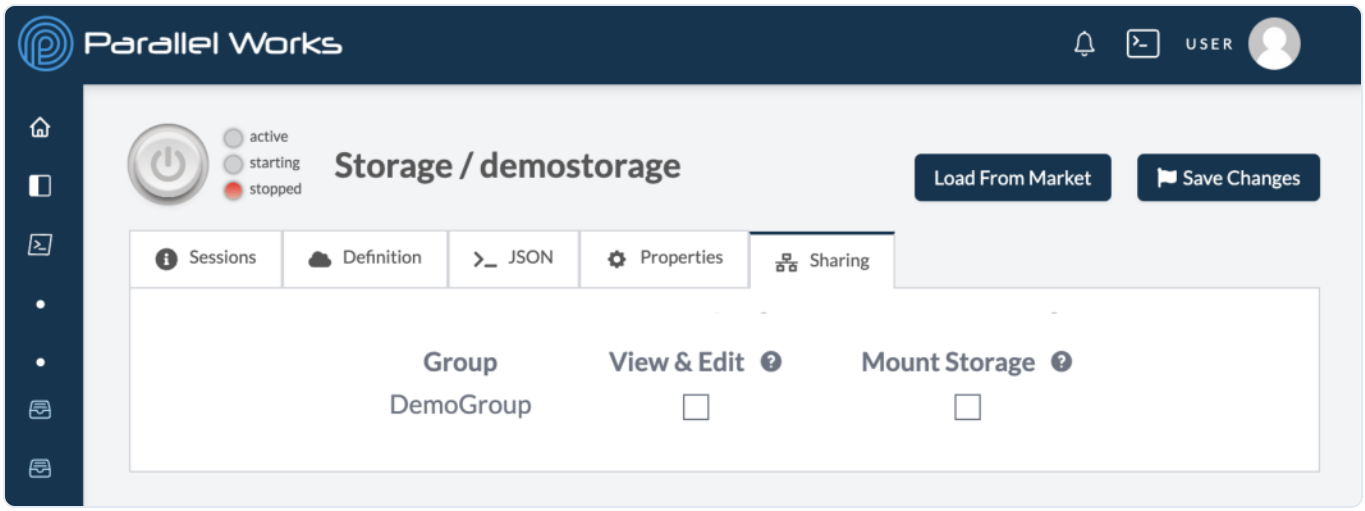
Properties

This tab shows the name that you entered when the storage was created. Here, you can change the name and upload a new thumbnail for the cluster.



Sharing

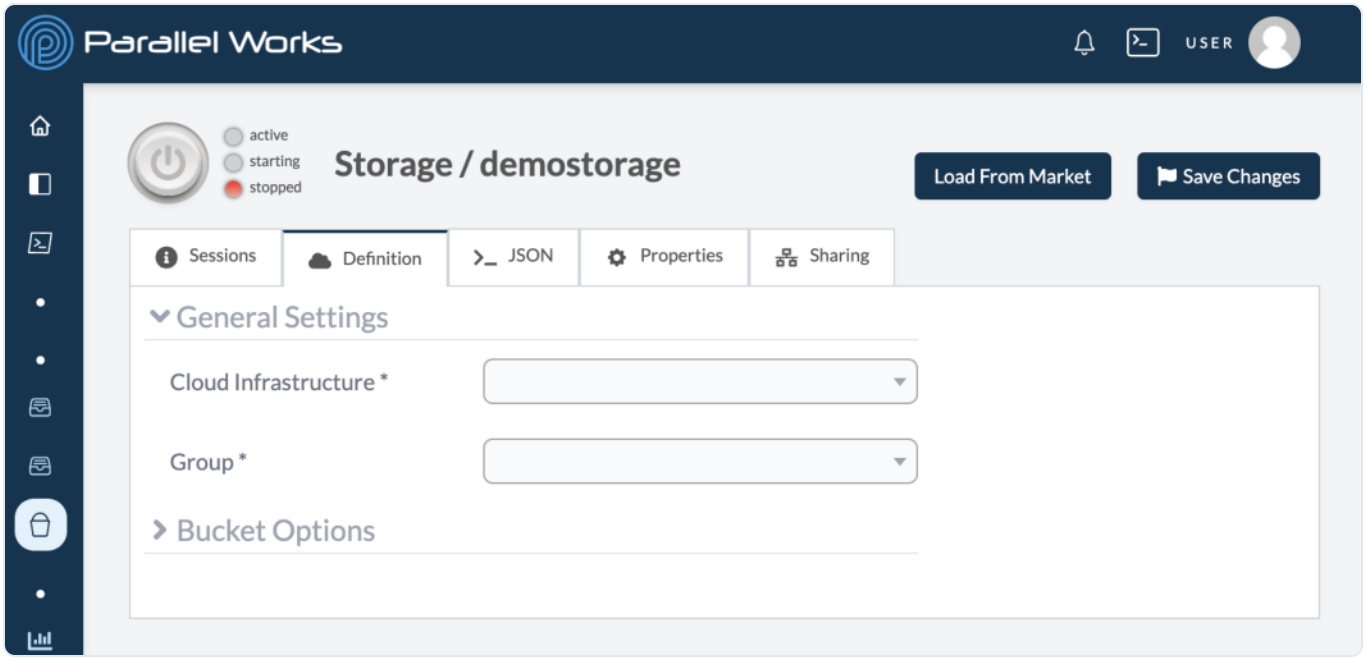
This tab lets you control storage sharing options for users in your organization who are assigned to the same group.



For more information, please see Sharing Storage.

General Settings

Storage resources have these settings in the **Definition** tab of the configuration page.



Cloud Infrastructure

Use this dropdown menu to choose your infrastructure, which determines which region your storage will be created in.

Group

Use this dropdown menu to select the group name that your organization uses to allocate costs. This menu is especially important if your organization uses multiple groups.

If you're not sure which group to select, you can contact us or your organization's ACTIVATE platform administrator.

> **STORAGE-SPECIFIC SETTINGS**
>
> For information about storage-specific settings, please select your storage type in the user guide sidebar.

Buckets

The configuration parameters on this page are exclusive to buckets.

Region

Use this dropdown menu to select the region that your storage will be deployed into. A region represents a geographic area.

Bucket Name*

**Currently available only on Azure buckets*

Use this field to name your storage. Please note that the bucket name must be globally unique, which means the name must not be currently in use on bucket's platform.

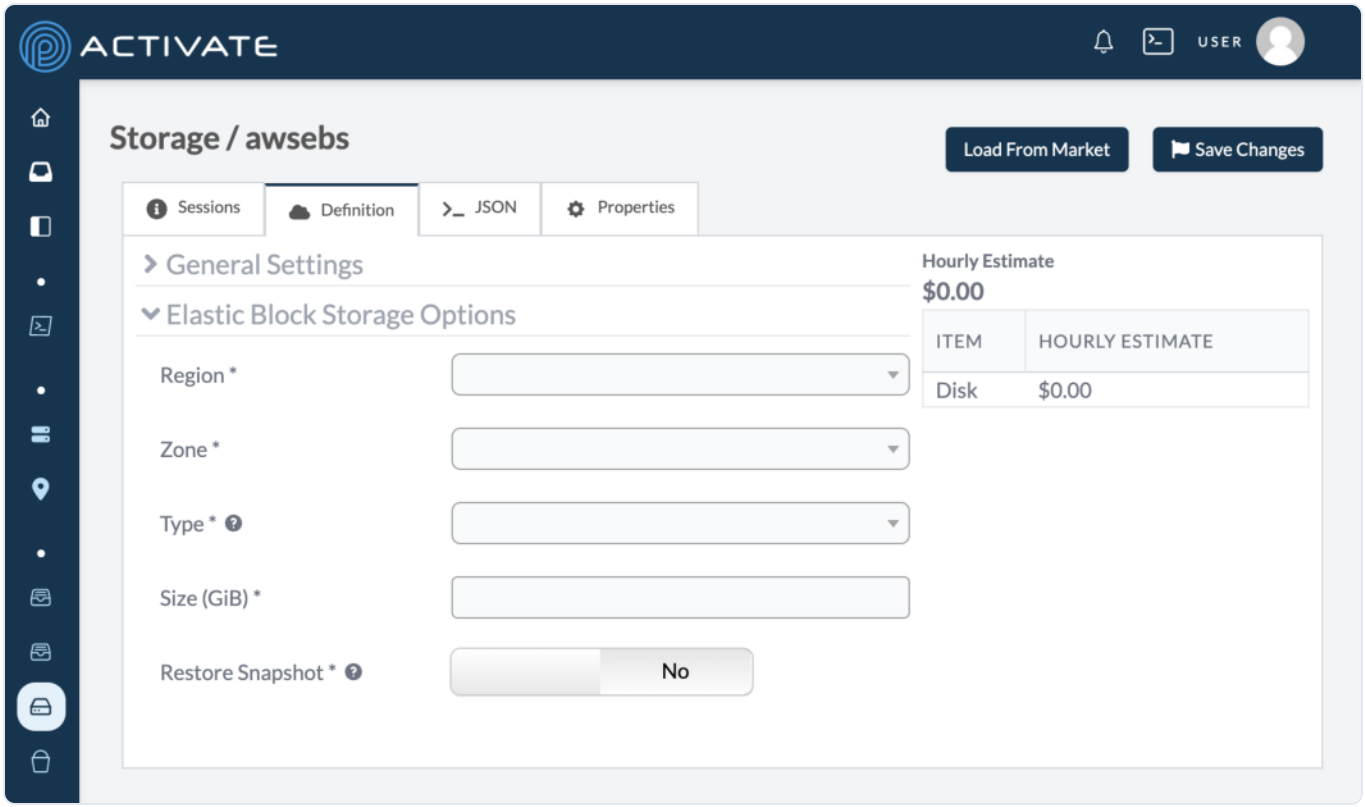
For more information about the naming conventions for bucket storage, please see the following pages:

- AWS S3
- Azure Blob Storage
- Google Cloud Bucket

Disks

AWS EBS Disk

The configuration parameters on this page are exclusive to AWS Elastic Block Store (EBS) Disks.



Region

Use this dropdown menu to select the region that your storage will be deployed into. A region represents a geographic area.

Zone

Use this dropdown menu to select the zone to use for your storage. A zone refers to an isolated location inside a region.

We recommend that your storage zone matches the zone of the cluster you're pairing the storage with.

Type

Use this dropdown menu to select the type of EBS storage.

Generally speaking:

- `gp2`, `gp3*`, `io1`, and `io2` are optimized for frequent read/write operations with small input/output size
- `sc1` and `st1` are optimized for large streaming workloads where the dominant performance attribute is throughput
- `standard` is suited for workloads with small datasets where data is accessed infrequently and performance is not of primary importance

*Please note that if you select `gp3`, you'll also need to specify its **Throughput**.

For more details about these types and their use cases, please see the AWS EBS documentation.

Size (GiB)

Use this field to enter the size of your storage. AWS measures disk capacity in both gibibytes (GiB) and tebibytes (TiB). A gibibyte is equivalent to 1,073,741,824 bytes, or 1.1 gigabytes (GB). A tebibyte is equivalent to 1,099,511,627,776 bytes, or 1.1 terabytes (TB).

For specifics on minimums and maximums for each disk type, please see this AWS page.

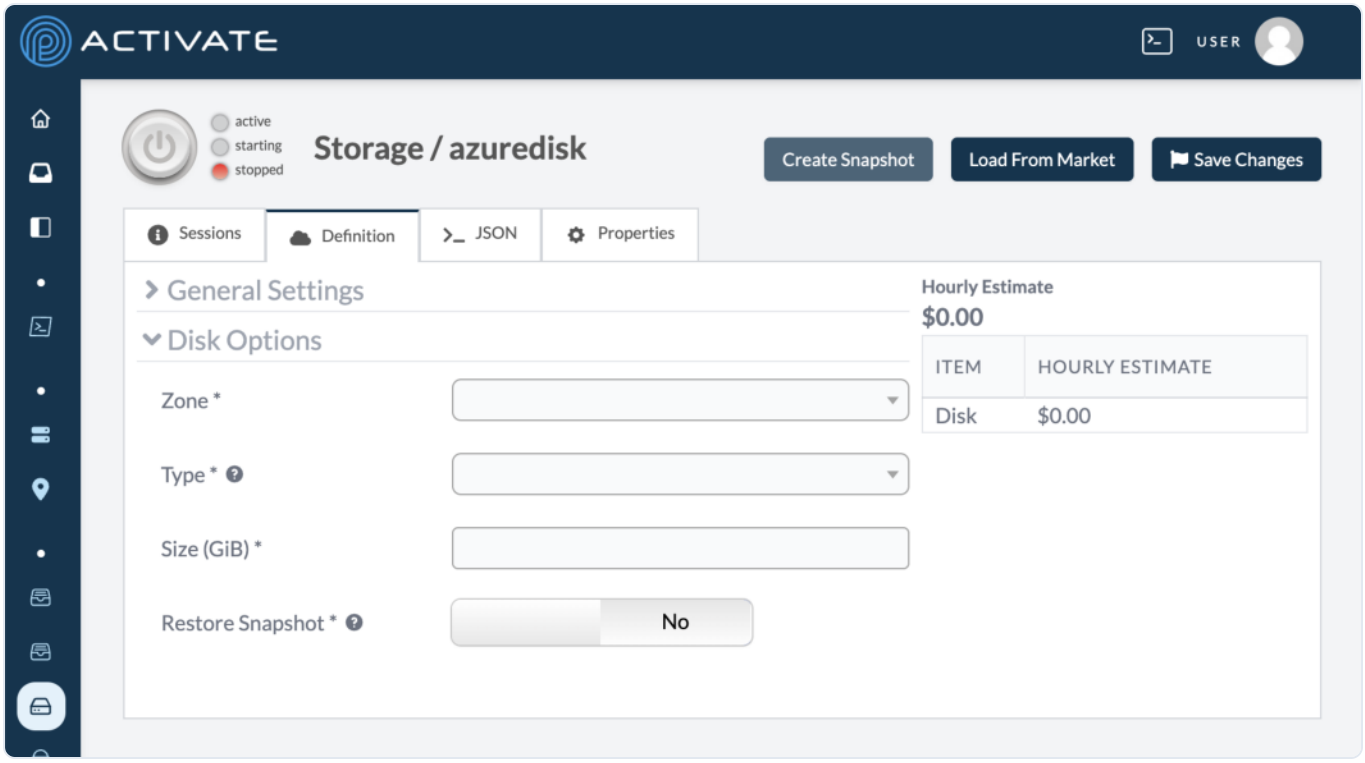
Restore Snapshot

Toggle this option to **Yes** if you want to restore a previously saved disk snapshot. A new dropdown menu, **Snapshot**, will appear so you can select your snapshot.

For more information about creating snapshots, please see **Working With Disk Snapshots**.

Azure Disk

The configuration parameters on this page are exclusive to Azure disks.



Zone

Use this dropdown menu to select the zone to use for your storage. A zone refers to an isolated location inside a region.

If you choose **Automatic**, the disk will be zone-agnostic (not assigned to any specific zone). A zone-agnostic Azure disk can only be attached to a zone-agnostic Azure cluster.

However, a zone-agnostic Azure cluster can attach both zone-agnostic disks and disks in any zone.

If both your Azure cluster and Azure disk have a zone, their zones must match.

Example scenarios:

- A zone-agnostic cluster can attach both zone 1 and zone-agnostic disks without issues.
- A cluster in zone 1 can attach a disk in zone 1 or a zone-agnostic disk.
- A cluster in zone 1 cannot attach a disk in zone 2.

Type

Use this dropdown menu to select the type of Azure disk.

Types for Azure disks include:

- `Standard_LRS`
- `StandardSSD_LRS`
- `StandardSSD_ZRS`
- `Premium_LRS`

- `Premium_ZRS`
- `PremiumV2_LRS`
- `UltraSSD_LRS`

In general, Azure disk types increase in both performance and expense according to the order listed above. You can read more about the specifics of each type in the Azure disk documentation.

Azure storage includes redundancy, meaning that Azure stores multiple copies of data to protect against data loss in the event of hardware failure, network outages, and natural disasters. For most storage types, you can choose the type of redundancy that best suits your work.

Locally redundant storage (LRS) copies your data synchronously three times within a single physical location in the primary region. Zone-redundant storage (ZRS) copies your data synchronously across three Azure availability zones in the primary region. LRS is the least expensive replication option, but isn't recommended for applications requiring high availability or durability.

Please see this Azure page to read more about their storage redundancy features.

Size (GiB)

Use this field to enter the size of your storage. Azure measures disk capacity in gibibytes (GiB); a gibibyte is equivalent to 1,073,741,824 bytes, or 1.1 gigabytes (GB).

For specific limits on each disk type, please see this Azure page.

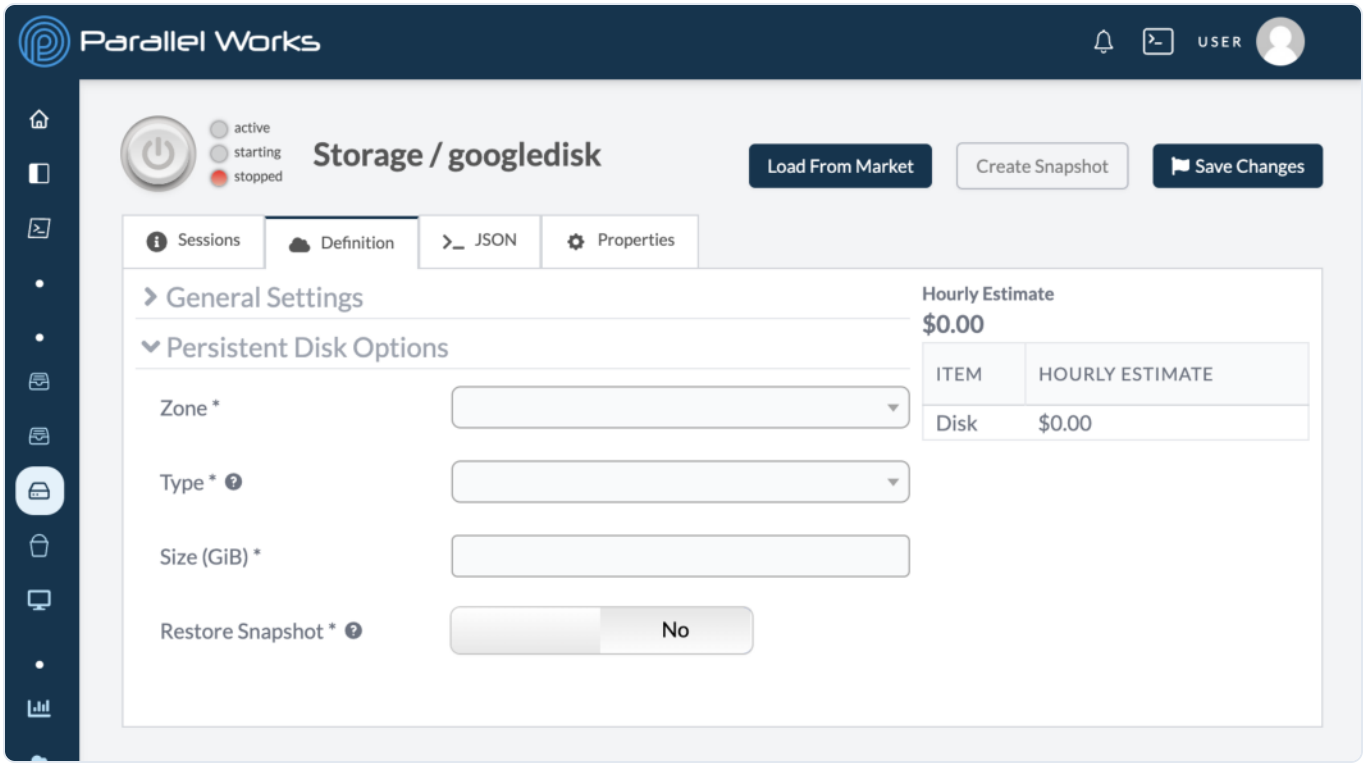
Restore Snapshot

Toggle this option to **Yes** if you want to restore a previously saved disk snapshot. A new dropdown menu, **Snapshot**, will appear so you can select your snapshot.

For more information, please see **Working With Disk Snapshots**.

Google Persistent Disk

The configuration parameters on this page are exclusive to Google Persistent Disks.



Zone

Use this dropdown menu to select the zone to use for your storage. A zone refers to an isolated location inside a region.

We recommend that your storage zone matches the zone of the cluster you're pairing the storage with.

Type

Use this dropdown menu to select the type. Currently, we offer `pd-ssd`, which is designed for single-digit millisecond latencies.

For more details about `pd-sdd`, please see the Google persistent disk documentation.

Size (GiB)

Use this field to enter the size of your storage in gibibytes.

The minimum capacity for `pd-sdd` is 10 GiB. For more information about disk capacity, please see this Google Cloud page.

Restore Snapshot

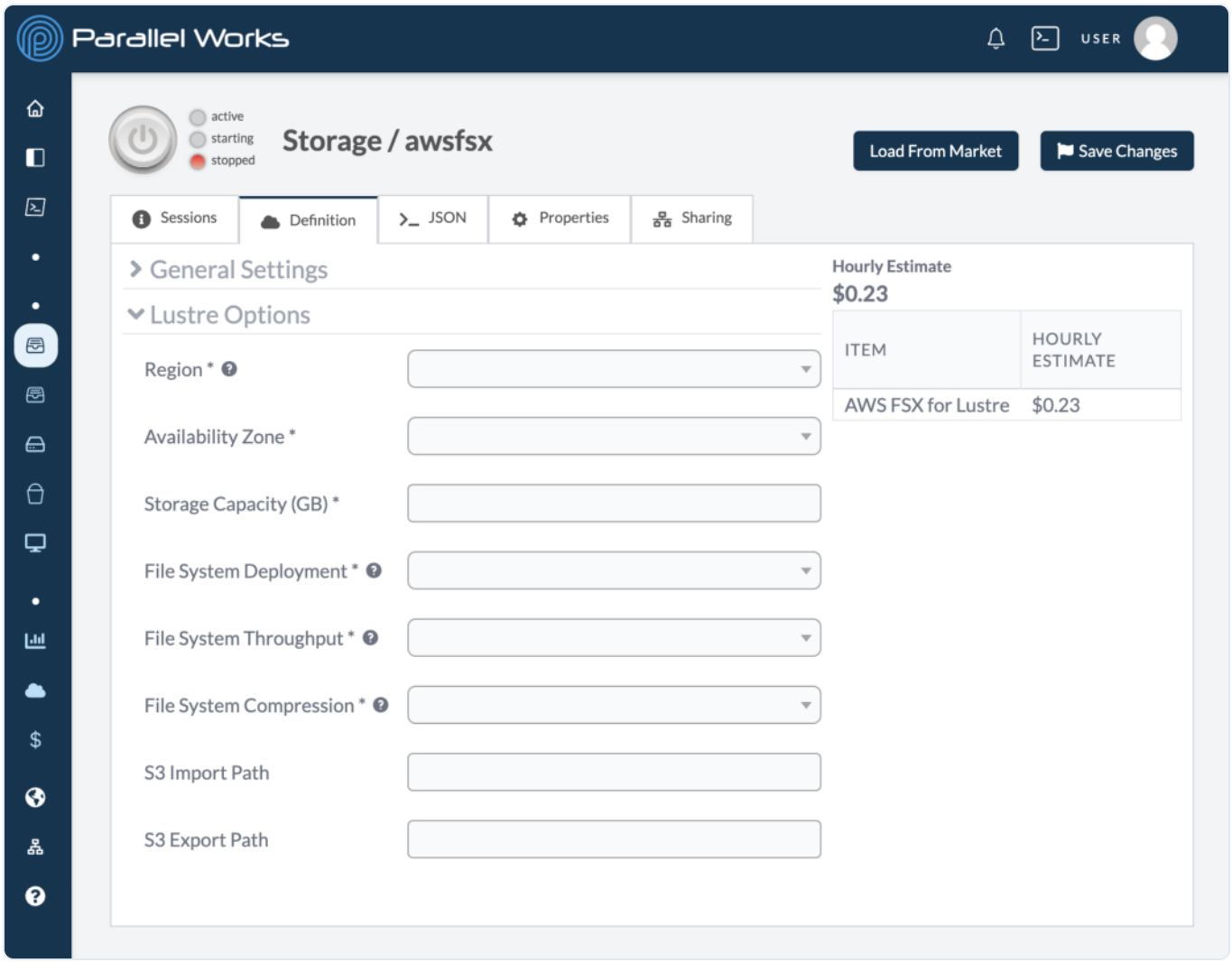
Toggle this option to **Yes** if you want to restore a previously saved disk snapshot. A new dropdown menu, **Snapshot**, will appear so you can select your snapshot.

For more information, please see **Working With Disk Snapshots**.

Lustre

AWS FSx for Lustre

The configuration parameters on this page are exclusive to AWS FSx for Lustre.



Region

This field identifies which region the storage will be deployed in. Please note that region is determined by **Cloud Infrastructure**. As a result, the **Region** parameter may be locked if your infrastructure is only configured for one region.

Availability Zone

Use this dropdown menu to select the Availability Zone that your infrastructure will be deployed in. An Availability Zone refers to an isolated location inside a region.

We recommend that your storage Availability Zone matches the Availability Zone of the cluster you're pairing the storage with.

Storage Capacity (GiB)

Use this field to enter the total capacity of your storage in gibibytes (GiB).

This field is set to `1200` by default. `1200` is the minimum amount of storage and must be increased in increments of `2400`.

Please note that altering the storage size will affect your estimated hourly cost.

File System Deployment

Use this dropdown menu to select which type of file system your storage will be deployed as. Currently, the options are `SCRATCH_1`, `SCRATCH_2`, `PERSISTENT_1`, and `PERSISTENT_2`.

Scratch filesystems are meant for shorter-term workloads. They provide a higher throughput per TiB of storage capacity. If a scratch filesystem fails, your data is not replicated.

Persistent filesystems are meant for long-term workloads. If this filesystem fails, your data will automatically replicate in the same Availability Zone.

File System Throughput*

**This parameter is only available for `persistent` deployments.*

Use this dropdown menu to select the throughput of your storage. Throughput is measured in number of megabytes (MB) per second, per tebibyte (TiB) of storage.

In general, as the throuphput number increases, the cost will decrease.

File System Compression

Use this dropdown menu to select whether your filesystem will be compressed. Currently, the options are `LZ4` and `NONE`.

`LZ4` compresses data when writing to Lustre. When reading from Lustre, `LZ4` decompresses data while minimally impacting performance. Using `LZ4` can improve read and write performance as well as reduce storage capacity.

S3 Import Path (Optional)

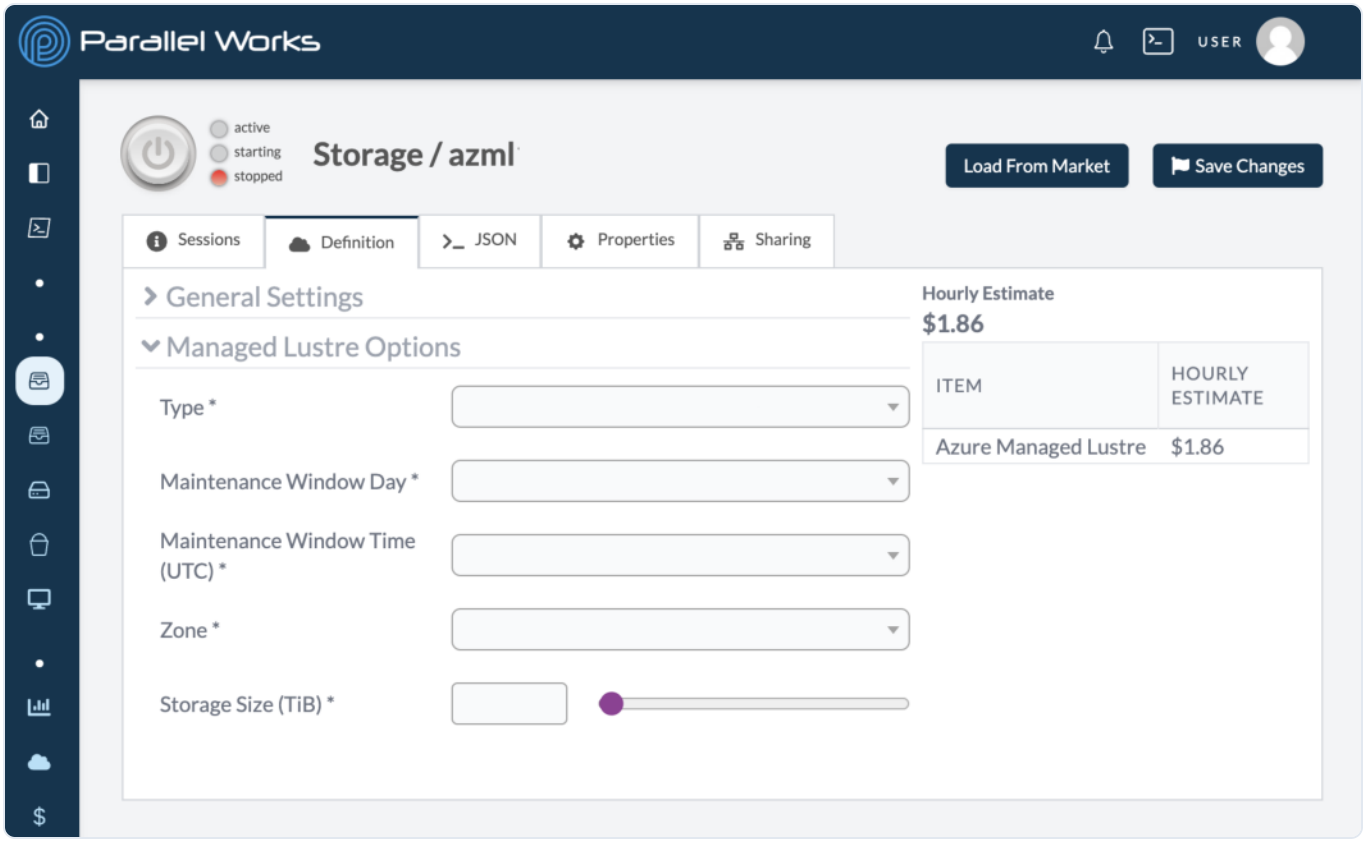
Use this field to enter an S3 bucket path to import data from. The format for this field is `s3://BucketName`.

S3 Export Path (Optional)

Use this field to enter an S3 bucket path to export data to. The format for this field is `s3://BucketName`.

Azure Managed Lustre

The configuration parameters on this page are exclusive to Azure Managed Lustre storage.



Type

Use this dropdown menu to select the configuration type for your storage. The numbers on each type represent data transfer speeds (throughput) in number of megabytes (MB) per second, per tebibyte (TiB) of storage.

In general, as the throughput number increases, the storage minimum and maximum as well as the cost will decrease.

For specific information about these numbers, please see the chart in the Azure documentation.

Maintenance Window Day

Use this dropdown menu to select a day when Azure can perform routine maintenance on your filesystem.

According to the Azure documentation, maintenance is typically performed less than once a month.

Maintenance Window Time (UTC)

Use this dropdown menu to select the time of day for your maintenance window. The times listed are in Coordinated Univeral Time (UTC) and 24-hour format (HH:MM).

Zone

Use this dropdown menu to select the zone to use for your storage. A zone refers to an isolated location inside a region. The region for Azure Managed Lustre is determined by the **Cloud Infrastructure** you select.

We recommend that your storage zone matches the zone of the cluster you're pairing the storage with.

Storage Size (TiB)

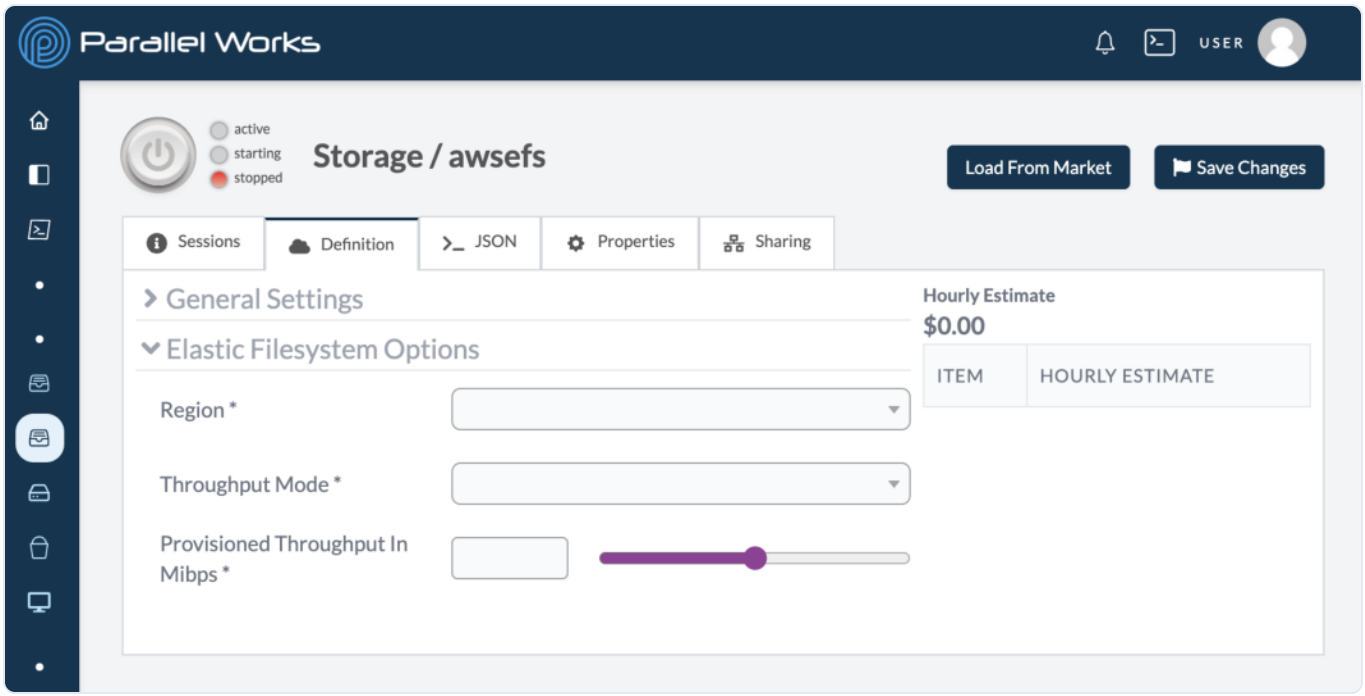
Use this field and slider to select the size of your storage in tebibytes (TiB).

Please note that the **Type** you select affects the increments by which you can increase your storage size. Specific increments for each **Type** are listed in the Azure documentation.

NFS

AWS Elastic File System

The configuration parameters on this page are exclusive to AWS Elastic File System (EFS).



Throughput Mode

Use this dropdown menu to select how much throughput will be available to your file system. AWS offers three throughput modes: `bursting`, `elastic`, and `provisioned`.

`Bursting` throughput scales with the amount of storage in your file system.

`Elastic` throughput (AWS recommended) is useful for spiky/unpredictable workloads, if your performance requirements are difficult to forecast, or when your application drives throughput at an average-to-peak ratio of 5% or less.

`Provisioned` throughput is useful when you know your workload's performance requirements, or when your application drives throughput at an average-to-peak ratio of 5% or more.

For more information, please see the AWS documentation on throughput modes.

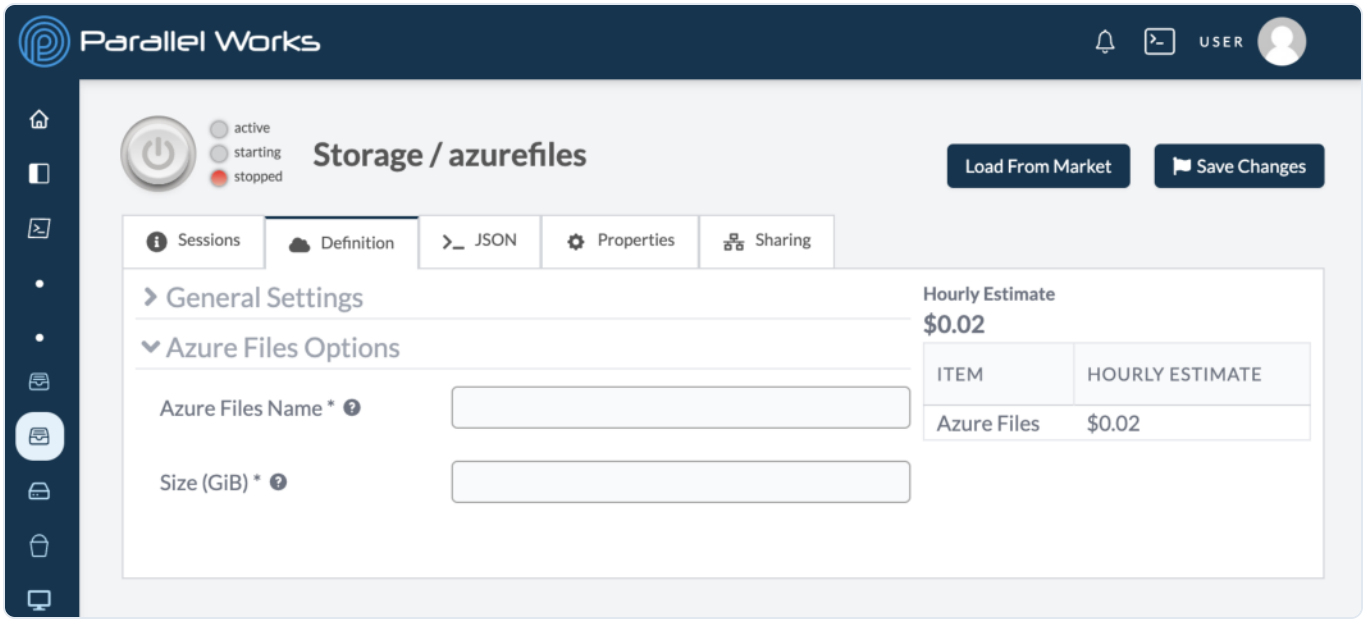
Provisioned Throughput *

This parameter will only appear if you select `provisioned` for your **Throughput Mode.*

Use this field and slider to choose the speed of your provisioned throughput in mebibytes per second (MiB/s).

Azure Files

The configuration parameters on this page are exclusive to Azure Files.



Filesystem Name

Use this field to name your storage. Please note that the filesystem name must be globally unique (that is, it has never been used on the Azure platform).

For more specific information about the naming conventions for Azure Files, please see this page of the Azure documentation.

Size

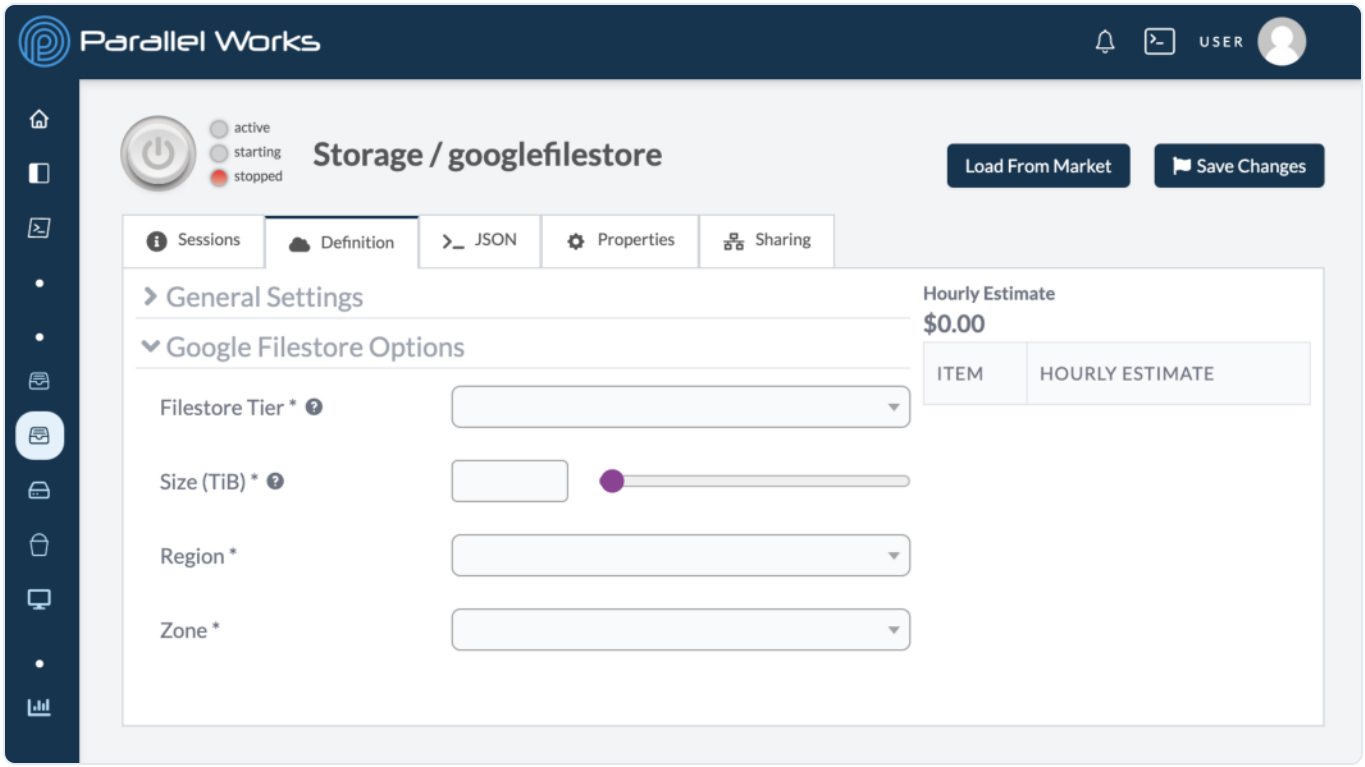
Use this field to enter the size of your storage in gibibytes (GiB). A gibibyte is equivalent to 1,073,741,824 bytes, or 1.1 gigabytes (GB).

The minimum size for Azure Files is 100 GiB. The maximum size is 10,240 GiB.

Google Filestore

The configuration parameters on this page are exclusive to Google Filestore.

Please note that Google Filestore measures storage in tebibytes (TiB); a tebibyte is equivalent to 1,099,511,627,776 bytes, or 1.1 teraybtes (TB).



Filestore Tier

Use this dropdown menu to select which service tier to use.

The tier `BASIC_HDD` supports 1 TiB to 10 TiB.

The tier `BASIC_SSD` supports 2.5 TiB to 10 TiB.

For more information, see the Google documentation on Filestore tiers.

Size

Use this field and slider to choose the size of your storage in TiB. The size must be changed in increments of 0.25.

Region

Use this dropdown menu to select the region to use for your storage.

Zone

Use this dropdown menu to select the zone to use for your storage. A zone refers to an isolated location inside a region.

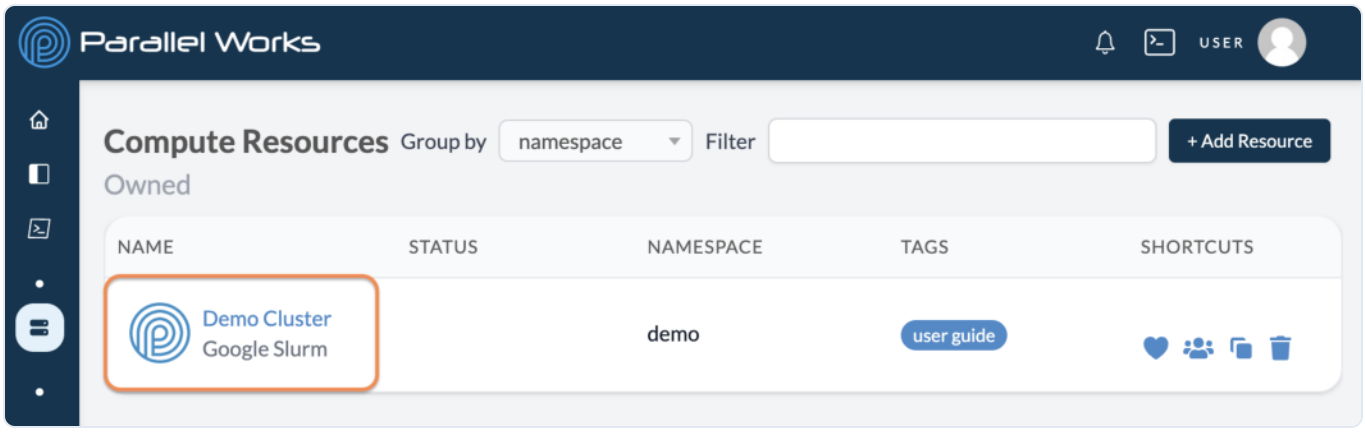
We recommend that your storage zone matches the zone of the cluster you're pairing the storage with.

Attaching Storage

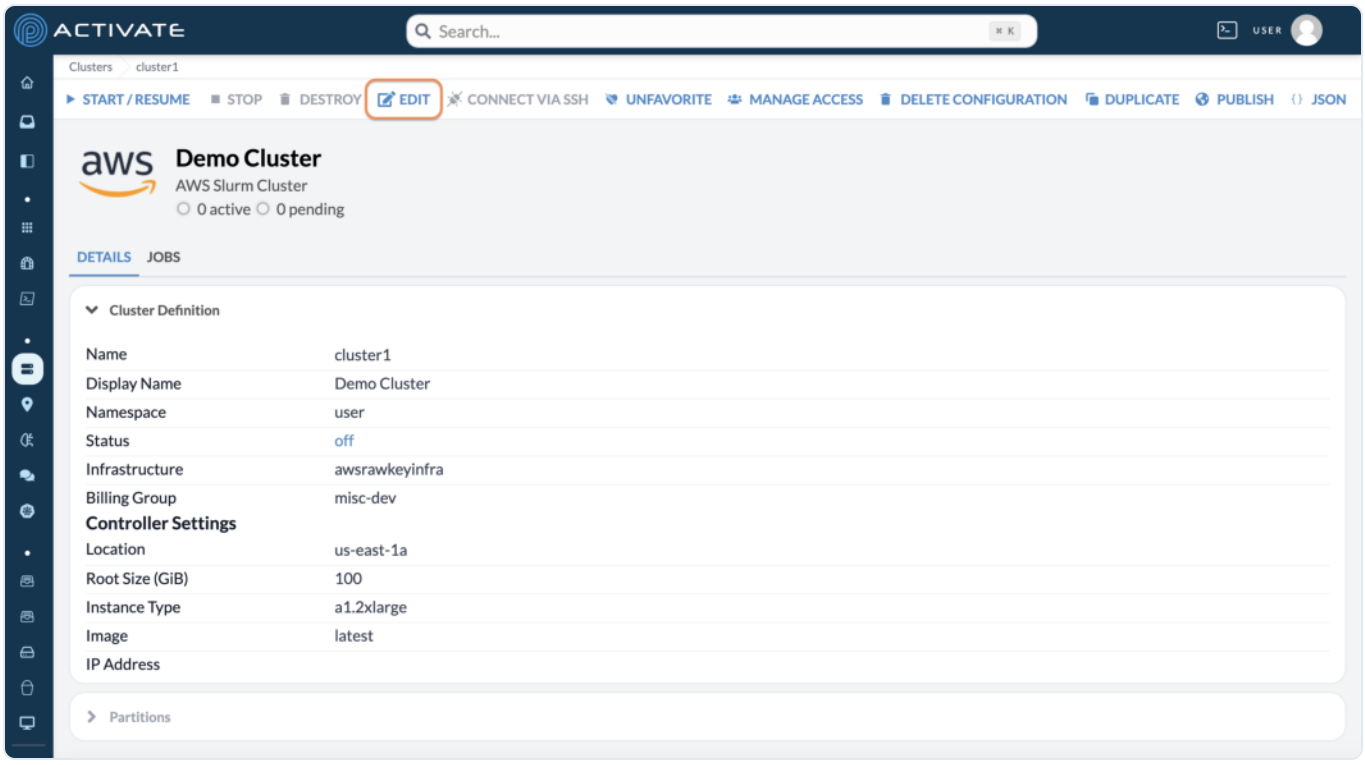
Attaching Storage

After you've created and configured a storage resource, you can attach it to one of your clusters.

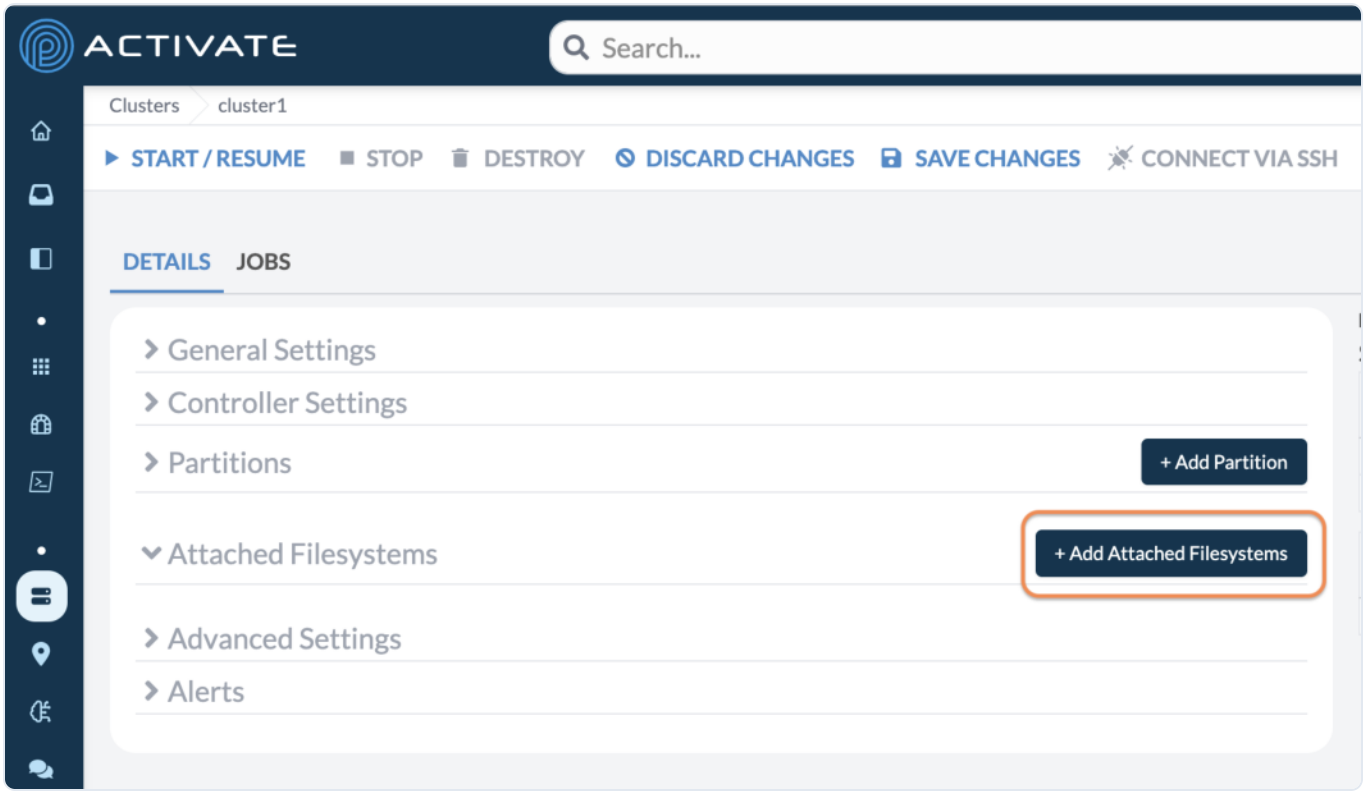
Navigate to the **Clusters** page. Click the cluster you want to work with.



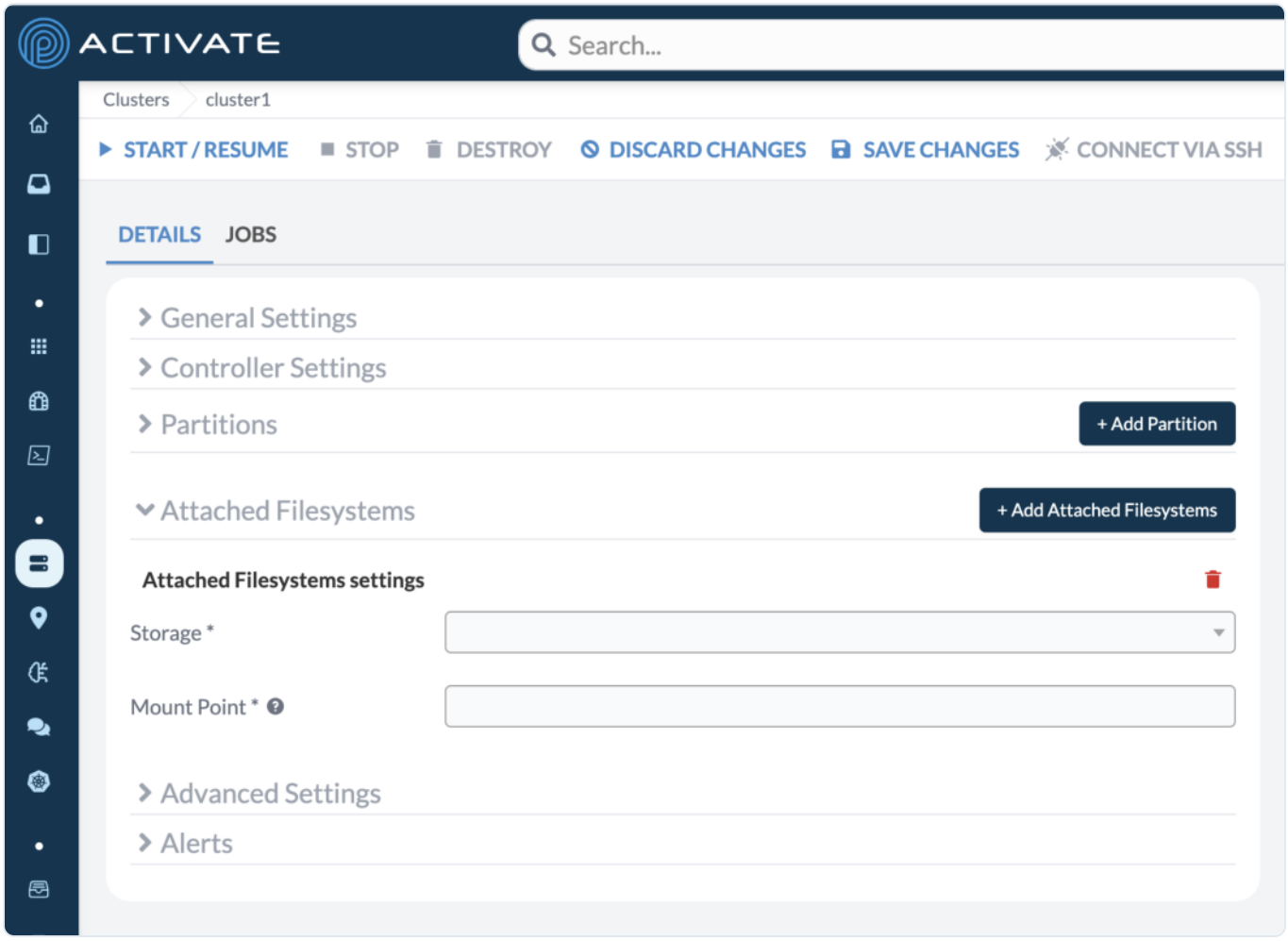
Click **Edit**.



In the **Attached Filesystems** section at the bottom of the page, click **+ Add Attached Filesystems**.



Two new parameters will appear: **Storage** and **Mount Point**.



Use the **Storage** dropdown menu to select which storage resource to attach to this cluster. Please note that only storage resources that match your cluster's CSP will be shown (for example, an AWS cluster will only show AWS FSx for Lustre or AWS Elastic Filesystem).

Use the **Mount Point** field to enter the directory where the storage will be mounted in the cluster's controller and compute nodes. Please note that the mount point should be an absolute path.

> **NOTE**
>
> You can add or remove persistent storage resources while a cluster is running. You cannot do so with ephemeral storages.

Verifying Storage Mounts

After you've logged in to your cluster, you can run the following command to verify that the storage is mounted correctly:

```
df -h
```

After you enter the command, you should see the storage mounted at the **Mount Point** you defined.

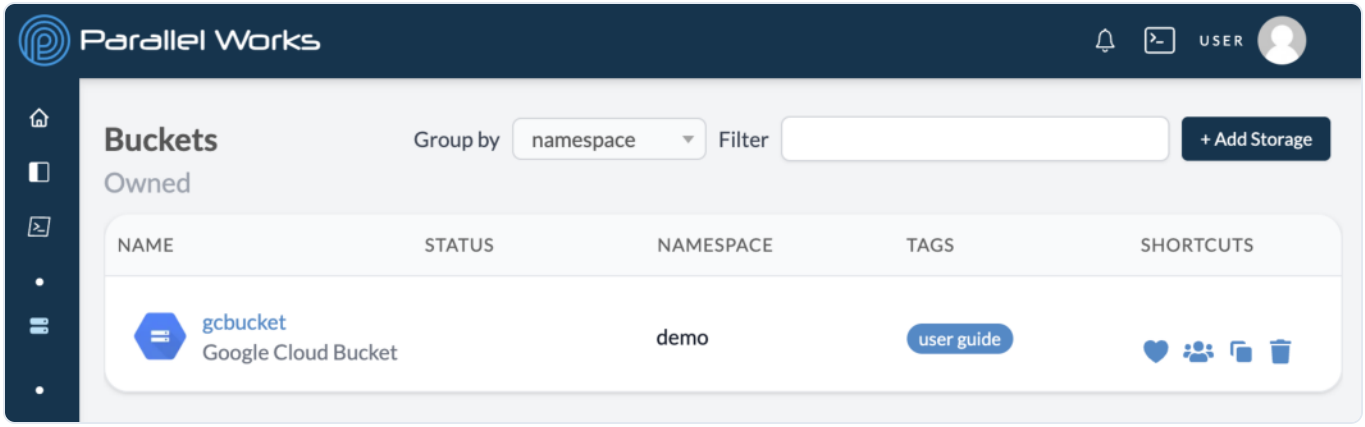
Mounts are updated every 45 seconds. If you very recently attached a storage resource to a cluster, there may be a short delay before you see your storage in the mount list.

Sharing Storage

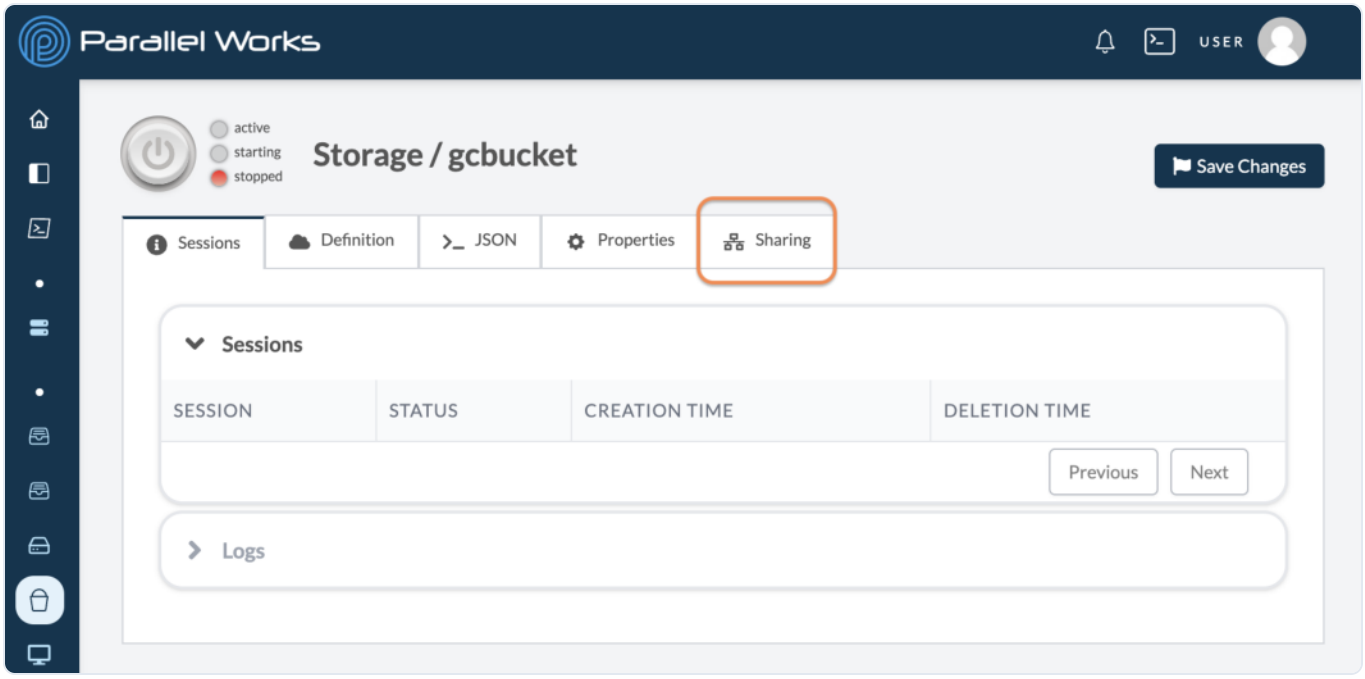
Sharing Storage

After you've created a storage resource, you can share it with other users in your organization.

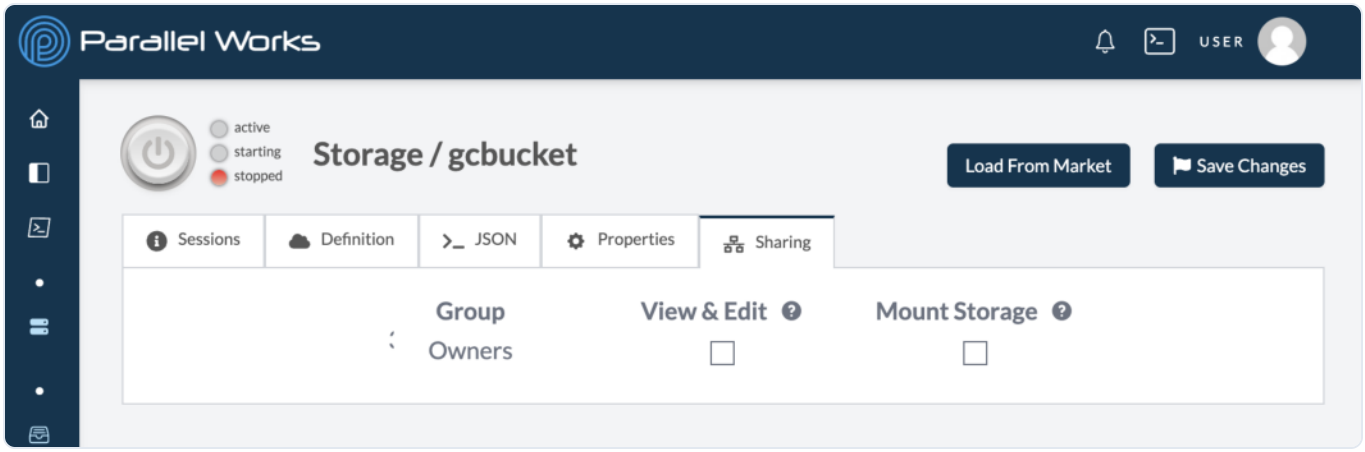
Click on your storage resource.



Click the **Sharing** tab.



You can manage the storage sharing options here.



If you check the **View & Edit** box, users in the selected group will be able to view and edit the storage's configuration. If it's a persistent storage resource, they'll also be able to start and stop the storage. Other users won't be able to change the storage's settings for **Cloud Infrastructure** or **Group**.

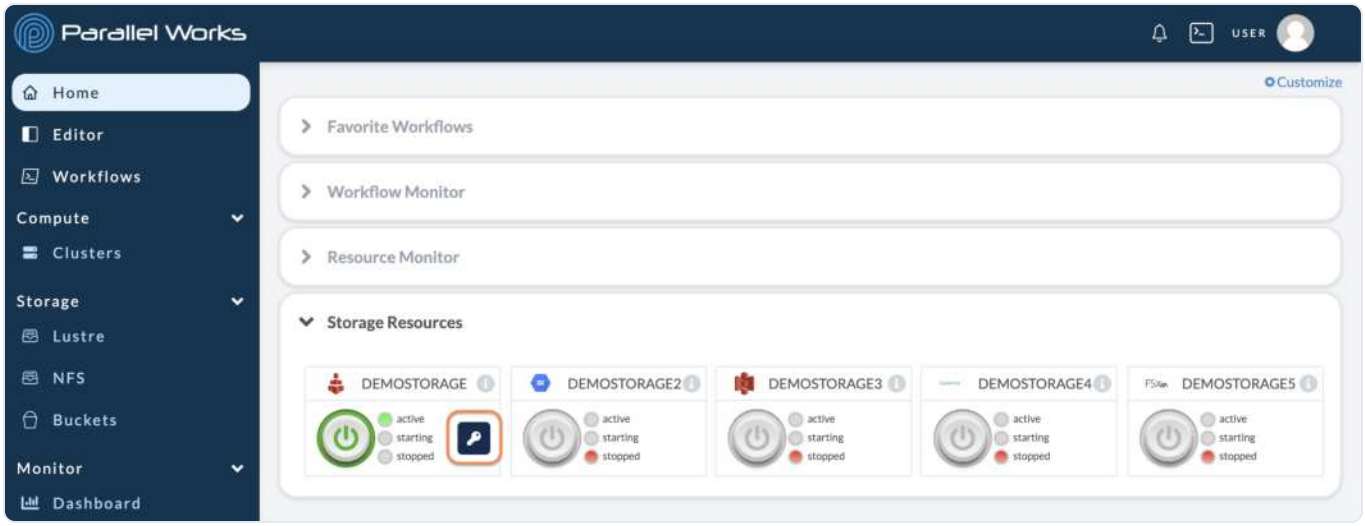
If you check the **Mount Storage** box, users in the selected group will be able to mount the storage to their cluster, but they won't be able to view or edit the storage's configuration. Users will be able to see whether the storage is currently running on the **Home** page, but they won't be able to start or stop it.

Transferring Data

Obtaining Credentials

You can use temporary credentials for bucket storage resources to make data transfers more secure.

Navigate to your **Home** page and start your storage resource. Once it's active, click the key icon.



A module will appear, where you can copy your storage resource's credentials and commands for the corresponding CLI.

AWS

```
Credentials for AWS buckets expire after 12 hours.
<img src="/content-images/docs/storage/transferring-data/obtaining-credentials/aws-bucket-
credentials.png" alt="Screenshot of AWS bucket credentials."/>
```

Azure

```
Credentials for Azure Storage Accounts expire after 12 hours.
<img src="/content-images/docs/storage/transferring-data/obtaining-credentials/azure-bucket-
credentials.png" alt="Screenshot of Azure bucket credentials."/>
```

Google

```
Credentials for Google Cloud Storage buckets expire after 1 hour.
<img src="/content-images/docs/storage/transferring-data/obtaining-credentials/google-bucket-
credentials.png" alt="Screenshot of Google bucket credentials."/>
```

Transferring Data with AWS S3

This page explains how to transfer data to/from your AWS S3 buckets with a terminal. You can use the methods on this page for all S3 buckets, whether you created them on the ACTIVATE platform or outside the platform.

To transfer data to/from S3 bucket storage, you'll use the AWS Command-Line Interface (CLI).

The AWS CLI is pre-installed on cloud clusters provisioned by ACTIVATE, so you can enter commands directly into the IDE after logging in to the controller of an active AWS cluster.

If you're transferring data between S3 buckets and your local machine or an on-premises cluster, you'll likely need to install the AWS CLI first.

Check for AWS CLI

Open a terminal or command line and navigate to your data's destination. Enter `which aws`.

If the AWS CLI is installed, you'll see a message that shows its location, such as `/usr/local/bin/aws`.

If the AWS CLI is not installed, you'll see a message such as `/usr/bin/which: no aws` or `aws not found`.

Install AWS CLI

If you need to install the AWS CLI, we recommend following the AWS installation guide, which includes OS-specific instructions for Linux, macOS, and Windows as well as troubleshooting tips.

Export Your AWS Credentials

You can see our page **Obtaining Credentials** for information on finding your AWS credentials.

In your terminal, enter `export AWS_ACCESS_KEY_ID="_____"` with your AWS access key ID in the blank space.

Enter `export AWS_SECRET_ACCESS_KEY="_____"` with your AWS access key ID in the blank space.

> **NOTE**
>
> Please be sure to include the quotation marks on both ends of your key IDs. There are characters inside AWS key IDs that, without quotation marks, systems will try to read as commands.

List Files in S3

In your terminal, enter `aws s3 ls s3://bucket_name` to display the files in your bucket. For this guide, we used a small text file named `test.txt`, so our command returned this message:

```
demo@pw-user-demo:~/pw/cloud-data-transfer$ aws s3 ls s3://cloud-data-test
2023-02-08 16:37:56         28 test.txt
```

Transfer a File To/From S3

The AWS CLI mimics the Linux `cp` command for transferring files. To transfer a file, enter `aws s3 SOURCE DESTINATION` in your terminal.

`SOURCE` and `DESTINATION` can be local files or S3 buckets. S3 buckets are formatted as `s3://bucket_name/file_name`.

Below is an example of the `aws s3` command.

In your terminal, enter `aws s3 cp s3://bucket_name/file_name ./` to download a file from S3 storage to your current directory. You'll see this message:

```
demo@pw-user-demo:~/pw/cloud-data-transfer$ aws s3 cp s3://cloud-data-test/test.txt ./
download: s3://cloud-data-test/test.txt to ./test.txt
```

To download a file from S3 storage to a specific directory, enter its absolute or relative path (e.g., `/home/username/` or `./dir_relative_to_current_dir`) in place of `./` with the `aws s3 cp` command. In the example below, we used the `storage` directory, which is located inside the current working directory:

```
demo@pw-user-demo:~/pw/cloud-data-transfer$ aws s3 cp s3://cloud-data-test/test.txt ./storage
download: s3://cloud-data-test/test.txt to ./storage
```

To upload, simply reverse the order of `SOURCE` and `DESTINATION` in the `aws s3` command.

Delete a File From S3

In your terminal, enter `aws s3 rm s3://bucket_name/file_name` to delete a file. You'll see this message:

```
demo@pw-user-demo:~/pw/cloud-data-transfer$ aws s3 rm s3://cloud-data-test/test_upload.txt
delete: s3://cloud-data-test/test_upload.txt
```

Further Reading

For a comprehensive guide to installing the AWS CLI in different environments, please see this page of the AWS documentation.

To configure your AWS CLI settings, please see this page.

For the full list of AWS CLI transfer commands, please see this reference guide.

Transferring Data with Azure Storage Accounts

This page explains how to transfer data to/from your Azure storage account with a terminal. You can use the methods on this page for all Azure storage accounts, whether you created them on the ACTIVATE platform or outside the platform.

Working With azcopy

To transfer data to/from an Azure storage account, you'll use Azure's command-line tool, azcopy.

Azcopy is pre-installed on cloud clusters provisioned by ACTIVATE, so you can enter commands directly into the IDE after logging in to the controller of an active Azure cluster.

If you're transferring data between Azure storage and your local machine or an on-premises cluster, you will need to install azcopy first.

Check for azcopy

Open a terminal and enter `which azcopy`.

If azcopy is installed, you'll see a message that shows its location, such as `/usr/local/bin/azcopy`. Otherwise, you'll see a message such as `/usr/bin/which: no azcopy` or `azcopy not found`.

Install azcopy

If you need to install azcopy, we recommend following the Azure installation guide, which includes OS-specific instructions for Linux, macOS, and Windows as well as troubleshooting tips.

Export Your Azure Credentials

You can see our page **Obtaining Credentials** for information on finding your Azure credentials.

In your terminal, enter the following commands with your own Azure credentials substituted:

```
azcopy login –service–principal –application–id $AZURE_CLIENT_ID –tenant–id $AZURE_TENANT_ID
```

List Files in an Azure Storage Account

In your terminal, enter `azcopy ls $AZCOPY_BUCKET_URL` to display the files in your bucket. For this guide, we used a small text file named `test.txt`, so our command returned this message:

```
demo@pw–user–demo:~/pw/cloud–data–transfer$ azcopy ls $AZCOPY_BUCKET_URL
2023–02–08 16:37:56         28 test.txt
```

Transfer a File To/From an Azure Storage Account

The `azcopy` command mimics the Linux `cp` command for transferring files. To transfer a file, enter `azcopy cp SOURCE DESTINATION` in your terminal.

Below is an example of the `azcopy cp` command.

In your terminal, enter `azcopy cp $AZCOPY_BUCKET_URL/file/in/bucket.txt fileName` to download a file from Blob storage to your current directory. You'll see this message:

```
demo@pw-user-demo:~/pw/cloud-data-transfer$ azcopy cp $AZCOPY_BUCKET_URL/test.txt
test_download.txt
download: https://test-storage.blob.core.windows.net/test-bucket/test.txt to test_download.txt
```

To upload a file to your Azure storage account, simply reverse the order of `SOURCE` and `DESTINATION` in the `azcopy cp` command.

Delete a File From an Azure Storage Account

In your terminal, enter `azcopy rm $AZCOPY_BUCKET_URL/file/in/bucket.txt` to delete a file. You'll see this message:

```
demo@pw-user-demo:~/pw/cloud-data-transfer$ azcopy rm $AZCOPY_BUCKET_URL/test_upload.txt
delete: https://test-storage.blob.core.windows.net/test-bucket/test_upload.txt
```

Further Reading

For the full list of azcopy commands, please see this reference guide.

Transferring Data with Google Cloud Storage Buckets

This page explains how to transfer data to/from your Google Cloud Storage (GCS) Buckets with a terminal. You can use the methods on this page for all GCS Buckets, whether you created them on the ACTIVATE platform or outside the platform.

To transfer data to/from GCS Bucket storage, you'll use the Google Cloud Command-Line Interface (CLI), gcloud.

Gcloud is pre-installed on cloud clusters provisioned by ACTIVATE, so you can enter commands directly into the IDE after logging in to the controller of an active Google cluster.

If you're transferring data between GCS Buckets and your local machine or an on-premises cluster, you'll likely need to install gcloud first.

Check for gcloud

Open a terminal and navigate to your data's destination. Enter `which gcloud`.

If gcloud is installed, you'll see a message that shows its location, such as `/usr/local/bin/gcloud`. Otherwise, you'll see a message such as `/usr/bin/which: no gcloud` or `gcloud not found`.

Install gcloud

To install gcloud, we recommend following the Google installation guide, which includes OS-specific instructions for Linux, macOS, and Windows as well as troubleshooting tips.

> **ABOUT `GSUTIL`**
>
> Google refers to `gsutil` commands as a legacy feature that is minimally maintained; instead, they recommend using `gcloud` commands. For this reason, we've used `gcloud` in this guide. Please see this page for Google's `gsutil` guide.

Export Your Google Credentials

You can see our page **Obtaining Credentials** for information on finding your Google credentials.

In your terminal, enter `export BUCKET_NAME=gs://` with your Bucket's name after the backslashes.

Next, enter `export CLOUDSDK_AUTH_ACCESS_TOKEN='_____'` with your Google access token in the blank space.

> **NOTE**
>
> Please be sure to include the quotes on both ends of your access token. There are characters inside Google tokens that, without quotation marks, systems will try to read as commands.

List Files in a GCS Bucket

In your terminal, enter `gcloud storage ls gs://$BUCKET_NAME` to display the files in your Bucket. For this guide, we used a small text file named `test.txt`, so our command returned this message:

```
demo@pw-user-demo:~/pw$ gcloud storage ls gs://$BUCKET_NAME
gs://pw-bucket/test.txt/
```

If your Bucket is empty, this `gcloud storage ls` command will not print anything.

Transfer a File To/From a GCS Bucket

gcloud mimics the Linux `cp` command for transferring files. To transfer a file, enter `gcloud storage cp SOURCE DESTINATION` in your terminal.

Below is an example of the `gcloud storage cp` command:

In your terminal, enter `gcloud storage cp gs://$BUCKET_NAME/file/in/bucket.txt fileName.txt` to copy a remote file to your current directory. You'll see this message:

```
demo@pw-user-demo:~/pw$ gcloud storage cp gs://$BUCKET_NAME/file/in/bucket.txt new-test.txt
Copying gs://pw-bucket/file/in/bucket.txt to file://new-test.txt
  Completed files 1/1 | 11.0B/11.0B
```

To download a file from GCS storage to a specific directory, enter its absolute or relative path (e.g., `/home/username/` or `./dir_relative_to_current_dir`) in place of ./ with the gcloud storage cp command.

To upload, simply reverse the order of `SOURCE` and `DESTINATION` in the `gcloud storage cp` command.

Delete a File From a GCS Bucket

In your terminal, enter `gcloud storage rm gs://$BUCKET_NAME/file_name` to delete a file. You'll see this message:

```
demo@pw-user-demo:~/pw$ gcloud storage rm  gs://$BUCKET_NAME/file/in/bucket.txt
Removing objects:
Removing gs://65441d55134e2d74c85726e4/file/in/bucket.txt...
  Completed 1/1
```

Further Reading

For a gcloud cheat sheet, see this page.

For the full list of gcloud `storage` parameters, please see this reference guide.

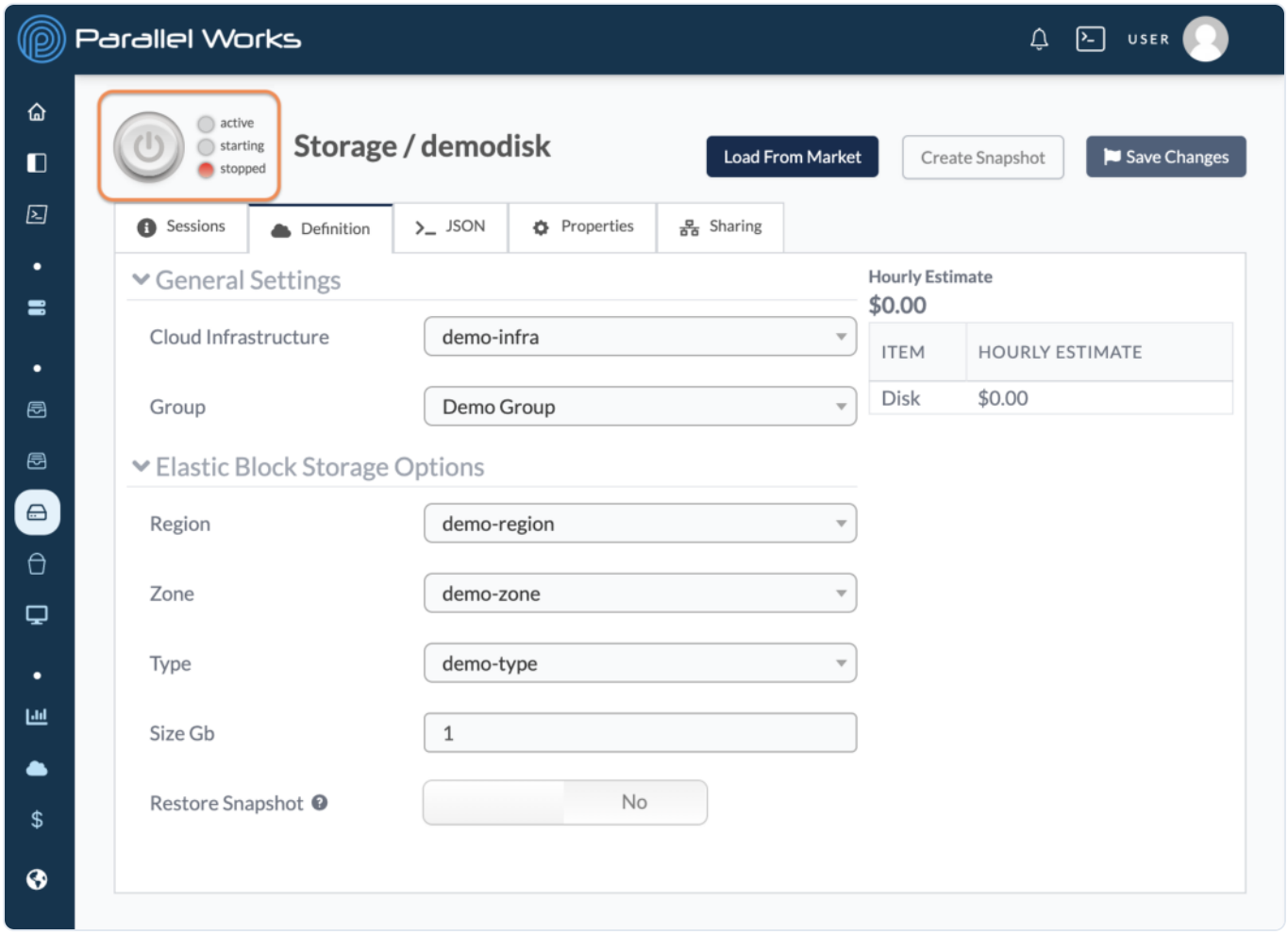
Working With Disk Snapshots

If you want to save a disk's configuration, you can create a snapshot to re-use later.

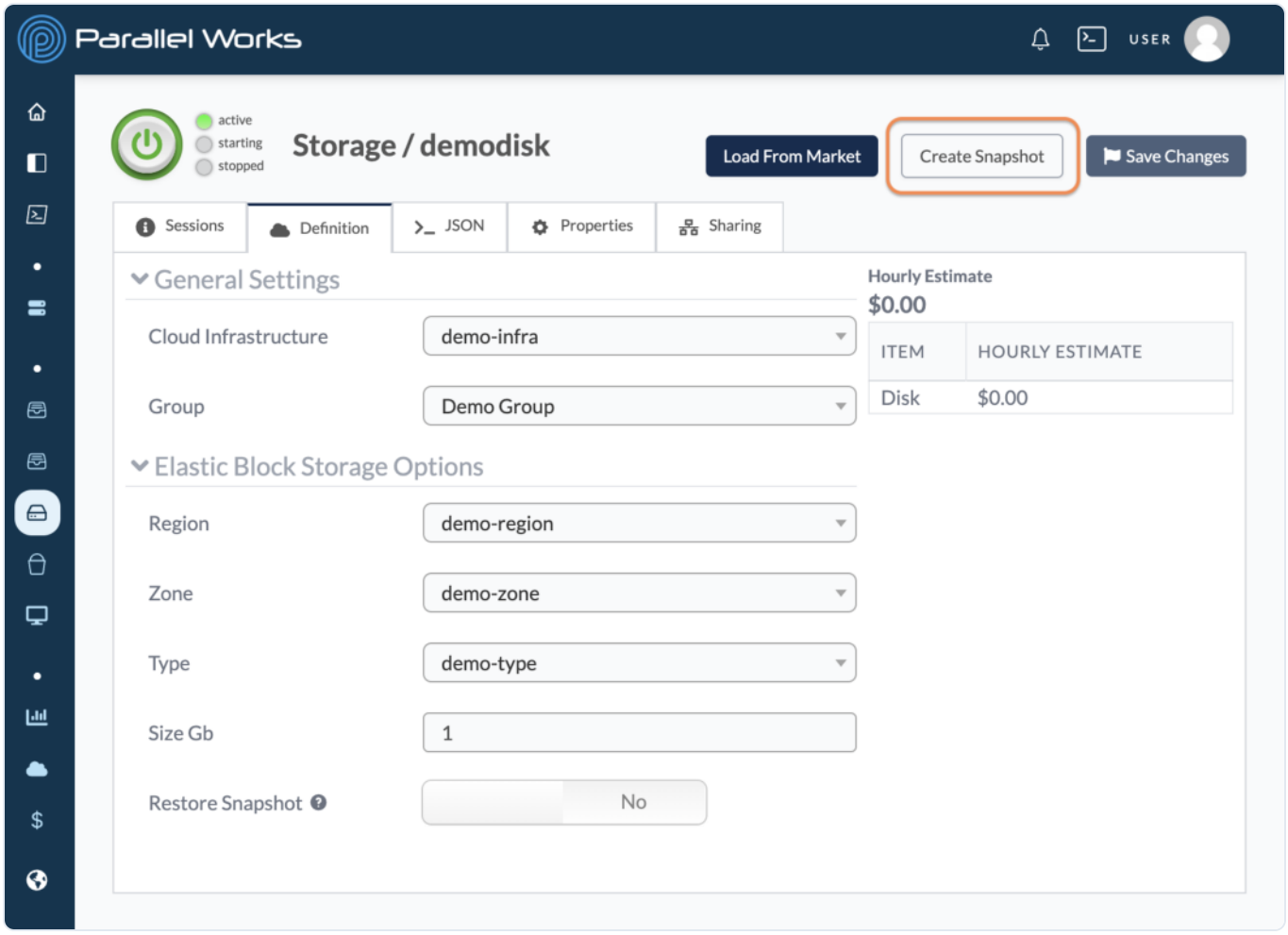
Please note that you can only create snapshots of *persistent* disks.

Creating a Snapshot

After configuring your disk, click the start button.

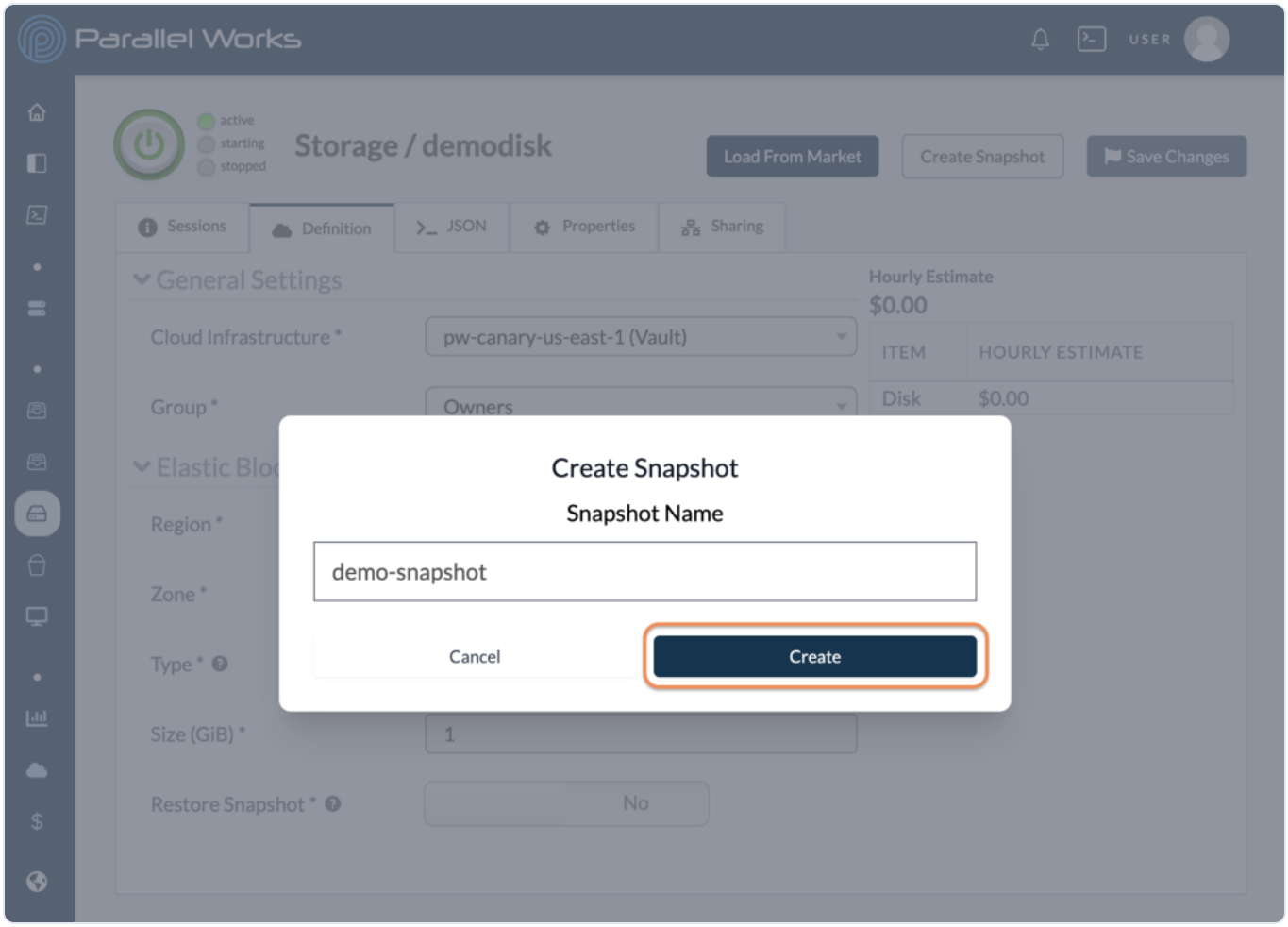


After the disk has started, click **Create Snapshot**.



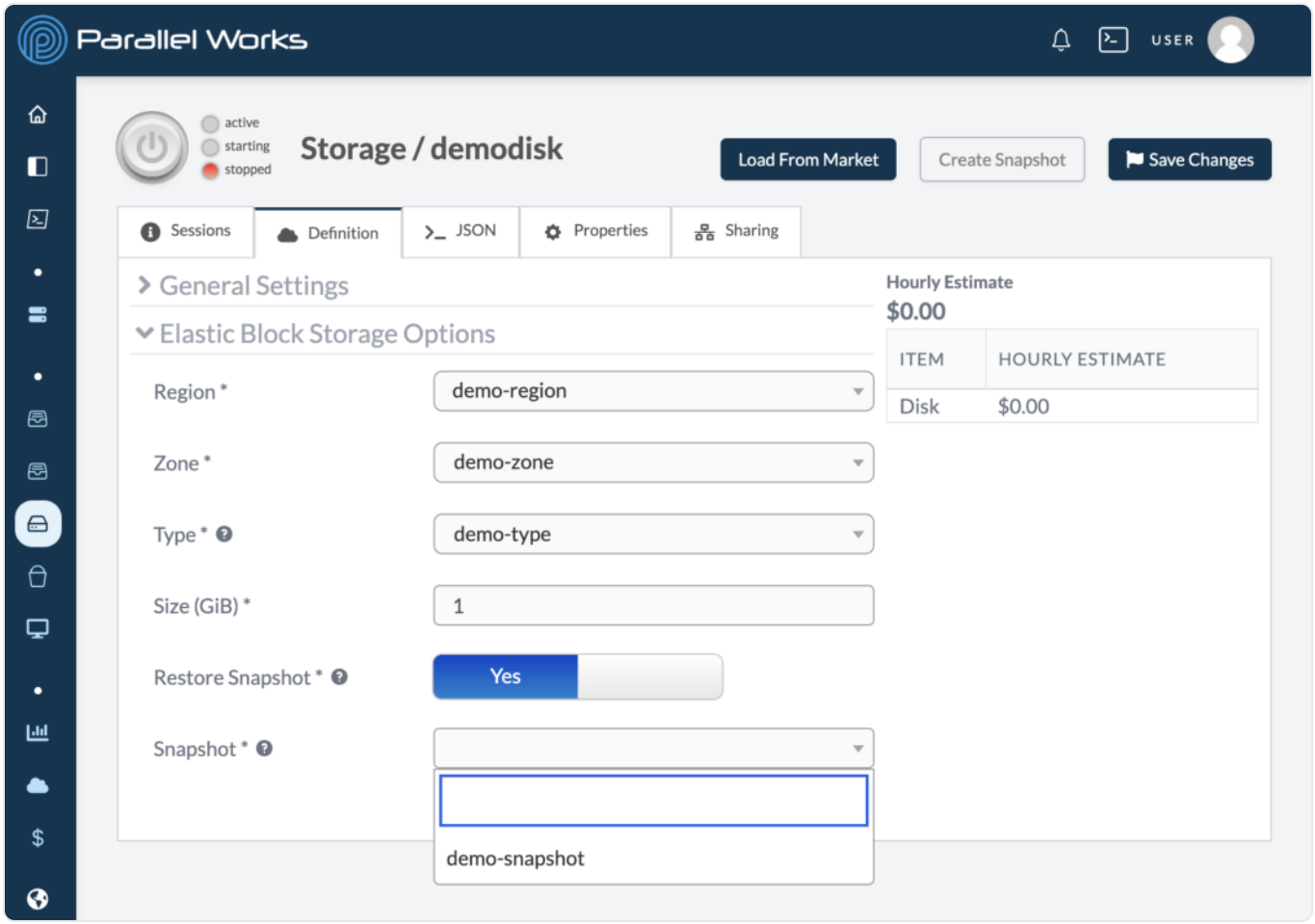
A new dialog box will appear. Enter a **Snapshot Name**. This field can only include lowercase letters, numbers, and single dashes.

Click **Create**.

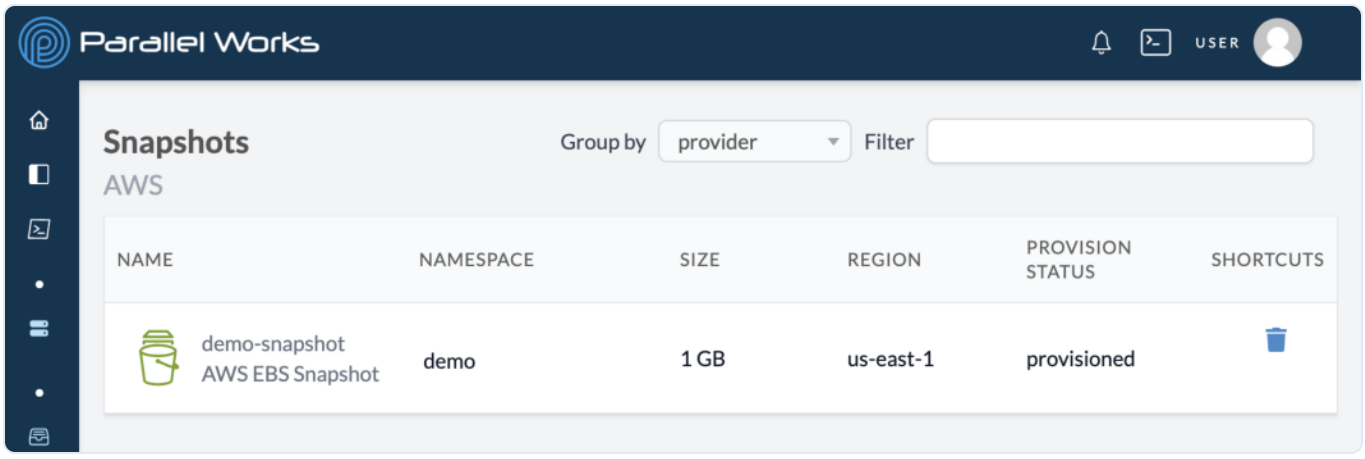


You'll see the message *The snapshot is being created. This process may take a few minutes.*

You'll be able to load your snapshot when configuring a disk of the same type (Azure Disk snapshots can only be loaded on other Azure Disks, for example). Click **Restore Snapshot**, then select a saved configuration from the **Snapshot** dropdown menu.

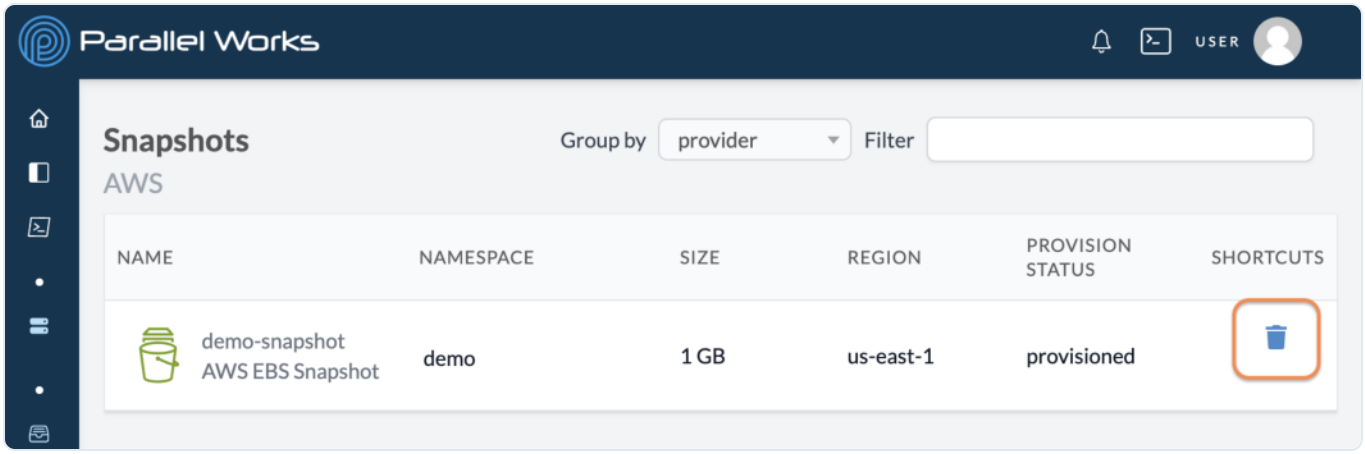


Your snapshot will also appear on the **Snapshots** page.



Deleting a Snapshot

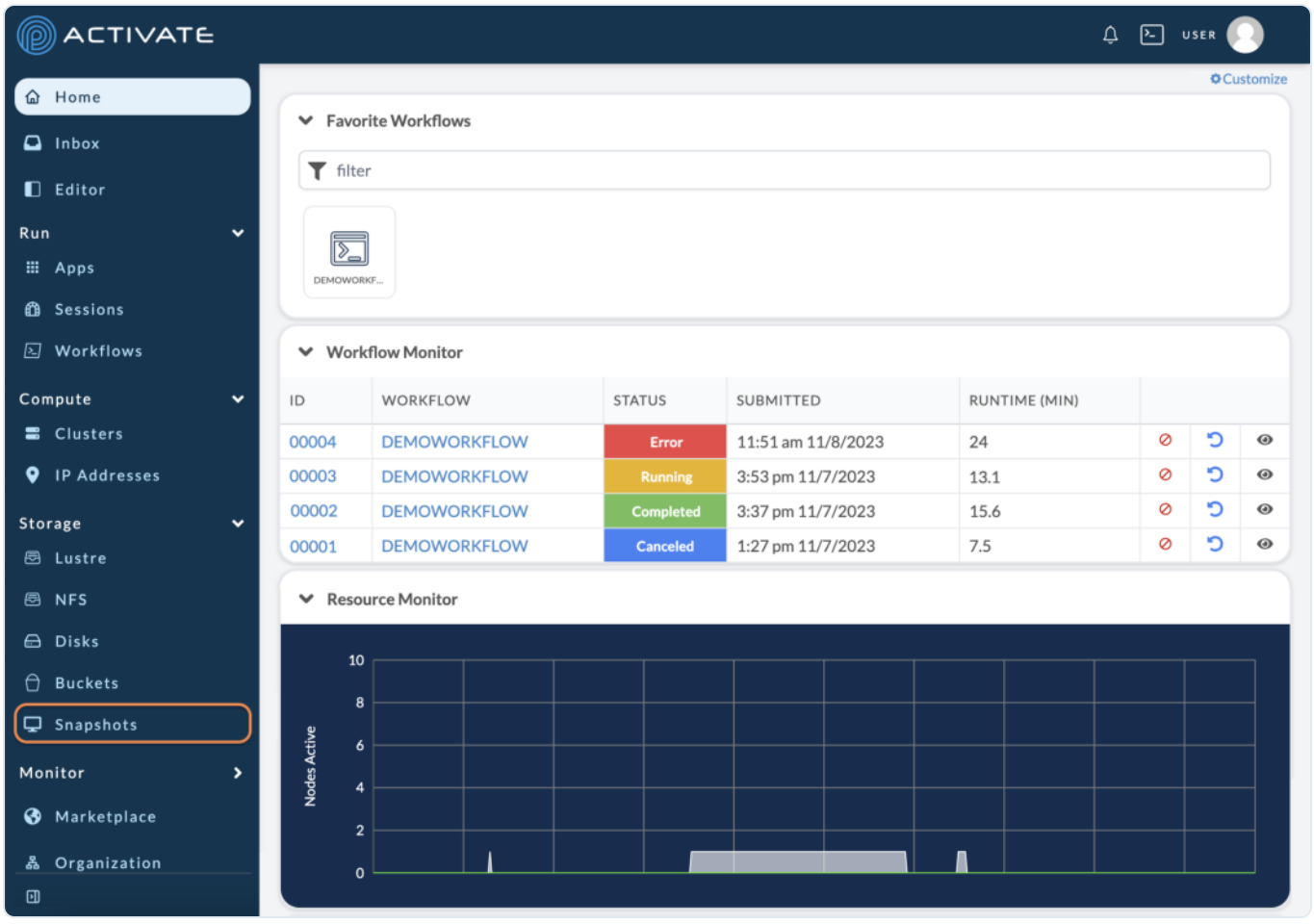
Navigate to **Shapshots**. Click the delete icon for the snapshot you'd like to remove.



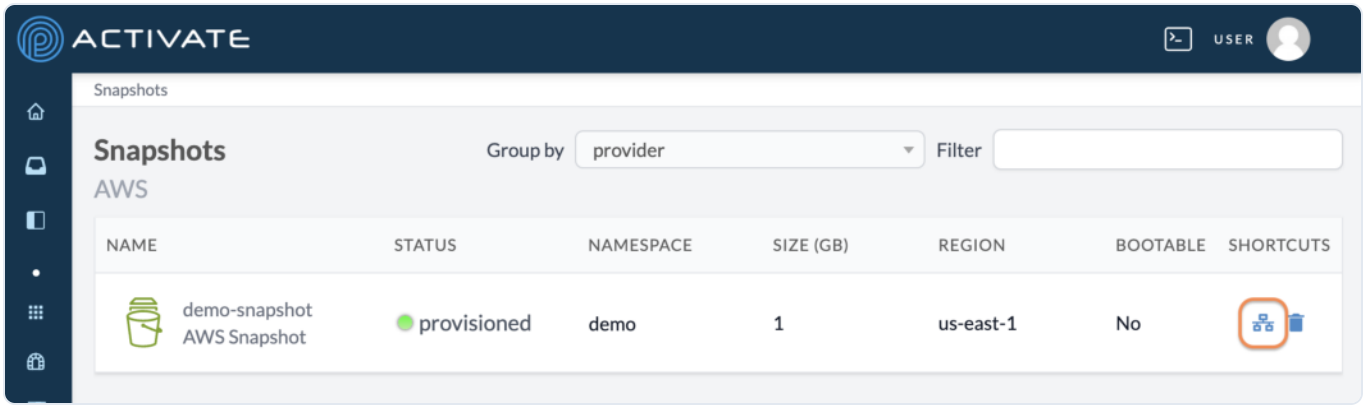
You'll see the message *The snapshot is being deleted. This process may take a few minutes.* The snapshot's **Provisioned Status** will display as **Deleted**. After one hour, the snapshot will be removed from your dashboard.

Sharing a Snapshot

Navigate to **Snapshots**.



Click the access icon.



On the next page, select the group(s) you want to share the snapshot with. You'll see the message *Permission updated successfully*.

Monitoring

The Monitor section provides visibility into your running resources, job status, and cloud spending.

Monitoring Areas

Instances & Jobs

Track your running compute resources and job status:

- View active and recently terminated instances
- Monitor cluster status and health
- Track job execution and completion

Costs

Understand and manage your cloud spending:

- View cost dashboards and trends
- Track spending by resource type
- Monitor allocation usage
- Generate cost reports

Quick Links

- Instances & Jobs - Resource and job monitoring
- Cost Dashboard - Cost tracking and analysis

Instances & Jobs

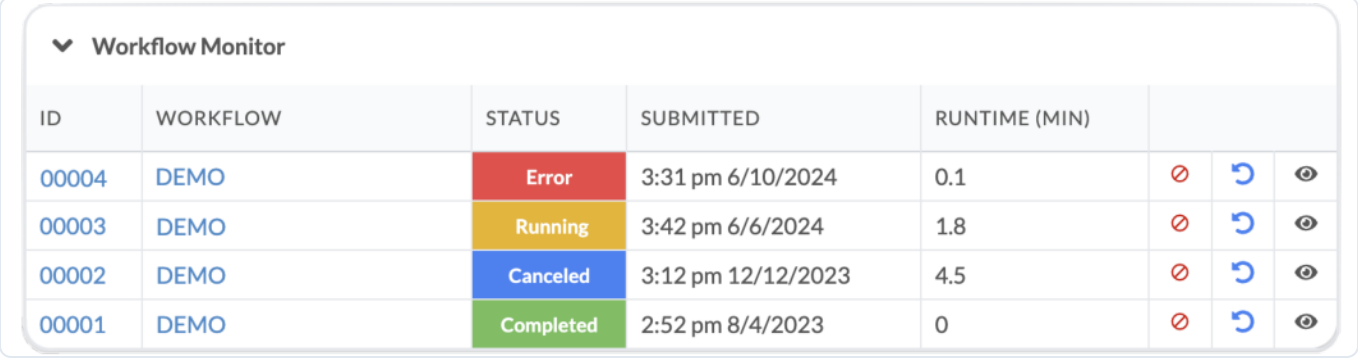
ACTIVATE features several data monitoring modules as well as a monitor dashboard to track your work.

Home

The **Home** page displays important data at a glance.

Workflow Monitor

This module shows a snapshot of your most recently run workflows.



Data columns for your workflows include:

- ID
- Workflow
- Status
- Submitted
- Runtime (Minutes)

If you click a workflow's **ID** number or its name in the **Workflow** column, you'll be taken to a more detailed view on the **Workflows** page.

A workflow's **Status** can be Started, Running, Completed, Canceled, or Error.

The **Workflow Monitor** also includes three important action buttons:

- Cancel Run
- Run Workflow Again
- View Active Workflow

If you click the icon to re-run a workflow, you'll be taken to the workflow's configuration form on the **Workflows** page.

Storage Resources

This module shows your favorited storage resources and their status (**active**, **starting**, or **stopped**).

If you click the information icon, you'll be taken to the storage's configuration form on the **Storage** page.

Compute

This module shows your favorited resources and their status (**active** nodes and **requested** nodes if active or **stopped** if inactive).

The navy bar reflects the number of maximum nodes a resource can have. In the screenshot above, the resource has the controller node and a partition that's configured for 10 maximum nodes, for a total of 11 possible nodes.

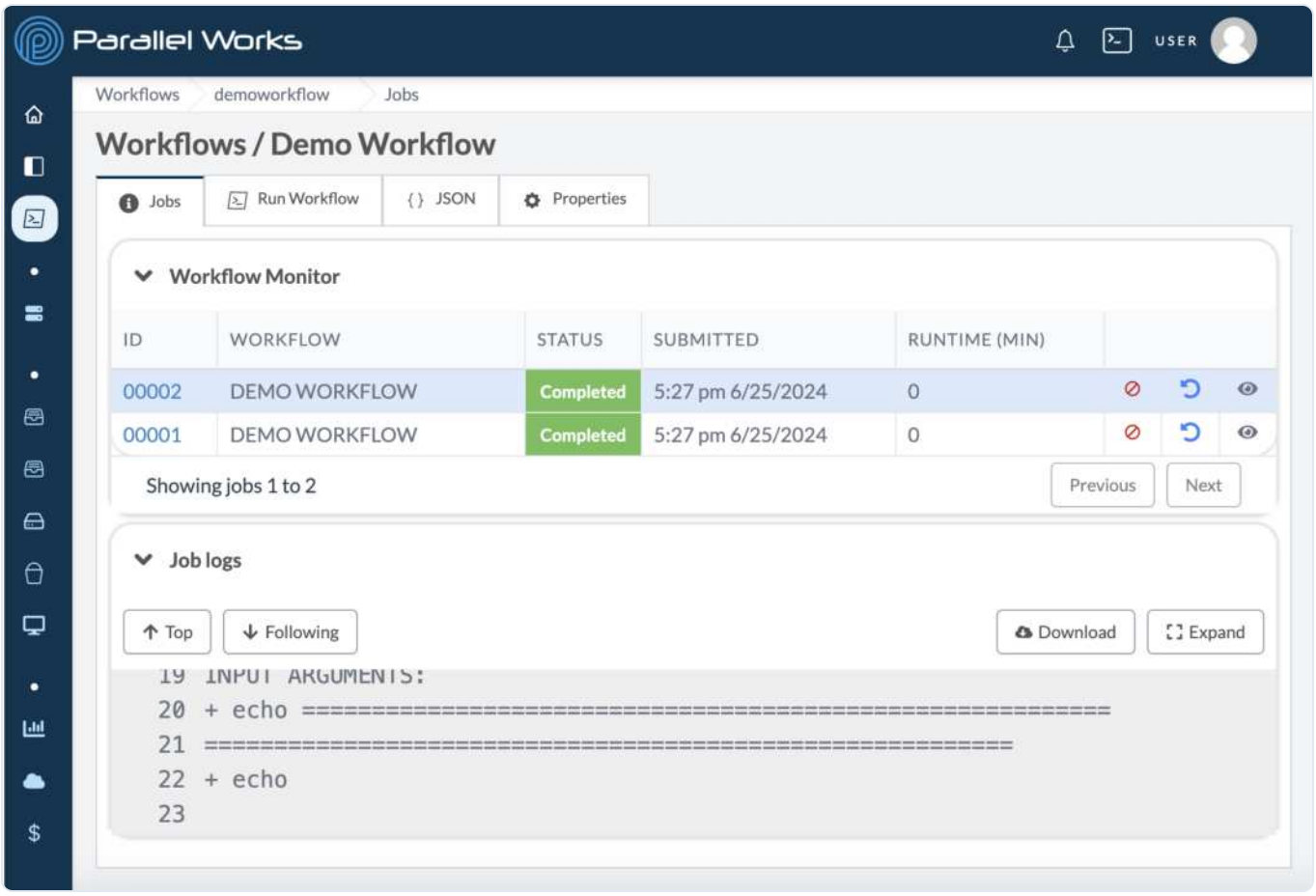
The green bar reflects the number of active nodes on a compute resource. In the screenshot above, the resource is active but not running any jobs, so there is 1 active node (the controller).

If you click the gear icon, you'll be taken to the resource's **Definition** tab on the **Clusters** page.

If you click the information icon, you'll be taken to the resource's **Sessions** tab on the **Clusters** page.

Workflows

On the **Workflows** page, click a workflow. You'll be taken to the workflow's **Jobs** tab.



The **Workflow Monitor** here mirrors the **Workflow Monitor** from the **Home** page.

The **Job logs** module shows details about specific workflow sessions. When you navigate to this page, this module shows `No log found` until you click a job number in the **ID** column.

If you navigate to this page after clicking the eye icon on a running workflow, the **Job logs** module will show details for that active session.

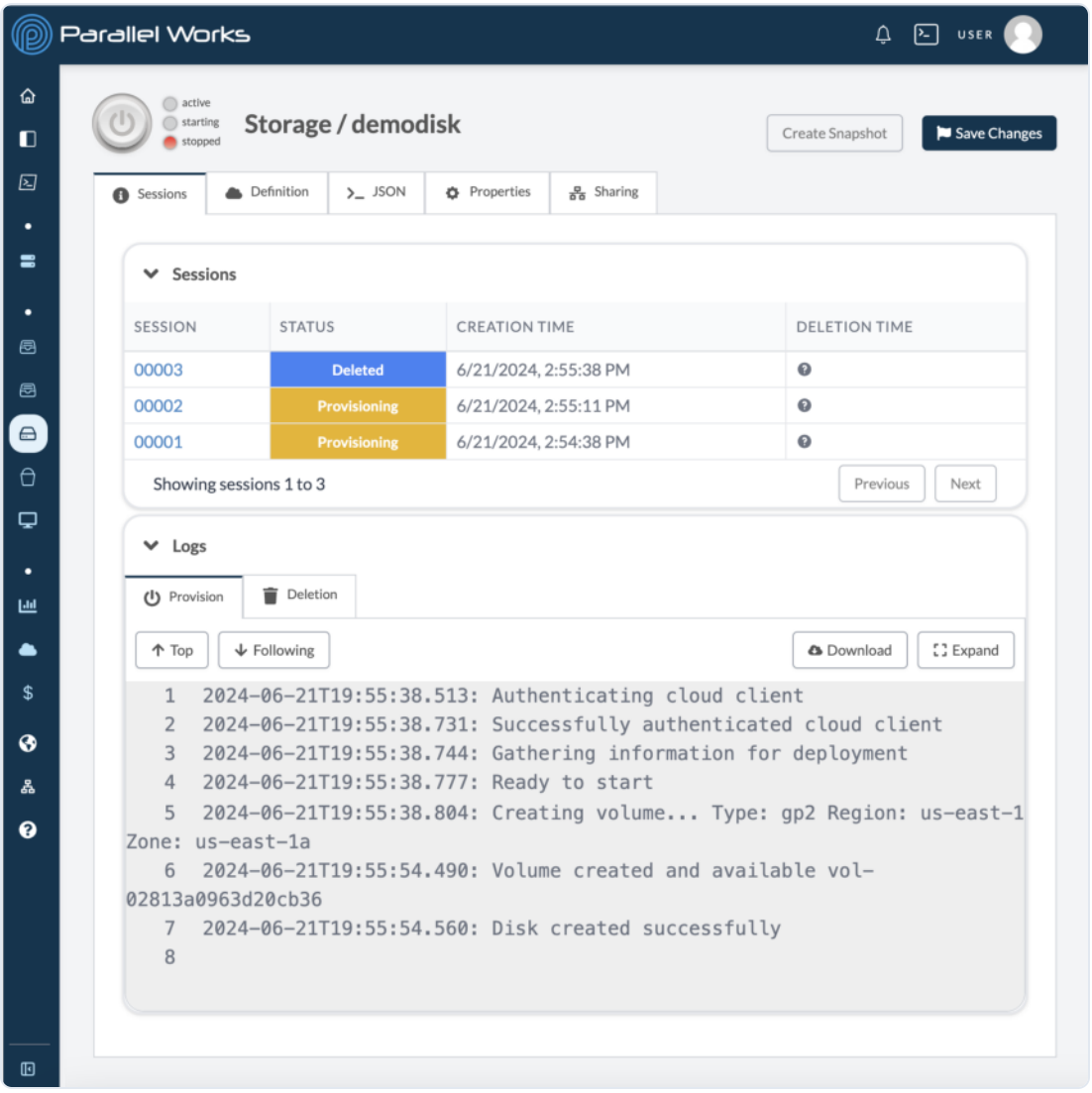
You can save workflow logs by clicking the **Download** button.

Clusters

When monitoring a cluster, the most important features are its **Status** and **Jobs** tab.

Storage

When you click on a persistent storage resource, you'll be taken to the storage's **Sessions** tab.



The **Sessions** module shows details about all your sessions with this storage resource, including:

- Session
- Status

- Creation Time
- Deletion Time

The **Logs** module shows details about specific storage sessions. When you navigate to this page, the **Provision** and **Deletion** tabs here show `Log not found` until you click a number in the **Session** column.

You can save storage logs by clicking the **Download** button.

> **EPHEMERAL STORAGE**
>
> Please note that ephemeral storage resources don't have this page because they're created and destroyed with a resource.
>
> You can see more details about ephemeral storage resources by navigating to their attached resource and clicking on the **Storages** tab of the **Logs** module.

Monitor

In the sidebar category **Monitor**, there are two pages: **Cost** and **Instances**.

Cost

The **Cost** page shows your cost data across ACTIVATE. For more information about this page, please see **Monitoring Costs**.

Instances

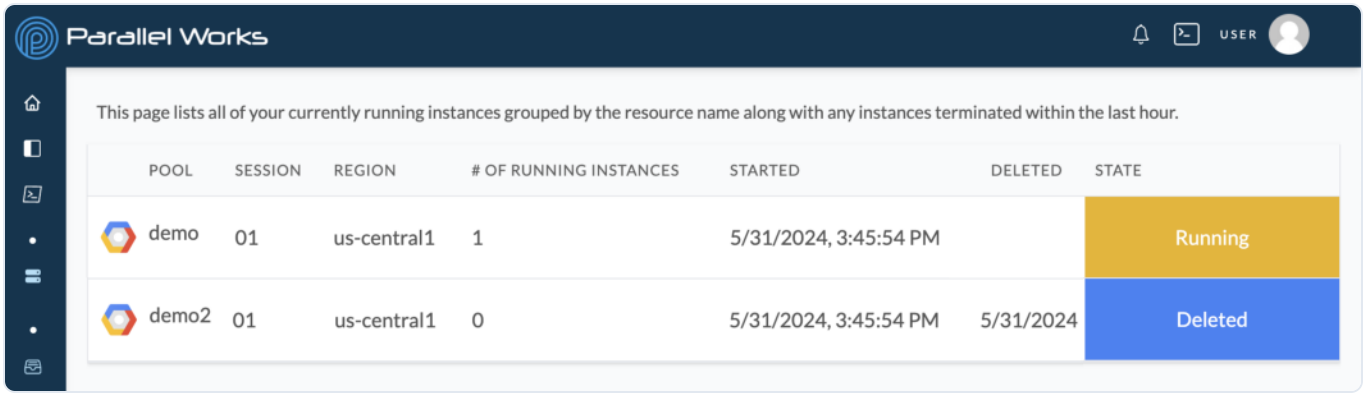
The **Instances** page shows your active and deleted clusters within the last hour.

Each instance listed here includes the following data:

- Pool
- Session
- Region
- Number of Running Instances
- Started
- Deleted
- State

If a cluster is active, its **State** will show **Running** in yellow.

If a cluster has recently been shut down, its **State** will show **Deleted** in blue.



If you haven't started or stopped a cluster within the last hour, the **Instances** tab will show the message *No instances found*.

Costs

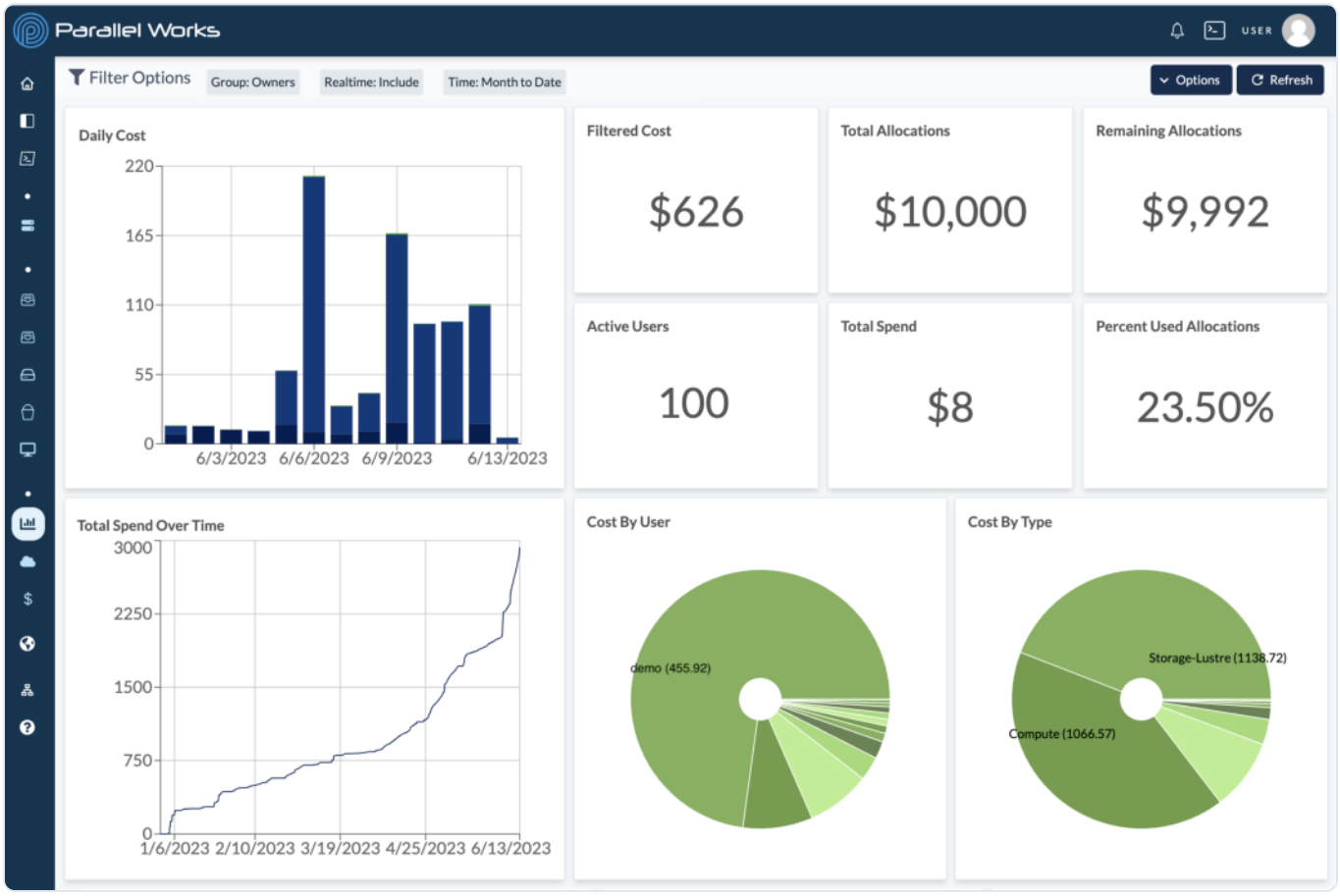
ACTIVATE features a dashboard that tracks costs across all cloud service providers (CSPs). The cost dashboard shows metrics down to an individual user's cluster session. Cost data is pulled directly from CSPs to show charges by usage type, such as compute instances and storage.

The dashboard's real-time cost tracking feature is designed to provide an accurate approximation of cloud costs between billing updates, capturing resource usage since the most recent official billing update. Official billing updates can be spaced by many hours; a large cluster can incur substantial expenses in that time window, so near real-time billing is an important feature to help users monitor their costs.

> **NOTE**
>
> Please note that you won't see the cost dashboard if your organization has not enabled it.

The Cost Dashboard

To reach the cost dashboard, navigate to the **Cost** page.



By default, you'll see the following data in the cost dashboard. You can change the view at any time; for more information, please see **Filtering Data** below.

Please note that all monetary values are listed in U.S. dollars (USD).

Graphs

The cost dashboard features several graphs to help users visualize data.

Unless otherwise noted, all filters can change a graph's displayed data. For more details, please see **Filtering Data** below.

Daily Cost

This graph shows the accumulated daily cost for users in a group within the selected **Time** filter.

You can mouse over any bar to see detailed costs for that day. For more details, please see **About Cost Types** below.

Total Spend Over Time

This graph shows a group's total expenditures to date.

Because this graph shows an accumulation of cost data for a group, only the **Group** filter changes its displayed data.

Cost by User

This graph shows a breakdown of costs by group users within the selected **Time** filter.

You can mouse over unlabeled pieces of the graph to see usernames.

Cost by Type

This graph shows a breakdown of costs by type within the selected **Time** filter.

You can mouse over unlabeled pieces of the graph to see cost types.

For more details about these costs, please see **About Cost Types** below.

Metrics

The cost dashboard features several data modules that display important usage metrics.

Unless otherwise noted, all filters can change a metric's displayed data. For more details, please see **Filtering Data** below.

Filtered Cost

This metric shows the costs for the filters you've applied.

For example, the cost dashboard defaults to data for the month to date and includes real-time tracking. The **Filtered Cost** module displays data only for those users within those parameters until you change the filters.

Total Allocations

This metric shows the total allocations that have been allotted for a group.

Allocations are set by ACTIVATE platform administrators; for more information, please see **Enforcing Allocations** (coming soon).

Remaining Allocations

This metric shows how much a group can spend before reaching its set cost allocation.

Active Users

This metric shows the active users for the filters you've applied.

For example, the cost dashboard defaults to data for the month to date and includes real-time tracking. There could be one hundred total users in your selected group, but with these filters, **Active Users** would show how many users used clusters and/or storage within that time period.

Total Spend

This metric shows the amount all users in your group have spent.

Because this graph shows an accumulation of cost data for a group, only the **Group** filter changes its displayed data.

If real-time tracking is enabled, this metric updates every three minutes.

If real-time tracking is disabled, update time is CSP dependent.

- AWS updates two to three times each day. They don't have a set billing time.
- Azure updates once each day. Their billing time depends on when your billing report was created.
- Google updates five to eight times each day. They don't have a set billing time.

Percent Used Allocations

This metric shows the percentage of allocations used by all users in a group.

Because this graph shows an accumulation of cost data for a group, only the **Group** filter changes its displayed data.

If real-time tracking is enabled, this metric updates every three minutes. If real-time tracking is disabled, update time is CSP dependent.

Tables

The cost dashboard features several tables that display important usage data.

All filters can change a table's displayed data. For more details, please see **Filtering Data** below.

Please note that all tables list data in descending order of costs.

User Cost

This table shows a breakdown of costs by user within the selected time frame.

Cost by Session

This table shows a breakdown of costs by users' sessions within the selected time frame.

Cost by Instance Type

This table shows a breakdown of costs by instance type within the selected time frame.

Cost by Type

This table shows a breakdown of costs by type within the selected time frame.

For more details about these costs, please see **About Cost Types** below.

Filtering Data

There are a variety of options for filtering the cost dashboard's data, including:

- Group
- Time
- Realtime
- User
- Type
- Pool
- Session

Three filters must have options selected: **Group**, **Realtime**, and **Time**. These options are pinned to the top of the **Cost** page. By default, these options are set to the first group you were added to, included real-time billing, and costs for the month to date. You can click any of these options to change them.

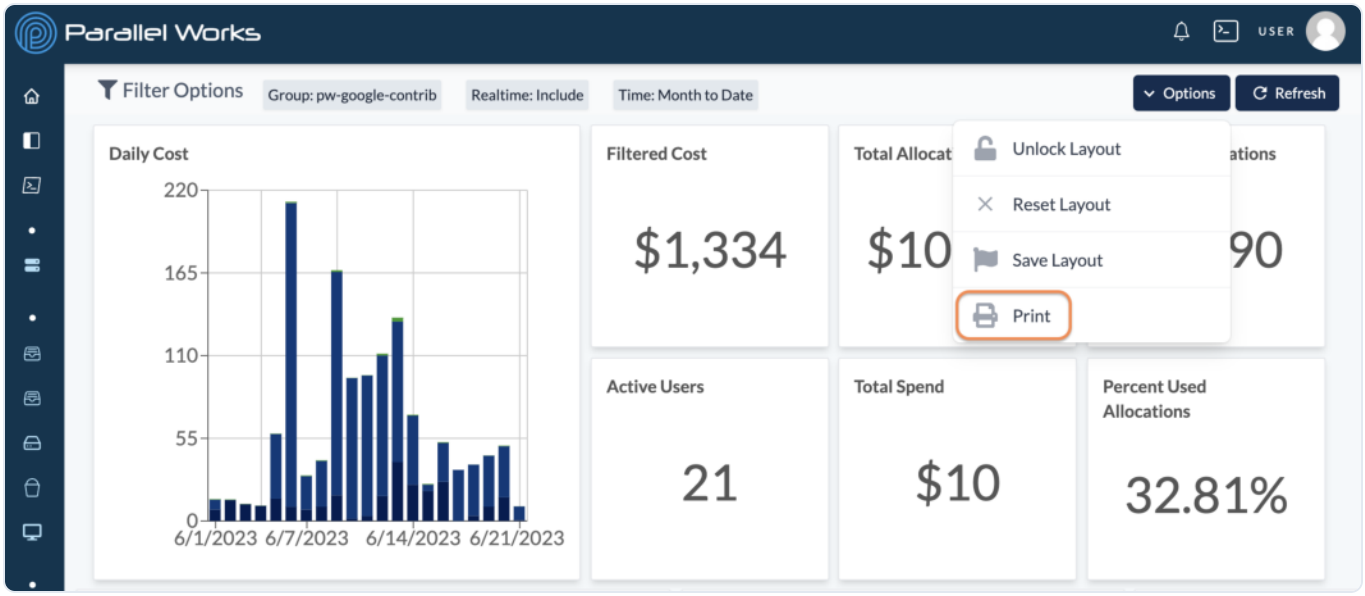
To add additional filters, click **Filter Options** and select any filter from the list. Next, use the dropdown menu to select the filter parameter. All filter dropdown menus include a search bar for quickly finding parameters.

Please note that some filters are conditional. For example, you must select a **User** before you can select a **Pool**. You must select a **Pool** before you can select a **Session**.

Printing Data

You can print the cost dashboard data at any time.

Click **Options**, then **Print**.



A **Print** window will appear. Select the option for **Save as PDF**. Click **Save**.

The cost dashboard page will be downloaded to your device as a PDF.

About Cost Types

Project costs can be divided into three broad categories: compute, storage, and network charges.

Compute costs include the controller, compute nodes, and custom images. Because compute nodes are capable of starting and stopping elastically based on scheduler load, charges can be minimized when a cluster is left on with a period of idle time between jobs.

Storage costs include disks attached to compute nodes, as well as optional storage deployed with a cluster. Other storage charges include persistent storage maintained outside of cluster configuration. For more information, please see **Storage Types**.

Network charges may accrue if data is transferred from the cloud or across regions.

Compute Charges

Compute

This cost type includes instance-related charges. Primarily, these costs include the instances themselves. Some CSPs also include attached items, such as external network interfaces.

Compute Nodes

This cost type is similar to **Compute** costs, but only includes data for compute nodes.

Storage Charges

Storage

This cost type is associated with `pwbilling` and `ca-cloudmgmt`, which applies only to AWS instances.

Storage-Disk

This cost type applies to disk storage on Azure instances.

Storage-EBS

This cost type applies to AWS instances. Elastic Block Storage (EBS) is a subset of the AWS service Elastic Cloud Compute (EC2). **Storage-EBS** includes instance local disks as well as snapshots.

Storage-EFS

This cost type applies to AWS instances. Elastic Filesystem (EFS) is an AWS feature.

Storage-File

This cost type applies to Azure instances with `Premium LRS All Other Operations` enabled.

Storage-GCS

This cost type applies to Google instances with Google Cloud Storage (GCS) enabled.

Storage-Image

This cost type includes snapshots on all CSPs.

Storage-Lustre

This cost type applies to Lustre storage resources.

Storage-PD

This cost type includes local boot disks for Google instances. PD stands for persistent disk.

Storage-S3

This cost type applies to AWS instances. Simple Storage Service (S3) is a bucket storage resource from AWS.

Network Charges

Egress

This cost type applies to data transfers out of the cloud.

Ingress

This cost type applies to data transfers into the cloud.

To better understand what counts as ingress and egress on a particular cloud, please see their respective documentation.

- AWS
- Azure
- Google

Other Charges

Infrastructure

This cost type includes cloud infrastructure charges, such as network transit gateways.

Other

This cost type includes miscellaneous charges that are not essential for tracking purposes. Charges in this category include items like API requests, security operations, and CSP-specific support fees.

Unknown

This cost type includes charges that haven't been labeled on the ACTIVATE platform yet.

Our cost types are based on the way CSPs name their charges. Because each CSP stores many different types of cost data, our engineering team must sort, consolidate, and map that data to corresponding ACTIVATE cost types. **Unknown** costs are re-assessed quarterly.

AI

ACTIVATE provides integrated AI capabilities, giving you access to AI-powered chat with configurable model providers directly within the platform.

AI Chat

AI Chat is a full-featured chat interface for interacting with large language models. You can have conversations with AI models, attach files for context, branch conversations to explore different approaches, and share conversations with teammates.

- AI Chat Overview — Get started with the chat interface, learn about available features
- AI Chat Providers — Configure the model providers that power AI Chat
- Best Practices — Tips for prompting, model selection, and security

Related Documentation

- Machine Learning — ML workspace resources
- Getting Started — Tutorial for new users

AI Chat

AI Chat provides a full-featured chat interface for interacting with large language models directly within ACTIVATE. You can have conversations with AI models from multiple providers, attach files for context, branch conversations to explore different approaches, and share conversations with your team.

> **FEATURE MAY NOT BE AVAILABLE ON ALL PLATFORMS**
>
> AI Chat requires at least one AI provider to be configured. If you do not see the **Chat** option in your ACTIVATE account, contact your administrator.

Access Methods

You can interact with AI Chat in three ways:

- **Web UI** -- Open **Chat** from the sidebar navigation to use the full-featured chat interface with file attachments, conversation branching, and message history.
- **CLI** -- Run `pw ai chats new` to start an interactive chat session from the command line. You can pick a model interactively or specify one directly. See Using the CLI for details.
- **API** -- Send requests to the OpenAI-compatible API endpoint at `/api/openai/v1/chat/completions` for programmatic access. Model IDs use the format `owner:provider-name/model-name`.

Capabilities

- **Multi-provider support** -- Connect to Azure OpenAI, custom OpenAI-compatible endpoints, or models running on your compute clusters via session tunnels.

- **File attachments** -- Attach documents, images, and other files to your messages for additional context. Regular files up to 25 MB and documents (PDF, text, CSV, Office files) up to 100 MB are supported.
- **Conversation branching** -- Edit previous messages or regenerate responses to create conversation branches. Navigate between branches to compare different approaches.
- **Sharing and collaboration** -- Share conversations with your team for read-only viewing or collaborative editing.
- **Reasoning models** -- Use reasoning models (such as o1 and o3) that show their thinking process and support configurable reasoning effort.
- **Streaming responses** -- Responses stream in real time as the model generates them, so you can read output as it arrives.

Related Documentation

- Getting Started -- Send your first message
- Attaching Files -- Add files to your conversations
- Branching Conversations -- Explore alternative conversation paths
- Sharing Conversations -- Collaborate with your team
- Using Reasoning Models -- Work with reasoning models
- Using the CLI -- Chat from the command line
- AI Chat Providers -- Configure AI providers
- Troubleshooting -- Resolve common issues

Getting Started

This guide walks you through sending your first message in AI Chat, selecting a model, and managing your conversation history.

Prerequisites

Before you can use AI Chat, at least one AI provider must be configured and accessible to you. Providers can be added by you or shared with you by an administrator or team member. See AI Chat Providers for setup instructions.

Opening the Chat Interface

Click **Chat** in the sidebar navigation to open the AI Chat interface. If no providers are configured, you will see a prompt to add one.

When providers are available, the chat page displays a greeting and a set of suggested prompts to help you get started.

Selecting a Provider and Model

Use the model selector in the toolbar at the top of the chat interface to choose which AI model to use. Models are grouped by provider and listed with human-friendly names (for example, **my-azure-provider / gpt-4o**). If a provider offers multiple models, each one appears as a separate option.

Session tunnel models (models running on your compute clusters) also appear in this list when an active tunnel is available.

Sending a Message

1. Type your message in the input field at the bottom of the chat interface, or click one of the suggested prompts.
2. Optionally attach files by clicking the attachment button or dragging files into the chat area.
3. Press **Enter** or click the send button.

Your message is sent to the selected model, and a new conversation is created automatically if one does not already exist.

Understanding Streaming Responses

AI Chat streams responses in real time. As the model generates its reply, text appears incrementally in the chat interface. A green indicator is shown while the response is streaming.

For reasoning models, a "Thought for..." indicator appears after the response completes, showing how long the model spent in its thinking phase. You can click this indicator to view the model's reasoning process.

Managing Conversation History

Your conversations are saved automatically and listed in the sidebar. Conversations are sorted by most recently updated.

Viewing Past Conversations

Click any conversation in the sidebar to reopen it. The full message history, including any branches, is restored.

Renaming a Conversation

Conversations can be renamed by updating the title. This helps you organize and find past conversations.

Deleting a Conversation

Only the conversation owner can delete a conversation. Deleting a conversation also removes all associated attachments and sharing permissions.

Next Steps

- Attaching Files -- Learn how to include files in your messages
- Branching Conversations -- Explore alternative responses
- Sharing Conversations -- Share with your team
- Using Reasoning Models -- Use models with extended thinking
- Using the CLI -- Chat from the command line

Related Documentation

- AI Chat Overview -- Feature overview and access methods
- AI Chat Providers -- Configure AI providers

- Troubleshooting -- Resolve common issues

Attaching Files

AI Chat supports file attachments, allowing you to include documents, images, and other files as context for your conversations. Attached files are sent alongside your messages so the AI model can reference their content when generating responses.

Supported File Types

You can attach the following file types:

Category	Formats
Documents	PDF, TXT, CSV, DOC, DOCX, XLS, XLSX, PPT, PPTX
Images	PNG, JPG, JPEG, GIF, WebP, and other common image formats

Size Limits

File size limits depend on the file type:

- **Documents** (PDF, TXT, CSV, XLSX, DOCX, PPTX) -- up to **100 MB** per file
- **All other files** (images, etc.) -- up to **25 MB** per file
- **Maximum files per message** -- up to **10 files**

If a file exceeds the size limit for its type, the upload is rejected with an error message indicating the maximum allowed size.

How to Attach Files

There are two ways to attach files to a message:

Drag and Drop

Drag one or more files from your file manager directly into the chat area. A drop zone overlay appears to confirm the drop target. Release the files to begin uploading.

File Picker

Click the attachment button in the chat input area to open a file picker. Select one or more files and confirm. The selected files are uploaded immediately.

Upload Process

When you add files, each file shows an upload progress indicator. Once uploaded, the file appears as a chip showing the filename and file size. You can remove an attachment before sending your message by clicking the remove button on the file chip.

Files can be uploaded before a conversation exists. These standalone attachments are associated with the conversation when you send your first message.

Viewing Attachments in Messages

Attachments appear below the message content:

- **Images** are displayed as inline thumbnails that you can view directly in the chat.
- **Documents and other files** appear as downloadable file chips showing the filename and size. Click on a file chip to download the attachment.

Managing Attachments

- You can view all attachments associated with a conversation.
- Attachments can be downloaded at any time by clicking on them in the message history.
- When a conversation is deleted, all of its attachments are also deleted.
- Only the attachment owner can delete individual attachments.

Limitations

- File content processing depends on the capabilities of the selected AI model. Not all models support all file types.
- Very large files may increase response times as the model processes the additional context.
- Image support requires a model that accepts multimodal input (text and images).

Related Documentation

- Getting Started -- Send your first message
- AI Chat Overview -- Feature overview
- Troubleshooting -- Resolve file upload issues

Branching Conversations

AI Chat uses a tree-based message structure rather than a simple linear history. This means you can create branches at any point in a conversation to explore alternative responses, compare different approaches, or refine your prompts without losing previous work.

How Branching Works

Every message in a conversation has a parent message (except the first message, which is the root). When you edit a previous message or regenerate an AI response, a new branch is created from that point. The original branch is preserved, and you can switch between branches at any time.

This tree structure means a single conversation can contain multiple parallel threads of discussion, all sharing the same earlier context.

Creating a Branch

There are two ways to create a new branch:

Editing a Message

1. Hover over one of your messages to reveal the action buttons.

2. Click the **Edit** button.
3. Modify the message content in the editor that appears.
4. Click **Send** to submit the edited message.

A new branch is created starting from the edited message. The AI generates a fresh response based on your updated input. The original message and its response remain available on the previous branch.

Regenerating a Response

1. Hover over an AI response to reveal the action buttons.
2. Click the **Regenerate** button.

The model generates a new response to the same prompt. This creates a sibling branch from the same parent message. You can regenerate multiple times to get different responses and compare them.

Navigating Between Branches

When a message has sibling branches (multiple responses to the same parent), a branch navigator appears below the message. The navigator shows the current branch position (for example, "2/3") and provides arrow buttons to switch between siblings.

Clicking the left or right arrow loads the sibling message and its entire downstream conversation thread. The branch navigator only appears at messages where the conversation actually diverges.

Active Branch

Each conversation tracks an **active branch**, which determines which path through the message tree is displayed by default when you open the conversation. The active branch updates automatically as you send messages or navigate to different branches.

Use Cases

Branching is useful in several scenarios:

- **Comparing approaches** -- Ask the same question with different phrasings to see how the model responds differently, without starting a new conversation.
- **Exploring alternatives** -- Regenerate a response multiple times and keep all versions for comparison.
- **Iterating on prompts** -- Edit an earlier message to refine your prompt while preserving the original conversation path.
- **A/B testing** -- Test different instructions or system configurations within the same conversation context.

Related Documentation

- Getting Started -- Send your first message
- AI Chat Overview -- Feature overview
- Sharing Conversations -- Collaborate with your team

Sharing Conversations

AI Chat allows you to share conversations with other members of your organization. You can grant read-only access for review purposes or full collaboration access that lets others contribute messages to the conversation.

Permission Levels

AI Chat supports two permission levels for shared conversations:

- **View** -- Recipients can read the entire conversation, including all messages and branches, but cannot add new messages or modify existing content.
- **Collaborate** -- Recipients can read the conversation and add their own messages. This is useful for team problem-solving or collaborative analysis sessions.

> **OWNER ONLY**
>
> Only the conversation owner can manage sharing settings. If you need access to a conversation, ask the owner to share it with you.

How to Share a Conversation

1. Open the conversation you want to share.
2. Click the **Share** icon in the conversation header.
3. In the **Share Conversation** dialog, configure access at the organization or group level.

Organization-Wide Access

To share with everyone in your organization:

1. Check **Anyone in the organization can view** to grant view access to all organization members.
2. Optionally check **Allow collaboration (edit messages)** to upgrade the permission from view-only to collaborate.

Group Access

To share with specific groups:

1. In the **Group Access** section, use the **Add groups** dropdown to select one or more groups.
2. For each group, choose the permission level using the dropdown next to the group name:
   - **View only** -- Members of the group can read the conversation.
   - **Can collaborate** -- Members of the group can read and add messages.
3. To remove a group's access, click the trash icon next to the group entry.

After configuring the desired permissions, click **Save** to apply the changes.

Accessing Shared Conversations

Conversations that have been shared with you appear in your conversation list. You can identify shared conversations by the presence of the share indicator. Depending on your permission level:

- **View access** -- You can open and read the conversation but cannot send messages.
- **Collaborate access** -- You can open the conversation and send messages as if it were your own.

Managing and Revoking Access

To modify or remove access for a shared conversation:

1. Open the conversation and click the **Share** icon.
2. Change permission levels using the dropdown next to each group.
3. Remove a group by clicking the trash icon.
4. Uncheck the organization-wide access checkbox to revoke access for the entire organization.
5. Click **Save** to apply changes.

Changes take effect immediately. Users who lose access will no longer see the conversation in their list.

Collaboration Tips

- **Use branches for parallel exploration** -- When collaborating, each team member can create branches from the same message to explore different approaches without interfering with each other's work.
- **Set clear permissions** -- Use view-only access for stakeholders who need to review progress and collaborate access for team members who need to actively contribute.
- **Share with groups, not individuals** -- Sharing at the group level makes it easier to manage access as team membership changes.

Related Documentation

- Branching Conversations - Create parallel conversation threads
- Getting Started with AI Chat - First-time setup walkthrough
- Best Practices - Tips for effective AI collaboration

Using Reasoning Models

Reasoning models such as OpenAI's o1 and o3 series use an extended thinking process before generating a response. AI Chat supports these models and surfaces their reasoning output so you can follow the model's chain of thought.

What Are Reasoning Models?

Standard chat models generate responses token by token in a single pass. Reasoning models add a **thinking phase** before the final response. During this phase, the model works through the problem step by step, considering different approaches and verifying its logic before producing an answer.

This thinking process is especially beneficial for:

- Complex math and logic problems
- Multi-step analysis and planning
- Code debugging and architectural decisions
- Tasks that require careful evaluation of tradeoffs

How Reasoning Models Differ

When you use a reasoning model in AI Chat, the experience differs from standard models in several ways:

- **Longer response times** -- The thinking phase adds processing time before the response begins streaming. This is normal and expected.
- **Reasoning content** -- The model's internal reasoning may be displayed alongside the final response, allowing you to follow how it arrived at its answer.
- **Reasoning tokens** -- Reasoning models consume additional tokens for the thinking phase. These reasoning tokens count toward the model's context window and usage.

Reasoning Effort

Some reasoning models support a **reasoning effort** parameter that controls how much time the model spends thinking before responding. AI Chat supports three effort levels:

- **Low** -- Minimal reasoning. The model spends less time thinking, resulting in faster responses. Best for straightforward questions where deep analysis is not needed.
- **Medium** -- Balanced reasoning. A good default for most tasks.
- **High** -- Maximum reasoning. The model takes more time to think through the problem thoroughly. Best for complex problems where accuracy is critical.

> **MODEL SUPPORT**
>
> Reasoning effort is only available for models that support it. If the selected model does not support reasoning effort, the setting has no effect.

When to Use Reasoning Models

Use reasoning models when:

- The task involves complex logic, math, or multi-step problem-solving.
- You need the model to carefully evaluate multiple options before answering.
- Accuracy matters more than response speed.
- You are debugging intricate code or analyzing system architectures.

Use standard models when:

- You need quick, conversational responses.
- The task is straightforward (summarization, simple Q&A, text generation).
- Low latency is important.
- You want to minimize token usage.

Working with Reasoning Output

When a reasoning model is used, AI Chat may display the model's reasoning process alongside the final response. This reasoning content shows the steps the model took to arrive at its answer. Reviewing this output can help you:

- Verify the model's logic and catch errors in its reasoning.
- Understand why the model chose a particular approach.
- Refine your prompt based on how the model interpreted your request.

Related Documentation

- Getting Started with AI Chat - First-time setup walkthrough
- Best Practices - Tips for model selection and effective prompting
- AI Chat Providers - Configure providers that offer reasoning models

Using the CLI

You can interact with AI Chat directly from your terminal using the PW CLI. The CLI provides an interactive TUI chat session with markdown rendering and streaming responses, as well as commands for managing chats, providers, and models.

Prerequisites

Before using AI Chat from the CLI, ensure:

- The **PW CLI** is installed and available on your system. See CLI Installation for setup instructions.
- You are **authenticated** with the CLI (`pw auth login`).
- At least one **AI Chat provider** is configured and accessible to you. See AI Chat Providers for setup details.

Listing Available Models

Before starting a chat, you can see which models are available to you:

```
pw ai models ls
```

This displays a table of all models across your configured providers. Each model is identified by a human-friendly identifier in the format `owner:provider-name/model-name` (for example, `me:my-azure-provider/gpt-4o`).

Starting a New Chat

To start an interactive chat session, run:

```
pw ai chats new
```

This opens a full-screen interactive TUI. If no model is specified, an interactive model picker is displayed so you can choose from your available models. You can also specify a model directly to skip the picker:

```
pw ai chats new me:my-azure-provider/gpt-4o
```

Non-Interactive Mode

To send a single prompt and print the response without entering the interactive TUI, use the `--prompt` flag. A model argument is required in non-interactive mode:

```
pw ai chats new -p "What is the capital of France?" me:my-azure-provider/gpt-4o
```

By default, non-interactive chats are not saved. To persist the conversation, add the `--save` flag:

```
pw ai chats new -p "Summarize TCP vs UDP" --save me:my-azure-provider/gpt-4o
```

Resuming a Chat

To continue a previous conversation:

```
pw ai chats resume <id>
```

The model is automatically detected from the conversation history. You can override it with the `--model` flag:

```
pw ai chats resume -m me:my-azure-provider/gpt-4o abc-123
```

You can also send a single prompt to an existing chat:

```
pw ai chats resume -p "Summarize our conversation" abc-123
```

Managing Chats

Listing Chats

```
pw ai chats ls
```

Lists your recent chats in a table, sorted with the most recent at the bottom. Use `--limit` and `--offset` for pagination, and `-o json` for JSON output.

Viewing a Chat

```
pw ai chats get <id>
```

Displays the full message history of a chat. Use `-o json` for JSON output.

Deleting a Chat

```
pw ai chats delete <id>
```

Managing Providers

The CLI also provides commands for managing AI Chat providers:

```
# List all providers
pw ai providers ls

# Get details of a specific provider
pw ai providers get my-provider

# List models available for a provider
pw ai providers models my-provider

# Create a custom provider
pw ai providers create --name my-provider --csp custom --endpoint https://api.example.com --api-key sk-xxx

# Create an Azure provider
pw ai providers create --name my-azure --csp azure --region eastus --model gpt-4 --group my-group --network my-network

# Delete a provider
pw ai providers delete my-provider
```

Interactive Session

Once the TUI starts, you can type messages and press **Enter** to send them. Responses stream in real time with markdown formatting.

Ending a Session

- Press **Ctrl+C** to quit the interactive session.

Examples

Basic question and answer

```
pw ai chats new me:my-azure-provider/gpt-4o
```

```
> Explain the difference between TCP and UDP in two sentences.

TCP is a connection-oriented protocol that guarantees reliable, ordered
delivery of data through acknowledgments and retransmissions. UDP is a
connectionless protocol that sends data without establishing a connection
or guaranteeing delivery, making it faster but less reliable.
```

Single prompt from a script

```
pw ai chats new -p "List three sorting algorithms" me:my-provider/gpt-4o
```

Resume and continue a conversation

```
# List your chats to find the ID
pw ai chats ls

# Resume the conversation
pw ai chats resume abc-123
```

Related Documentation

- Getting Started with AI Chat - Web-based chat walkthrough
- AI Chat Providers - Configure and manage providers
- CLI Reference: pw ai - Full CLI command reference
- CLI Reference: pw ai chats new - New chat command reference
- CLI Reference: pw ai chats resume - Resume chat command reference

Troubleshooting

This page covers common issues you may encounter when using AI Chat and how to resolve them.

No Providers Available

Symptom: The provider or model dropdown is empty, or you see a message indicating no providers are configured.

Cause: No AI Chat providers have been set up for your organization, or you do not have permission to use the available providers.

Solution:

- Contact your organization administrator and ask them to configure an AI Chat provider. See AI Chat Providers for setup instructions.
- If providers exist but you cannot see them, ask the administrator to grant your group access to the provider. See Managing Provider Permissions.

Connection Errors

Symptom: You receive an error when sending a message, such as "Failed to connect" or "Provider endpoint unreachable."

Cause: The underlying AI provider endpoint may be unavailable, misconfigured, or experiencing an outage.

Solution:

- Verify that the provider endpoint is correct and accessible. For custom OpenAI-compatible providers, confirm that the endpoint URL is reachable from the ACTIVATE platform.
- For session tunnel providers, ensure the tunnel is active and the compute session is running. Tunnels are only available while the associated session is active. See Session Tunnels.
- For Azure OpenAI providers, verify the API key and endpoint in the provider configuration. Check the Azure status page for any ongoing outages.
- Try sending another message after a few moments. Transient network issues may resolve on their own.

Slow or Stalled Responses

Symptom: The AI response takes a long time to appear or the loading indicator runs indefinitely.

Cause: This can happen for several reasons:

- **Reasoning models** use an extended thinking phase before generating a response. This is expected behavior and can take significantly longer than standard models. See Using Reasoning Models.
- **High model load** on the provider side can cause delays, especially with shared or rate-limited endpoints.
- **Large conversation context** can increase processing time as the model processes more tokens.

Solution:

- If using a reasoning model, wait for the response to complete. The thinking phase may take 30 seconds or longer for complex prompts.
- For standard models, try starting a new conversation to reduce context size.
- If the issue persists, check with your administrator whether the provider endpoint is under heavy load.

File Upload Failures

Symptom: A file upload fails or you see an error when attaching a file to a message.

Cause: The file may exceed size limits or be in an unsupported format.

Solution:

- Check the file size. Regular files have a maximum size of 25 MB, while documents have a maximum size of 100 MB.
- Verify that the file type is supported. Common supported types include text files, code files, PDFs, images, and office documents.
- Try uploading a smaller file or splitting large files into smaller parts.
- See Attaching Files for detailed information on supported formats and size limits.

Model Not Listed

Symptom: A specific model you expect to see is not available in the model dropdown.

Cause: The model may not be deployed on the provider, or the provider configuration may not include it.

Solution:

- Verify with your administrator that the desired model is deployed and available on the provider endpoint.
- For Azure OpenAI providers, confirm that the model deployment name is correctly configured.
- For custom providers, ensure the endpoint's `/v1/models` response includes the model you are looking for.
- Check that you have permission to use the provider that hosts the model.

CLI Chat Issues

Symptom: The `pw ai chats new` command fails or cannot connect.

Cause: Authentication, network, or model identifier issues.

Solution:

- Verify you are authenticated: run `pw auth login` if needed.
- Confirm the model identifier is correct. It must use the format `owner:provider/model-name` (for example, `me:my-provider/gpt-4o`). Run `pw ai models ls` to see available models.
- Check your network connection to the ACTIVATE platform.
- See Using the CLI for detailed setup instructions.

Getting Help

If the solutions above do not resolve your issue:

1. Check the AI Chat overview for general guidance on available features.
2. Review the AI Chat Providers documentation to verify your provider configuration.
3. Contact your organization administrator for platform-specific assistance.
4. Reach out to Parallel Works support for further troubleshooting.

Related Documentation

- Getting Started with AI Chat - First-time setup walkthrough
- Attaching Files - File upload details and limits
- Using Reasoning Models - Understanding reasoning model behavior
- AI Chat Providers - Provider configuration and management

AI Chat Providers

AI Chat Providers are the model backends that power AI Chat in ACTIVATE. Each provider connects the platform to a large language model endpoint, allowing users to send messages and receive AI-generated responses.

ACTIVATE supports three types of providers:

- **Azure OpenAI** — A managed provider provisioned through your organization's Azure infrastructure. ACTIVATE handles deployment and configuration automatically.
- **Custom OpenAI-Compatible** — Any endpoint that implements the OpenAI chat completions API. Use this to connect services like vLLM, Ollama, or other compatible inference servers.
- **Session Tunnels** — A secure tunnel from the platform to a model running inside an ACTIVATE compute session. Useful for private or fine-tuned models running on your own hardware.

> **PERMISSIONS REQUIRED**
>
> Creating and managing AI Chat providers requires appropriate platform permissions. In most organizations, providers are set up by administrators or team leads and then shared with other users.

Provider Permissions

Each provider has its own access controls. The provider owner and organization administrators can share a provider with specific groups or the entire organization using the `aiprovider:use` permission. Users who have been granted access can select the shared provider in AI Chat.

See Managing Permissions for details on sharing and revoking access.

Next Steps

- Azure OpenAI — Set up a managed Azure OpenAI provider
- Custom OpenAI-Compatible — Connect any OpenAI-compatible endpoint
- Session Tunnels — Route chat through a model in a compute session
- Managing Permissions — Control who can use your providers

Related Documentation

- AI Chat — Using the chat interface
- Adding an ML Resource — Provisioning ML workspaces

Azure OpenAI

The Azure OpenAI provider is a managed integration that provisions an Azure OpenAI deployment through your organization's cloud infrastructure. ACTIVATE handles the deployment lifecycle, so you do not need to manually configure Azure resources.

Prerequisites

Before creating an Azure OpenAI provider, ensure that:

- Your organization has at least one **provisioned network** in ACTIVATE
- You have a **billing group** associated with your account
- Your organization's Azure subscription supports Azure OpenAI deployments in your target region

Creating a Provider

1. Navigate to **AI Chat Providers** in the ACTIVATE sidebar.
2. Click **Add Provider**.
3. Select **Azure** as the cloud service provider.
4. Fill in the required fields:
   - **Name** — A descriptive name for the provider (for example, "Azure GPT-4o").
   - **Network** — Select the provisioned network to deploy with.
   - **Group** — Select the billing group to associate with this provider.
   - **Region** — The Azure region for the deployment (for example, "eastus2").
   - **Model** — The model to deploy (for example, "gpt-4o").
5. Optionally add a **Description** and **Tags**.
6. Click **Create**.

The provider will enter a **provisioning** status while ACTIVATE configures the Azure resources. Once provisioning completes, the status changes to **provisioned** and the provider is ready to use in AI Chat.

Verifying the Provider

After provisioning completes:

1. Open **AI Chat**.
2. Select your new Azure provider from the provider dropdown.
3. Send a test message to confirm the model responds.

If the provider does not appear in the dropdown, verify that your user account has access. See Managing Permissions for details.

Updating a Provider

Azure providers are managed by the platform. The model and deployment configuration are set during provisioning. To change the model or region, delete the existing provider and create a new one.

Deleting a Provider

1. Navigate to **AI Chat Providers**.
2. Find the provider you want to remove.
3. Click the delete action for that provider.
4. Confirm the deletion.

Deleting an Azure provider tears down the associated Azure resources. This action cannot be undone.

Related Documentation

- AI Chat Providers Overview — Overview of all provider types
- Managing Permissions — Sharing providers with your team
- Custom OpenAI-Compatible — Connecting non-Azure endpoints

Custom OpenAI-Compatible

A custom provider connects ACTIVATE AI Chat to any endpoint that implements the OpenAI chat completions API. This gives you flexibility to use self-hosted models, third-party inference services, or any other compatible backend.

Endpoint Requirements

Your endpoint must support the following:

- **POST** `/v1/chat/completions` — The standard OpenAI chat completions endpoint. ACTIVATE sends all chat messages to this route.
- **GET** `/v1/models` (optional) — If your endpoint exposes this route, ACTIVATE can list available models dynamically so users can switch models without reconfiguring the provider.

The endpoint must accept an API key via the `Authorization: Bearer <key>` header.

Prerequisites

Before creating a custom provider, ensure that:

- Your inference endpoint is running and reachable from the ACTIVATE platform
- You have the endpoint URL and a valid API key

Creating a Provider

1. Navigate to **AI Chat Providers** in the ACTIVATE sidebar.
2. Click **Add Provider**.
3. Select **Custom** as the cloud service provider.
4. Fill in the required fields:
   - **Name** — A descriptive name for the provider (for example, "vLLM Llama 3").
   - **Endpoint** — The full URL to your chat completions endpoint (for example, `https://my-server.example.com/v1/chat/completions`).
   - **API Key** — The authentication key for your endpoint.
5. Optionally add a **Description** and **Tags**.
6. Click **Create**.

Custom providers are available immediately after creation — no provisioning step is required.

Verifying the Provider

1. Open **AI Chat**.
2. Select your new custom provider from the provider dropdown.
3. Send a test message to confirm the endpoint responds correctly.

ACTIVATE automatically discovers available models by querying your endpoint's `/v1/models` route. If the model list dropdown is populated, your endpoint supports this route and ACTIVATE has successfully queried it.

Updating a Provider

You can update the endpoint and API key for a custom provider at any time:

1. Navigate to **AI Chat Providers**.
2. Select the provider you want to update.
3. Modify the fields as needed.
4. Save your changes.

This is useful for rotating API keys or pointing the provider at a different endpoint without creating a new provider.

Notes for Common Inference Servers

vLLM — Start vLLM with the `--api-key` flag to enable API key authentication. The default port is 8000. Your endpoint URL will be `http://<host>:8000/v1/chat/completions`.

Ollama — Ollama exposes an OpenAI-compatible API on port 11434. Set the endpoint to `http://<host>:11434/v1/chat/completions`. The model name should match the Ollama model tag (for example, "llama3").

Text Generation Inference (TGI) — TGI provides an OpenAI-compatible Messages API. Set the endpoint to `http://<host>:8080/v1/chat/completions`.

Deleting a Provider

1. Navigate to **AI Chat Providers**.
2. Find the provider you want to remove.
3. Click the delete action for that provider.
4. Confirm the deletion.

Deleting a custom provider only removes it from ACTIVATE. It does not affect the external inference endpoint.

Related Documentation

- AI Chat Providers Overview — Overview of all provider types
- Managing Permissions — Sharing providers with your team
- Session Tunnels — Tunneling to a model running in a compute session

Session Tunnels

A session tunnel provider routes AI Chat traffic through a secure tunnel to a model running inside an ACTIVATE compute session. This allows you to use models hosted on your own cluster hardware without exposing them to the public internet.

How Session Tunnels Work

When you start a session in ACTIVATE with an OpenAI-compatible model server running on it, you can create a tunnel that makes the model available to AI Chat. The tunnel establishes a secure connection between the ACTIVATE platform and the model endpoint inside your session.

Session tunnel providers appear in the AI Chat provider list with the type `openai-tunnel`. They are automatically discovered when a tunnel with the OpenAI flag enabled is active for your user account.

Prerequisites

Before using a session tunnel provider:

- You have an active ACTIVATE session running on a cluster
- An OpenAI-compatible model server is running inside the session (for example, vLLM, Ollama, or TGI)
- A tunnel has been created from the session with the **OpenAI** option enabled

Setting Up a Session Tunnel

1. Start a **session** on your cluster with enough resources to run your model (GPU, memory).
2. Inside the session, start an OpenAI-compatible inference server on a known port.

3. Create a **tunnel** for that session and enable the **OpenAI** toggle.
4. Once the tunnel status shows as **running**, the provider appears automatically in your AI Chat provider list.

You do not need to manually create a provider entry. ACTIVATE detects tunnels with the OpenAI flag and includes them in your available providers.

Using a Session Tunnel Provider

1. Open **AI Chat**.
2. In the provider dropdown, look for your tunnel. It will show with the status of the tunnel (for example, "running" or "stopped").
3. Select the tunnel provider and send a message.

If the tunnel is in a "stopped" state, you must restart the session and tunnel before it can serve requests.

Use Cases

- **Air-gapped environments** — Run models on infrastructure that has no direct internet access. The tunnel provides a secure path from ACTIVATE to the model.
- **Sensitive data** — Keep model inference on your own hardware to ensure data does not leave your environment.
- **Custom fine-tuned models** — Serve models you have fine-tuned on proprietary data without uploading them to a cloud provider.
- **Experimentation** — Quickly test different models by starting sessions with different configurations.

Limitations

- **Session dependency** — The tunnel provider is only available while the session and tunnel are active. If the session stops, the provider becomes unavailable.
- **No sharing** — Session tunnel providers are tied to the user who created the tunnel. They cannot be shared with other users via the permissions system.
- **Single user** — Each tunnel is associated with one user account.

Related Documentation

- AI Chat Providers Overview — Overview of all provider types
- Custom OpenAI-Compatible — Connecting a persistent external endpoint
- Sessions — Managing compute sessions

Managing Permissions

AI Chat providers support a permission system that lets you control who can use your providers. By default, a provider is private to its owner. You can share it with specific groups or your entire organization.

Permission Model

ACTIVATE uses the `aiprovider:use` permission to control access to AI Chat providers. This is the only permission level for providers — granting it allows the recipient to select the provider in AI Chat and send messages through it.

Access to a provider is determined by the following rules, evaluated in order:

1. **Owner** — The user who created the provider always has full access, including the ability to manage permissions.
2. **Organization administrators** — Org admins can access and manage permissions for any provider in the organization.
3. **Granted permissions** — Users who belong to a group that has been granted `aiprovider:use` on the provider, or users in an organization where the provider is shared organization-wide.

Who Can Manage Permissions

Only the provider **owner** and **organization administrators** can view or modify provider permissions. Other users with `aiprovider:use` access can use the provider but cannot change its sharing settings.

Sharing a Provider with a Group

1. Navigate to **AI Chat Providers**.
2. Select the provider you want to share.
3. Open the **Permissions** settings for the provider.
4. Under **Groups**, find the group you want to share with.
5. Enable the `aiprovider:use` permission for that group.
6. Save your changes.

All members of the selected group will now see the provider in their AI Chat provider list.

Sharing a Provider with the Entire Organization

1. Navigate to **AI Chat Providers**.
2. Select the provider you want to share.
3. Open the **Permissions** settings for the provider.
4. Under **Organization**, enable the `aiprovider:use` permission.
5. Save your changes.

Every user in your organization will now have access to this provider.

Revoking Access

To remove access for a group or the organization:

1. Open the **Permissions** settings for the provider.
2. Disable the `aiprovider:use` permission for the group or organization.
3. Save your changes.

Users who lose access will no longer see the provider in their AI Chat provider list. Existing conversations that used the provider will remain, but users will not be able to send new messages through it.

Best Practices

- **Start with group-level sharing.** Share providers with specific teams that need them rather than the entire organization. This limits unnecessary usage and cost.
- **Use descriptive provider names.** When sharing across groups, a clear name like "Azure GPT-4o (East US)" helps users identify the right provider.
- **Audit permissions periodically.** Review which groups have access to each provider, especially for providers connected to paid endpoints.
- **Rotate API keys regularly.** For custom providers, update the API key in the provider settings when you rotate credentials on the backend.

> **SESSION TUNNELS**
>
> Session tunnel providers do not support the permissions system. They are private to the user who created the tunnel.

Related Documentation

- AI Chat Providers Overview — Overview of all provider types
- Azure OpenAI — Setting up managed Azure providers
- Custom OpenAI-Compatible — Connecting external endpoints

Best Practices

This page covers recommended practices for getting the most out of AI Chat on the ACTIVATE platform.

Writing Effective Prompts

The quality of AI responses depends heavily on how you frame your requests.

Be Specific

Vague prompts produce vague answers. Include relevant details, constraints, and the desired format in your prompt.

Instead of	Try
"Explain Kubernetes"	"Explain how Kubernetes pod scheduling works, including the role of the scheduler and node affinity rules, in about 200 words."
"Write a script"	"Write a Bash script that finds all CSV files in a directory and combines them into a single file with one header row."

Provide Context

Give the model the background it needs to produce a useful response:

- Describe the environment or system you are working with.
- Mention relevant constraints (language, framework, version).
- Reference earlier parts of the conversation if building on previous responses.

Use System Messages

If your provider supports system messages, use them to set the model's behavior for the entire conversation. For example, you can instruct the model to respond as a domain expert, use a particular coding style, or avoid certain topics.

Iterate and Refine

If the first response is not quite right, refine your prompt rather than starting from scratch. Add clarifications, ask for a different format, or request that the model focus on a specific aspect.

Choosing the Right Model

Different models have different strengths. Selecting the right model for your task improves both response quality and efficiency.

Standard Models

Standard models (such as GPT-4o, GPT-4o-mini) are best for:

- General-purpose questions and conversation
- Text summarization and generation
- Code generation for straightforward tasks
- Tasks where speed matters more than deep analysis

Reasoning Models

Reasoning models (such as o1, o3) are best for:

- Complex math, logic, and multi-step problem-solving
- Detailed code review and debugging
- Architectural decisions that require evaluating tradeoffs
- Tasks where accuracy is more important than speed

See Using Reasoning Models for details on reasoning effort configuration.

Cost Considerations

Larger and more capable models consume more tokens and may have higher usage costs. When working on routine tasks, consider using a smaller or faster model to conserve resources. Reserve more powerful models for tasks that genuinely benefit from their capabilities.

Working with File Attachments

File attachments allow you to provide the model with additional context beyond what fits in a text prompt.

Keep Documents Focused

Rather than uploading a large document and asking a broad question, extract the relevant section or upload a smaller, targeted file. This helps the model focus on the content that matters and reduces token usage.

Use the Right File Type

- **Code files** -- Attach source code files directly for code review, debugging, or analysis tasks.
- **Text and Markdown** -- Use plain text or Markdown for structured information, notes, or specifications.
- **PDFs and documents** -- Upload reports, papers, or documentation that the model can reference when answering your questions.

Be Aware of Size Limits

Regular files have a maximum size of 25 MB, and documents have a maximum size of 100 MB. If a file exceeds these limits, split it into smaller parts or extract the relevant sections.

See Attaching Files for supported formats and detailed usage instructions.

Collaboration Tips

AI Chat's sharing and branching features support team workflows.

Share with Appropriate Permissions

- Use **View** permission for stakeholders who need to review the conversation without modifying it.
- Use **Collaborate** permission for team members who need to actively contribute messages.
- Share at the group level rather than managing individual access, so permissions stay current as team membership changes.

See Sharing Conversations for detailed instructions.

Use Branching for Exploration

When evaluating different approaches, create branches from the same message rather than cluttering a single thread. Each branch maintains its own context, allowing you to compare model responses to different prompts or explore alternative solutions side by side.

See Branching Conversations for details.

Document Decisions

Use the conversation itself as a record of your analysis. When you reach a conclusion, summarize the decision in a follow-up message so that anyone reviewing the shared conversation can quickly understand the outcome.

Security Considerations

Sensitive Data

Be mindful of the data you include in prompts and file attachments. Messages are sent to the configured AI provider endpoint, which may be hosted externally.

- For sensitive or proprietary data, use a **session tunnel** provider that routes requests to a model running on your own compute infrastructure. See Session Tunnels.
- Avoid including credentials, API keys, or personally identifiable information in prompts.

API Key Management

If you manage AI Chat providers, follow these practices for API key security:

- **Rotate keys regularly** -- Update API keys in the provider configuration on a regular schedule.
- **Use the minimum required permissions** -- Configure provider API keys with only the permissions needed for chat completions.
- **Monitor usage** -- Review token usage and access patterns through your provider's dashboard to detect unexpected activity.

Provider Permissions

Limit provider access to the groups and users who need it. Review and audit provider permissions periodically to ensure that only authorized teams have access.

See Managing Provider Permissions for details.

Related Documentation

- AI Chat Overview - Feature summary and capabilities
- Getting Started with AI Chat - First-time setup walkthrough
- AI Chat Providers - Provider configuration and management
- Troubleshooting - Common issues and solutions

Kubernetes

ACTIVATE provides built-in Kubernetes cluster management that lets you connect external clusters, manage workloads, deploy Helm charts, track costs, and control resource quotas — all from a single interface.

Key Capabilities

- **Connecting Clusters** — Register external Kubernetes clusters and share them with your groups
- **kubectl Access** — Authenticate to clusters using the PW CLI with federated OIDC credentials
- **Managing Workloads** — View, edit, and delete Deployments, StatefulSets, Jobs, Pods, and more across clusters
- **Pod Logs** — Stream and search container logs in real time
- **Workload Metrics** — Monitor CPU, memory, and storage usage with built-in charts
- **Helm Charts** — Install, upgrade, roll back, and uninstall Helm releases
- **Resource Quotas** — Set CPU, memory, and GPU limits per namespace
- **Cost Tracking** — Calculate and monitor per-namespace compute costs
- **Services & Storage** — Inspect services, persistent volumes, ConfigMaps, and Secrets
- **Nodes & GPUs** — View node capacity and configure NVIDIA MIG GPU profiles

Architecture

ACTIVATE connects to your Kubernetes clusters using OIDC token federation. When a user accesses a cluster through the UI or CLI:

1. ACTIVATE issues a short-lived OIDC token (10-minute TTL) containing the user's identity and group memberships
2. The target cluster validates the token against ACTIVATE's OIDC provider
3. Kubernetes RBAC policies map the user's groups to cluster roles and namespace access

This approach requires no long-lived credentials to be stored and ensures that access is always scoped to the user's current permissions and group memberships.

> **OIDC REQUIREMENT**
>
> Clusters must be configured to trust ACTIVATE as an OIDC identity provider. See Connecting Clusters for setup details.

Getting Started

1. **Connect a cluster** — Add your first Kubernetes cluster in the cluster connection page
2. **Set up CLI access** — Install the PW CLI and run `pw kube auth` to configure kubectl access
3. **Explore workloads** — Browse running workloads in the workloads view

Connecting Clusters

Parallel Works lets you bring your own Kubernetes clusters and manage them through a unified interface. Once connected, you can deploy workloads, manage access through groups, and monitor resources across all of your clusters from a single pane of glass.

Prerequisites

Before connecting a cluster, make sure you have the following:

- **Cluster endpoint URL** — The public HTTPS endpoint of your Kubernetes API server (e.g. `https://k8s.example.com:6443`).
- **CA certificate** — The Certificate Authority (CA) certificate for the cluster. This is used to establish a trusted TLS connection to the API server.
- **Network connectivity** — The Parallel Works platform must be able to reach your cluster endpoint over the network. Make sure that any firewalls or security groups allow inbound connections to the API server from the platform.
- **Organization admin permissions** — Only organization admins can add or modify clusters.

> **OIDC REQUIREMENT**
>
> Your cluster must be configured to accept OIDC-based authentication. Parallel Works uses OIDC token federation to authenticate users against connected clusters. See Security Model below for details.

Adding a Cluster

1. Navigate to **Kubernetes > Clusters** in the sidebar.
2. Click **Add Cluster** to open the cluster registration drawer.
3. Fill in the required fields:
   - **Cluster Name** — A lowercase alphanumeric name for the cluster (special characters are removed automatically).
   - **Cluster Endpoint** — The public HTTPS URL of your Kubernetes API server.
   - **CA Certificate** — Paste the full PEM-encoded CA certificate for your cluster.
4. Click **Add Cluster** to save.

After the cluster is added, you are redirected to the cluster definition page where you can verify the connection and configure access.

> **NAMING RESTRICTIONS**
>
> The cluster name cannot contain special characters. Only lowercase letters and numbers are allowed (e.g. `prodcluster01`).

Verifying Connectivity

Once a cluster is added, Parallel Works verifies the connection by querying the Kubernetes API for the cluster's node list. On the **Clusters** page, each connected cluster displays:

- **vCPUs** — Total CPU capacity across all nodes.
- **Memory** — Total memory capacity across all nodes.

If the platform cannot reach the cluster, the cluster still appears in the list but shows zero resources. This typically indicates a network connectivity issue or an invalid CA certificate.

> **CONNECTIVITY CHECK**
>
> If a cluster shows 0 CPUs and 0 memory, double-check that the endpoint is reachable from the platform and that the CA certificate is correct.

Sharing Clusters with Groups

Organization admins can share clusters with specific groups to control who has access. When a cluster is shared with a group:

1. A **Kubernetes namespace** is automatically created on the cluster, named after the group.
2. A **RoleBinding** is created in that namespace granting the group `admin`-level access within the namespace.
3. Members of the group can then view and manage resources within their namespace.

To share a cluster:

1. Navigate to the cluster's **definition** page.
2. Select the groups you want to share the cluster with.
3. Save the updated configuration.

When a group is removed from a cluster, the associated RoleBinding is deleted but the namespace is preserved to prevent accidental data loss.

> **GROUP PREREQUISITES**
>
> Groups must be created in the organization before they can be shared with a cluster. The reserved group name `cluster-admins` cannot be used as a regular group name — it is managed internally by the platform.

Multi-Cluster Management

The Kubernetes section provides a unified view across all connected clusters. From the sidebar, you can access:

- **Clusters** — View and manage all connected clusters.
- **Nodes** — Inspect node details and GPU operator configuration (admin only).

- **Workloads** — List deployments, stateful sets, daemon sets, replica sets, jobs, cron jobs, and pods across clusters.
- **Services** — View services across clusters.
- **Storages** — Manage PersistentVolumeClaims and PersistentVolumes.
- **ConfigMaps & Secrets** — Browse configuration resources.
- **Helm Charts** — Install, upgrade, rollback, and manage Helm releases.
- **Quotas** — Set resource quotas per namespace (admin only).

All resource views support filtering by cluster and namespace, making it straightforward to manage resources across a large fleet of clusters.

Security Model

Parallel Works uses **OIDC token federation** to authenticate users against connected Kubernetes clusters. This eliminates the need to distribute long-lived kubeconfig files or service account tokens.

How It Works

1. When a user accesses a connected cluster, the platform generates a short-lived **OIDC JWT token** on behalf of the user.
2. The token is signed by the platform's OIDC issuer (`https://<platform-host>/api/oidc`).
3. The token is scoped to the specific cluster (using the cluster ID as the JWT audience).
4. The connected cluster validates the token using the platform's OIDC discovery endpoint and public keys.

Token Properties

Property	Value
TTL	10 minutes
Audience	The specific cluster ID
Subject	`user:<username>`
Issuer	`https://<platform-host>/api/oidc`

Group-Based Access Control

Each OIDC token includes group claims that map to Kubernetes RBAC:

- `cluster-scope` — Included in every token. Grants read-only access to cluster-scoped resources like storage classes and runtime classes via a dedicated ClusterRole.
- `cluster-admins` — Included for organization admins and platform admins. Grants full administrative access to the cluster.
- **Group names** — If the user belongs to groups that the cluster is shared with, those group names are included as claims. These map to namespace-scoped RoleBindings that grant `admin` access within the corresponding namespace.

This model ensures that regular users can only access namespaces belonging to their groups, while organization admins retain full cluster-level access. All tokens are short-lived, reducing the risk associated with credential leakage.

CoreWeave

This guide walks through connecting a CoreWeave Kubernetes cluster to ACTIVATE so you can manage its workloads, storage, and access through the platform. It builds on the generic Connecting Clusters flow with the specifics of CoreWeave's console, endpoint naming, and OIDC configuration.

> **TWO WAYS TO CONNECT COREWEAVE**
>
> CoreWeave can be connected in two distinct ways. This guide covers connecting it as a **Kubernetes provider**. If you instead want to attach a CoreWeave **Slurm** cluster (SUNK) so users can submit batch jobs, see Connecting a CoreWeave Slurm Cluster.

Prerequisites

- **Organization admin permissions** in ACTIVATE - only org admins can add clusters.
- **Access to the CoreWeave Cloud Console** for the target cluster, including its kubeconfig.
- **Network reachability** - ACTIVATE must be able to reach the cluster's Kubernetes API server over HTTPS.

Download the kubeconfig

In the CoreWeave Cloud Console, open the cluster, click the ⋮ (three dots) menu, and download the kubeconfig file.

You can sanity-check it by prefixing `kubectl` with the downloaded file:

```
KUBECONFIG="CWKubeconfig_ue01a" kubectl get pods
```

Determine the API endpoint

The CoreWeave console shows the cluster's endpoint, for example:

```
mycluster-1a2b3c4d.k8s.us-east-01a.coreweave.com
```

ACTIVATE connects to the Kubernetes **API server**, which is reached by prefixing the endpoint with `api.`:

```
api.mycluster-1a2b3c4d.k8s.us-east-01a.coreweave.com
```

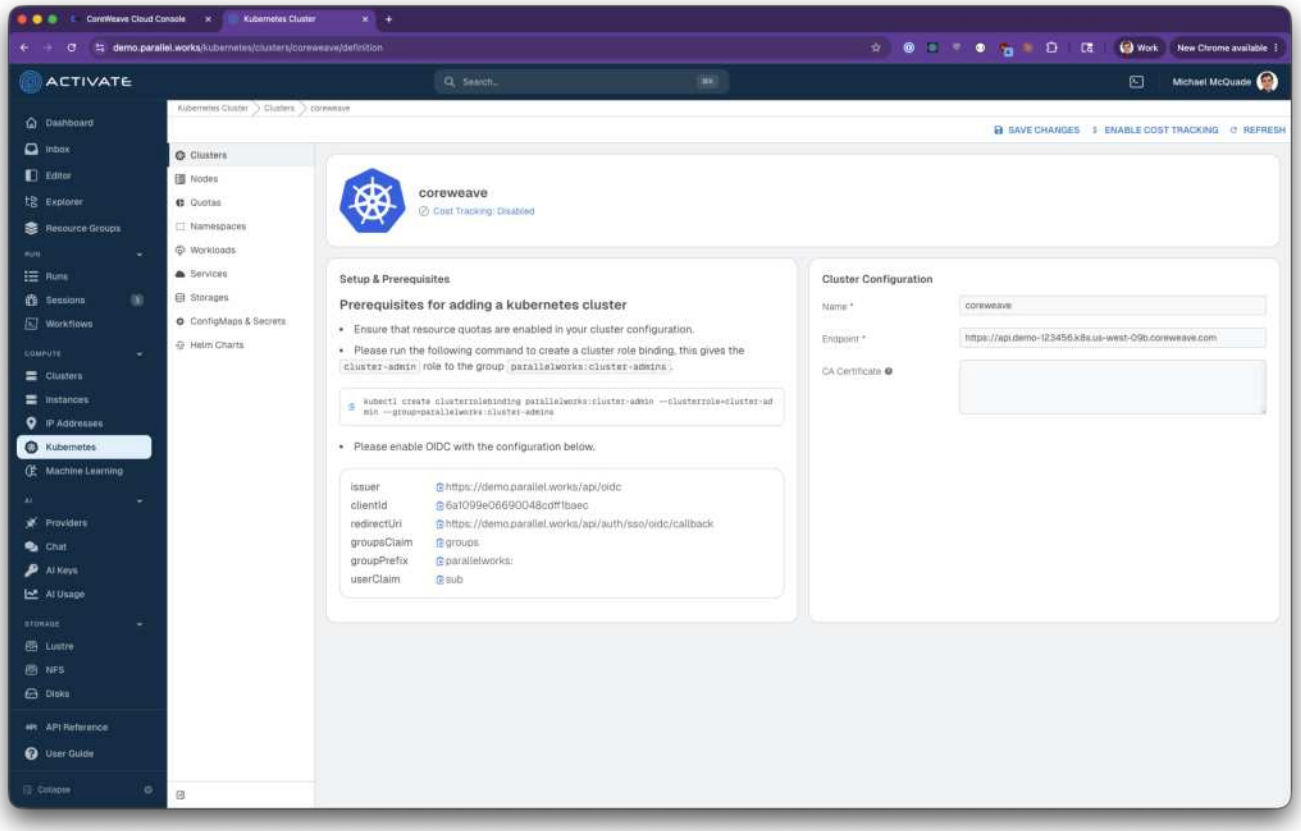
You will enter this as an HTTPS URL (`https://api.<endpoint>`) when configuring the cluster.

Add the cluster in ACTIVATE

1. Navigate to **Kubernetes → Clusters** in the sidebar.
2. Click **Add Cluster**.
3. Fill in the registration form:

- **Cluster Name** - a lowercase alphanumeric name (e.g. `coreweave`).
- **Cluster Endpoint** - the `https://api.<endpoint>` URL derived above.
- **CA Certificate** - leave this blank. CoreWeave's API server presents a publicly-trusted certificate, so ACTIVATE validates it against the system trust store with no custom CA bundle.

4. Save. You are taken to the cluster's **Definition** page.



Complete the cluster prerequisites

The **Definition** page shows a **Setup & Prerequisites** panel. Complete each item against your CoreWeave cluster:

Enable resource quotas

Ensure resource quotas are enabled in your CoreWeave cluster configuration. ACTIVATE relies on namespace quotas to track and bound per-group usage.

Create the cluster-admin role binding

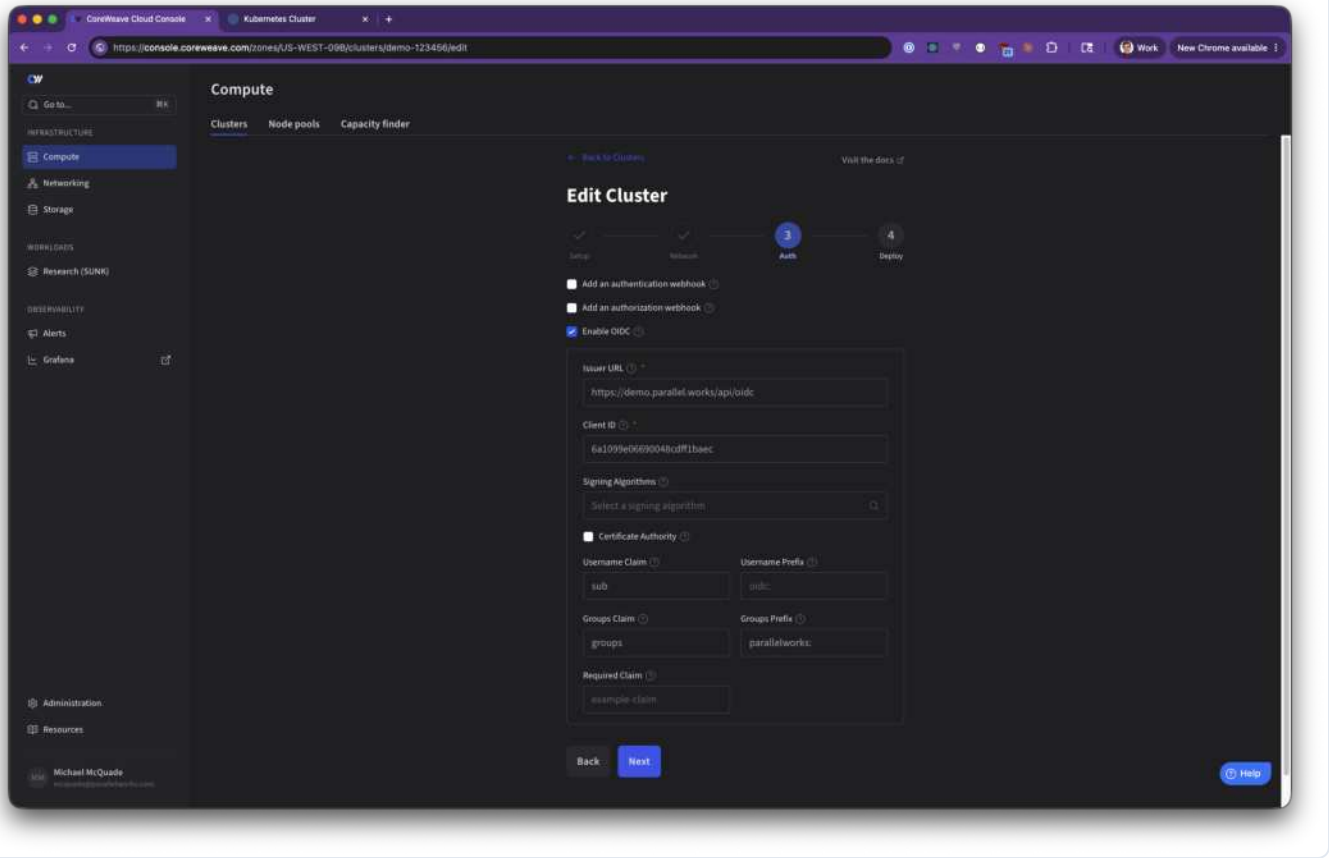
Using the kubeconfig you downloaded, grant the `cluster-admin` role to the `parallelworks:cluster-admins` group:

```
kubectl create clusterrolebinding parallelworks:cluster-admin \
  --clusterrole=cluster-admin \
  --group=parallelworks:cluster-admins
```

This is what lets ACTIVATE org and platform admins administer the cluster once OIDC is wired up.

Configure OIDC on CoreWeave

ACTIVATE authenticates users to the cluster with short-lived OIDC tokens rather than long-lived kubeconfigs. The **Setup & Prerequisites** panel displays the exact values to use - copy each into the matching field of CoreWeave's **Enable OIDC** configuration (found in the cluster's **Auth** step in the CoreWeave Cloud Console).



ACTIVATE value	Example	CoreWeave field
`issuer`	`https://<platform-host>/api/oidc`	**Issuer URL**
`clientId`	your cluster's ID	**Client ID**
`userClaim`	`sub`	**Username Claim**
`groupsClaim`	`groups`	**Groups Claim**
`groupPrefix`	`parallelworks:`	**Groups Prefix**

Additional CoreWeave fields:

- **Signing Algorithms** - `RS256` (the only algorithm ACTIVATE advertises in its OIDC discovery document at `https://<platform-host>/api/oidc/.well-known/openid-configuration`).
- **Username Prefix** - leave empty unless your environment requires one.

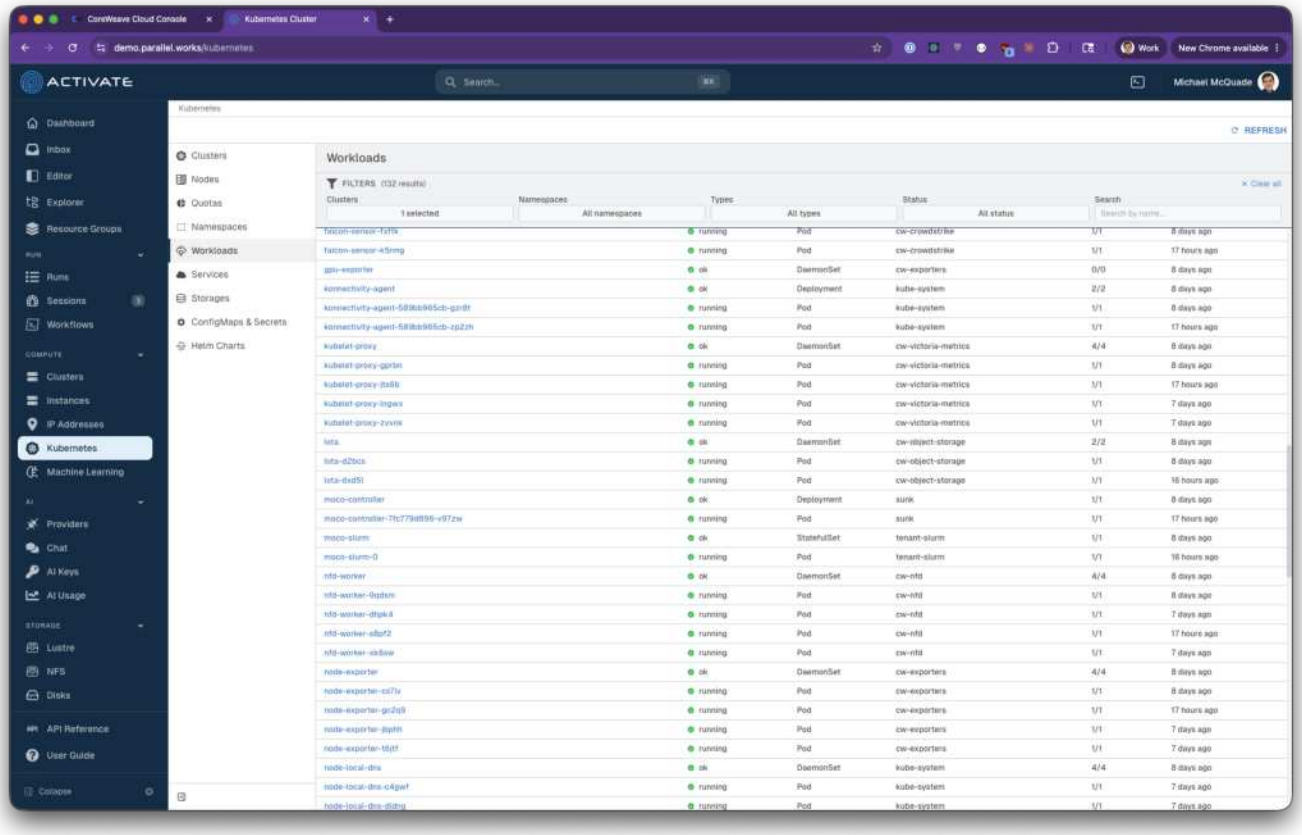
- **Certificate Authority** - not required; ACTIVATE's OIDC issuer presents a publicly trusted certificate.

> **WHY THE GROUP PREFIX MATTERS**
>
> The `parallelworks:` group prefix is what ties the `cluster-admin` role binding from the previous step to ACTIVATE's identity tokens. ACTIVATE emits the `parallelworks:cluster-admins` group claim for admins, and CoreWeave prepends the prefix to every group it receives - so the binding and the claim line up.

Verify the connection

Back on the **Clusters** page, a successfully connected cluster reports its total **vCPUs** and **Memory**. If it shows zeros, re-check the endpoint URL and that the API server is reachable from ACTIVATE.

Once connected, you can browse the cluster's live workloads under **Kubernetes → Workloads**:



Share the cluster with groups

To give your teams access, share the cluster with the appropriate ACTIVATE groups. Sharing automatically provisions a namespace and a namespace-scoped role binding for each group. See Sharing Clusters with Groups for details.

kubectl Access

Use the PW CLI to authenticate with Kubernetes clusters managed by Parallel Works. The CLI handles kubeconfig generation, certificate management, and OIDC token-based authentication so you can use `kubectl` against your clusters without manual configuration.

Prerequisites

- The PW CLI installed and configured
- `kubectl` installed on your local machine
- Access to at least one Kubernetes cluster in your organization

Listing Available Clusters

To see which Kubernetes clusters you have access to, run:

```
pw kube ls
```

This lists all clusters available in your organization. You can control the output format:

```
# Output as JSON
pw kube ls -o json

# Output as a table (shows name, total vCPUs, total memory)
pw kube ls -o table
```

> **QUICK OVERVIEW**
>
> Use `pw kube ls -o table` for a quick overview of cluster capacity.

Authenticating with a Cluster

The `pw kube auth` command configures your local kubeconfig with everything needed to connect to a cluster:

```
pw kube auth <cluster-name>
```

For example, to set up access to a cluster named **demo**:

```
pw kube auth demo
```

This command performs the following steps automatically:

1. Retrieves the cluster's CA certificate and API server endpoint from the platform
2. Adds or updates a cluster entry in your kubeconfig with the CA certificate embedded
3. Configures user credentials using the OIDC exec-credential plugin (calls `pw kube token` automatically when `kubectl` needs a token)

4. Creates a context named `pw#<cluster-name>` and switches to it

After running `pw kube auth`, you can immediately use `kubectl`:

```
kubectl get namespaces
kubectl get pods -n <namespace>
```

Options

Flag	Description
`--no-context-switch`	Configure the cluster without switching your active kubectl context

> **CONTEXT NAMING**
>
> The context is named `pw#<cluster-name>`. You can switch between contexts with `kubectl config use-context pw#<cluster-name>`.

Manual Token Generation

If you need a raw OIDC token (for example, to use with a custom tool or API client), use:

```
pw kube token <cluster-name>
```

This outputs an `ExecCredential` JSON object compatible with the Kubernetes client-go credential plugin protocol:

```
{
  "apiVersion": "client.authentication.k8s.io/v1",
  "kind": "ExecCredential",
  "status": {
    "token": "<oidc-token>"
  }
}
```

> **TOKEN EXPIRY**
>
> Tokens expire after **10 minutes**. The exec-credential plugin handles renewal automatically when you use `pw kube auth`, but if you use tokens manually you will need to regenerate them before they expire.

How Authentication Works

Parallel Works uses OIDC (OpenID Connect) tokens to authenticate users with Kubernetes clusters. Here is how the flow works:

1. When you run `pw kube auth <cluster-name>`, the CLI configures kubectl to use `pw kube token <cluster-name>` as an exec-based credential plugin.
2. Each time `kubectl` makes a request, it invokes `pw kube token` to obtain a fresh OIDC token.
3. The Kubernetes API server validates the token against the platform's OIDC provider.

Token Claims

Each OIDC token contains the following claims:

Claim	Value
`sub` (subject)	`user:<username>`
`aud` (audience)	The cluster ID
`iss` (issuer)	`https://<platform-host>/api/oidc`
`groups`	List of group names the user belongs to, plus `cluster-scope`
`platform_host`	The platform hostname
`exp` (expiry)	10 minutes from issuance

Namespace Access

Access to namespaces is determined by your group memberships:

- Each group that has been granted access to a cluster gets a corresponding namespace (named after the group).
- You can access namespaces for any group you belong to that is shared with the cluster.
- Organization admins have access to all namespaces on the cluster.

To see which namespaces are available to you:

```
kubectl get namespaces
```

Admin vs Regular User Permissions

	Regular User	Organization Admin
Cluster groups	`cluster-scope` + group names	`cluster-scope` + `cluster-admins` + group names
Namespace access	Only namespaces matching their group memberships	All namespaces
Use case	Day-to-day workload management within group namespaces	Full cluster administration

Organization admins are automatically added to the `cluster-admins` group, which grants elevated permissions across the cluster.

Troubleshooting

"kubectl not found"

The `pw kube auth` command requires `kubectl` to be installed and available on your `PATH`. Install it by following the official Kubernetes documentation.

Token Errors

If you see authentication errors when running `kubectl` commands:

1. Verify that you are logged in to the PW CLI (`pw auth status`).
2. Re-run `pw kube auth <cluster-name>` to refresh your kubeconfig.
3. Confirm the cluster is still available with `pw kube ls`.

Wrong Context

If `kubectl` is targeting the wrong cluster, check and switch your context:

```
kubectl config current-context
kubectl config use-context pw#<cluster-name>
```

See Also

- CLI Reference: pw kube auth
- CLI Reference: pw kube token
- CLI Reference: pw kube ls

Managing Workloads

The Workloads view provides a unified interface for monitoring and managing Kubernetes workloads across all your connected clusters. You can inspect resource status, view detailed metadata, edit configurations, and delete workloads directly from the platform.

Navigating the Workloads View

Access the workloads view by navigating to **Kubernetes > Workloads** in the sidebar. The page displays a filterable, sortable table of all workloads across your connected clusters.

Supported Workload Types

The platform supports the following Kubernetes workload types:

Type	Description
Deployment	Manages stateless application replicas with rolling updates
StatefulSet	Manages stateful applications with persistent storage and ordered deployment
DaemonSet	Ensures a pod runs on all (or selected) nodes in a cluster
Job	Runs a task to completion with a specified number of completions
CronJob	Schedules Jobs to run on a recurring cron-based schedule

Pod	The smallest deployable unit in Kubernetes
ReplicaSet	Maintains a stable set of replica pods (typically managed by a Deployment)

List View

The workloads list view presents all workloads in a sortable table. Click any column header to sort by that field.

Filtering

The filter bar at the top of the table provides several ways to narrow down the displayed workloads:

- **Cluster** -- Select one or more clusters to show workloads from only those clusters. When no cluster is selected, workloads from all connected clusters are shown.
- **Namespace** -- Filter by one or more Kubernetes namespaces.
- **Type** -- Filter by workload type (Deployment, StatefulSet, DaemonSet, Job, CronJob, Pod).
- **Search** -- Type a name to filter workloads by name. The search is case-insensitive and matches partial names.

Use the **Clear all** button to reset all active filters at once.

Table Columns

Each row in the workloads table displays the following information:

Column	Description
Name	The workload name. Click to open the detail view.
Namespace	The Kubernetes namespace the workload belongs to.
Cluster	The cluster where the workload is running.
Type	The workload type (Deployment, StatefulSet, etc.).
Status	A color-coded status indicator showing the current state of the workload.
Pods	The number of ready pods versus total pods (e.g., `3/3`).
CPU	Current CPU usage, when metrics are available.
Memory	Current memory usage, when metrics are available.
Restarts	Total container restart count across all pods.

Status Indicators

Workload status is shown with a color-coded icon:

- **Ok / Running** -- The workload is healthy and all expected pods are ready.
- **Pending** -- The workload is starting up or waiting for resources.
- **Failed** -- The workload has encountered an error.

- **Completed** -- The workload (Job) has finished successfully.
- **Unknown** -- The status could not be determined.

Detail View

Click on any workload name in the list to open its detail view. The detail page shows comprehensive information about the selected resource.

Resource Metadata

The top panel displays key metadata fields that vary by workload type:

Common fields across all types:

- Name, Namespace, Resource Version, and Creation Timestamp

Deployment-specific fields:

- Replica counts: updated, ready, available, and unavailable

StatefulSet-specific fields:

- Desired replicas, observed generation, associated service name, and persistent volume claim specifications

DaemonSet-specific fields:

- Pod counts: desired, current, ready, available, and unavailable
- Update strategy type

Job-specific fields:

- Parallelism, completions, backoff limit, active/ready/succeeded pod counts, start and completion times
- Job conditions table showing message, reason, type, status, and transition times

CronJob-specific fields:

- Cron schedule expression, last schedule time, concurrency policy, and starting deadline
- Job template specification with parallelism and completion settings

Pod-specific fields:

- Phase, host IP, pod IP, start time, and labels
- Container statuses table with name, ready state, restart count, image, and started state

ReplicaSet-specific fields:

- Owner references (API version, kind, name, UID, controller, block owner deletion)
- Desired, current, and ready replica counts

Pod Template

For Deployments, StatefulSets, DaemonSets, Jobs, and CronJobs, the detail view includes a **Pod Template** section showing:

- Labels applied to pods
- Termination grace period
- Restart policy
- Init containers (if any)
- Containers
- Volumes

Pod List

For workload types that manage pods (Deployments, StatefulSets, DaemonSets, ReplicaSets, and Jobs), the detail view includes a table of associated pods with columns for:

Column	Description
Name	Pod name (links to the pod detail view)
Status	Current pod phase (Running, Pending, etc.)
Ready	Ready containers vs. total containers (e.g., `1/1`)
Restarts	Total restart count across all containers
Age	Time since the pod was created

ReplicaSet History

For **Deployments**, the detail view includes a ReplicaSet history table showing all associated ReplicaSets sorted by revision number:

Column	Description
Revision	Deployment revision number (links to the ReplicaSet detail view)
Name	ReplicaSet name
Desired	Desired replica count
Current	Current replica count
Ready	Number of ready replicas
Age	Time since the ReplicaSet was created

Viewing JSON and YAML

The detail view provides both a structured view and a raw representation of the resource:

- The structured view displays the resource metadata and status in a readable format.
- Click **View / Edit YAML** to open a side drawer containing the full YAML representation of the resource.
- Use the **Copy** button in the YAML drawer to copy the resource definition to your clipboard.

Editing Resources

You can edit any workload resource directly from the platform:

1. Navigate to the workload detail view.
2. Click **View / Edit YAML** to open the YAML editor drawer.
3. Modify the YAML in the built-in editor.
4. Click **Update resource** to apply the changes.

The platform sends the updated YAML to the Kubernetes API, which validates and applies the changes. If the update fails (for example, due to a validation error), an error message is displayed.

FULL REPLACEMENT

Editing a resource YAML replaces the entire resource specification. Make sure you understand the impact of your changes before applying them.

ROLLING UPDATES

For Deployments, editing the pod template spec triggers a rolling update by default, creating a new ReplicaSet while scaling down the old one.

Deleting Workloads

To delete a workload, use the delete action on the workload. The delete operation supports two important options:

Cascade vs. Orphan

- **Cascade** (default) -- Deletes the workload and all of its dependent resources. For example, deleting a Deployment with cascade enabled also deletes its ReplicaSets and Pods.
- **Orphan** -- Deletes only the workload itself, leaving dependent resources running. Orphaned pods continue to run but are no longer managed by a controller.

Grace Period

The grace period controls how long Kubernetes waits for pods to shut down gracefully before force-terminating them. The default is **30 seconds**. During this period, pods receive a `SIGTERM` signal and can perform cleanup tasks such as draining connections or saving state.

PROTECTED NAMESPACE

Deleting workloads in the `kube-system` namespace is not permitted. This protects critical system components from accidental removal.

SUPPORTED TYPES

> Supported workload types for deletion: Deployment, StatefulSet, DaemonSet, ReplicaSet, Job, CronJob, and Pod.

Observability

The workload detail view includes an **Observability** panel with real-time metrics charts for:

- **CPU** usage over time
- **Memory** usage over time (in GB)
- **Storage** usage over time (in GiB)

You can select a time range of **1 Hour**, **3 Hours**, or **8 Hours** using the dropdown in the panel header. Metrics refresh automatically.

For more details on metrics, see Workload Metrics.

Pod Logs

ACTIVATE lets you view container logs for any running pod directly in the web UI, so you can debug and monitor workloads without switching to a terminal.

Viewing Logs

1. Go to **Kubernetes > Workloads**
2. Select a cluster and namespace
3. Click on a pod name to open its detail page
4. Scroll down to the **Logs** panel

The Logs panel displays the most recent log output from the pod's container in a scrollable viewer.

> **POD-LEVEL ONLY**
>
> The Logs panel only appears when viewing an individual pod. For higher-level workloads such as Deployments or StatefulSets, navigate to one of their child pods to access logs.

Log Controls

Tail Lines

By default, ACTIVATE loads the most recent **100 lines** of log output. You can request up to **10,000 lines** to see more history.

Time-Based Filtering

You can filter logs to only show entries after a specific timestamp. This is useful when you know roughly when an issue occurred and want to skip older output.

Real-Time Streaming

Once logs are loaded, ACTIVATE automatically polls for new lines. New output is appended to the bottom of the viewer as it arrives — you don't need to refresh the page.

Duplicate lines are automatically filtered out, so you'll always see a clean, continuous log stream.

> **SESSION PERSISTENCE**
>
> Logs are accumulated in the browser for the duration of your session. If you navigate away from the pod and return, the log viewer resets and fetches the latest lines again.

See Also

- Managing Workloads — Browse and manage pods, Deployments, and other workload types
- Workload Metrics — Monitor CPU, memory, and storage usage for workloads

Workload Metrics

ACTIVATE provides built-in observability charts for Kubernetes workloads. CPU, memory, and storage usage are collected from the Kubernetes metrics-server and displayed as time-series charts on each workload's detail page.

Accessing Metrics

Metrics are available on the detail page for workload-level resources (Deployments, StatefulSets, DaemonSets, Jobs, CronJobs, and ReplicaSets). To view them:

1. Go to **Kubernetes > Workloads**
2. Select a cluster and namespace
3. Click on a workload name to open its detail page
4. The **Observability** panel is displayed at the top of the page

> **WORKLOAD-LEVEL ONLY**
>
> Metrics are shown for workload-level resources, not individual pods. To view metrics for a Deployment, click on the Deployment itself rather than one of its pods.

Available Charts

The Observability panel displays three charts side by side:

CPU Usage

Shows CPU consumption in **cores** over time. The chart plots:

- **Usage** — Actual CPU consumed by all containers in the workload's pods
- **Request** — The sum of CPU requests configured across containers (displayed when set)
- **Limit** — The sum of CPU limits configured across containers (displayed when set)

Memory Usage

Shows memory consumption in **GB** over time. The chart plots:

- **Usage** — Actual memory used by all containers in the workload's pods
- **Request** — The sum of memory requests configured across containers (displayed when set)
- **Limit** — The sum of memory limits configured across containers (displayed when set)

Storage Usage

Shows ephemeral storage consumption in **GiB** over time. The chart plots:

- **Usage** — Actual ephemeral storage used by the workload's pods
- **Limit** — The node's ephemeral storage capacity (displayed when available)

> **EPHEMERAL STORAGE**
>
> Storage metrics are collected from the Kubernetes node's `/stats/summary` endpoint and reflect ephemeral storage usage, not persistent volume usage. For persistent volume information, see Services & Storage.

Time Range Selection

A dropdown in the Observability panel header lets you select the time window for all three charts:

Option	Description
1 Hour	Show metrics from the last 1 hour
3 Hours	Show metrics from the last 3 hours
8 Hours	Show metrics from the last 8 hours (default)

The selected time range applies to all three charts simultaneously. The default view is **8 Hours**, giving you a broad view of workload behavior.

Auto-Refresh

Metrics data refreshes automatically. There is no need to manually reload the page — the charts update in place with the latest data points from the metrics-server.

Data Source

ACTIVATE collects metrics from the Kubernetes metrics-server, which must be installed on each connected cluster. The metrics-server provides point-in-time CPU and memory usage for pods and nodes.

The platform's metrics collection process works as follows:

1. **Pod metrics** — CPU and memory usage are read from the metrics-server's `PodMetrics` API (`metrics.k8s.io/v1beta1`)

2. **Resource requests and limits** — CPU, memory, and ephemeral storage requests/limits are read from each pod's container spec
3. **Storage metrics** — Ephemeral storage usage is read from each node's `/stats/summary` proxy endpoint
4. **Aggregation** — Metrics are aggregated at the workload level (Deployment, StatefulSet, etc.) by resolving each pod's parent owner reference
5. **Storage** — Collected metrics are stored with timestamps and can be queried over the selected time range

> **METRICS SERVER REQUIRED**
>
> If the metrics-server is not installed on a cluster, the Observability panel will not display any data. Ensure that `metrics-server` is deployed and running in the `kube-system` namespace of each cluster you want to monitor.

Workload Types

Metrics aggregation resolves each pod's owner to determine which workload it belongs to:

Pod Owner	Resolved Workload
ReplicaSet (owned by Deployment)	Deployment
StatefulSet	StatefulSet
Job	Job
DaemonSet	DaemonSet (excluded from metrics)
Standalone pod	Excluded

> **EXCLUDED WORKLOADS**
>
> DaemonSet pods and standalone pods (pods with no owner reference) are excluded from workload-level metrics collection because they are not typically associated with a user-managed workload.

See Also

- Pod Logs — Stream and view container logs for individual pods
- Managing Workloads — Browse and manage Deployments, StatefulSets, and other workload types
- Cost Tracking — Monitor per-namespace compute costs

Helm Charts

Parallel Works provides a built-in Helm chart manager that lets you install, upgrade, roll back, and uninstall Helm releases on your Kubernetes clusters directly from the web UI. You can configure chart repositories, customize values with the integrated YAML editor, and track release history without leaving the platform.

Installing a Helm Chart

Navigate to **Kubernetes > Helm Charts** and click **Install Helm Chart** in the action bar. A configuration drawer opens with two steps: chart configuration and values customization.

Step 1: Configure the Chart

Fill in the following fields:

Field	Required	Description
Cluster Name	Yes	The target Kubernetes cluster
Repository Name	Yes	A name for the Helm repository (e.g., `bitnami`)
Repository URL	Yes	The URL of the Helm repository (e.g., `https://charts.bitnami.com/bitnami`)
Chart Name	Yes	The name of the chart to install (e.g., `nginx`)
Chart Version	No	A specific chart version. Leave empty to install the latest version
Release Name	Yes	A unique name for this Helm release (lowercase letters, numbers, and hyphens only)
Namespace	Yes	The target namespace for the release
Load default values	No	Toggle to load the chart's default `values.yaml` for editing

Click **Next** to proceed to the values editor.

Step 2: Customize Values

The second step presents a full-screen YAML editor where you can customize the chart's configuration values.

- If **Load default values** is enabled, the editor is pre-populated with the chart's default `values.yaml`. Edit any values you want to override before installing.
- If **Load default values** is disabled, the editor starts empty. Enter only the values you want to set.

> **DEFAULT VALUES**
>
> Loading default values gives you a complete reference of every configurable option the chart supports. This is useful when you are installing a chart for the first time and want to see all available settings.

Click **Back** to return to the configuration step, or click **Install** to deploy the chart. The platform adds the repository, resolves the chart, parses your YAML values, and runs the Helm install operation against the selected cluster and namespace.

> **INSTALL TIMEOUT**
>
> The install operation has a 15-minute timeout. For large or complex charts, the release may continue reconciling in the background after the initial install completes.

Viewing Helm Releases

The **Kubernetes > Helm Charts** page displays a table of all Helm releases across your clusters. Each row shows:

Column	Description
Name	The release name (click to view details)
Status	Current release status: Deployed, Pending, Failed, Superseded, or Uninstalled
Namespace	The Kubernetes namespace the release is installed in
Cluster	The cluster the release belongs to (hidden when filtering by a single cluster)
Chart	The chart name
Version	Chart version and application version
Rev	Current revision number
Updated	Time since the last update

Use the filter bar at the top of the page to narrow results by cluster, namespace, or status. You can also search by release name, namespace, or chart name.

Release Details and History

Click a release name to open its detail page. The detail view shows the current release status, chart version, revision number, and last update time.

The release history lists every revision of the release, including:

- **Revision number** -- incremented with each install, upgrade, or rollback
- **Status** -- the outcome of that revision (e.g., deployed, superseded, failed)
- **Chart** -- the chart name and version used for that revision
- **App version** -- the application version from the chart metadata
- **Updated** -- when the revision was created

Upgrading a Release

To upgrade an existing Helm release to a new chart version or with updated values, use the upgrade operation. The upgrade accepts the same fields as installation (repository URL, chart name, version, release name, namespace, and YAML values). If the target namespace does not exist and namespace creation is enabled, the platform creates it automatically.

> **VALUES REPLACEMENT**
>
> Upgrading replaces the release's current configuration with the values you provide. Make sure to include all values you want to keep, not just the ones you are changing.

Rolling Back a Release

From the release detail page, click **Rollback** in the action bar. Enter the revision number you want to roll back to in the dialog that appears, then confirm.

- The revision number must be less than the current revision and greater than zero.
- A successful rollback creates a new revision with the configuration from the target revision.

Uninstalling a Release

From the release detail page, click **Delete** in the action bar. Confirm the deletion in the dialog. This uninstalls the Helm release from the namespace and removes all associated Kubernetes resources created by the chart.

> **IRREVERSIBLE ACTION**
>
> Uninstalling a Helm release permanently removes its resources from the cluster. This action cannot be undone.

Chart Repository Configuration

When you install a Helm chart, the platform automatically registers the repository using the **Repository Name** and **Repository URL** you provide. There is no separate repository management step -- the repository is configured as part of the install flow.

To install charts from different repositories, provide the appropriate repository name and URL each time you install. Common public repositories include:

Repository	URL
Bitnami	`https://charts.bitnami.com/bitnami`
Ingress-NGINX	`https://kubernetes.github.io/ingress-nginx`
Jetstack (cert-manager)	`https://charts.jetstack.io`
Prometheus Community	`https://prometheus-community.github.io/helm-charts`

Resource Quotas

Resource quotas let you set limits on CPU, memory, and GPU usage per namespace across your Kubernetes clusters. Use quotas to prevent any single namespace from consuming more than its fair share of cluster resources.

What Quotas Control

Each quota is scoped to a specific namespace on a specific cluster and enforces limits on three resource types:

Resource	Unit	Description
CPU	Millicores (e.g., `1000m` = 1 core)	Maximum CPU that workloads in the namespace can consume

Memory	Mebibytes (e.g., `1Gi`, `512Mi`)	Maximum memory that workloads in the namespace can consume
GPU	Count (e.g., `1`)	Maximum number of NVIDIA GPUs that workloads in the namespace can request

Kubernetes enforces these limits at the scheduling level. Pods that would exceed the quota are rejected until resources are freed.

Creating a Quota

1. Navigate to **Kubernetes > Quotas** and click **Add Quota** in the action bar.
2. Fill in the quota form:

Field	Required	Description
Cluster Name	Yes	The target Kubernetes cluster
Namespace	Yes	The namespace to apply the quota to
CPU	Yes	CPU limit (e.g., `1000m` for 1 core, `500m` for half a core)
Memory	Yes	Memory limit (e.g., `1Gi`, `2048Mi`)
GPU	Yes	GPU limit (integer, e.g., `0` for no GPUs, `2` for two GPUs)

3. Click **Create Quota**.

> **MANAGED BY PLATFORM**
>
> Quotas are created as Kubernetes `ResourceQuota` objects in the target namespace. The platform manages these objects for you -- you do not need to write YAML manifests.

Viewing Quota Utilization

The **Kubernetes > Quotas** page displays a table of all quotas across your clusters. Each quota row shows the current usage relative to the configured limit for CPU, memory, and GPU.

Usage is displayed as a progress bar with the format **used / limit**:

Utilization	Color	Meaning
0 -- 50%	Green	Healthy usage, plenty of headroom
50 -- 70%	Yellow	Moderate usage, consider monitoring
Above 70%	Red	High usage, namespace is approaching its limit

Use the cluster and namespace filters at the top of the page to narrow the view.

> **HIGH UTILIZATION**

> If a quota's CPU or memory bar is consistently red, consider increasing the limit or moving workloads to a less utilized namespace.

Editing a Quota

1. On the **Quotas** page, select a single quota by clicking its checkbox.
2. Click **Edit** in the action bar.
3. Update the CPU, memory, or GPU limits in the form.
4. Click **Update Quota**.

> **EXISTING WORKLOADS**
>
> Lowering a quota below current usage does not terminate running workloads, but it prevents new pods from being scheduled until usage drops below the new limit.

Deleting Quotas

1. On the **Quotas** page, select one or more quotas by clicking their checkboxes.
2. Click **Delete** in the action bar.
3. Confirm the deletion in the dialog.

Deleting a quota removes the `ResourceQuota` object from the namespace. Workloads in the namespace are no longer subject to the deleted limits.

Cost Tracking

ACTIVATE can track per-namespace compute costs on your Kubernetes clusters. When enabled, a cost-tracking agent runs inside the cluster, periodically sampling CPU and memory usage for every pod. These samples are combined with configurable pricing rates to produce cost records broken down by namespace.

Enabling Cost Tracking

To enable cost tracking for a cluster:

1. Navigate to **Kubernetes** in the sidebar and open the **Cost Management** drawer.
2. Select the cluster from the **Cluster Name** dropdown.
3. Click **Enable Cost Tracking**.
4. Confirm the prompt — this will deploy new pods to the cluster to collect cost data.

The cluster's cost tracking status will change to **Provisioning** while the agent is being deployed. Once the agent is running, the status changes to **Enabled**.

> **CLUSTER REQUIREMENT**
>
> Enabling cost tracking deploys a cost-tracking agent as pods inside the target cluster. The agent requires the Kubernetes Metrics Server to be available on the cluster.

Status Indicators

The cost tracking status for a cluster is displayed next to the cluster name in the Cost Management drawer. Possible statuses are:

Status	Meaning
Enabled	Cost tracking is active and collecting data
Disabled	Cost tracking is not running on this cluster
Provisioning	The cost-tracking agent is being deployed to the cluster
Deleting	The cost-tracking agent is being removed from the cluster

Click the status indicator to view detailed provisioning steps and progress.

Configuring Pricing

Once cost tracking is enabled, you can set custom pricing rates that determine how resource usage is converted into cost values. Two rates are configurable:

Rate	Description	Default
CPU Price (per core/min)	Cost charged per vCPU core per minute of usage	$0.00055
Memory Price (per GB/min)	Cost charged per GiB of memory per minute of usage	$0.00007

To update pricing:

1. Open the **Cost Management** drawer and select the cluster.
2. Enter new values in the **CPU Price** and **Memory Price** fields.
3. Click **Update Prices**.

The new rates are sent to the cost-tracking agent running on the cluster and take effect immediately for all future cost calculations.

> **PRICING GUIDANCE**
>
> Set prices to match your infrastructure costs. For cloud-hosted clusters, you can derive per-core and per-GiB minute rates from your cloud provider's pricing for the instance types in your node pools.

How Costs Are Calculated

The cost-tracking agent samples resource data at regular intervals. For each sampling interval, costs are calculated as follows:

1. Collect Pod-Level Usage and Requests

For every pod in every namespace, the agent collects:

- **CPU usage** — actual CPU cores consumed (from the Metrics Server)

- **CPU requests** — CPU cores reserved in the pod spec
- **Memory usage** — actual bytes of memory consumed (from the Metrics Server)
- **Memory requests** — bytes of memory reserved in the pod spec

2. Determine Billable Resources

For each pod, the billable amount for each resource is the **greater of usage or requests**:

```
billable_cpu    = max(cpu_usage, cpu_requested)
billable_memory = max(memory_usage, memory_requested)
```

This means you are charged for at least what you requested, even if actual usage is lower. If usage exceeds the request, you are charged for the higher actual usage.

3. Aggregate by Namespace

Billable CPU and memory values are summed across all pods within each namespace. This produces a single CPU (in cores) and memory (in GiB) total per namespace per sampling interval.

4. Apply Pricing Rates

The final cost for each namespace is:

```
cpu_cost    = billable_cpu_cores * cpu_price_per_core_per_minute
memory_cost = billable_memory_gib * memory_price_per_gib_per_minute
total_cost  = cpu_cost + memory_cost
```

Each cost record is stored with a timestamp and the namespace (group) it belongs to.

> **MEMORY UNITS**
>
> Memory values are collected in bytes and converted to GiB (divided by 1024^3) before pricing is applied.

Disabling Cost Tracking

To disable cost tracking for a cluster:

1. Open the **Cost Management** drawer and select the cluster.
2. Click **Disable Cost Tracking**.
3. Confirm the prompt.

> **DATA LOSS**
>
> Disabling cost tracking **removes all collected cost data** for the cluster. This action cannot be undone. If you need to retain historical cost data, export it before disabling tracking.

The status will change to **Deleting** while the cost-tracking agent pods are removed from the cluster. Once removal is complete, the status returns to **Disabled**.

See Also

- Resource Quotas — set CPU, memory, and GPU limits per namespace
- Workload Metrics — monitor real-time resource usage for individual workloads

Services & Storage

ACTIVATE provides a unified view of Kubernetes services, persistent storage, ConfigMaps, and Secrets across all connected clusters. You can filter, search, and inspect these resources without switching between clusters or namespaces.

Services

Navigate to **Kubernetes > Services** in the sidebar to view all services across your clusters.

Service Table Columns

Column	Description
Name	The service name. Click to view full resource details.
Type	The Kubernetes service type: `ClusterIP`, `NodePort`, `LoadBalancer`, or `ExternalName`.
Namespace	The namespace the service belongs to.
Cluster	The cluster hosting the service (hidden when filtering by a single cluster).
Cluster IP	The internal cluster IP address assigned to the service.
External IP	The external IP or hostname assigned by a load balancer, or the explicitly configured external IPs. Displays `–` if none.
Ports	Port mappings in `port:targetPort/protocol` format (e.g., `80:8080/TCP`).
Created	Relative timestamp showing when the service was created.

Filtering Services

Use the filter bar at the top of the services table to narrow results:

- **Clusters** — Show services from specific clusters only
- **Namespaces** — Filter by one or more namespaces
- **Types** — Filter by service type (`ClusterIP`, `NodePort`, `LoadBalancer`, `ExternalName`)
- **Search** — Free-text search across service name, namespace, cluster, cluster IP, and external IP

Selectors

Each service includes a **selector** field that shows which pods the service targets. Selectors are displayed as comma-separated `key=value` pairs (e.g., `app=nginx,tier=frontend`). Services without selectors show an empty value.

Persistent Volume Claims (PVCs)

Navigate to **Kubernetes > Storages** in the sidebar to view PVCs and PVs across your clusters.

PVCs represent storage requests made by pods. The storage table displays the following information for each PVC:

Column	Description
Name	The PVC name. Click to view full resource details.
Status	Binding status: `Bound`, `Pending`, or `Failed`.
Type	Displays `PVC`.
Namespace	The namespace the PVC belongs to.
Cluster	The cluster hosting the PVC.
Capacity	The provisioned storage capacity (e.g., `10Gi`).
Access	Access mode abbreviations: `RWO` (ReadWriteOnce), `ROX` (ReadOnlyMany), `RWX` (ReadWriteMany), `RWOP` (ReadWriteOncePod).
Storage Class	The storage class used for provisioning (e.g., `standard`, `gp3`).
Used By	The pod currently using this PVC. Click the pod name to navigate to the pod details.
Created	Relative timestamp showing when the PVC was created.

Access Mode Reference

Abbreviation	Full Name	Description
`RWO`	ReadWriteOnce	Volume can be mounted as read-write by a single node
`ROX`	ReadOnlyMany	Volume can be mounted as read-only by many nodes
`RWX`	ReadWriteMany	Volume can be mounted as read-write by many nodes
`RWOP`	ReadWriteOncePod	Volume can be mounted as read-write by a single pod

Persistent Volumes (PVs)

PVs are cluster-scoped storage resources. They appear alongside PVCs in the same storage table.

Column	Description
Name	The PV name.
Status	Volume phase: `Available`, `Bound`, `Released`, or `Failed`.
Type	Displays `PV`.
Namespace	Displays `–` (PVs are cluster-scoped).

Cluster	The cluster hosting the PV.
Capacity	The total storage capacity of the volume.
Access	Supported access modes (same abbreviations as PVCs).
Storage Class	The storage class of the volume.
Reclaim Policy	What happens when the PVC is deleted: `Retain`, `Delete`, or `Recycle`.
Created	Relative timestamp showing when the PV was created.

Filtering Storage

Use the filter bar to narrow storage results:

- **Clusters** — Show storage from specific clusters only
- **Namespaces** — Filter by namespace (applies to PVCs only)
- **Types** — Filter by `PVC` or `PV`
- **Status** — Filter by status: `Bound`, `Pending`, `Available`, `Released`, or `Failed`
- **Search** — Free-text search across name, namespace, cluster, storage class, volume name, and claim reference

> **ADMIN ACCESS REQUIRED**
>
> PVs are cluster-scoped resources and only appear when querying across all namespaces (admin users). Non-admin users see only PVCs within their accessible namespaces.

ConfigMaps & Secrets

Navigate to **Kubernetes > ConfigMaps & Secrets** in the sidebar to view configuration data across your clusters.

ConfigMap and Secret Table Columns

Column	Description
Name	The resource name. Click to view full details including key-value data.
Type	Either `ConfigMap` or `Secret`.
Namespace	The namespace the resource belongs to.
Cluster	The cluster hosting the resource.
Data	Number of keys in the resource (e.g., `3 keys`).
Created	Relative timestamp showing when the resource was created.

ConfigMaps

ConfigMaps store non-confidential configuration data as key-value pairs. When you click a ConfigMap name, you can view:

- All keys and their values
- Binary data keys (marked with a `(binary)` suffix)
- Labels attached to the ConfigMap
- Which deployments reference the ConfigMap (via volume mounts or `envFrom`)

Secrets

Secrets store sensitive data such as passwords, tokens, and TLS certificates. ACTIVATE displays:

- The list of keys contained in the secret (values are not exposed in the listing)
- The secret type (e.g., `Opaque`, `kubernetes.io/tls`, `kubernetes.io/dockerconfigjson`)
- Which deployments reference the secret
- Labels attached to the secret

> **FILTERED TOKENS**
>
> Service account tokens are automatically filtered out from the secrets list to reduce noise. Only application-level secrets are shown.

Filtering ConfigMaps & Secrets

- **Clusters** — Show configs from specific clusters only
- **Namespaces** — Filter by one or more namespaces
- **Types** — Filter by `ConfigMap` or `Secret`
- **Search** — Free-text search across name, namespace, and cluster

Cross-Cluster Queries

All resource views in this section aggregate data from every connected cluster by default. The response metadata includes:

- **Total clusters queried** — How many clusters were contacted
- **Successful clusters** — How many clusters responded successfully
- **Total resources** — The combined count of resources returned

If a cluster is unreachable, the other clusters still return their results and the error is noted in the response metadata. This ensures partial availability does not block the entire view.

See Also

- Managing Workloads — View and manage Deployments, StatefulSets, and other workload types
- Connecting Clusters — Add clusters to enable cross-cluster resource views

Nodes & GPUs

ACTIVATE provides visibility into the nodes running across your Kubernetes clusters and tools for managing NVIDIA GPU configurations, including Multi-Instance GPU (MIG) partitioning.

> **ADMIN ONLY**
> The Nodes view is available to organization admins and platform admins only.

Viewing Cluster Nodes

Navigate to **Kubernetes > Nodes** in the sidebar to view all nodes across your connected clusters.

Node Table Columns

Column	Description
Name	The node hostname. Click to open the node detail page.
Cluster	The cluster the node belongs to (hidden when filtering by a single cluster).
Kubernetes Version	The kubelet version running on the node (e.g., `v1.28.4`).
Container Runtime	The container runtime and version (e.g., `containerd://1.7.2`).
Internal IP	The node's internal network IP address.
Architecture	The CPU architecture (e.g., `amd64`, `arm64`).

Filtering Nodes

Use the filter bar to narrow results:

- **Clusters** — Show nodes from specific clusters only
- **Search** — Free-text search across node name and cluster name

Node Detail Page

Click a node name to open its detail page. This page displays comprehensive information about the selected node.

System Information

The detail page shows the following system-level properties:

Property	Description
OS Image	The operating system image (e.g., `Ubuntu 22.04.3 LTS`).
Kernel Version	The Linux kernel version.
Operating System	The OS type (e.g., `linux`).
Architecture	The CPU architecture.
Container Runtime Version	The container runtime and version.
Kubernetes Version	The kubelet version.
Internal IP	The node's internal IP address.

Capacity and Allocatable Resources

Each node reports two sets of resource quantities:

- **Capacity** — The total physical resources available on the node
- **Allocatable** — The resources available for pod scheduling (capacity minus system-reserved resources)

Both sets include:

Resource	Format	Example
CPU	Number of cores	`8`
Memory	Gigabytes	`32Gi`
Ephemeral Storage	Gigabytes	`100Gi`
Pods	Maximum pod count	`110`
NVIDIA GPUs	GPU count (if present)	`4`

> **RESOURCE OVERHEAD**
>
> Compare the capacity and allocatable values to understand how much overhead is reserved for system components like the kubelet and OS processes.

Node Labels

The detail page displays all labels assigned to the node. Labels commonly include:

- `kubernetes.io/hostname` — The node hostname
- `kubernetes.io/arch` — CPU architecture
- `kubernetes.io/os` — Operating system
- `node.kubernetes.io/instance-type` — Instance type (on cloud providers)
- `nvidia.com/gpu.product` — GPU model name (on GPU nodes)
- `nvidia.com/mig.config` — Current MIG configuration label

GPU Management

For nodes equipped with NVIDIA GPUs, ACTIVATE provides tools to install and manage the NVIDIA GPU Operator and configure MIG partitioning directly from the node detail page.

NVIDIA GPU Operator

The GPU Operator automates the management of GPU drivers, container toolkits, and device plugins on Kubernetes. From the node detail page, you can install, upgrade, or roll back the GPU Operator Helm chart.

Installing the GPU Operator

1. Navigate to the node detail page for a GPU-equipped node
2. Click the **GPU Operator** button in the action bar

3. Fill in the installation form:

Field	Description	Default
Helm Chart Version	The GPU Operator chart version to install	`v25.3.0`
Namespace	The namespace for the GPU Operator deployment	`gpu-operator`
Create Namespace	Whether to create the namespace if it does not exist	`true`
Containerd Config	Path to the containerd configuration file (optional)	—
Containerd Socket	Path to the containerd socket (optional)	—

4. Click **Install NVIDIA GPU Operator**

The operator is installed from the `https://helm.ngc.nvidia.com/nvidia` Helm repository using the `nvidia/gpu-operator` chart.

Upgrading the GPU Operator

If the GPU Operator is already installed, the same form appears with an **Upgrade NVIDIA GPU Operator** button instead. The upgrade uses the same Helm chart configuration.

Rolling Back

When the GPU Operator is installed, the drawer also shows the **Release History** table with all previous revisions. Click the rollback button next to any revision to revert to that version.

MIG Configuration

NVIDIA Multi-Instance GPU (MIG) allows a single physical GPU to be partitioned into multiple isolated GPU instances, each with dedicated compute, memory, and bandwidth resources.

MIG Strategies

ACTIVATE supports two MIG strategies:

Strategy	Description
Single	All GPU instances on the node use the same MIG profile. Use this when all workloads on the node have identical GPU requirements.
Mixed	Different MIG profiles can coexist on the same GPU. Use this for heterogeneous workloads with varying GPU requirements.

Configuring MIG

1. Navigate to the node detail page for a GPU node that has the GPU Operator installed
2. Click the **NVIDIA MIG** button in the action bar
3. Select a **MIG Strategy** (`single` or `mixed`)
4. Enter a **MIG Strategy Config** value that specifies the MIG profile to apply

Common configuration values include:

- `all-1g.6gb` — All instances configured as 1g.6gb (smallest slice)
- `all-2g.12gb` — All instances configured as 2g.12gb
- `all-3g.24gb` — All instances configured as 3g.24gb
- `all-balanced` — A balanced mix of MIG instance sizes (for mixed strategy)

5. Click **Configure MIG**

> **GPU-DEPENDENT PROFILES**
>
> The available MIG profiles depend on the GPU model. The configuration drawer displays the default MIG partitioning options for the detected GPU type, loaded from the `default-mig-parted-config` ConfigMap managed by the GPU Operator.

What Happens During MIG Configuration

When you apply a MIG configuration, ACTIVATE performs two operations:

1. **Patches the cluster policy** — Updates the `clusterpolicies.nvidia.com/cluster-policy` CRD to set the MIG strategy (e.g., `mixed` or `single`) at `/spec/mig/strategy`
2. **Labels the node** — Applies the `nvidia.com/mig.config` label to the target node with the specified configuration value (e.g., `all-balanced`)

The GPU Operator detects these changes and automatically reconfigures the GPU partitioning on the node.

> **WORKLOAD DISRUPTION**
>
> Changing MIG configuration may temporarily disrupt GPU workloads running on the node. Plan MIG reconfiguration during maintenance windows when possible.

Cross-Cluster Queries

The nodes list view aggregates data from all connected clusters by default. The response metadata includes:

- **Total clusters queried** — How many clusters were contacted
- **Successful clusters** — How many responded successfully
- **Total nodes** — The combined count of nodes returned

If a cluster is unreachable, the remaining clusters still return their results.

See Also

- Resource Quotas — Set GPU limits per namespace
- Helm Charts — Manage Helm releases including the GPU Operator
- Managing Workloads — View GPU workloads running across your clusters

Machine Learning

ACTIVATE lets you provision managed machine learning environments connected to cloud infrastructure such as Azure Machine Learning Studio.

ML Workspaces

- Adding an ML Resource — Create and manage ML workspace resources

Related Documentation

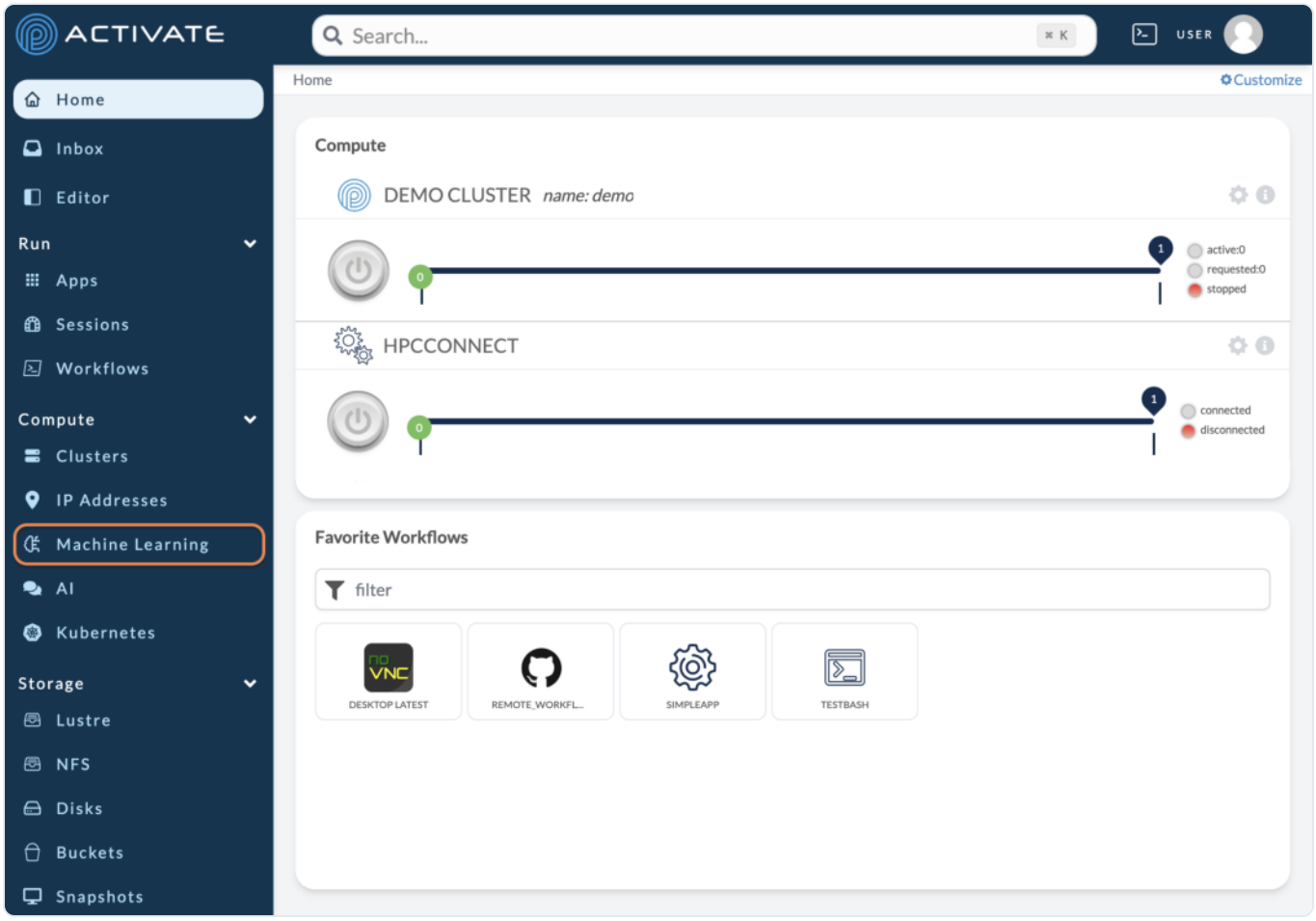
- AI — AI Chat, providers, and best practices
- Getting Started — Tutorial for new users

Adding an ML Resource

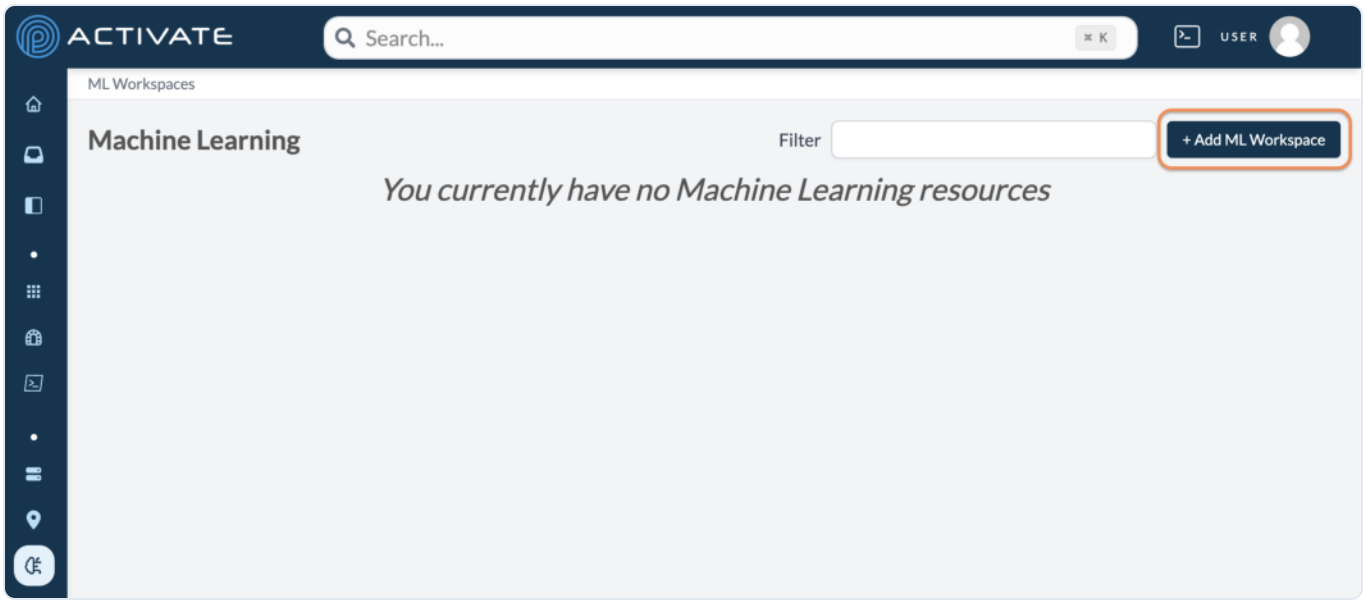
> **FEATURE MAY NOT BE AVAILABLE ON ALL PLATFORMS**
>
> Not all deployments of ACTIVATE support the Machine Learning feature. If you do not see the **Machine Learning** option in your ACTIVATE account, please contact support for more information.

You can add machine learning resources to your ACTIVATE account. In this example, we'll be adding a resource that connects to Azure Machine Learning Studio.

Navigate to **Machine Learning**.

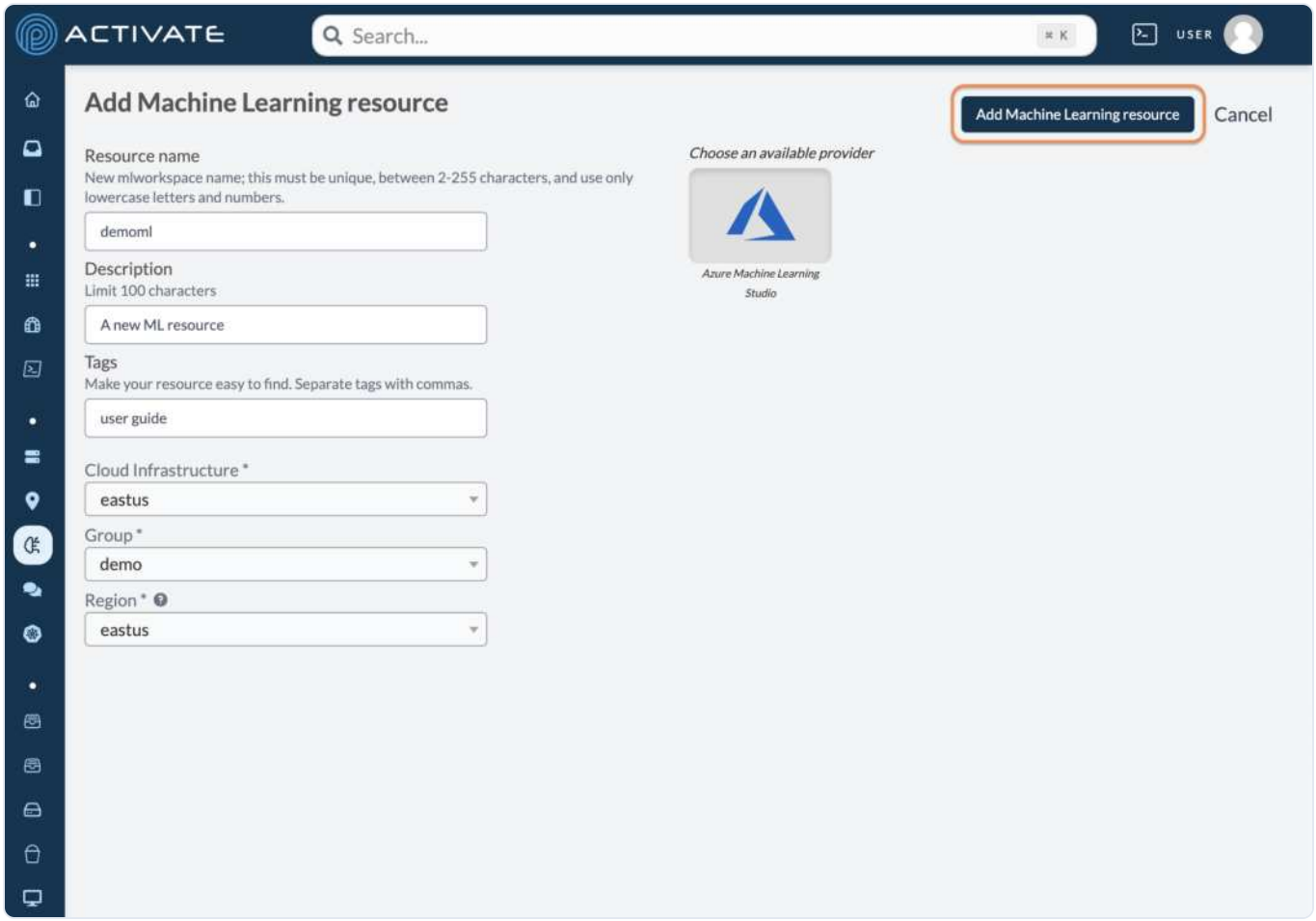


Click **Add ML Workspace**.

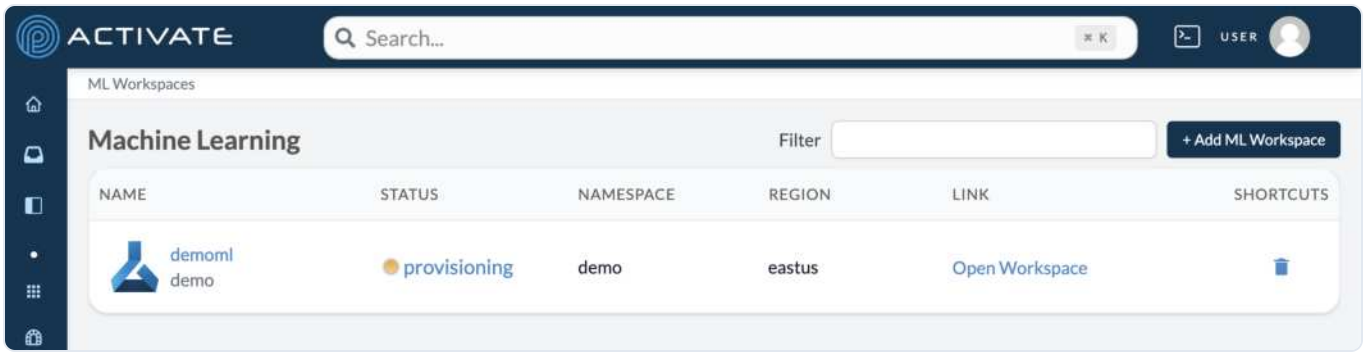


You must enter a **Resource Name** and choose a **Cloud Infrastructure**, **Group**, and **Region**. The **Description** and **Tags** are optional.

Click **Add Machine Learning resource**.



A notification will appear with the message *Machine Learning resource created*. You'll also be redirected to your list of Machine Learning resources.



Once the resource has been created, the page will display the status, namespace, and region. From this page, you can:

- click the name of your resource to see more information about it.
- click the status to see a detailed log of its progress.
- click **Open Workspace** to access Azure Machine Learning Studio, which will open in a new browser tab.

Deleting a Resource

On the **Machine Learning** page, click the delete icon.

A dialog box will appear with the message *Are you sure you want to delete this Machine Learning resource?*

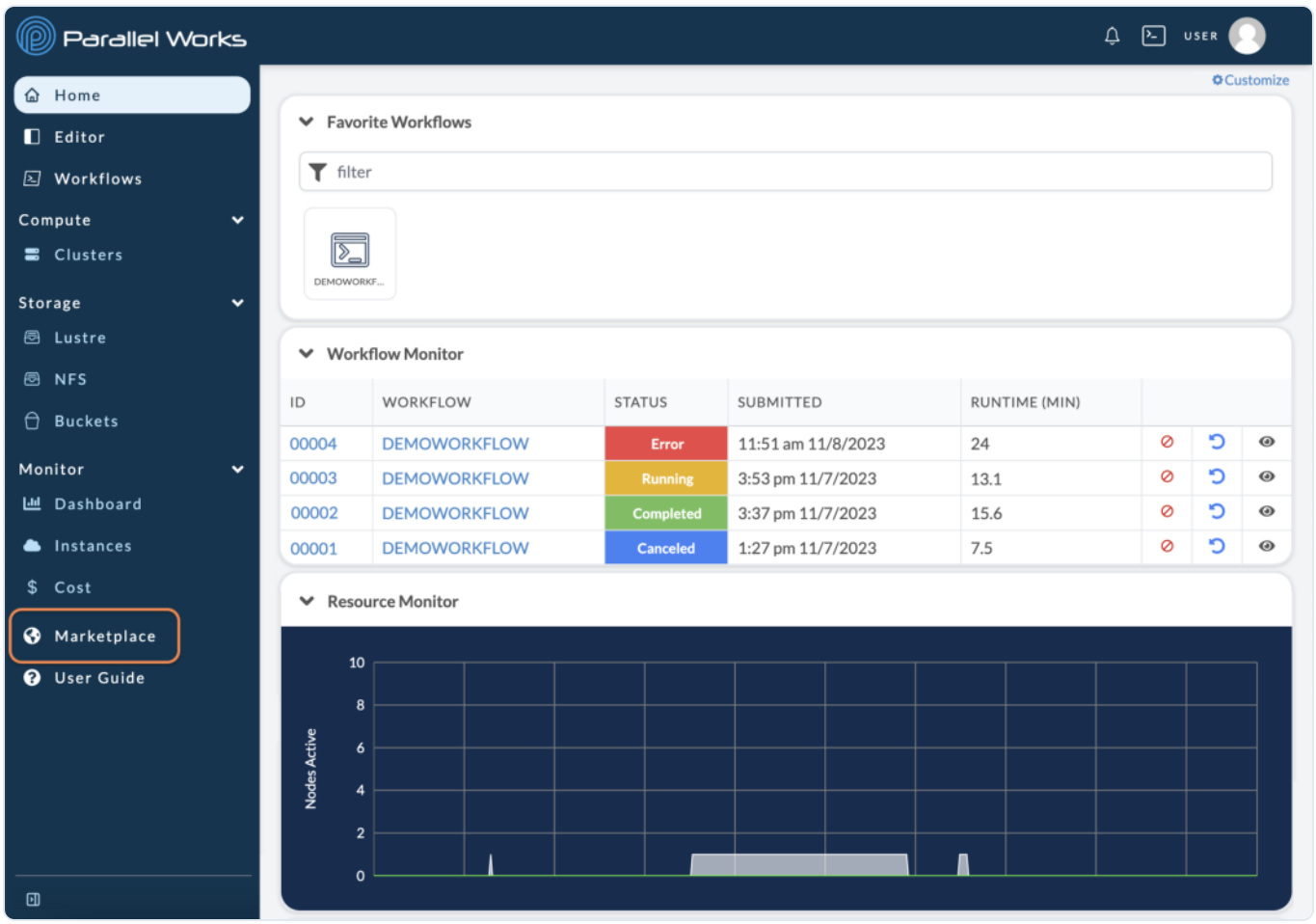
Click **Delete**.

A notification will appear with the message *Machine Learning resource deleted successfully*.

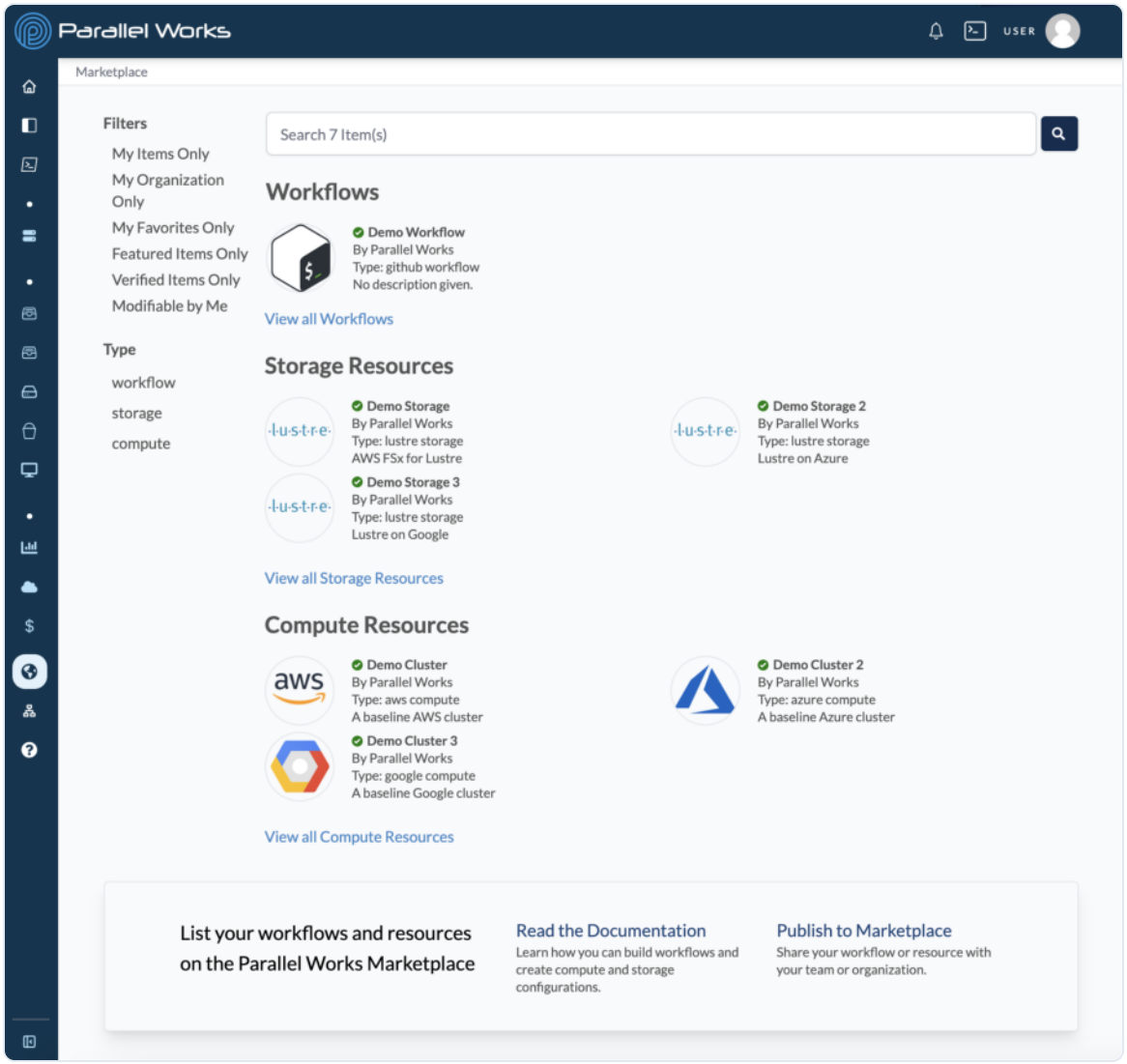
Marketplace

In the ACTIVATE Marketplace, you can access pre-configured workflows, storage resources, and compute resources from other users in your organization.

Click your username, then **Marketplace**.



You'll be taken to the Marketplace home page.



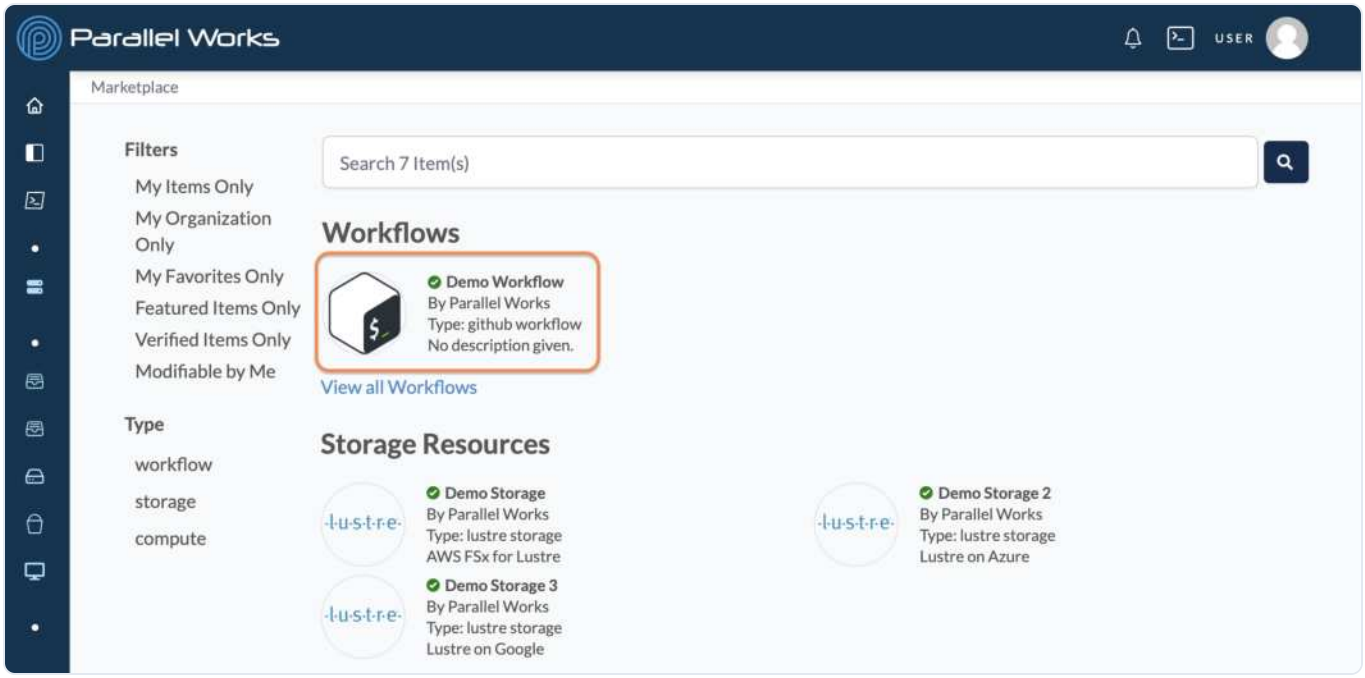
All resource categories are limited to ten items by default. Click **View all** to see a complete item list.

Use the search bar at the top of the page to find items by creator, type, or name. To filter viewable items, click an item in **Filters** or **Type** on the left.

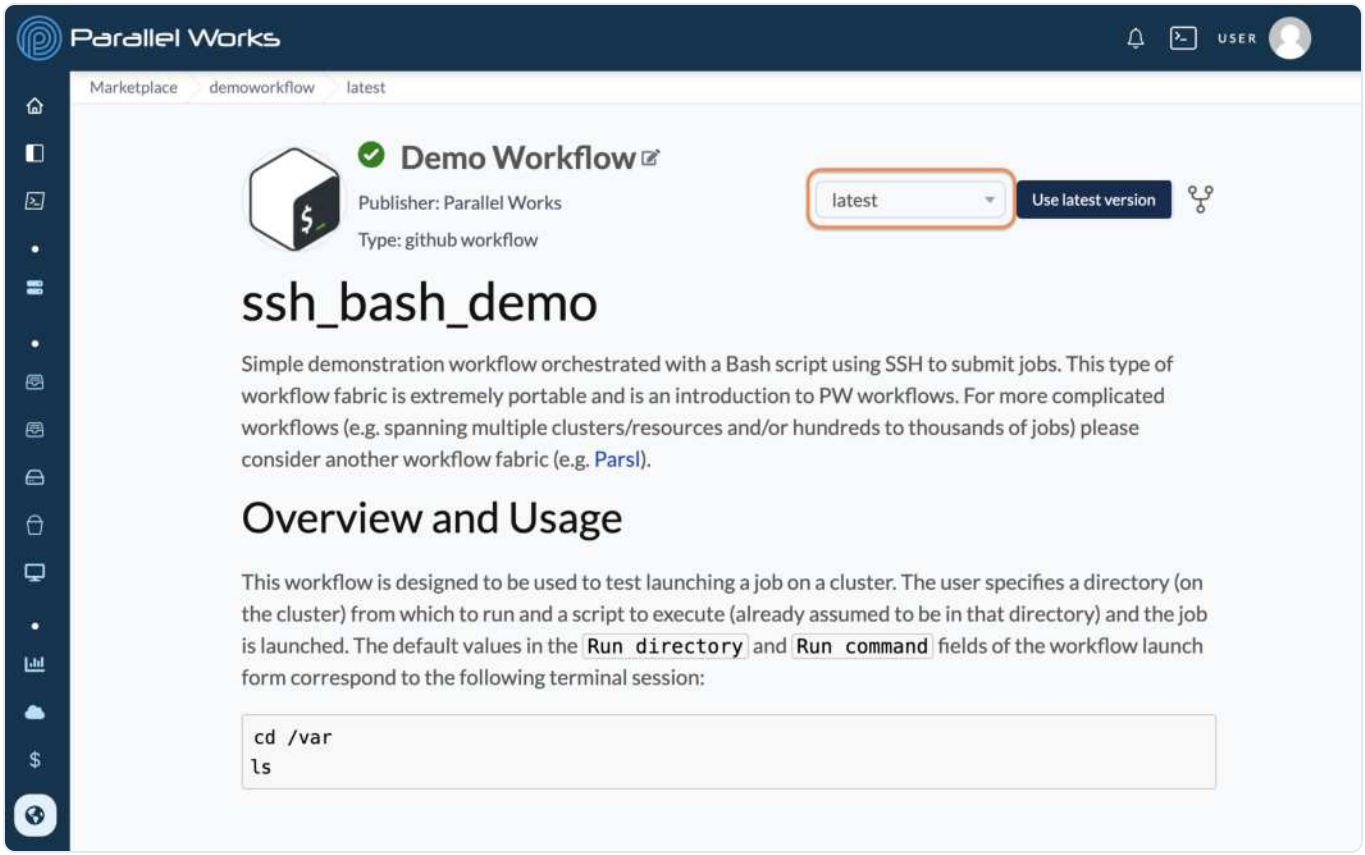
Any Marketplace item with a green check icon is verified by your organization.

Favoriting Workflows

Click the worfklow you'd like to use.



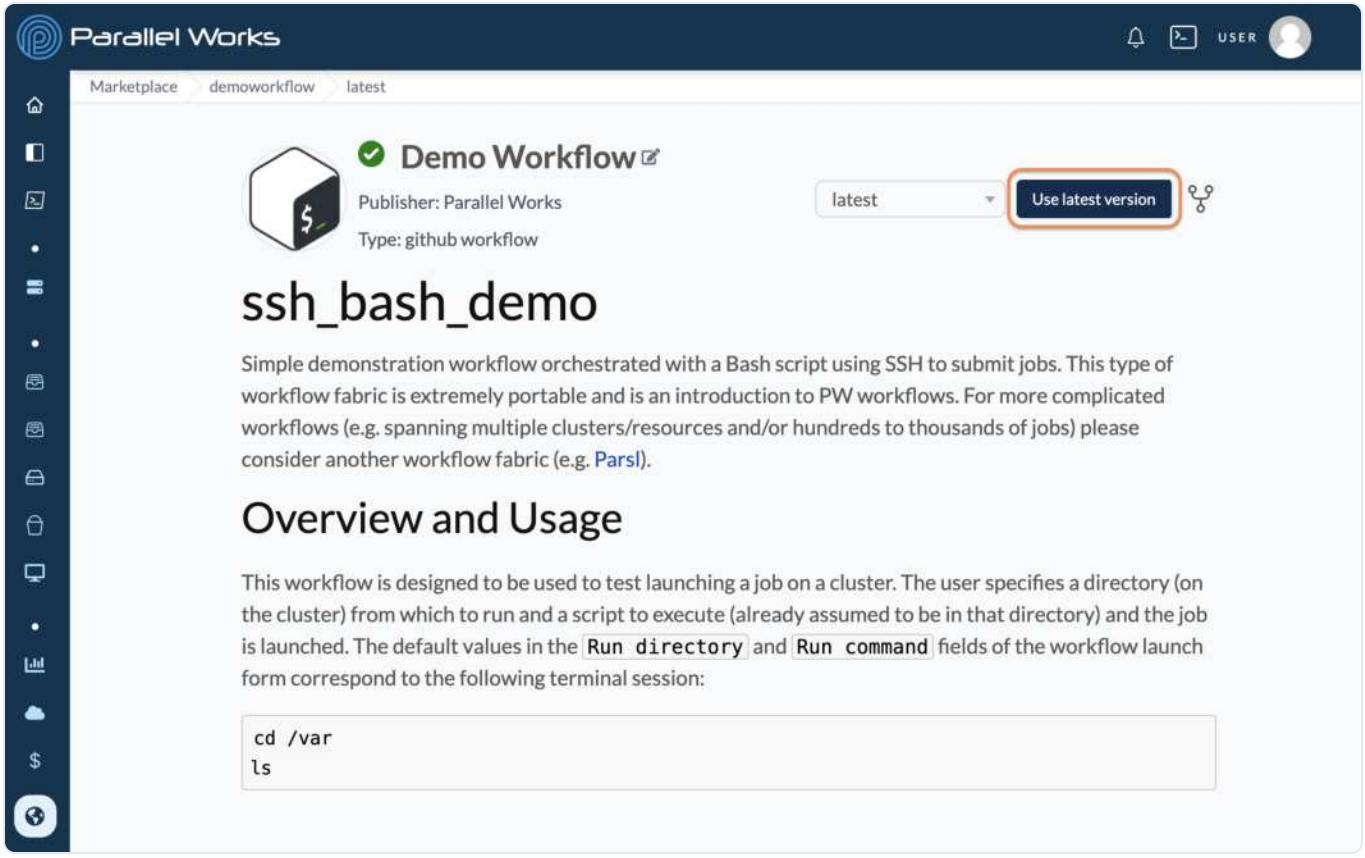
On the next page, click **latest** to open the dropdown menu of versions. Choose the version you'd like to use.



ABOUT VERSIONS

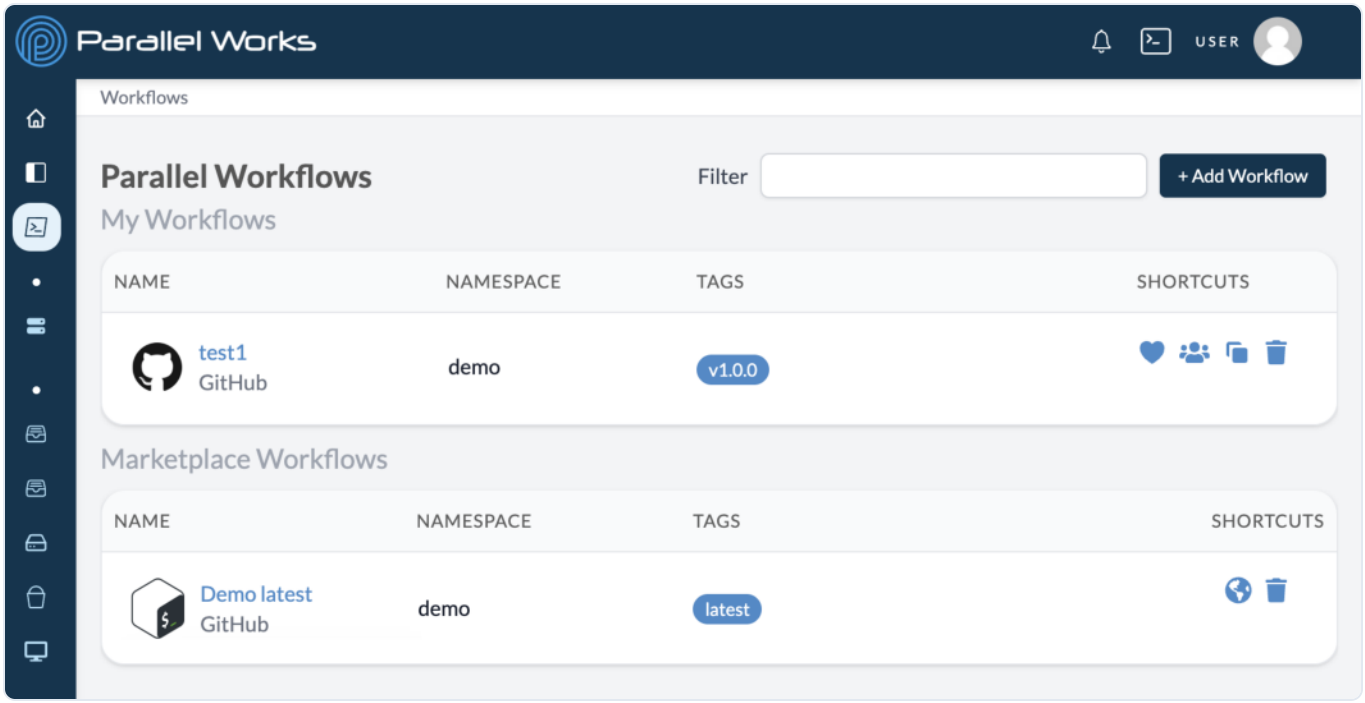
> If you select the latest version, your Marketplace item will be updated whenever the item's creator makes a change or a new version.
>
> Alternatively, you can select any of the versions listed in the dropdown menu. If you use a specific version of an item, it will not be updated in your account if the item's creator makes changes.

Click the **Use** button.



The **Use** button will turn green. The message *Using version successfully* will appear in the bottom right corner of your screen.

The workflow will appear under **Marketplace Workflows** on your **Workflows** page. The version number or *latest* will be included in the item's **Name** and **Tags**.

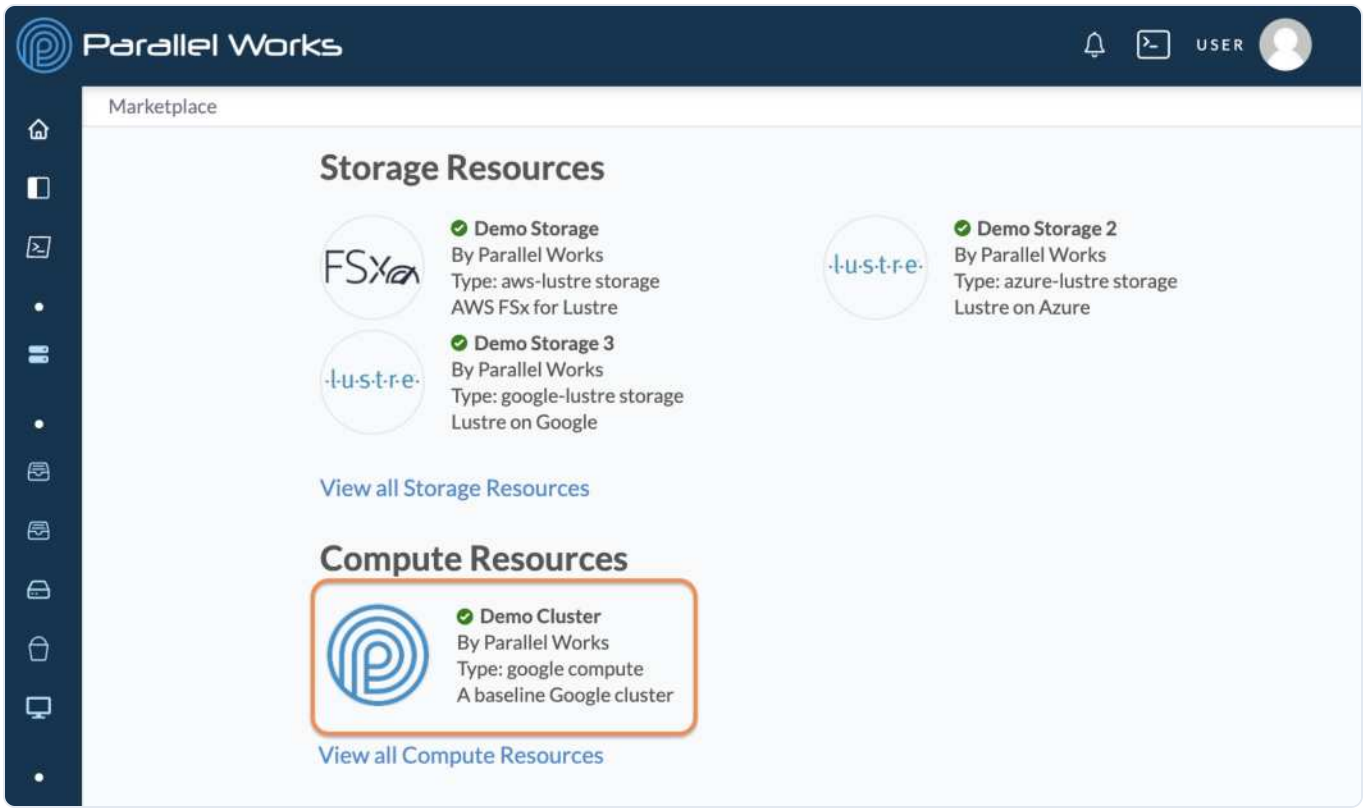


Forking an Item

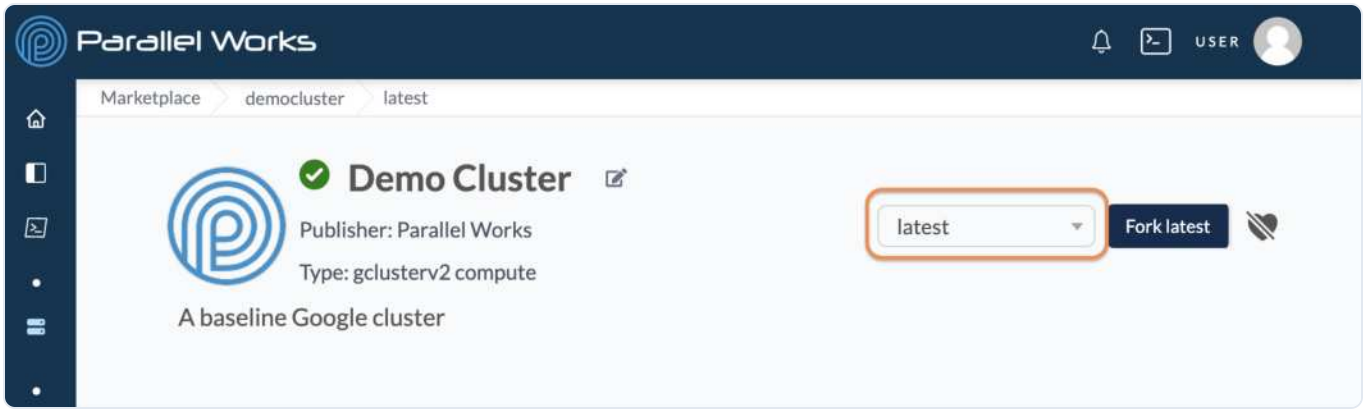
This method allows you to copy an item into your ACTIVATE account.

Forking Compute Resources

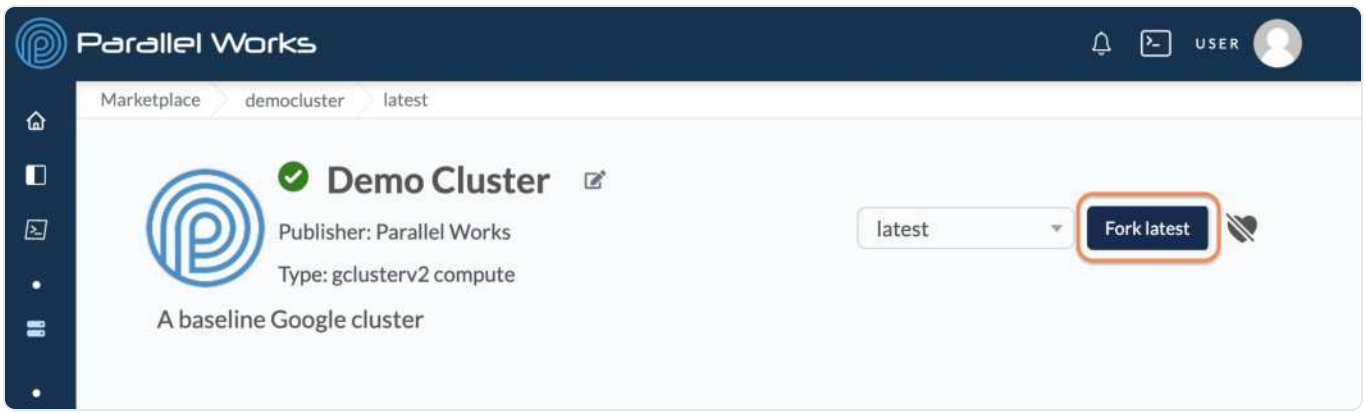
Click the compute resource you'd like to use.



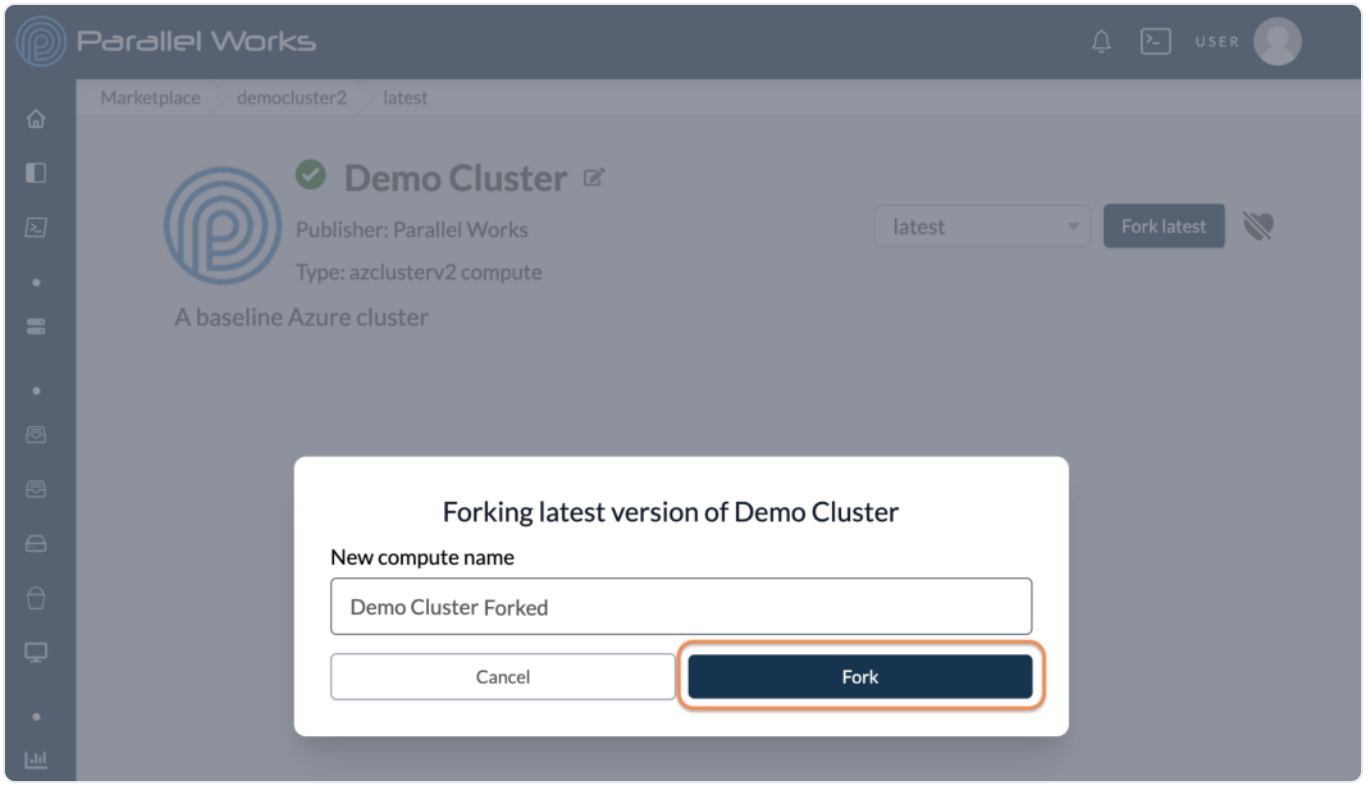
On the next page, click **latest** to open the dropdown menu of versions. Choose the version you'd like to use.



Click the **Fork** button.

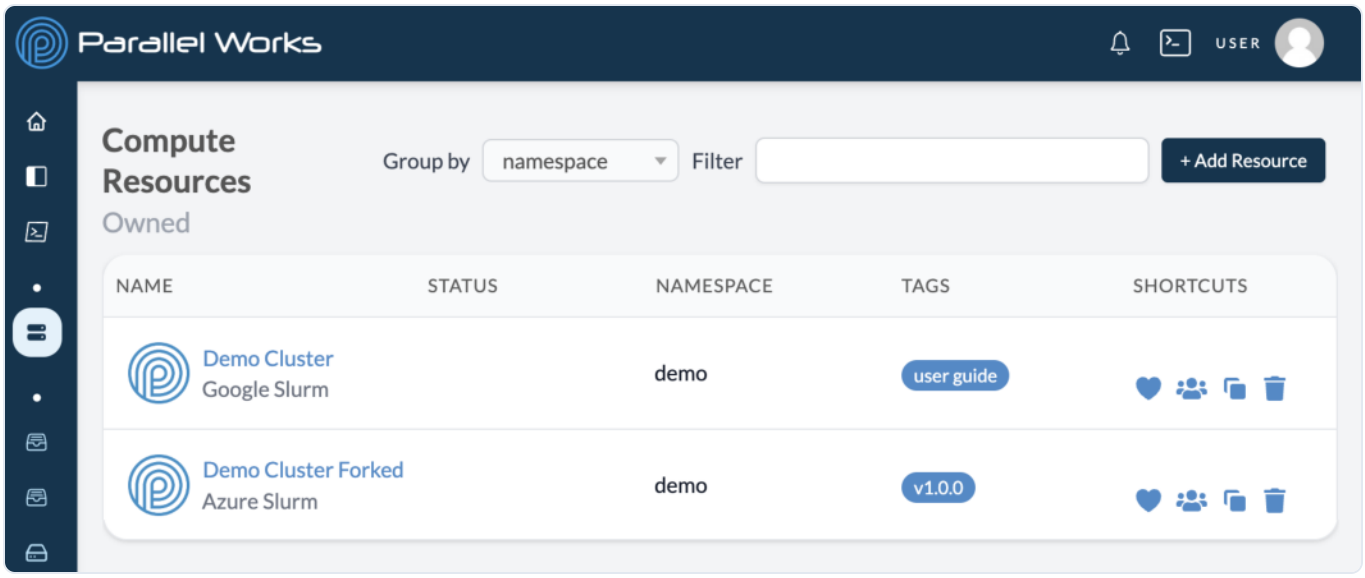


A dialog box will appear. Enter a **New compute name**. Click **Fork**.



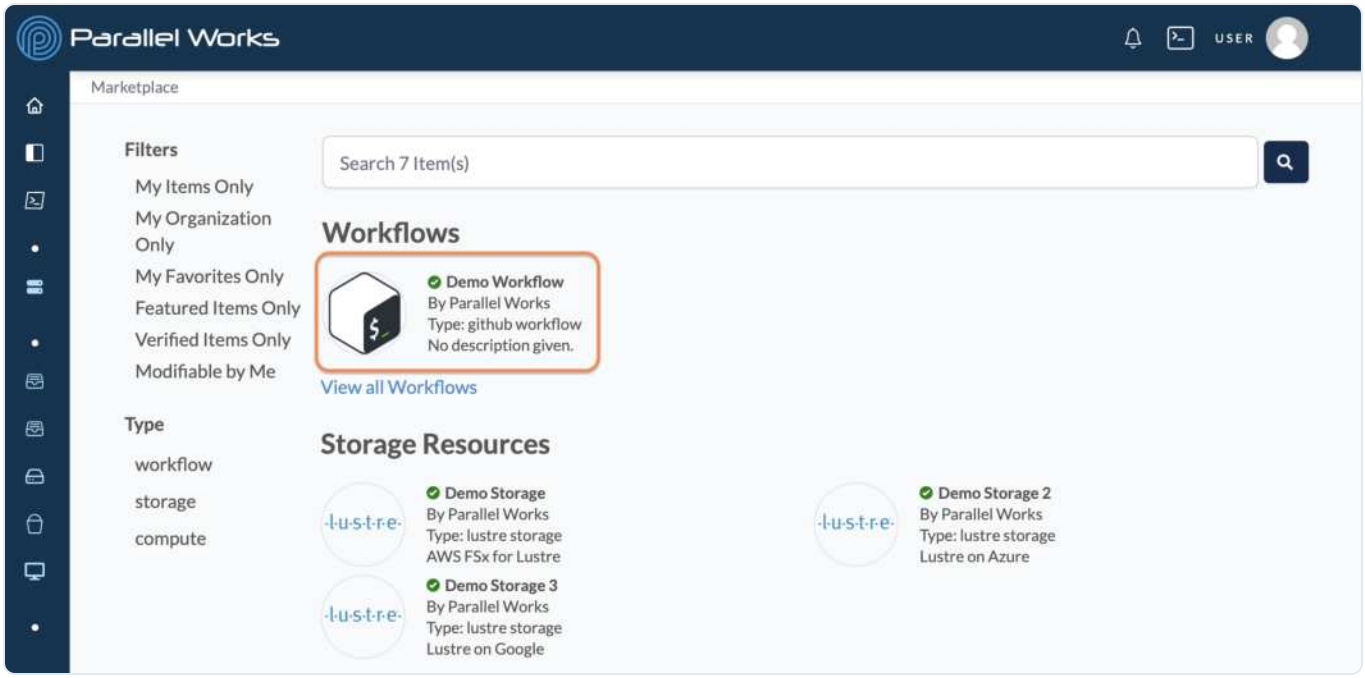
You'll be taken to the new compute resource's configuration page.

The compute resource will also appear on the **Clusters** page. The version number you created the fork from will be included in **Tags**.

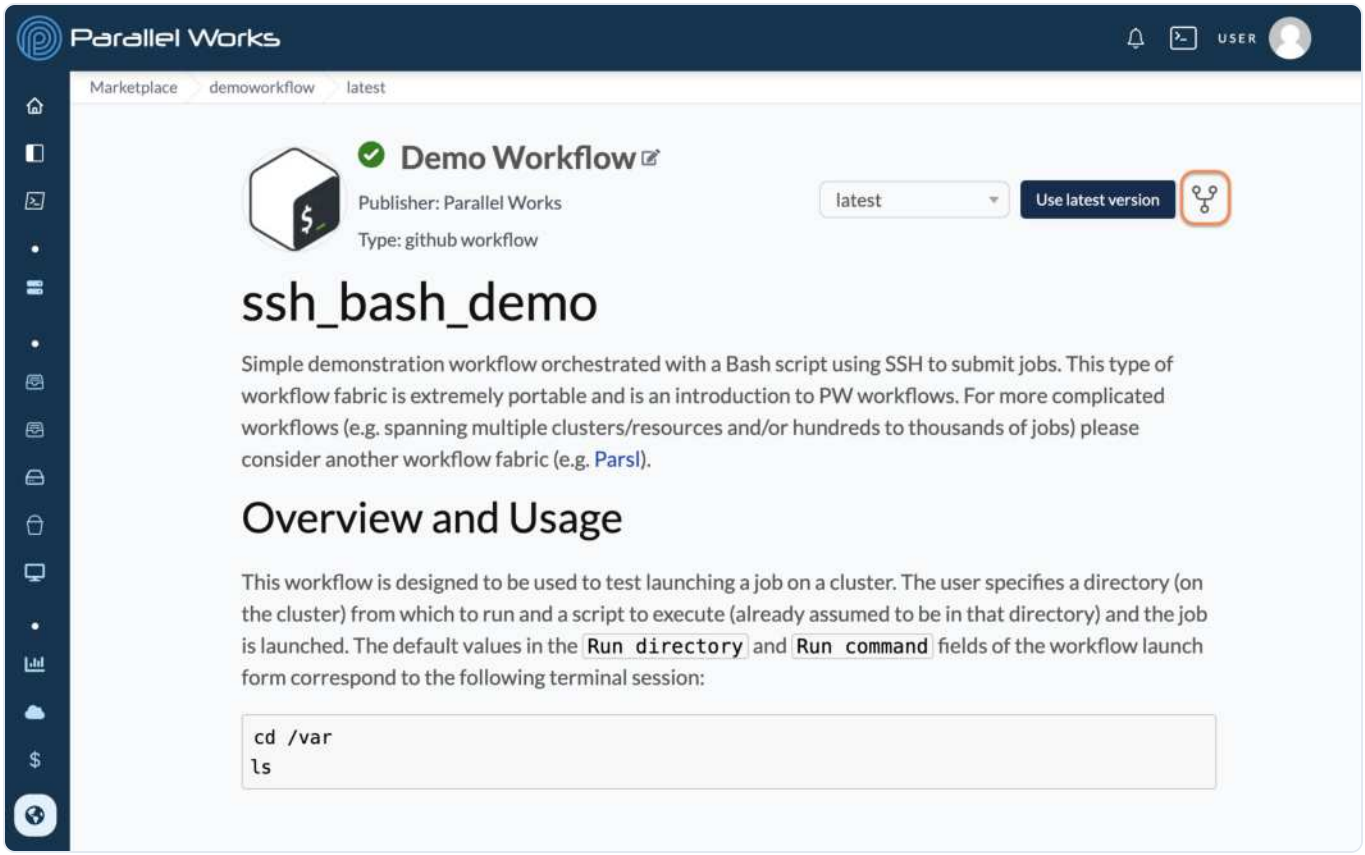


Forking Workflows

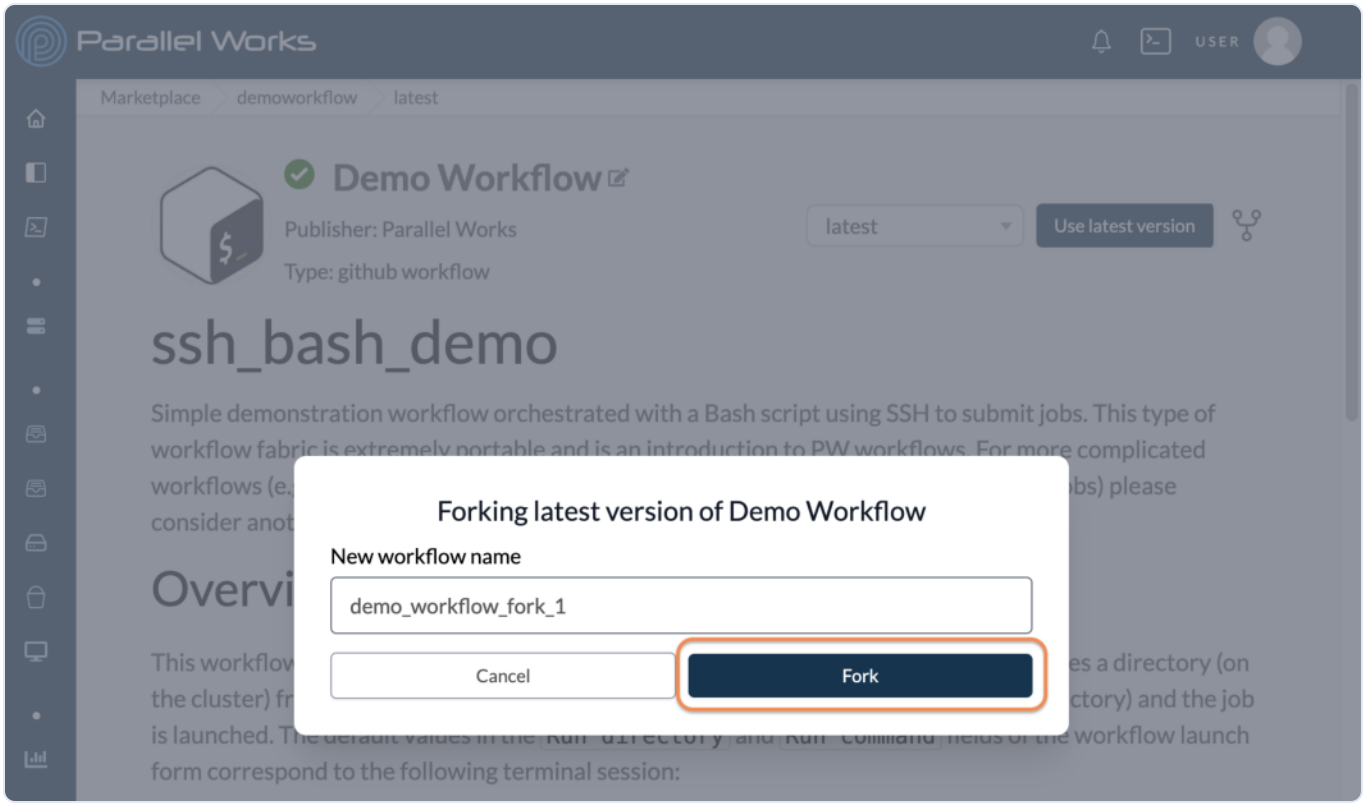
Click the workflow you'd like to use.



On the next page, click the fork icon.



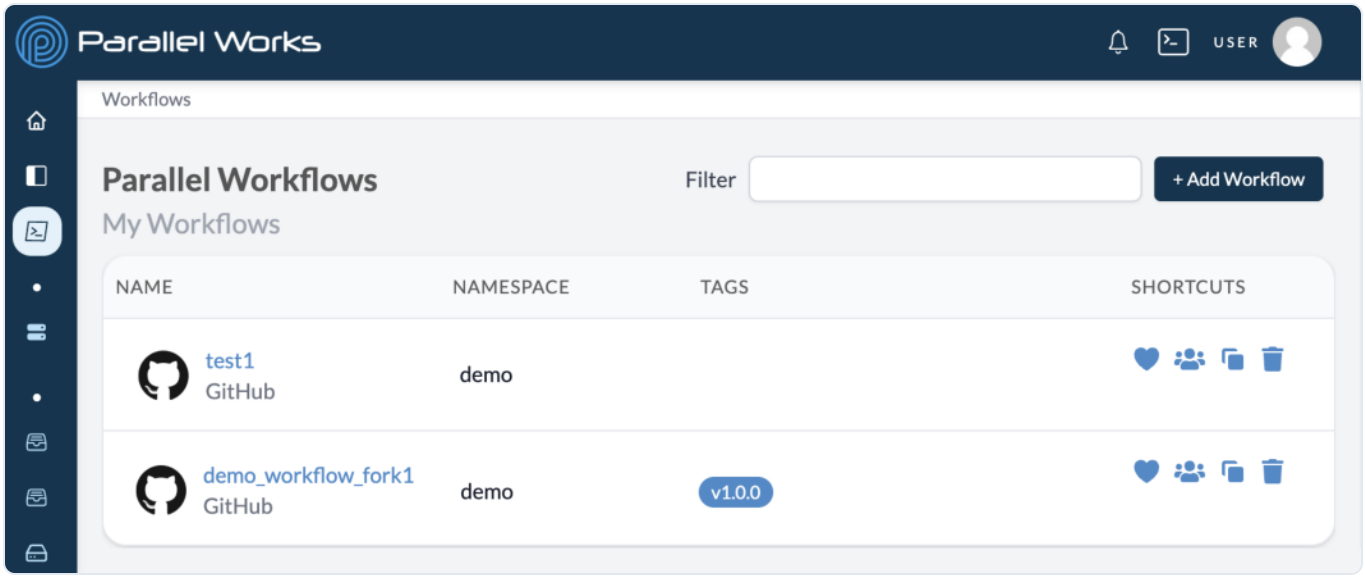
A dialog box will appear. Enter a **New workflow name**. Click **Fork**.



You'll be taken to the new workflow's configuration page.

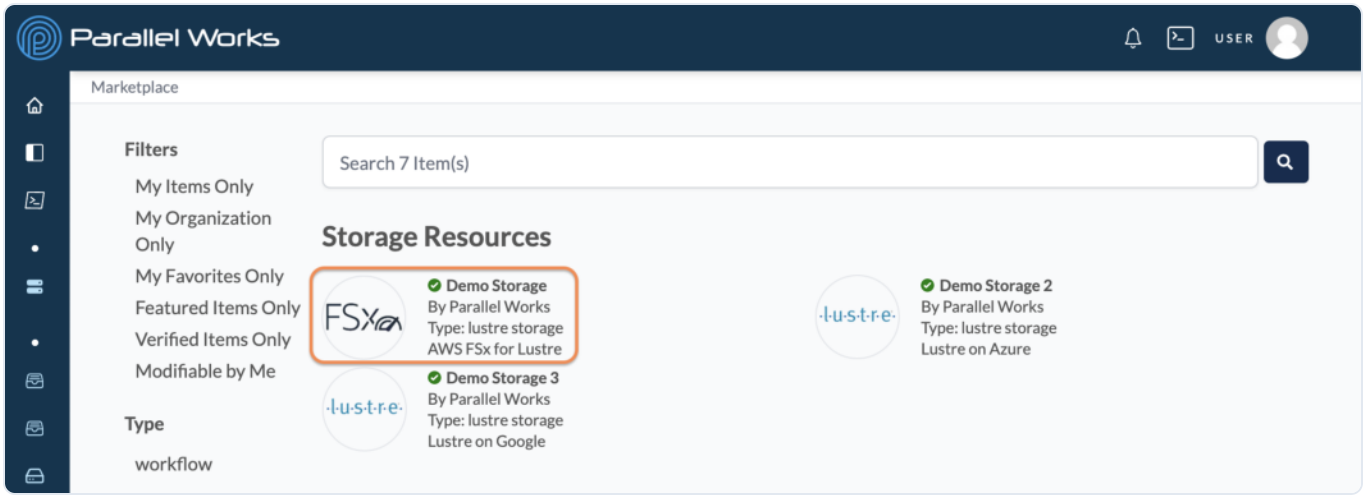
The workflow will also appear on the **Workflows** page beneath **My Workflows**.

Forking automatically creates the new item from the latest published version; the version number you created the fork from will be included in **Tags**.

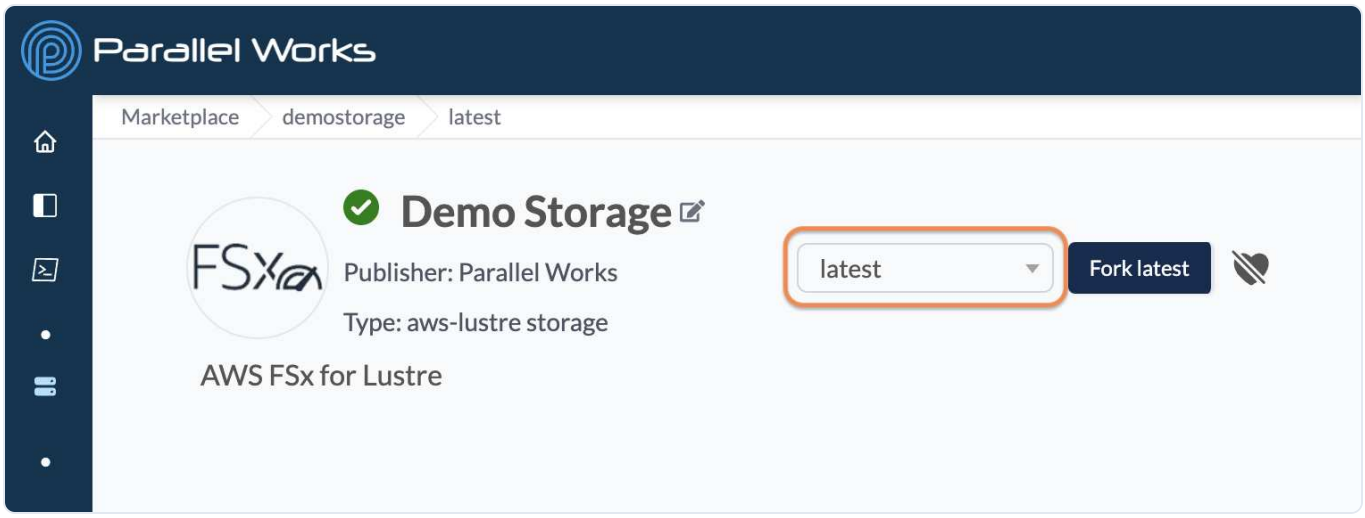


Forking Storage Resources

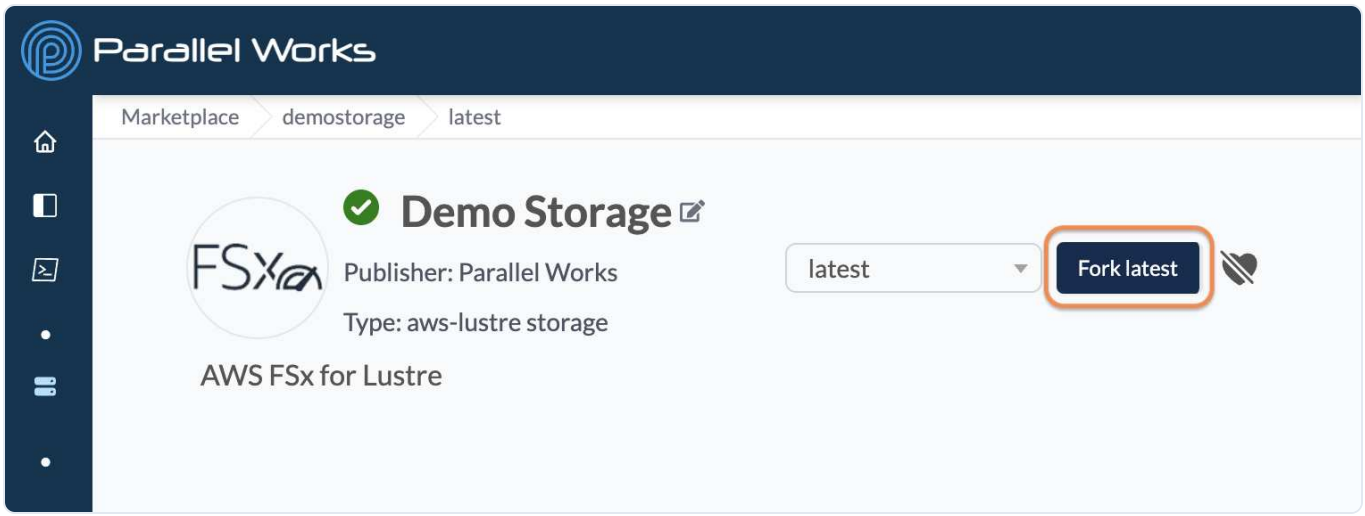
Click the storage resource you'd like to use.



On the next page, click **latest** to open the dropdown menu of versions. Choose the version you'd like to use.

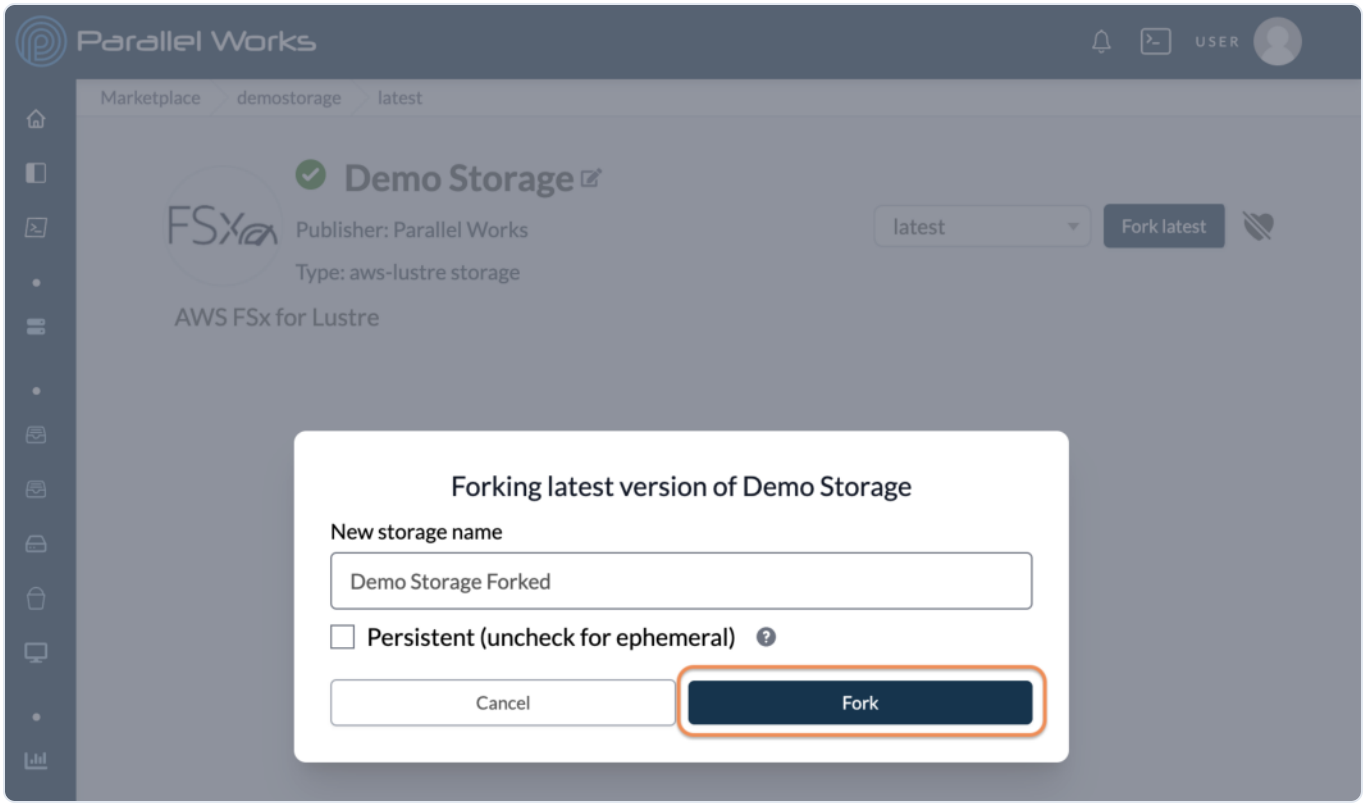


Click the **Fork** button.



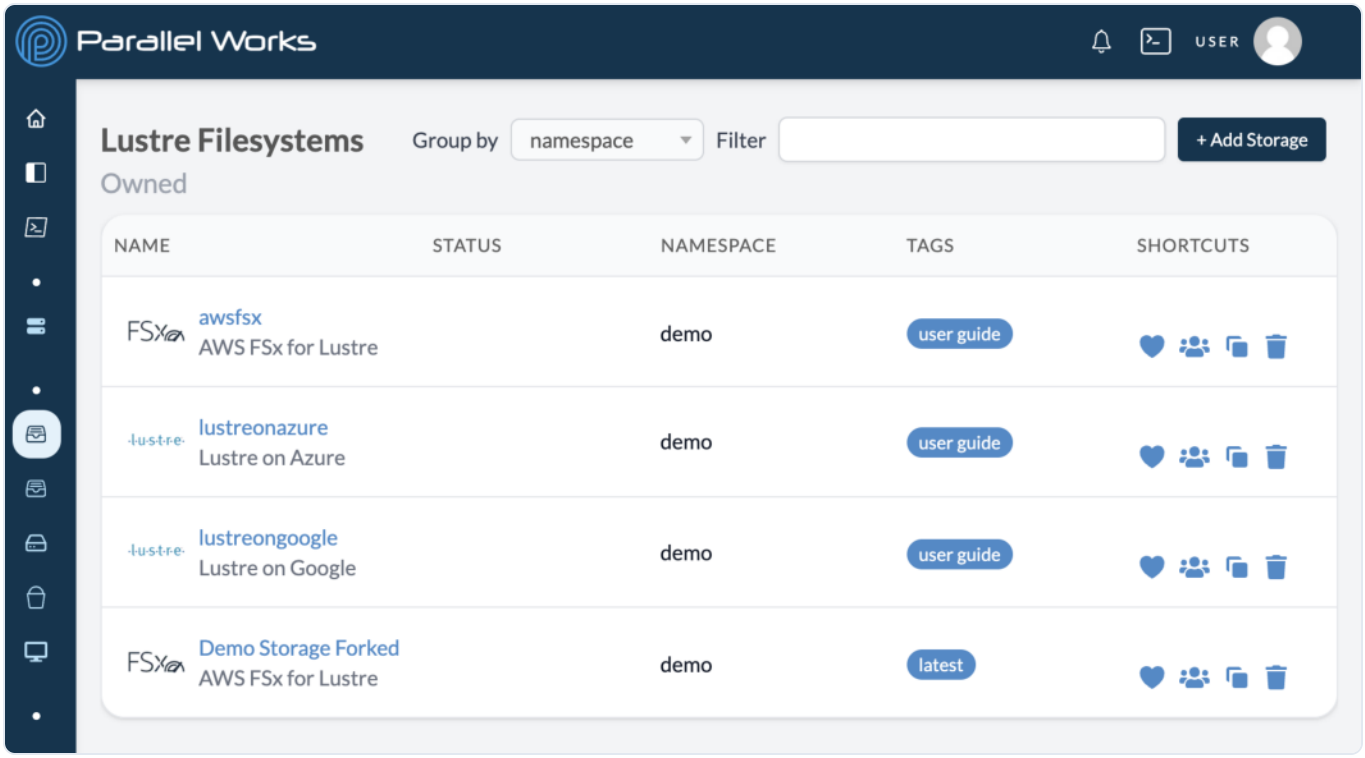
A dialog box will appear. Enter a **New storage name**. Click the checkbox if you want your new resource to be **Persistent**.

Click **Fork**.



You'll be taken to the new storage resource's configuration page.

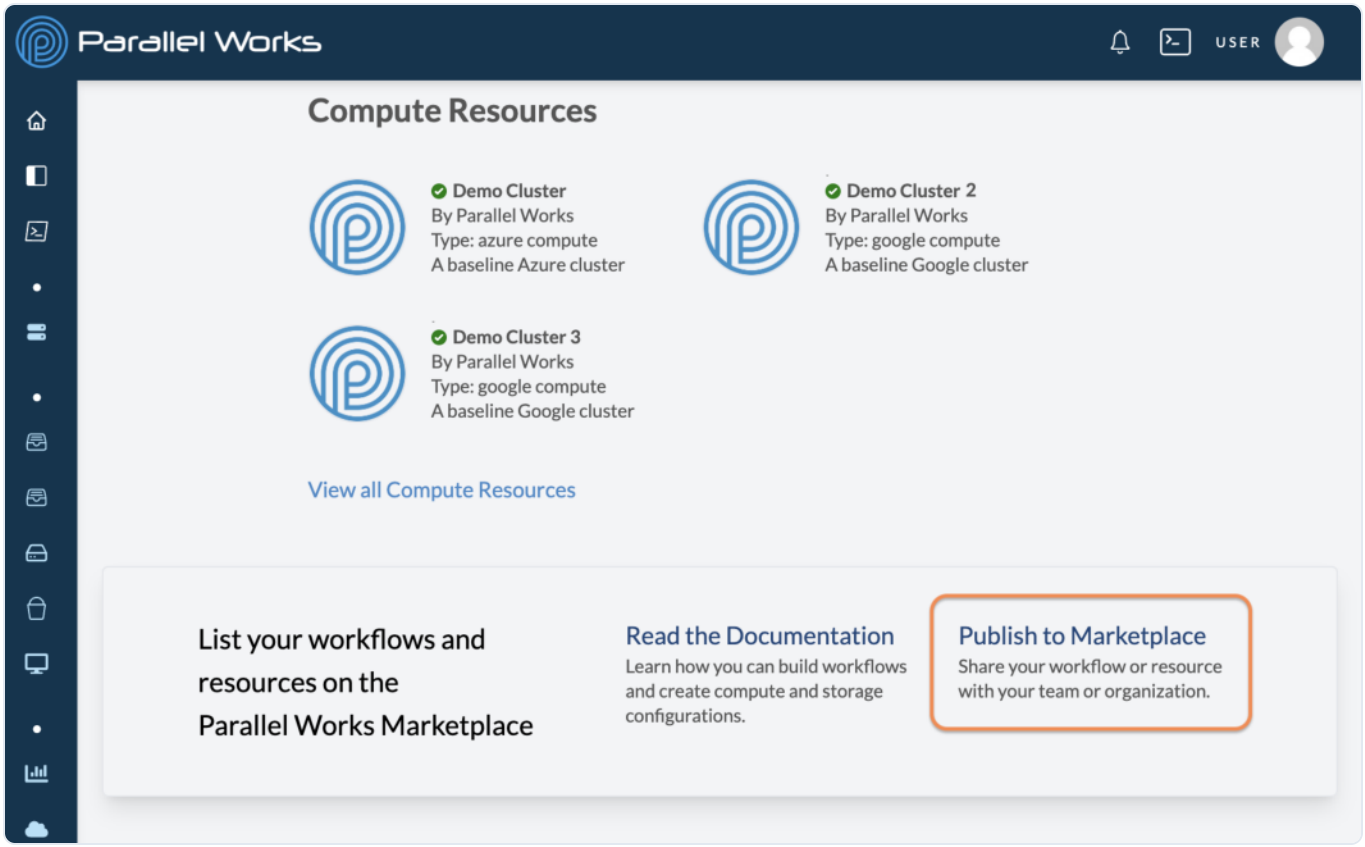
The storage resource will also appear on the **Storage** page with your other storage resources. The version number you created the fork from will be included in **Tags**.



Publishing an Item

You can quickly and easily share an item in your account with other users in your organization.

At the bottom of the Marketplace page, click **Publish to Marketplace**.



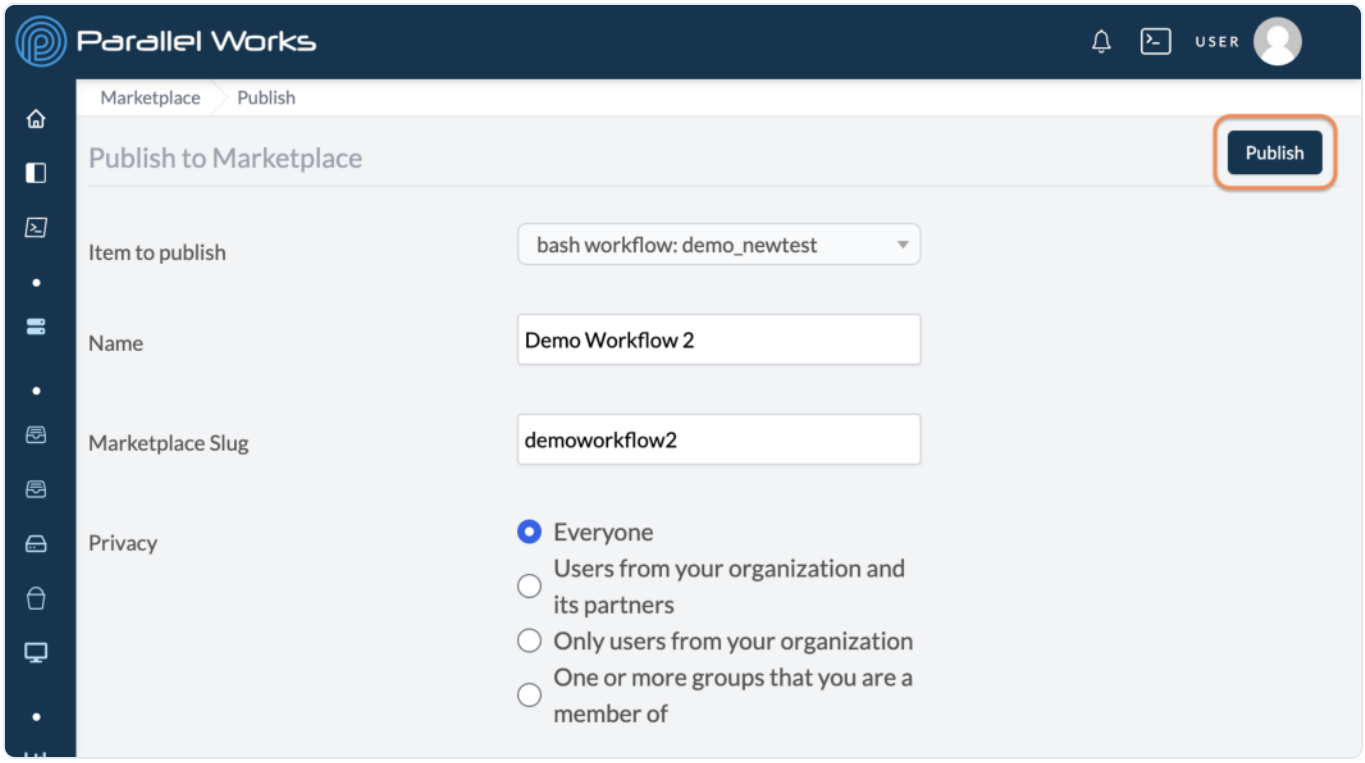
On the next page, choose the **Item to publish** from your account.

Enter a **Name**. This is the item's display name and must be a unique value.

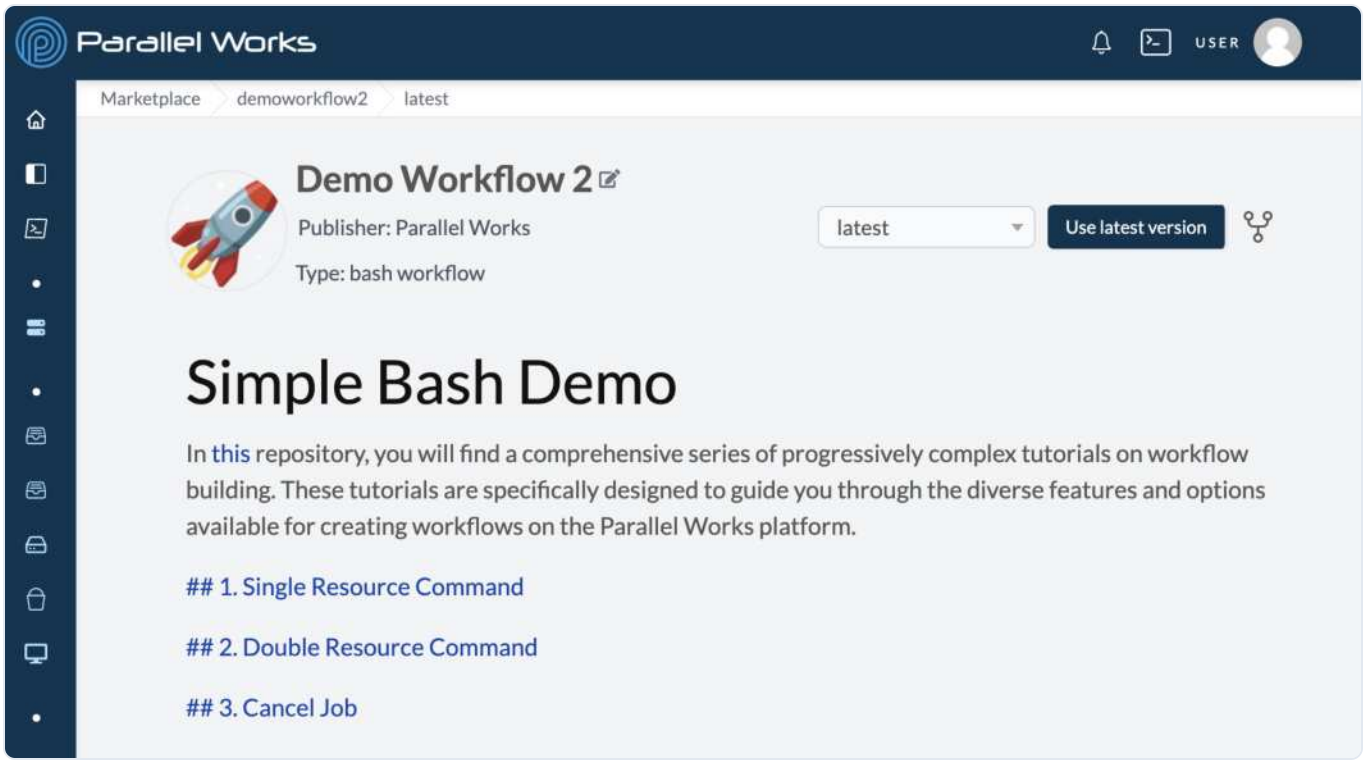
Enter a **Marketplace Slug**. This is a unique identifier that appears at the end of the Marketplace URL when others navigate to the item's page.

Choose the **Privacy** level for your item.

When you're done, click **Publish**.

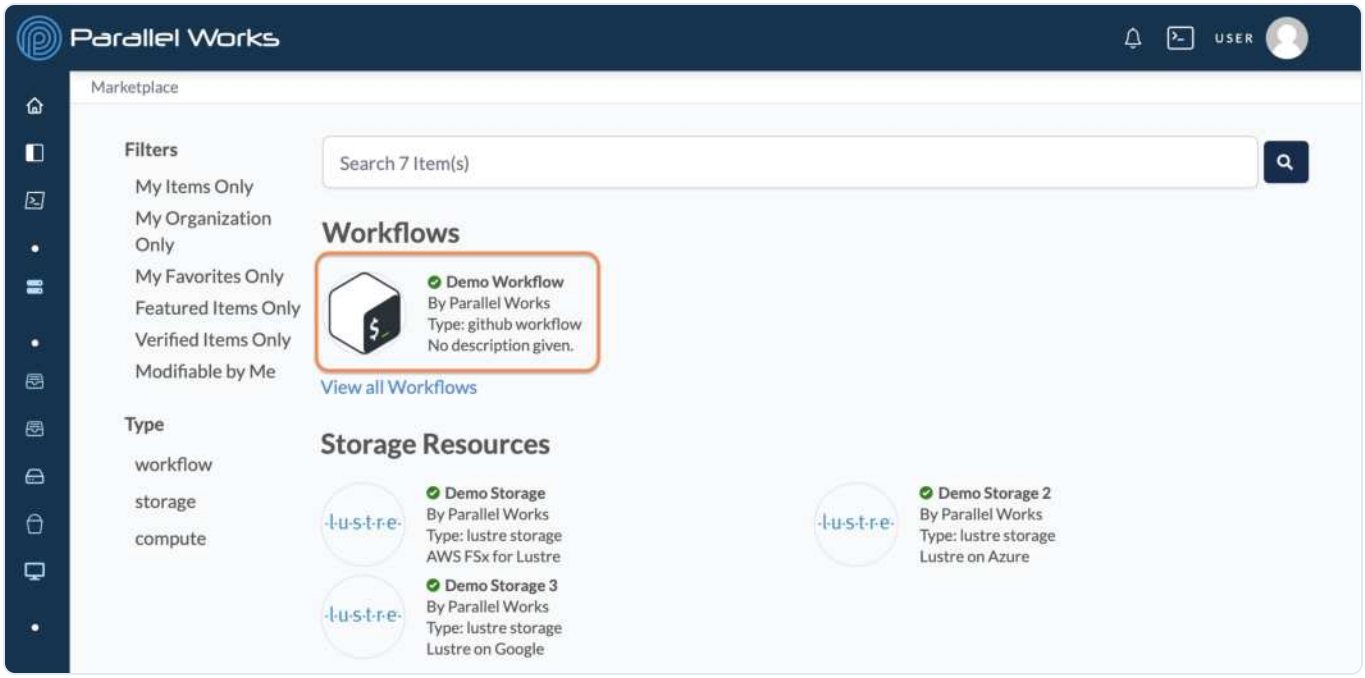


The message *Successfully published item* will appear in the top right corner of your screen, and you'll be taken to the item's Marketplace page.

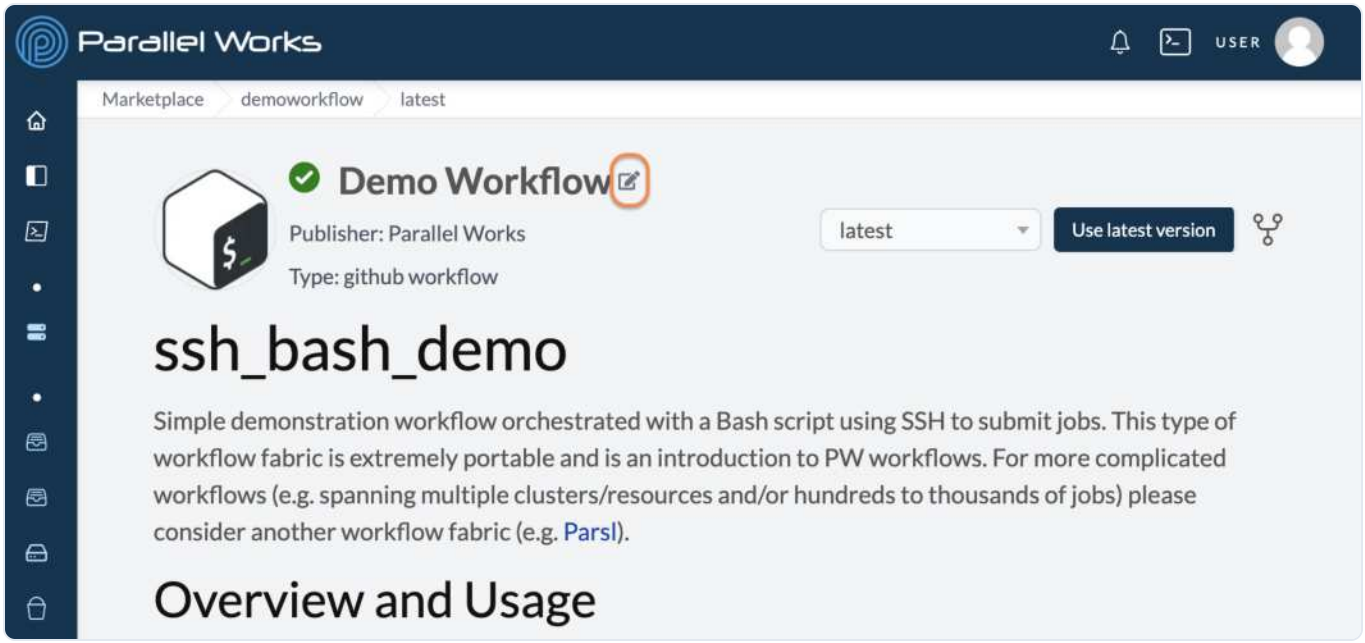


Editing an Item

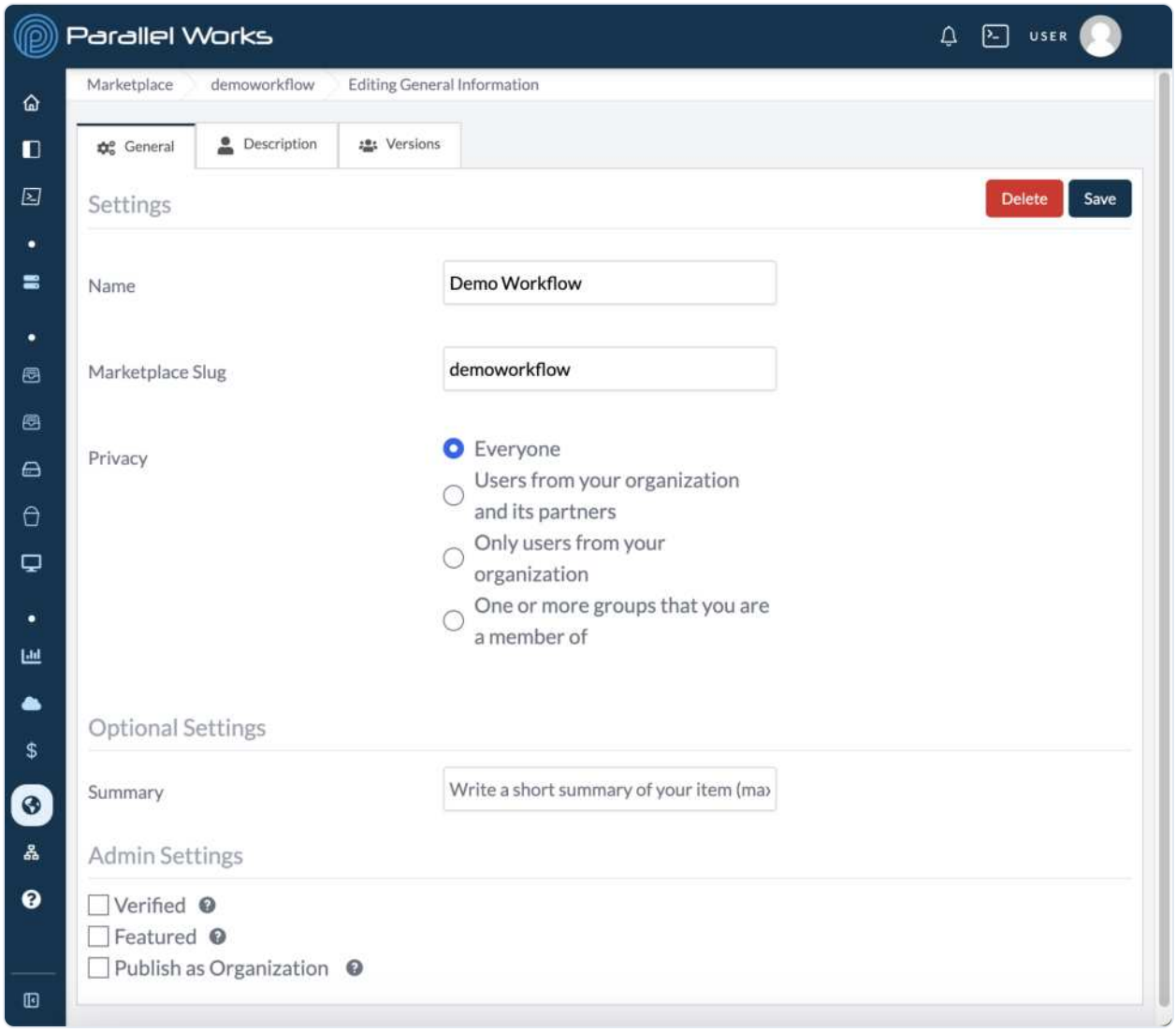
Click the Marketplace item you'd like to edit.



Click the edit icon.



You'll be taken to the item's **General** settings.



You can change the same settings that were chosen when the item was created: **Name**, **Marketplace Slug**, and **Privacy**.

Use the **Summary** field to add a short description that will be displayed beneath the item's thumbnail on the Marketplace.

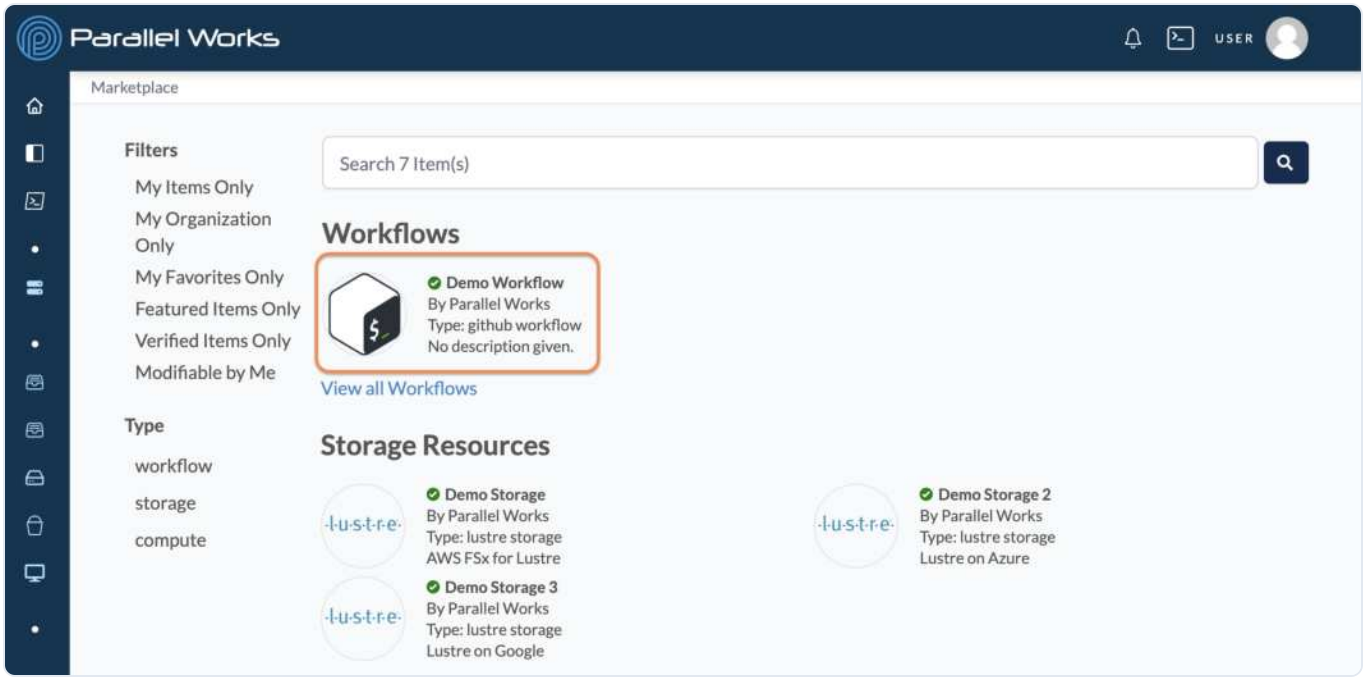
Admin Settings

- If you check **Verified**, a green checkmark will be displayed next to the item's name on the Marketplace.
- If you check **Featured**, the item will be displayed at the top of the Marketplace for all users in your organization.
- If you check **Publish as Organization**, the item's creator will be your organization's name instead of your username.

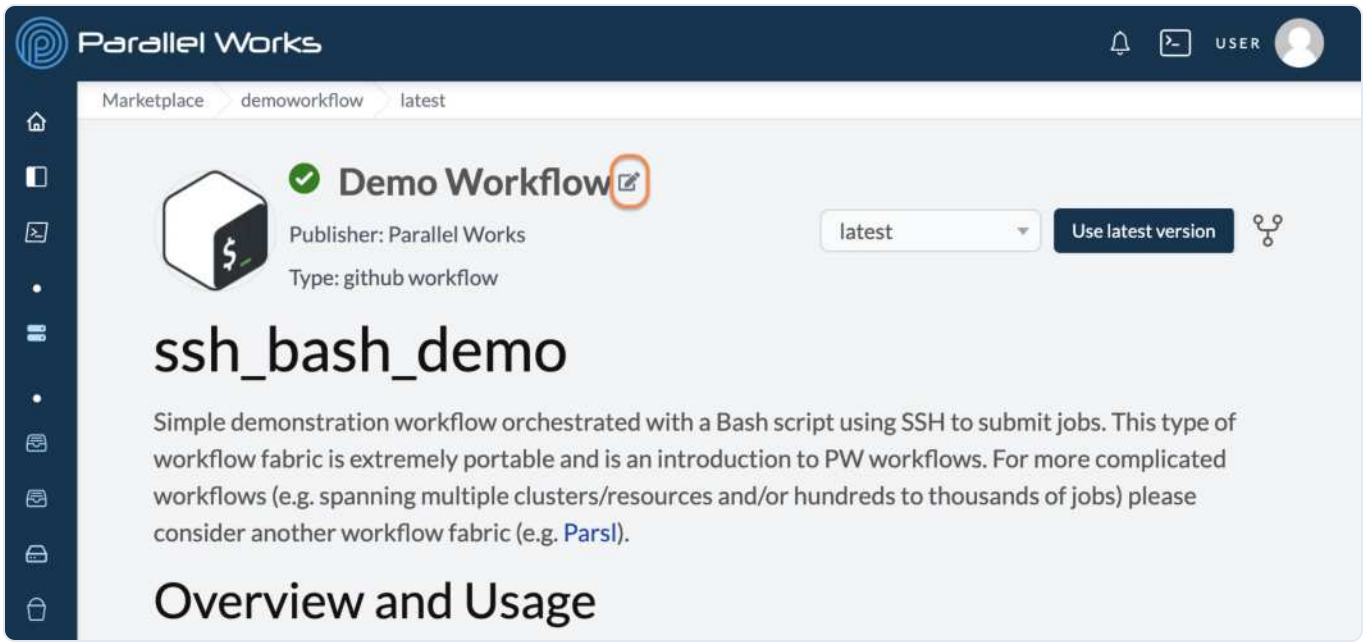
If you click the **Description** tab, you can add a markdown description that will be displayed on the item's Marketplace page.

Publishing a New Item Version

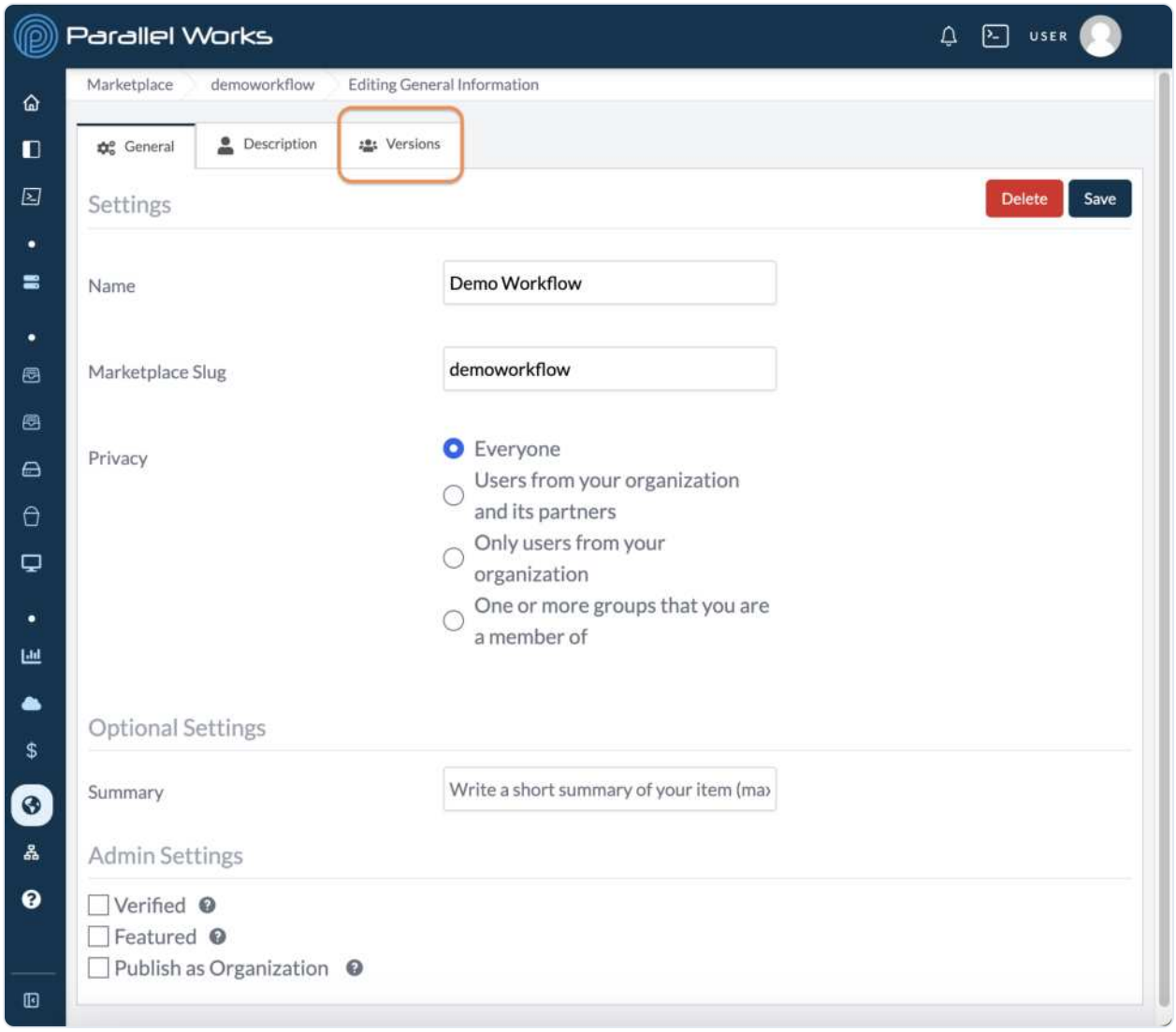
Click the Marketplace item you'd like to version.



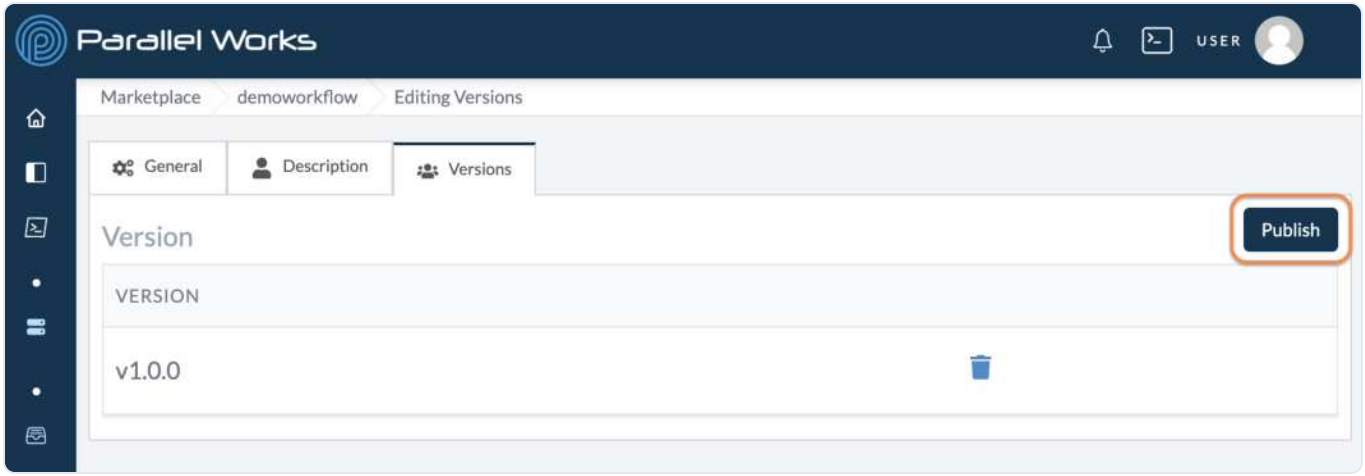
Click the edit icon.



Click **Versions**.



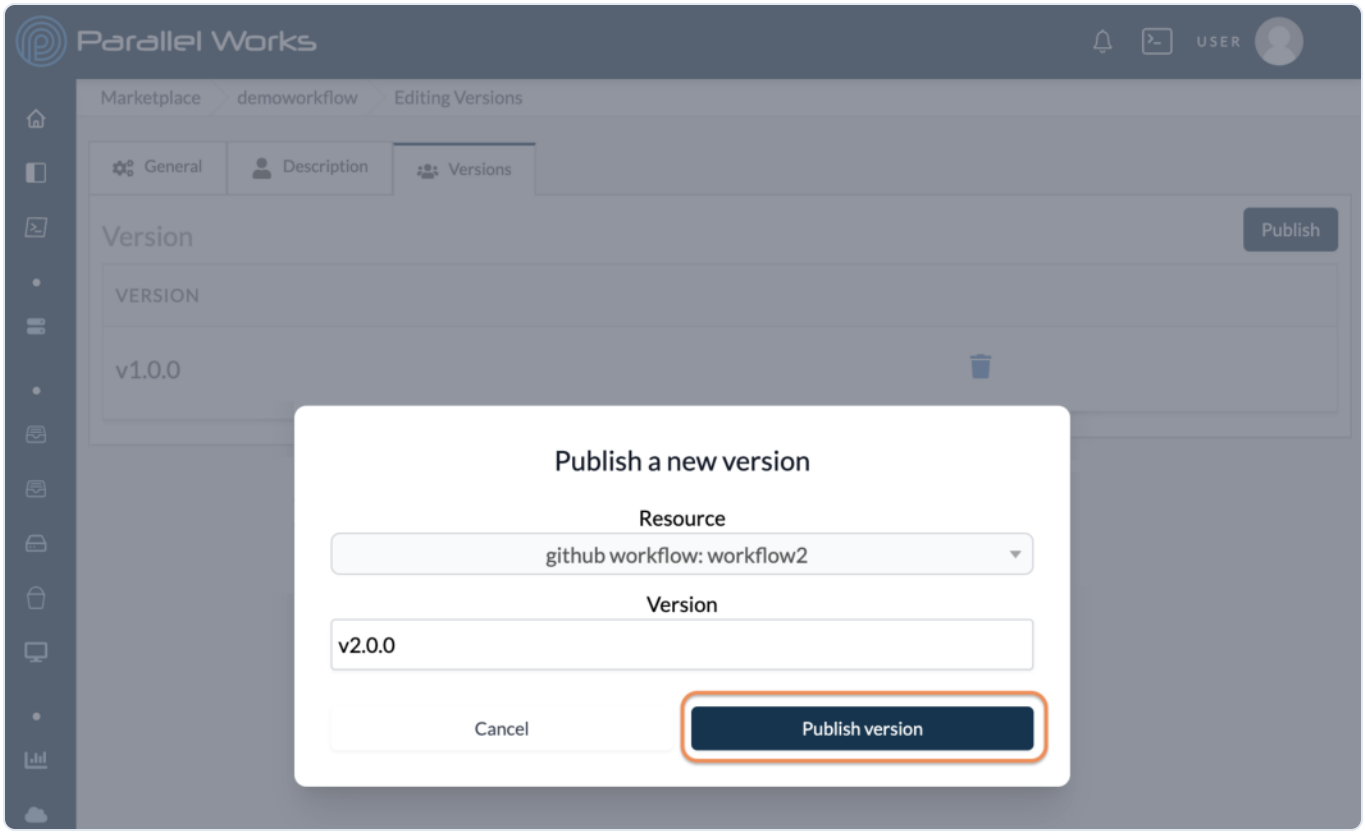
Click **Publish**.



Choose the **Resource** that will be this item's new verison.

Enter a **Version** number. This value must start with "v" followed by at least two characters. We recommend the format v0.0.0 to accomodate a range of version changes.

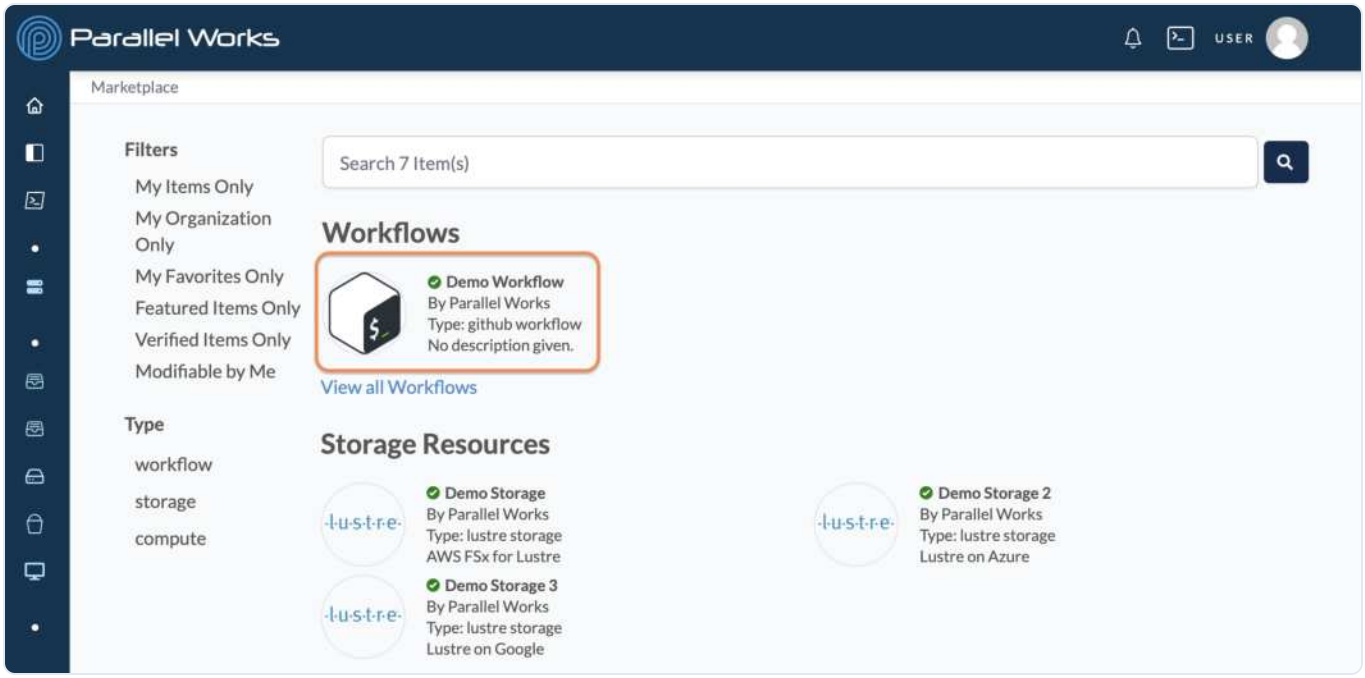
Click **Publish version**.



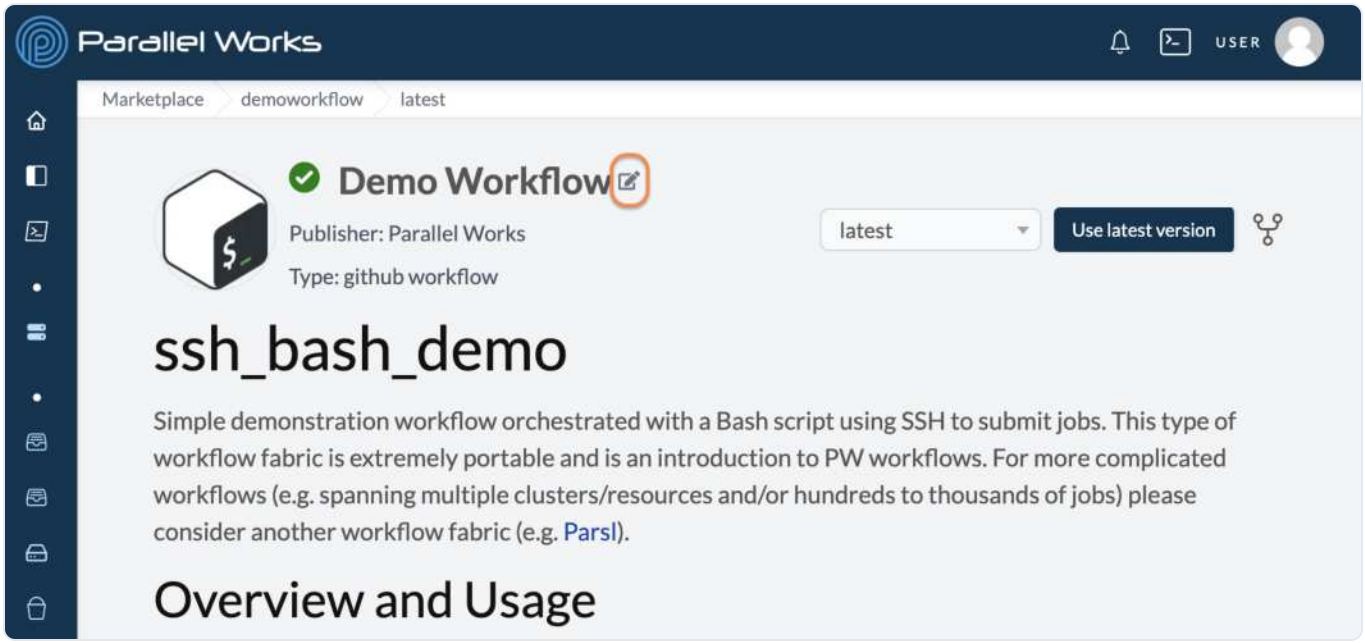
The message *Version published successfully* will appear in the top right corner of your screen.

Deleting an Item

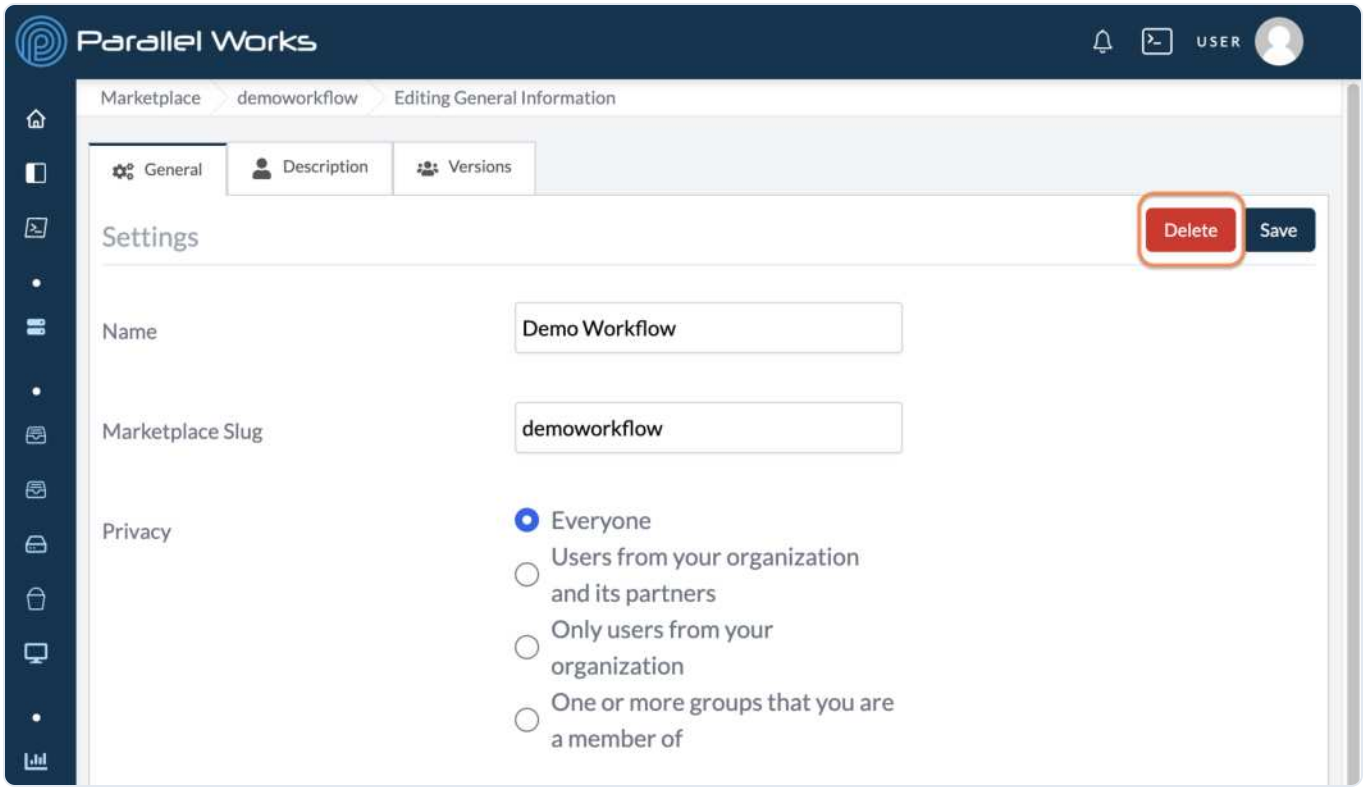
Click the Marketplace item you'd like to delete.



Click the edit icon.



Click the **Delete** button.



The message *Item deleted successfully* will appear in the top right corner of your screen.

Further Reading

For more information about creating your own workflows, please see **Building Workflows**.

Account Settings

Manage your personal account preferences and settings in ACTIVATE.

Settings Areas

Authentication

Manage your authentication credentials:

- SSH keys for cluster access
- API keys for programmatic access
- Token generation

Cloud Snapshots

Create and manage snapshots of your cloud resources for backup and recovery.

Customizing Layouts

Personalize your ACTIVATE workspace:

- Customize your home dashboard widgets
- Configure sidebar navigation preferences

Managing Notifications

Control how and when you receive notifications about your resources and workflows.

Resetting Password

Update your account password when needed.

Restarting Your Workspace

Restart your user workspace if you encounter issues.

Quick Links

- Authentication - Manage SSH and API keys
- Customizing Layouts - Personalize your dashboard
- Managing Notifications - Configure alerts

Authentication

Managing API Keys

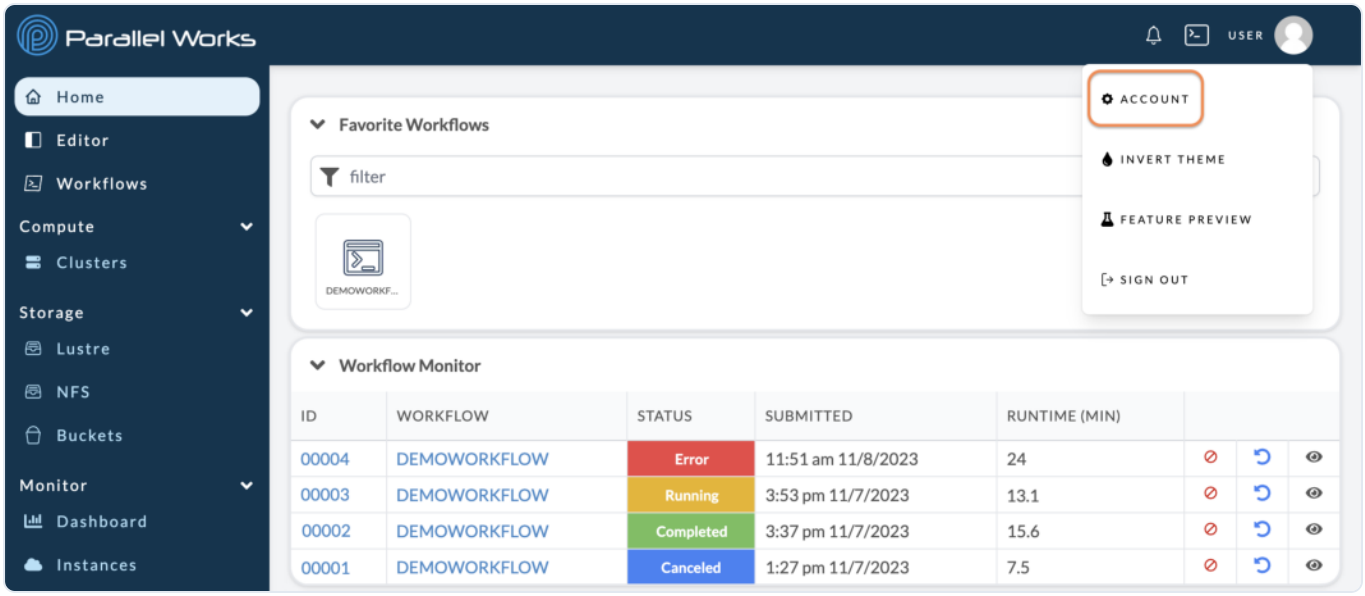
You can use an API key to access Parallel Works via its REST API.

> **NOTE**

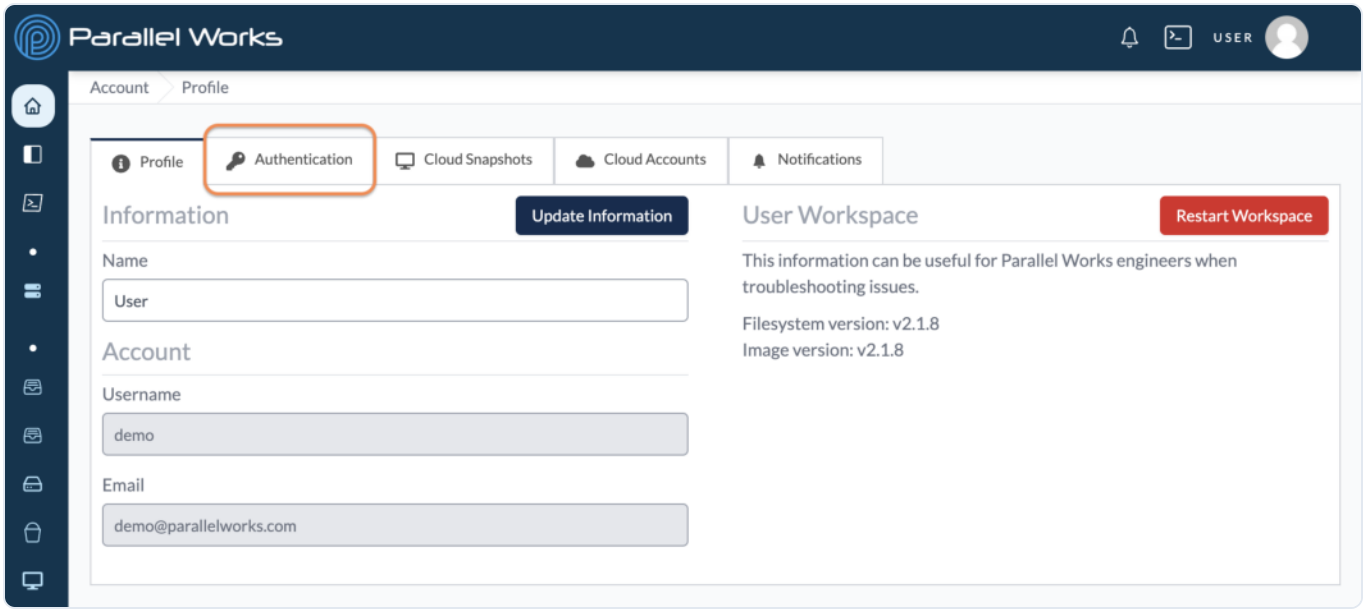
> Please note that API keys act as an alternate means to access your account and should be treated with the same care as your login password.

Create an API Key

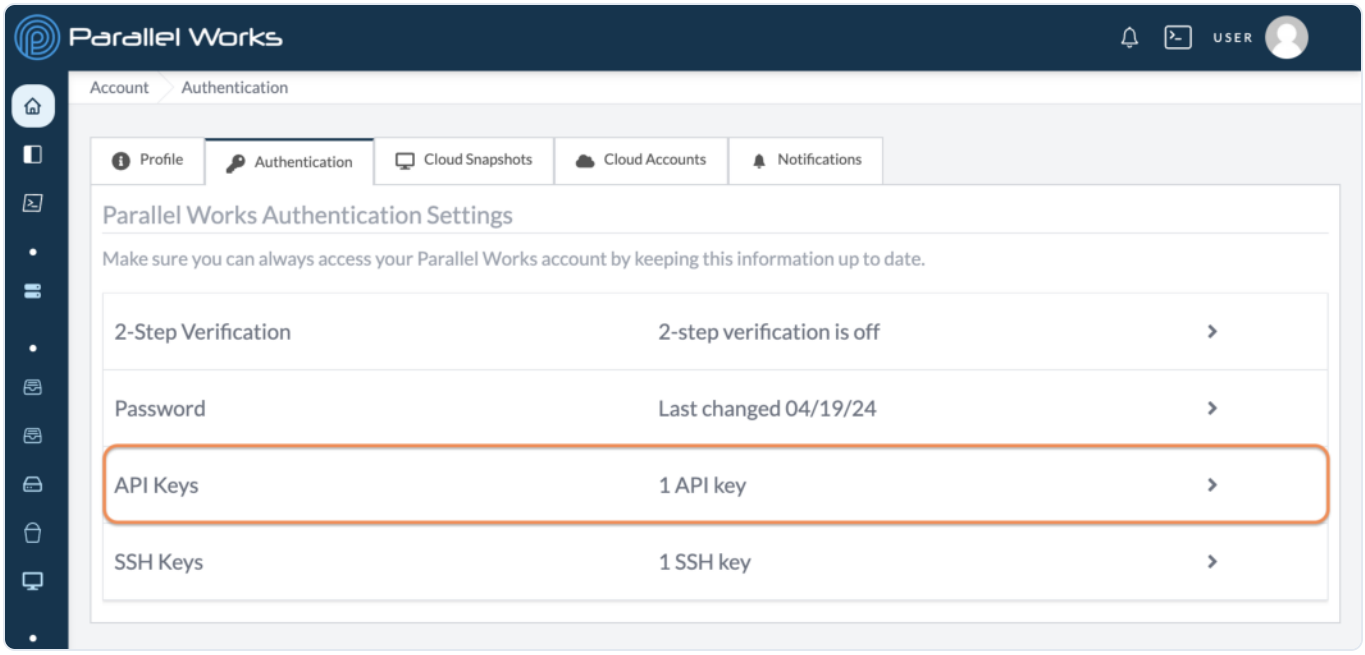
Navigate to your account settings (**Username** > **Account**).



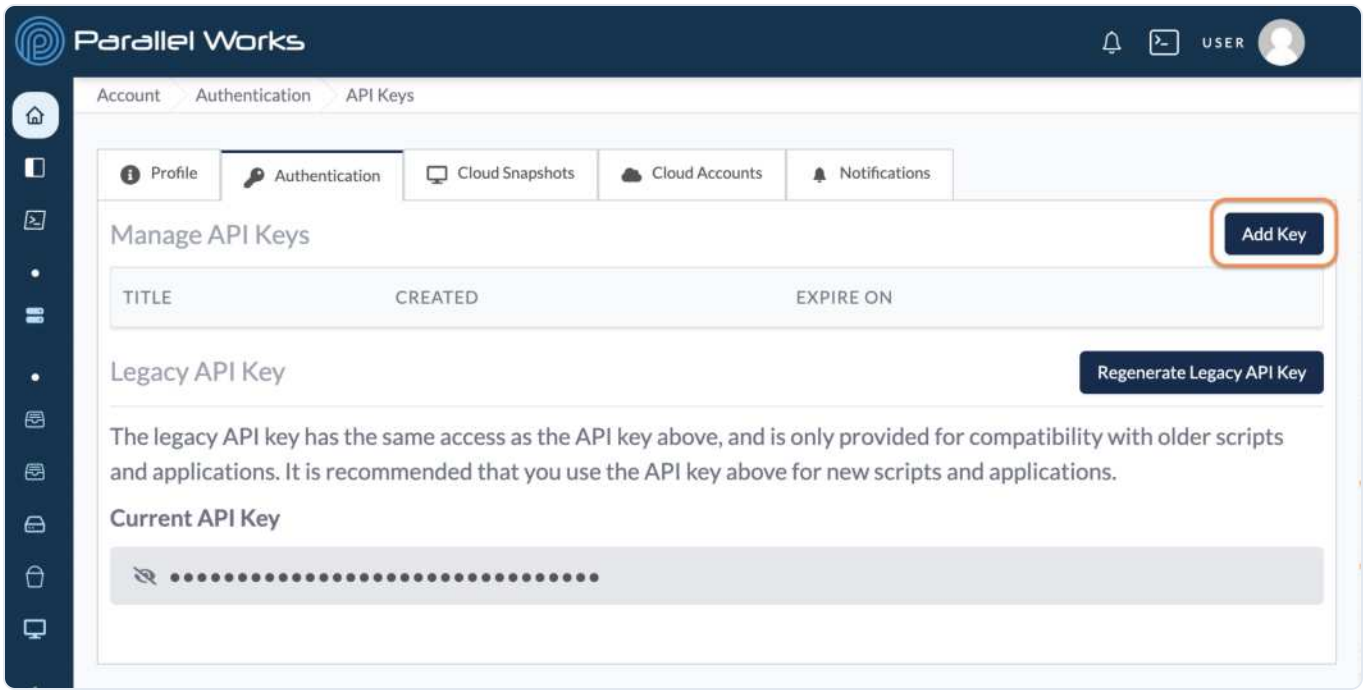
On the next page, click **Authentication**.



Click **API Keys**.



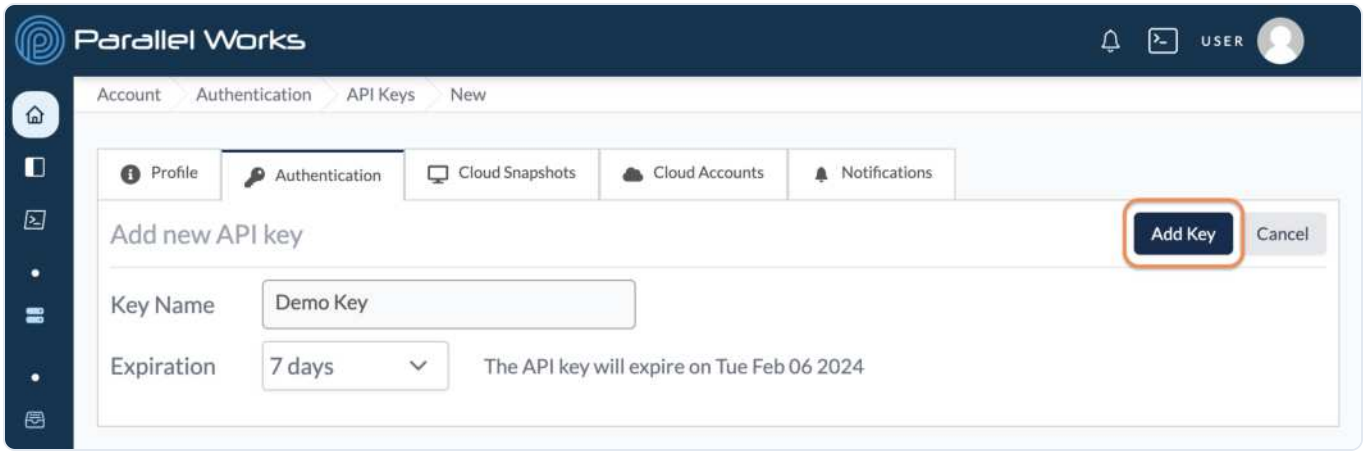
Click **Add Key**.



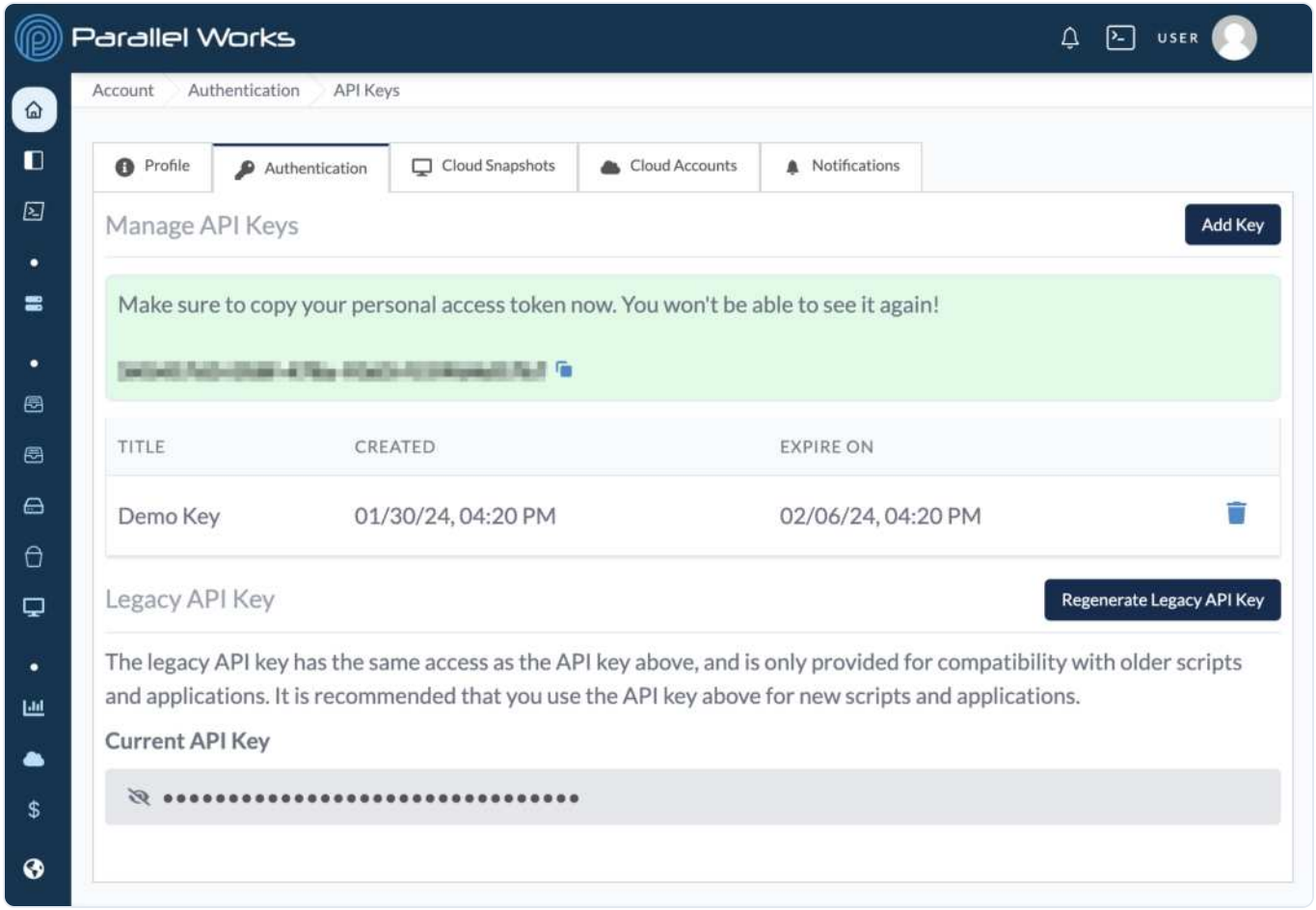
Enter a **Key Name** for the new API key.

If your organization permits users to select their own expiration date for API keys, use the **Expiration** dropdown menu to select when your key will expire. You can choose no expiration or a date in 7, 30, 60, or 90 days. Shorter expiration periods are more secure.

Click **Add Key**.

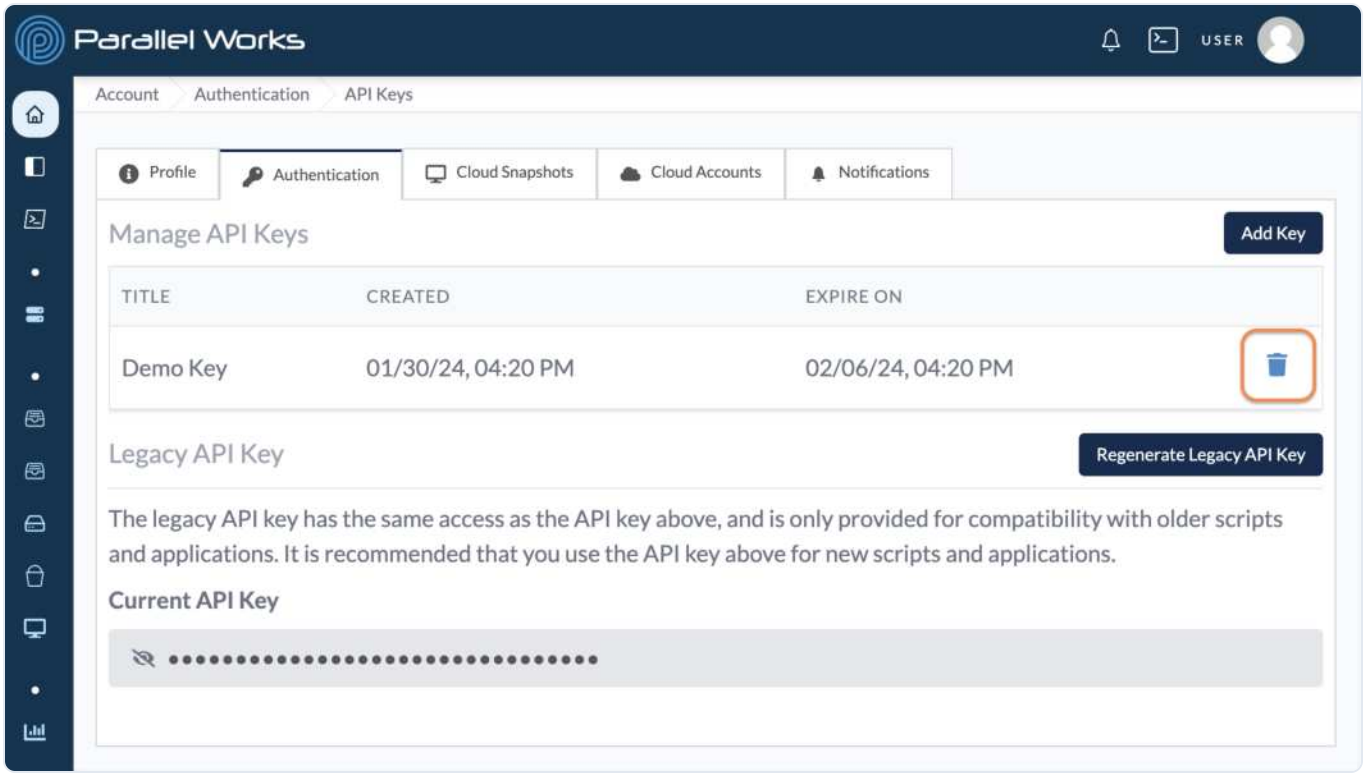


Your new API key will be displayed on the next page. You can use the copy button to copy and save the key in a secure place. Once you refresh or navigate away from this page, you won't be able to see the key again.



Delete an API Key

Navigate to **Account** > **Authentication** > **API Keys** and click the delete icon to remove an API key from your ACTIVATE account.



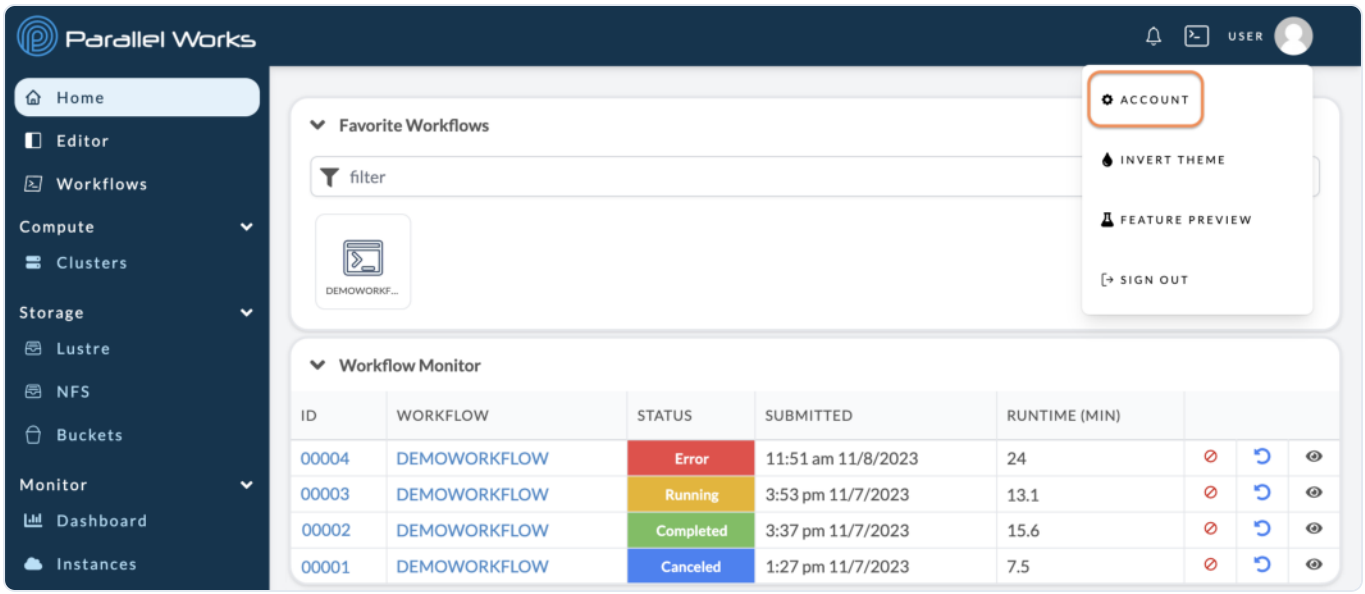
A dialog box will appear with more information about deleting your key. Click **Delete API Key** to confirm the deletion. The key will be removed from your account immediately.

Managing Tokens

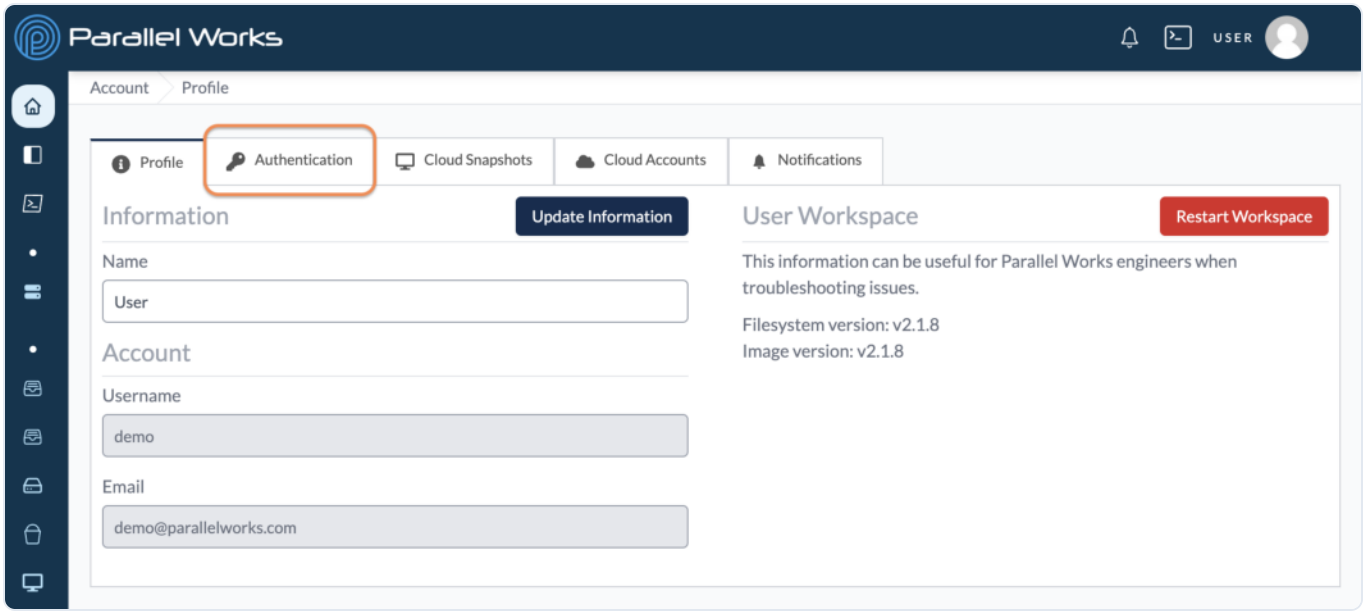
You can create a short-term token and use it to authenticate our CLI after installing locally. These tokens last 24 hours, after which you will need to re-authenticate.

Create a Token

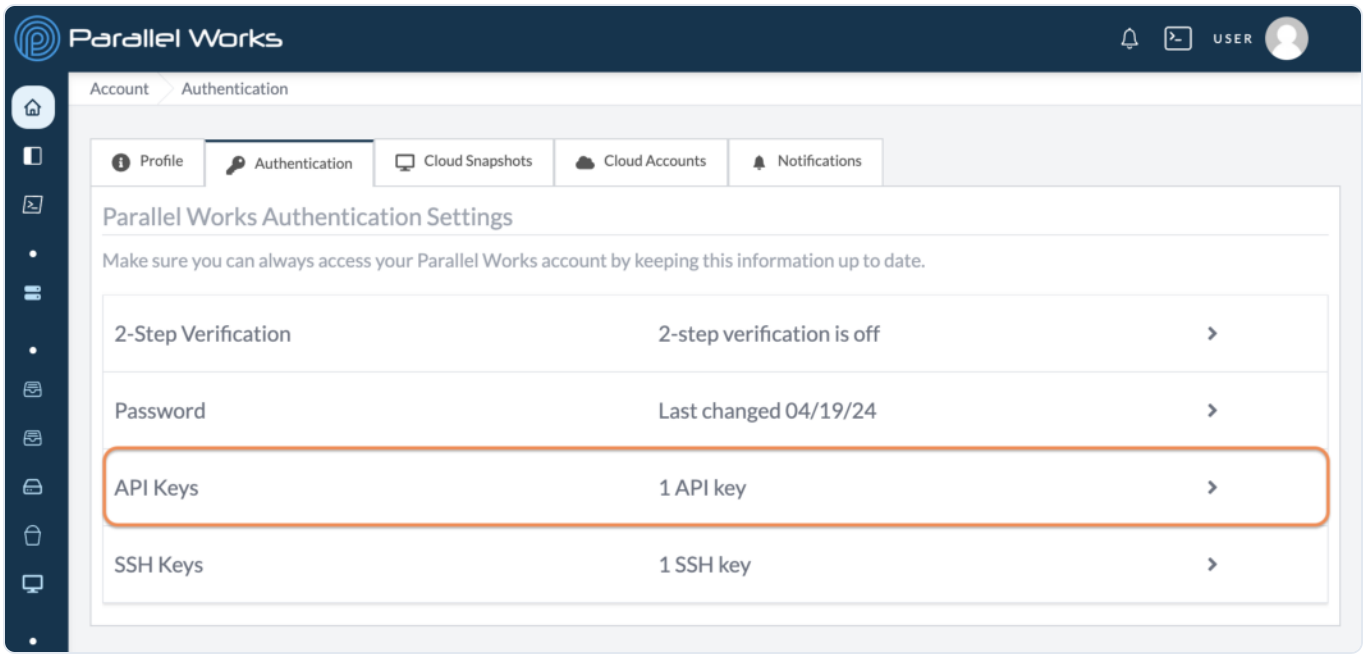
Navigate to your account settings (**Username** > **Account**).



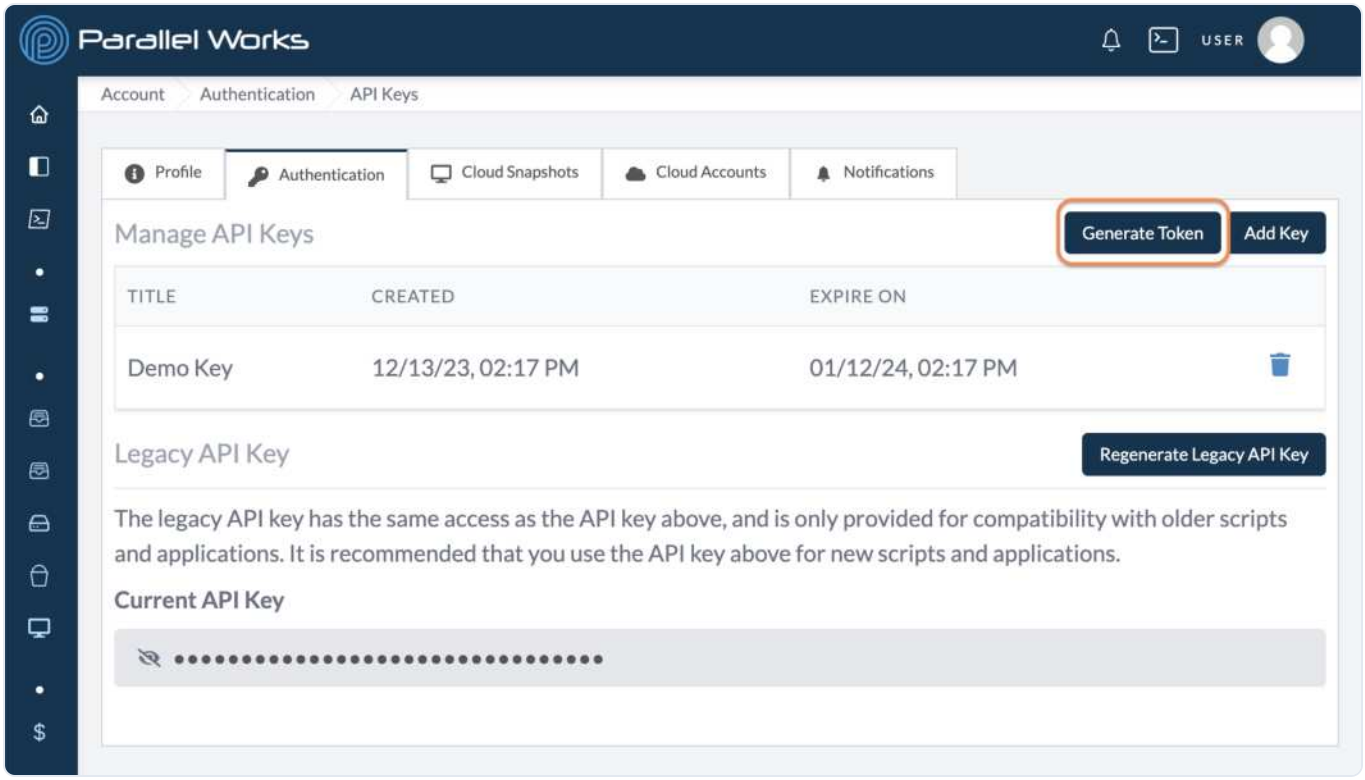
On the next page, click **Authentication**.



Click **API Keys**.

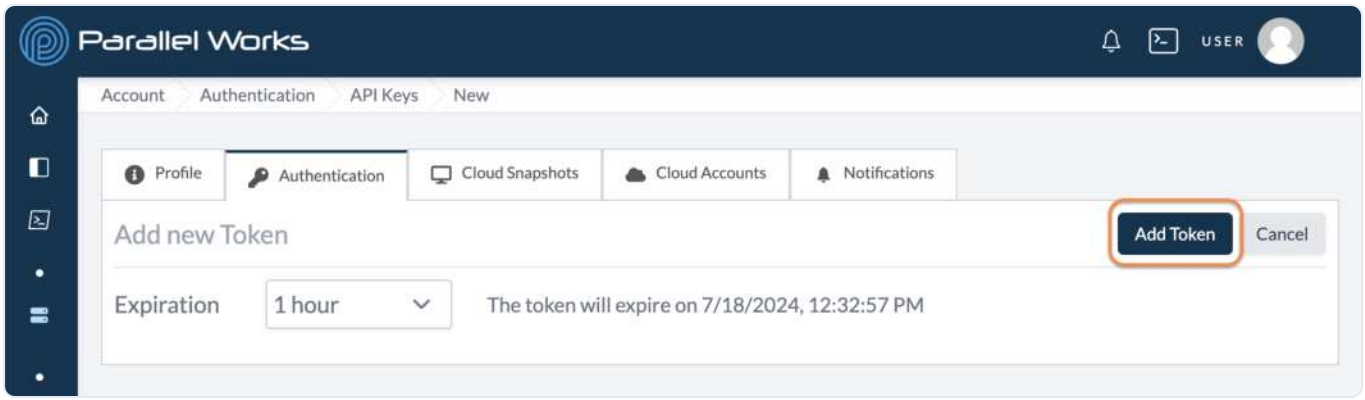


Click **Generate Token**.

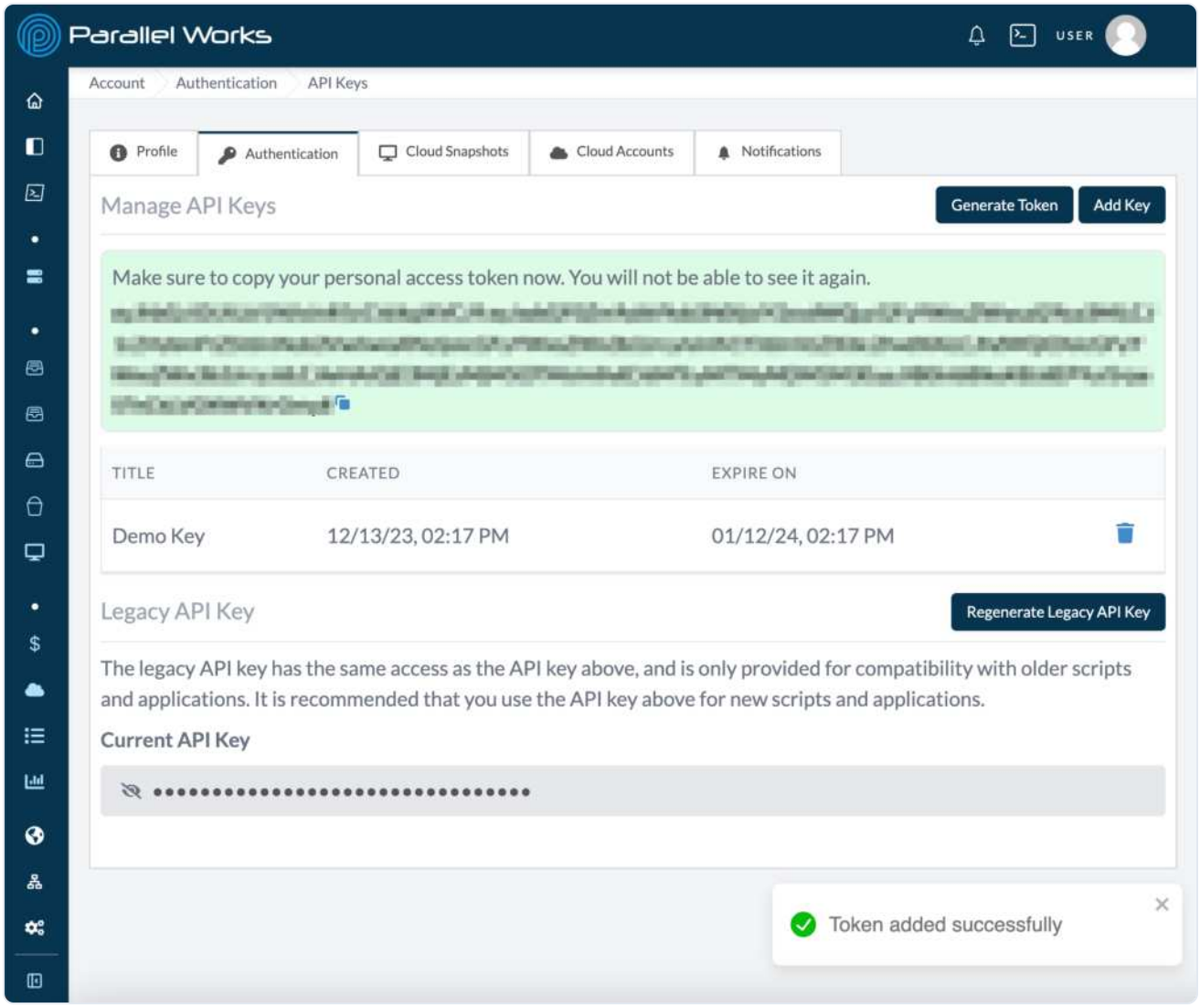


If your organization permits users to select their own expiration date for tokens, use the **Expiration** dropdown menu to select when your key will expire. You can choose from 1 hour up to 24 hours in 1-hour increments. Shorter expiration periods are more secure.

Click **Add Token**.



Your new token will be displayed on the next page. You can use the copy button to copy and save the key in a secure place. Once you refresh or navigate away from this page, you won't be able to see the key again.

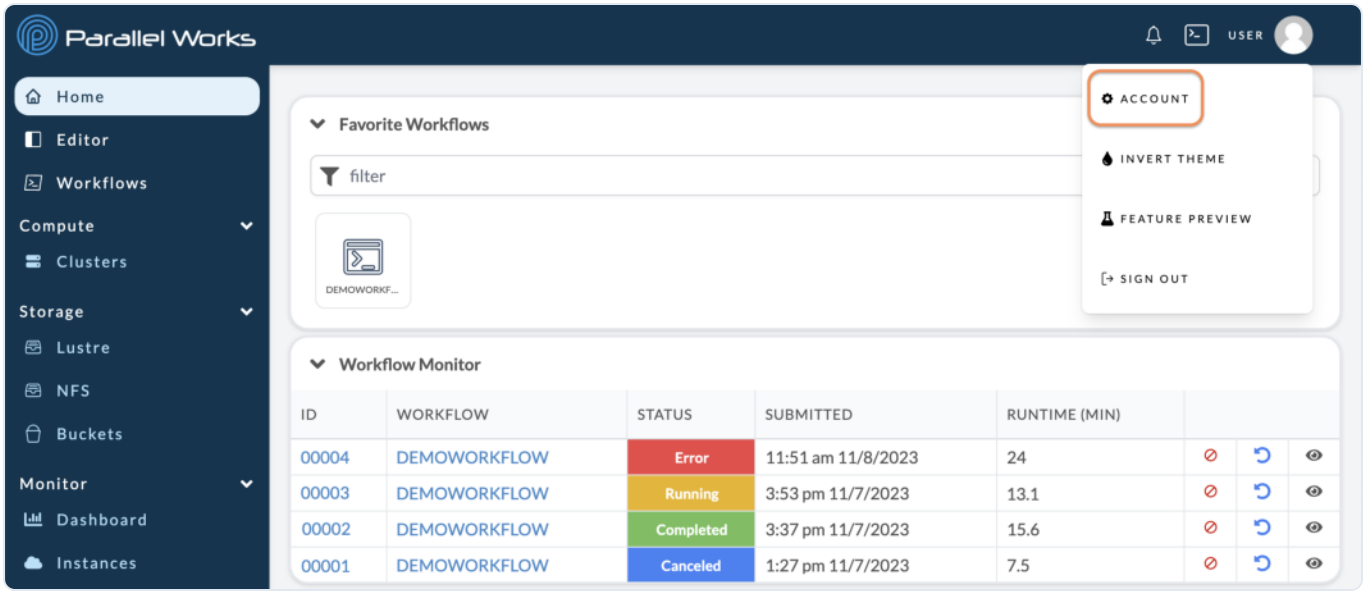


Managing SSH Keys

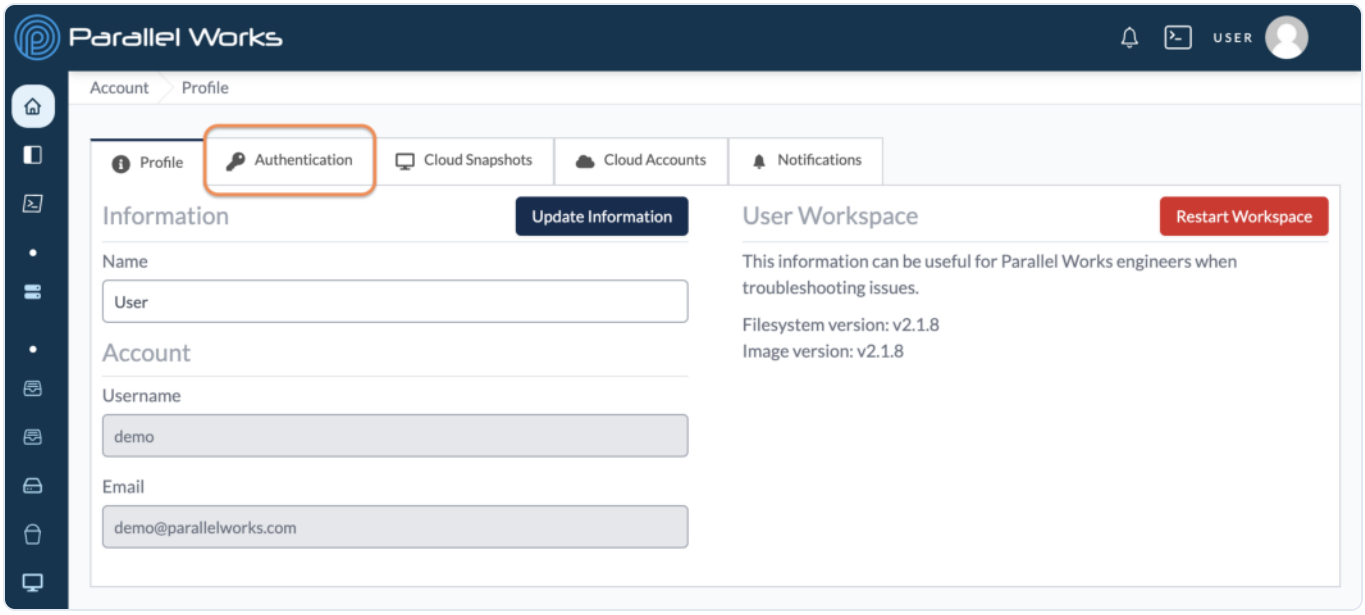
You can add an SSH key to your ACTIVATE account, which will allow you to log in to any active cluster from any device with that SSH key on it.

Create an SSH Key

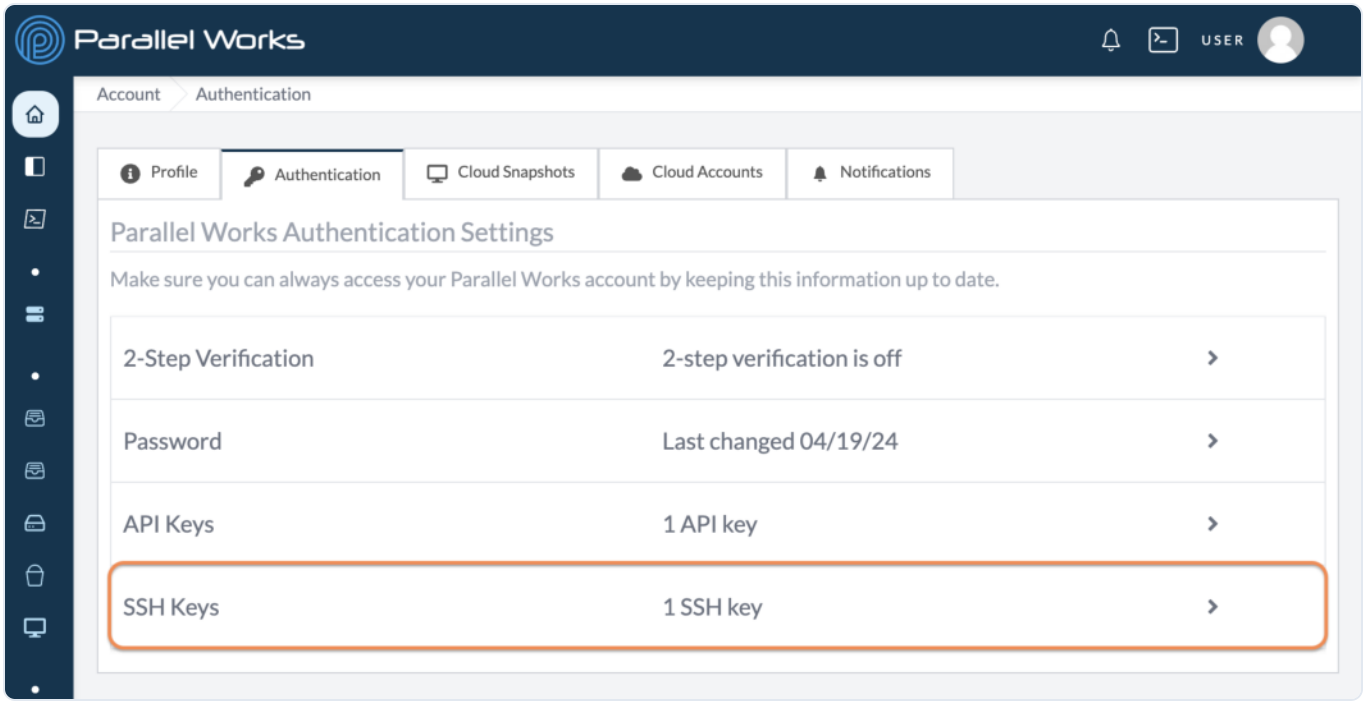
Navigate to your account settings (**Username** > **Account**).



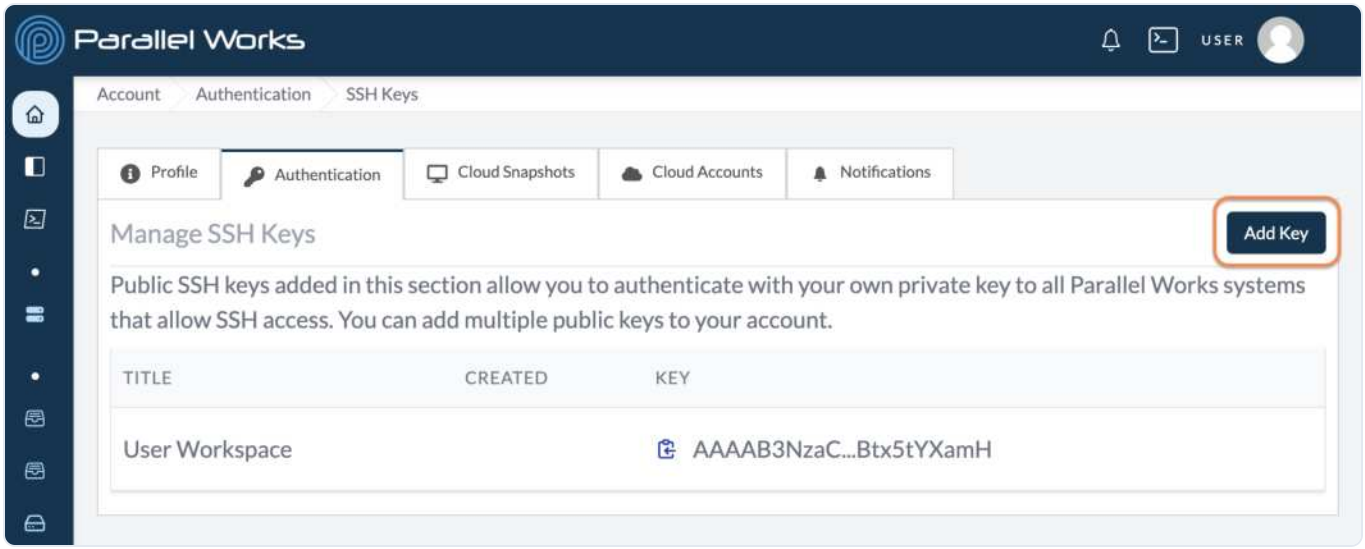
On the next page, click **Authentication**.



Click **SSH Keys**.

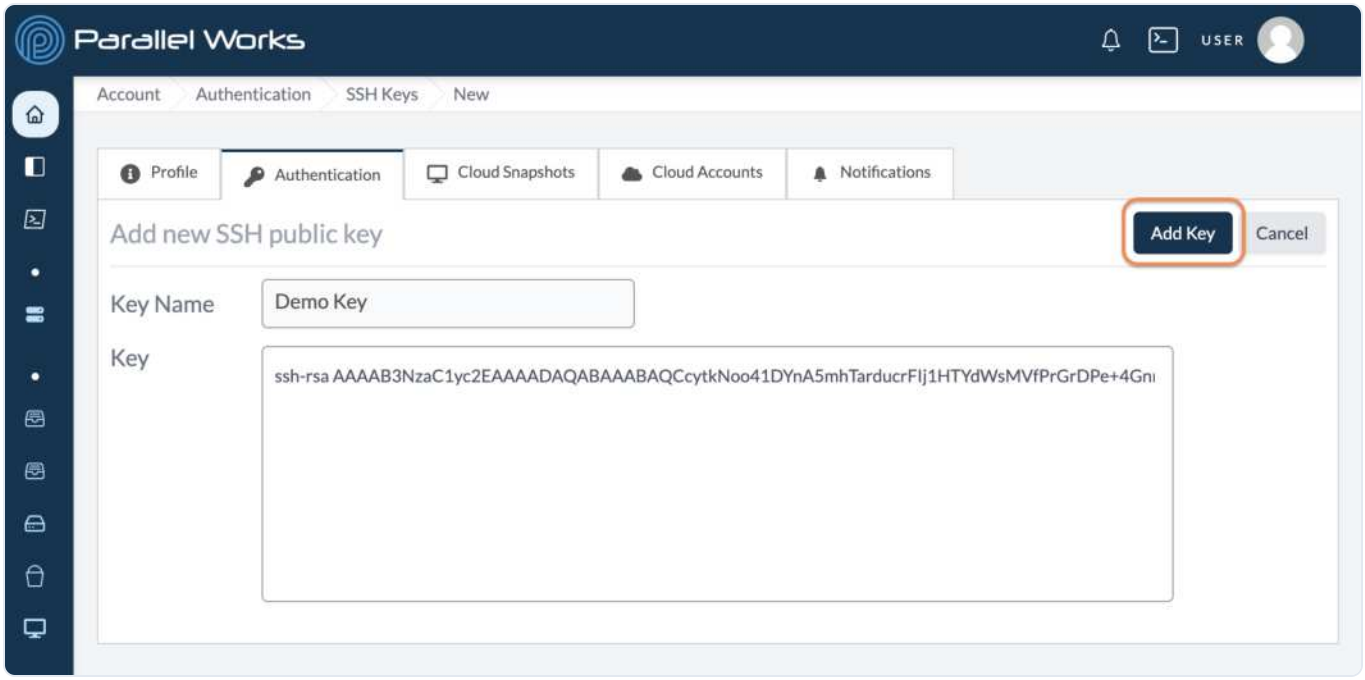


Click **Add Key**.



Use the **Key Name** field to give the key a unique name. Paste your public SSH key in the **Key** field.

Click **Add Key**.

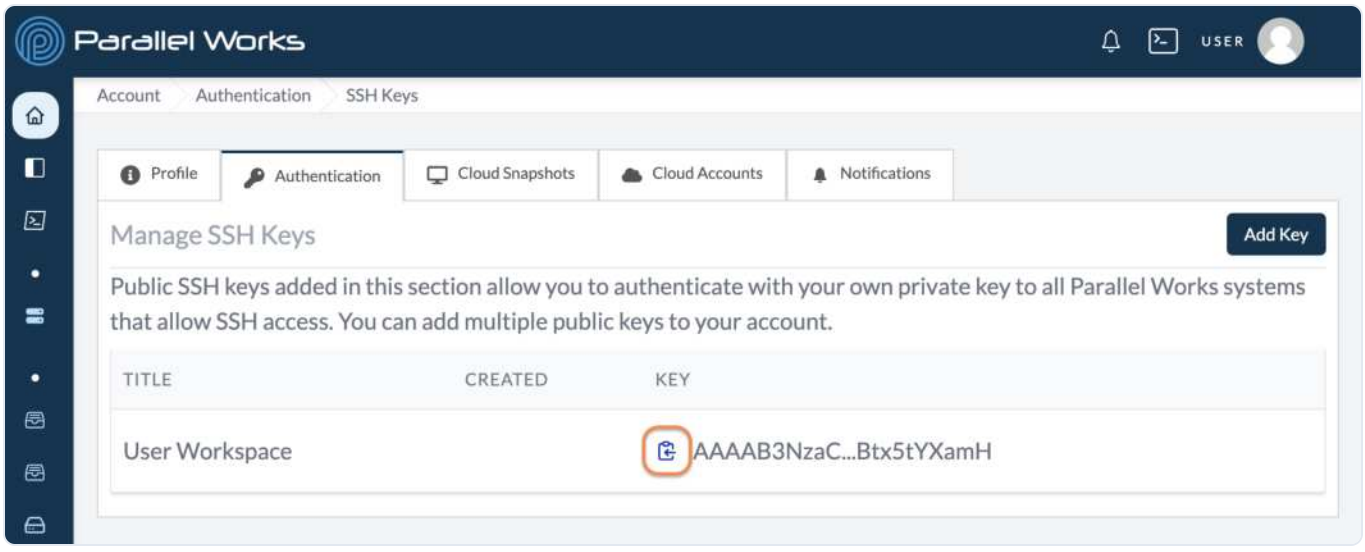


You can now log in to any active cluster. Navigate to the terminal or command line on your device and enter `ssh -i /path/to/ssh/private/key username@IPaddress`:

```
parallelworks@Parallels-Macbook-Air sshtest % ssh -i ./id_rsa demo@34.135.24.147
Last failed login: Thu Mar  2 09:48:20 UTC 2023 from 60.94.197.104.bc.googleusercontent.com on
ssh:notty
[demo@democluster-70 ~]$
```

Copy an SSH Key

In **Account** > **Authentication** > **SSH Keys**, click the clipboard icon to copy your SSH key to your computer's clipboard.



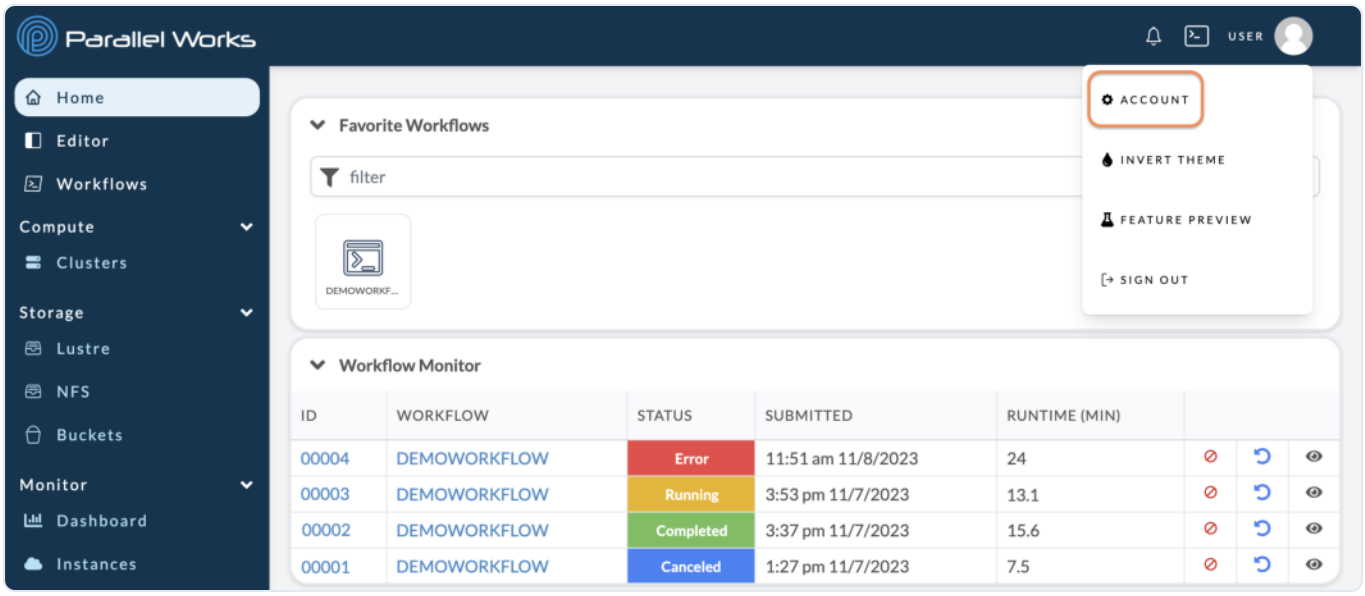
Restarting Your Workspace

Your user workspace is your dedicated environment for starting clusters, running workflows, and accessing your data on the ACTIVATE control plane. Each user's workspace is separate from each other, so the work you do on ACTIVATE will never affect another user's work.

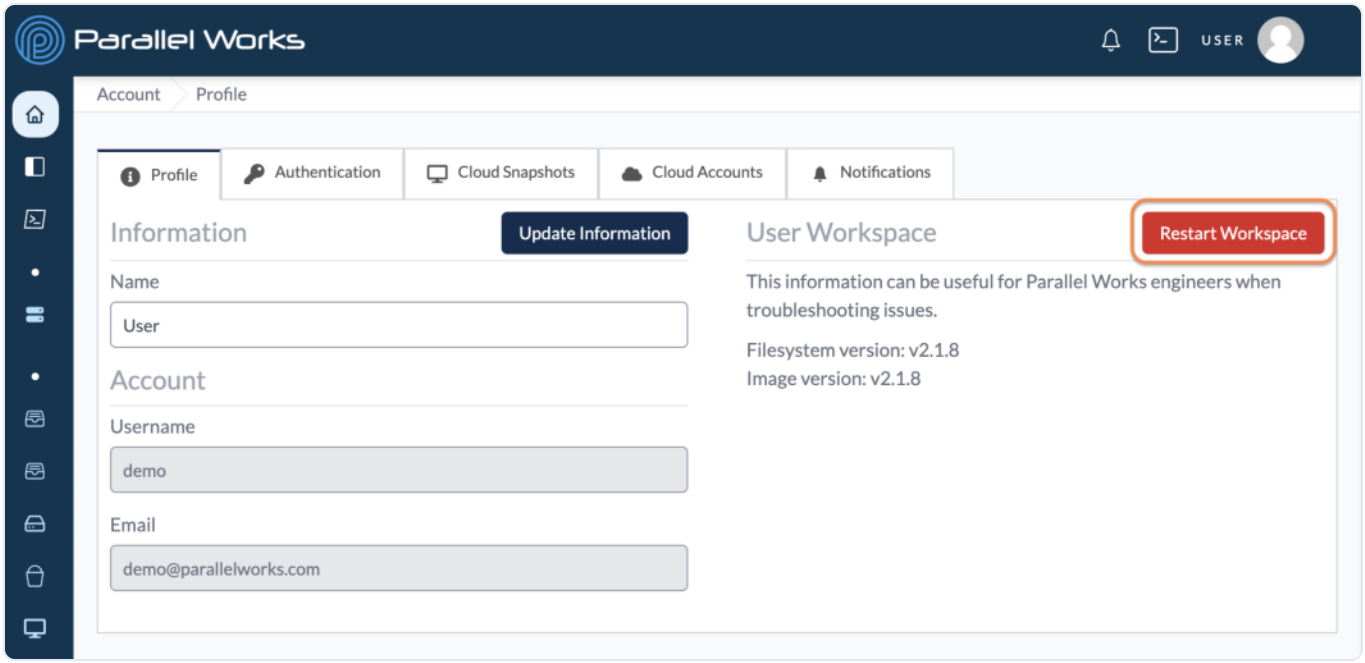
This page explains how to restart your user workspace, which is useful if you're experiencing issues like a cluster that won't start or a workflow that won't run.

> **NOTE**
>
> All running clusters and jobs will be disrupted if you restart your workspace.

Navigate to your account settings (**Username** > **Account**).



In your profile settings, click **Restart Workspace**.



A dialog box will appear with the message *Are you sure you want to restart your workspace?*

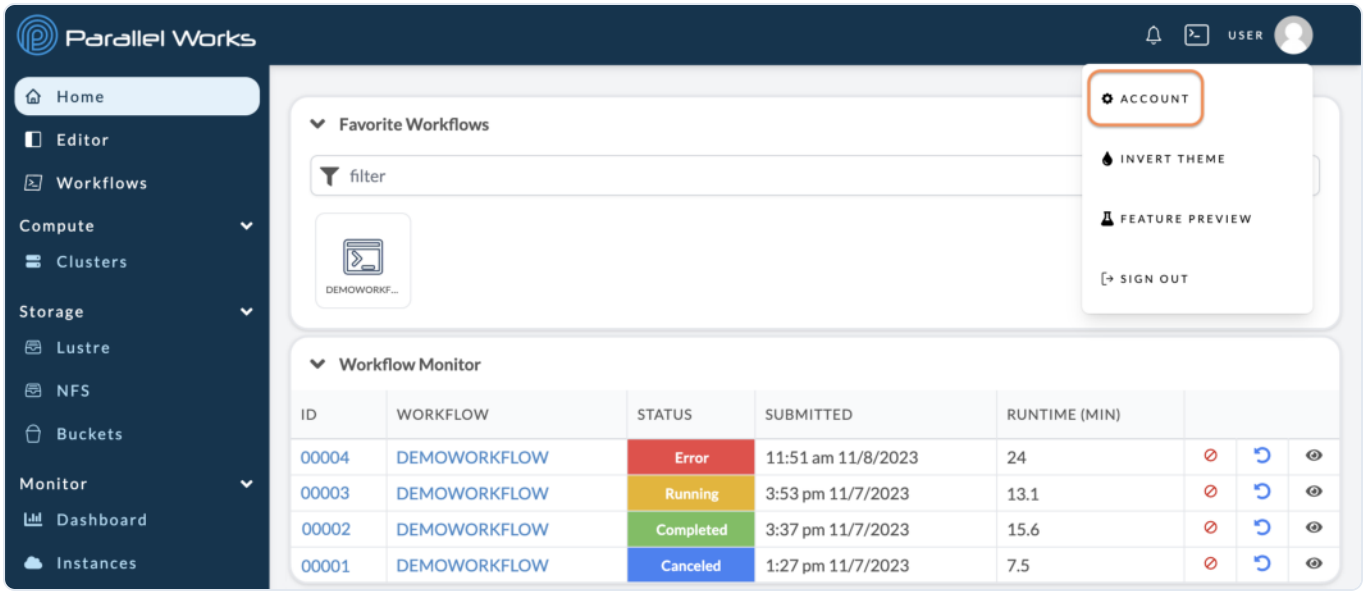
Click **Restart**. After the workspace has restarted, you'll see the message *User workspace killed*.

> **TROUBLESHOOTING**
>
> If you restarted your workspace because you were experiencing errors and they aren't resolved after restarting, please contact us or your ACTIVATE administrator.

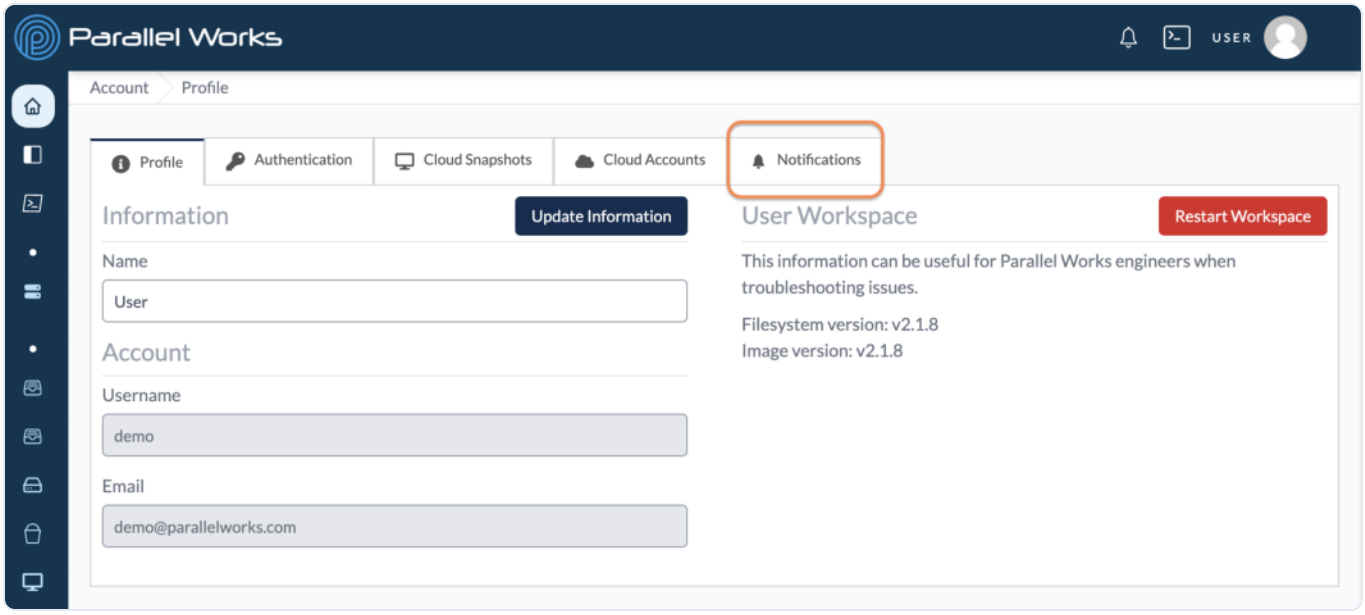
Managing Notifications

The ACTIVATE control plane includes a customizable notification system for both our web interface and your ACTIVATE-linked email address.

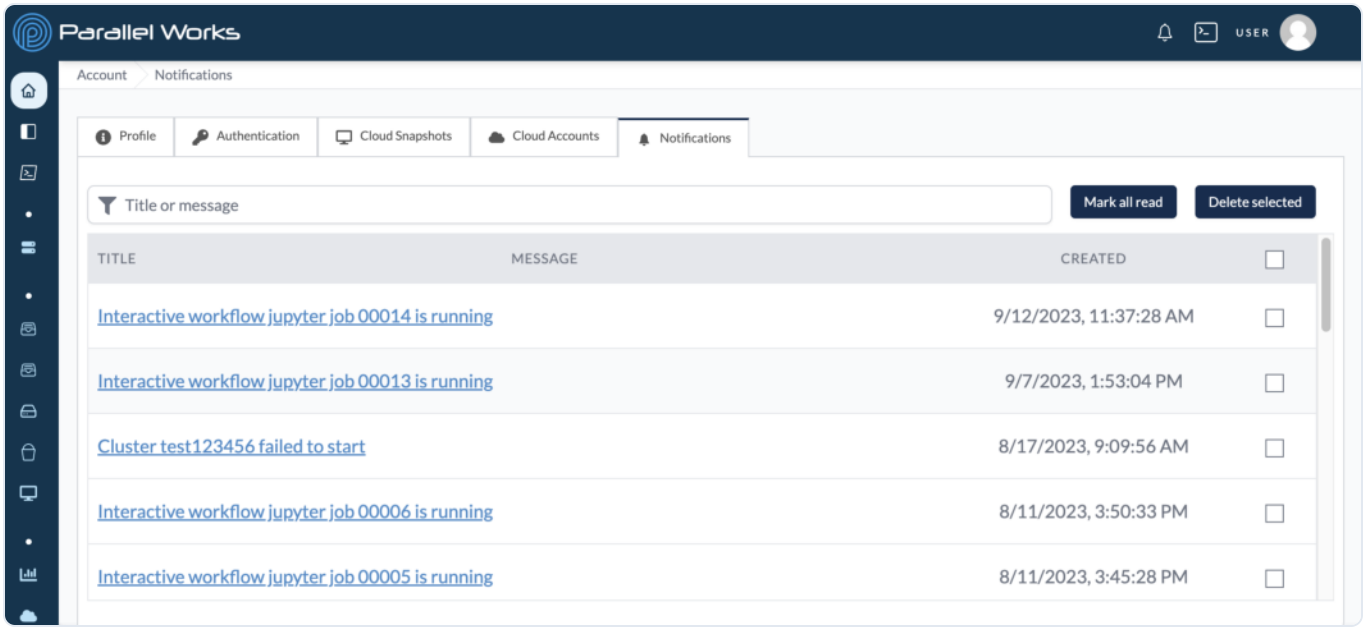
Navigate to your account settings (**Username** > **Account**).



In **Profile Settings**, click **Notifications**.



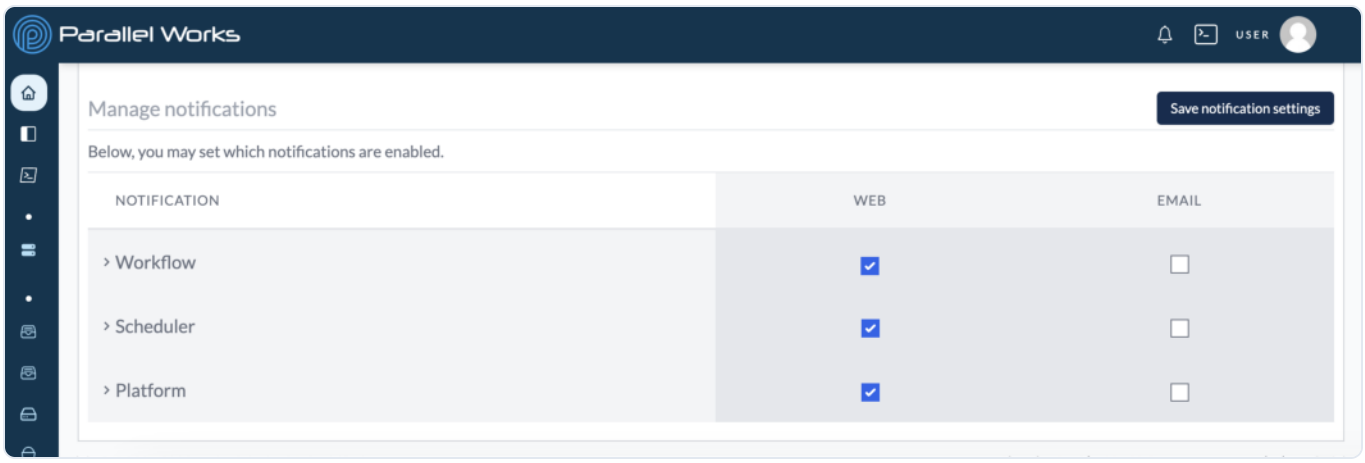
Here, you can manage past notifications as well as configure your notification settings.



You can also reach the **Notification Hub** by clicking the bell icon in the navigation bar, then clicking **Show More**.

Changing Notification Settings

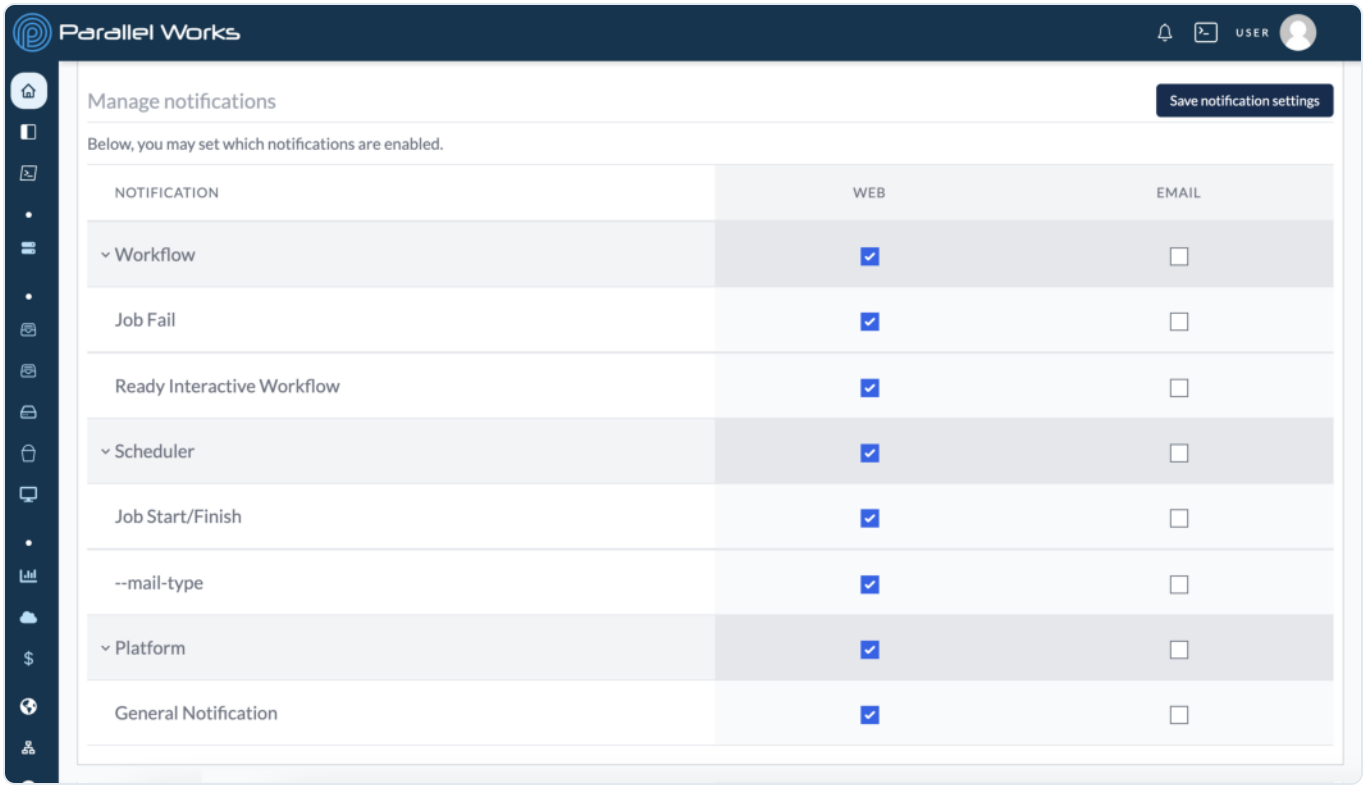
Scroll down to the **Manage notifications** section.



You can select how you would like to receive your notifications here. **Web** notifications appear under the bell icon and in the **Notification Hub** on the ACTIVATE control plane. **Email** notifications are sent directly to your ACTIVATE-linked email address.

By default, all options for **Workflow**, **Scheduler**, and **Platform** notifications are enabled as **Web** notifications. Click the checkboxes for **Web** and **Email** to enable or disable notifications.

For more customized settings, you can click a notification type and enable or disable its sub-types.



When you're done, click **Save notification settings**.

About Notification Types

Our notification system includes the types outlined below.

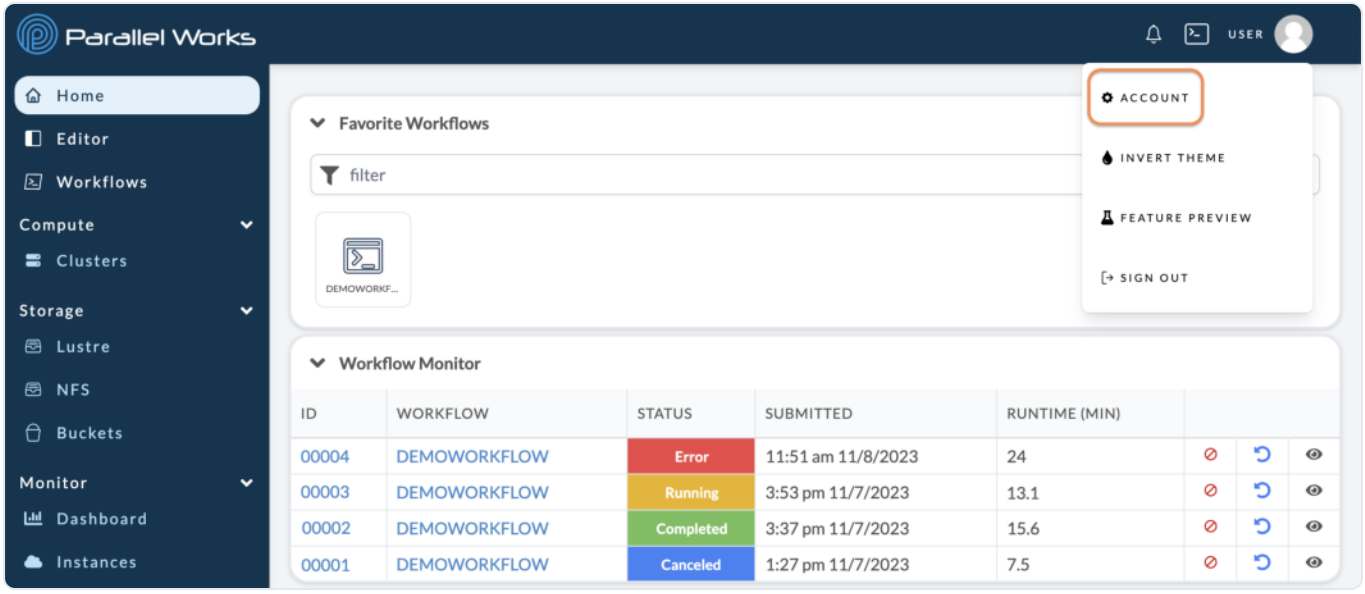
- Workflows
  - **Job Fail**: If a workflow fails during start-up or run-time, this notification will be triggered.
  - **Ready Interactive Workflow**: Because interactive workflows can have longer start-up times, we've included this notification to send when an interactive session is ready to use.
- Scheduler
  - **Job Start/Finish**: Job status notifications will be sent from the scheduler's prolog and epilog scripts. This method requires no cluster side user action, and will send notifications about your jobs when selected.
  - **—mail-type**: This notification uses the Slurm flag `--mail-type`, which is set by the user on the cluster in order to send notifications when certain events occur. This method requires users to enter the `--mail-type` flag when calling `salloc`, `sbatch`, `srun`, or in their job scripts. For more information about these Slurm commands, please see **Submitting Jobs via Slurm**.
- Platform
  - **General Notification**: These notifications include cluster start-up messages and run-time errors.

Working With Cloud Snapshots

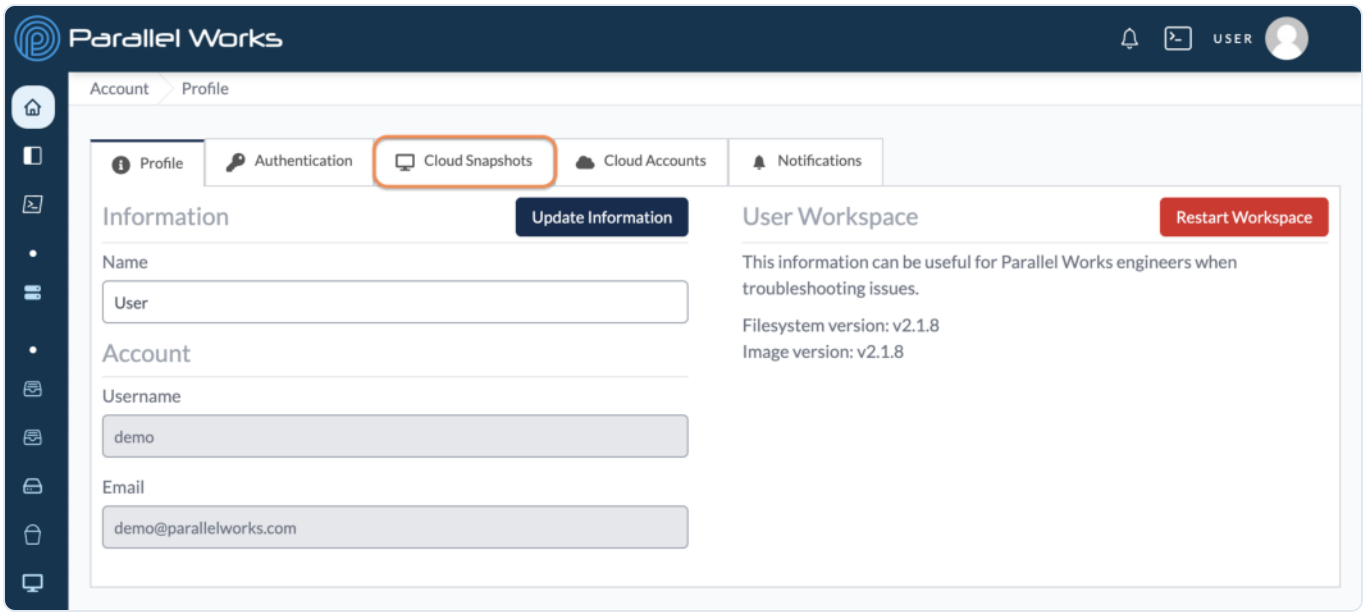
Cloud snapshots let you make persistent changes to the operating system image running on your cluster nodes. By using an image provided by Parallel Works as a base, you can stage automations, install additional software, or enable additional services.

Creating Cloud Snapshots

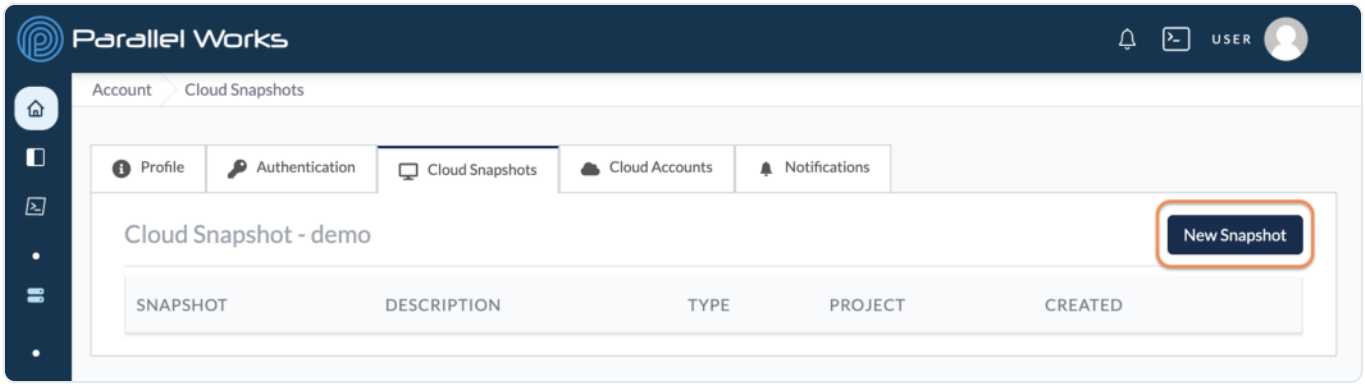
Navigate to your account settings (**Username** > **Account**).



On the next page, click **Cloud Snapshots**.

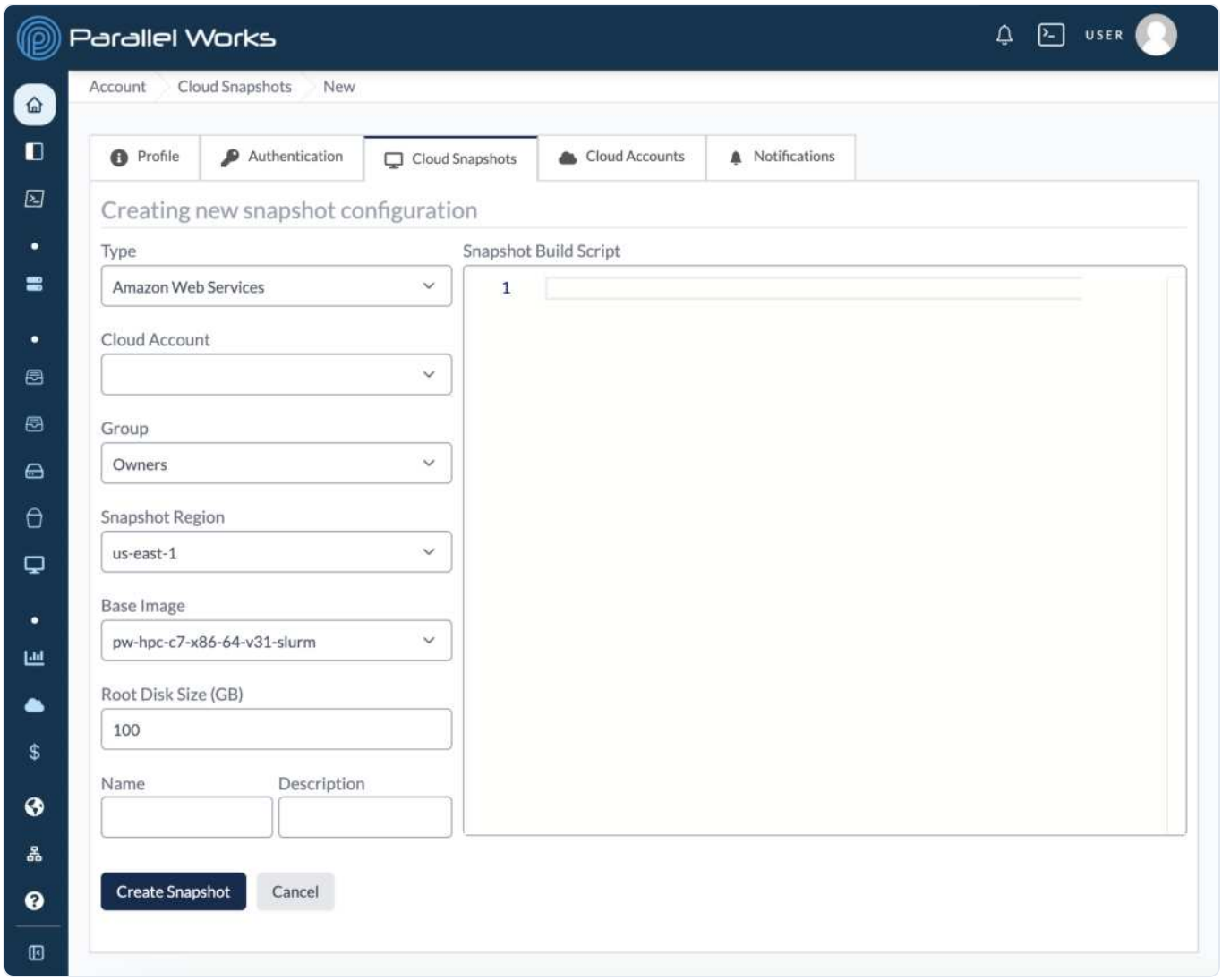


In **Cloud Snapshots**, click **New Snapshot**.



Snapshot Configuration Settings

There are several configurable parameters for cloud snapshots, which are outlined below.



Type

Use this dropdown menu to select whether your snapshot will be built for AWS, Azure, or Google.

Cloud Account

Use this dropdown menu to select which cloud account will provision your snapshot. For most users, this menu should be left as the default option, unless you're a member of multiple organizations.

Group

Use this dropdown menu to select the group name that your organization uses to allocate costs. This menu is especially important if your organization uses multiple groups.

If you're not sure which group to select, you can contact us or your organization's ACTIVATE administrator.

Snapshot Region

Use this dropdown menu to select the region your snapshot will be provisioned in.

If you run in multiple regions, you will want to provision the snapshot wherever you run your clusters.

Base Image

Use this dropdown menu to select the ACTIVATE image that will be the base for your snapshot. Typically, this menu will be populated with the suggested version by default.

Root Disk Size (GB)

Use this field to enter the size of the snapshot's root disk. By default, this field is set to `100`. Typically, you won't have to change this value unless you run into capacity issues.

Name

Use this field to enter the name of your snapshot as it will be listed on the ACTIVATE control plane.

Description

Use this field to enter a description to provide additional details about your snapshot.

Snapshot Build Script

You can enter a script in **Snapshot Build Script**, which will run to make additions to the base ACTIVATE image. You can use this script to install additional packages from `yum` repositories, clone GitHub repositories, or download files from a URL. An example build script is provided below.

```
#!/bin/bash

# Get useful environment info
echo "my username is $USER"
echo "my home dir is $HOME"
echo "my workdir is $PWD"

# Clone a public GitHub repository
git clone https://github.com/hpc/ior.git /usr/local/src/ior;

# Install Anaconda and have it initialize on login
wget https://repo.anaconda.com/archive/Anaconda3-2023.03-Linux-x86_64.sh;
sh ./Anaconda3-2023.03-Linux-x86_64.sh -b -p /usr/local/anaconda3;
```

```
# Symlink the Conda executable to the main /usr/local/bin directoryln -s
/usr/local/anaconda3/bin/conda /usr/local/bin/conda;# Install Jupyterconda install -y jupyter;#
Packages will be installed under the Anaconda root directory, so symlink what you needln -s
/usr/local/anaconda3/bin/jupyter-lab /usr/local/bin/jupyter-lab;ln -s
/usr/local/anaconda3/bin/jupyter /usr/local/bin/jupyter;# Install R and R Studio Server# Specify
the R version you want# Download and install R packagescurl -O https://cdn.rstudio.com/r/centos-
7/pkgs/R-${R_VERSION}-1-1.x86_64.rpm;yum -y install R-${R_VERSION}-1-1.x86_64.rpm; # NOTE: this
package installs R to /opt/R/4.1.3/bin/R# Install R Studio packageswget
https://download2.rstudio.org/server/centos7/x86_64/rstudio-server-rhel-2023.03.0-386-
x86_64.rpm;yum -y install rstudio-server-rhel-2023.03.0-386-x86_64.rpm;
```

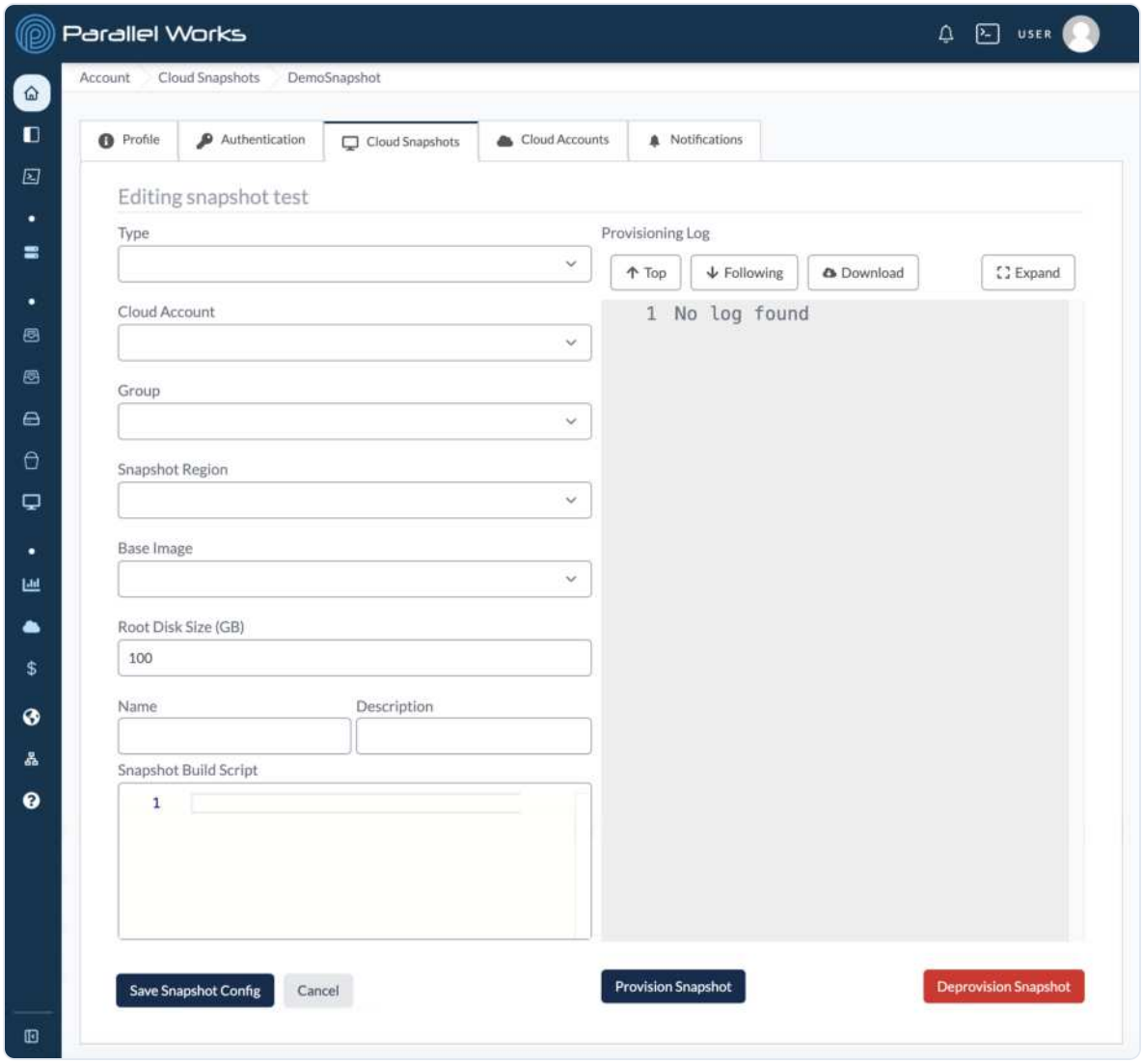
This build script will:

- print some basic environment information.
- clone the public IOR repository from GitHub into `/usr/local/src/ior`.
- download and run the `Conda` installer.
- download and install R and R Studio Server.
- link executables to a common `$PATH` location.

Once you've provided your build script, click **Create Snapshot** to save the configuration.

Provisioning Cloud Snapshots

After clicking **Create Snapshot**, the page will update with a **Provisioning Log** and two new buttons: **Save Snapshot Config** and **Provision Snapshot**.



Click **Save Snapshot Config** when making configuration changes that you want to save, including changes to the main parameters.

Click **Provision Snapshot** to start building the snapshot image. Doing so will replace the **Snapshot Build Script** with the **Provisioning Log**, where you can monitor the snapshot build.

The snapshot build process works by creating a temporary virtual machine (VM) that runs the ACTIVATE base image. Then, the provided script runs to make additions to the image, which is saved as a separate snapshot.

When the snapshot is done building, the **Provisioning Log** will show a message identifying the name of the image in the CSP account. If there were errors in the build that prevented it from completing, those will also be visible in the log.

If your build is successful, the **Provisioning Log** will show a message such as:

```
Packer build completed successfully
{"snapshot_id":"ami-0a5854d925e6e558b", "snapshot_name":"pw-Demo-hello", "csp":"aws",
```

```
"region":"us-east-1" }https://parallel.works/api/v2/machineimages
```

If your build is unsuccessful, the **Provisioning Log** will show a message such as:

```
Build 'amazon-ebs.aws' errored after 1 minute 57 seconds: Script exited with non-zero exit status:
1. Allowed exit codes are: [0]
==> Wait completed after 1 minute 57 seconds
==> Some builds didn't complete successfully and had errors:
--> amazon-ebs.aws: Script exited with non-zero exit status: 1. Allowed exit codes are: [0]
==> Builds finished but no artifacts were created.
Packer build failed
```

> **NOTE**
>
> If you reprovision an existing snapshot and the new build has errors, your previously working snapshot will not be overwritten.

Using Cloud Snapshots

Snapshots you have created for any CSP will automatically be added to your cluster configuration settings. You can see user-created snapshots on a cluster's configuration page by selecting the **Image** in **Controller Settings** and the **Elastic Image** in **Partition Settings**.

You can also configure a different snapshot for your controller and each of your partitions.

Deleting Cloud Snapshots

To delete a custom snapshot, navigate to its configuration page and click **Deprovision Snapshot**.

A dialog box will appear with the message *Are you sure you want to deprovision the snapshot test?*

Click **Deprovision snapshot**. The **Provisioning Log** will then display the deletion process:

```
Starting delete of custom image...

CLOUD: aws
REGION: us-east-1
PROJECT: ca-cloudmgmt
IMAGE: pw-Demo-hello
AMI:  ami-023b3386461fdfad6
SnapshotId: snap-01d59ca427982c1d8
aws ec2 deregister-image --image-id ami-023b3386461fdfad6
aws ec2 delete-snapshot --snapshot-id snap-01d59ca427982c1d8

Delete completed...
```

If you use the method above, you will still have your snapshot listed in **Cloud Snapshots** (**Username** > **Account** > **Cloud Snapshots**), and you will be able to reconfigure and reprovision the snapshot at any time.

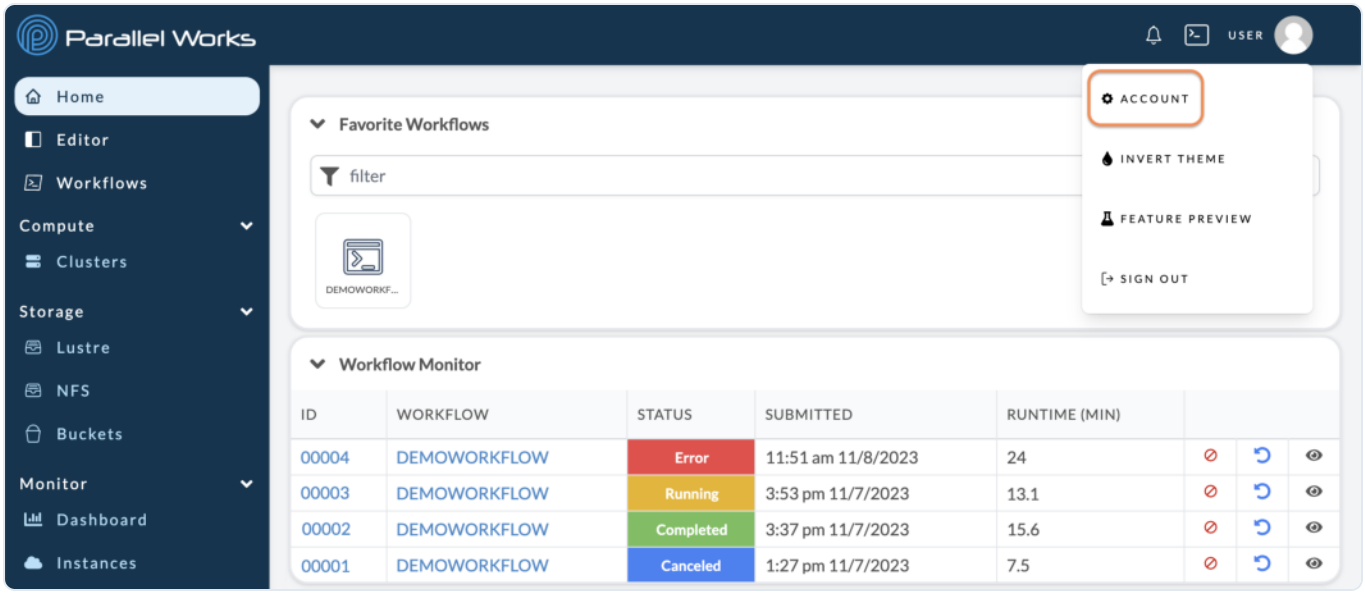
Alternatively, you can delete a snapshot from the list in **Cloud Snapshots**. If you click the delete icon, a dialog box will appear with the message *Are you sure you want to delete the snapshot [snapshot name]?*

Clicking **Delete snapshot** will permanently remove the snapshot from your account and from the list in **Cloud Snapshots**.

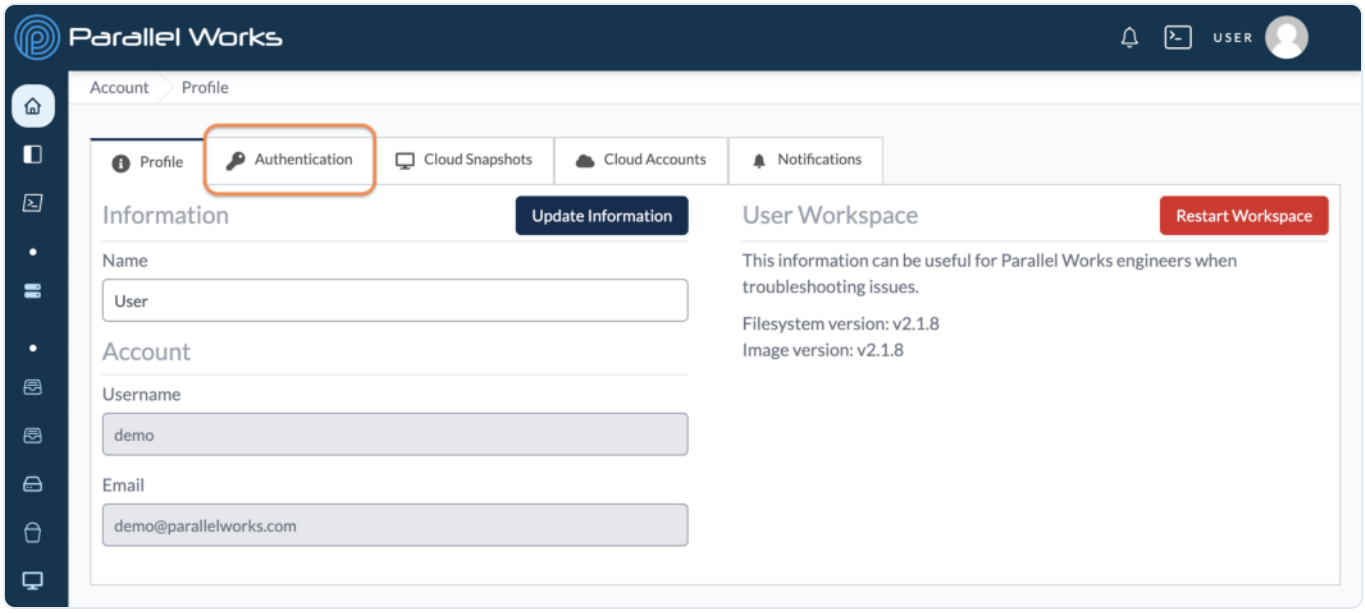
> **ABOUT DELETION**
>
> If you want to delete a snapshot from the list in **Cloud Snapshots**, you first need to deprovision the snapshot on its configuration page. If you don't, the snapshot will still exist in your CSP account.

Resetting Your Password

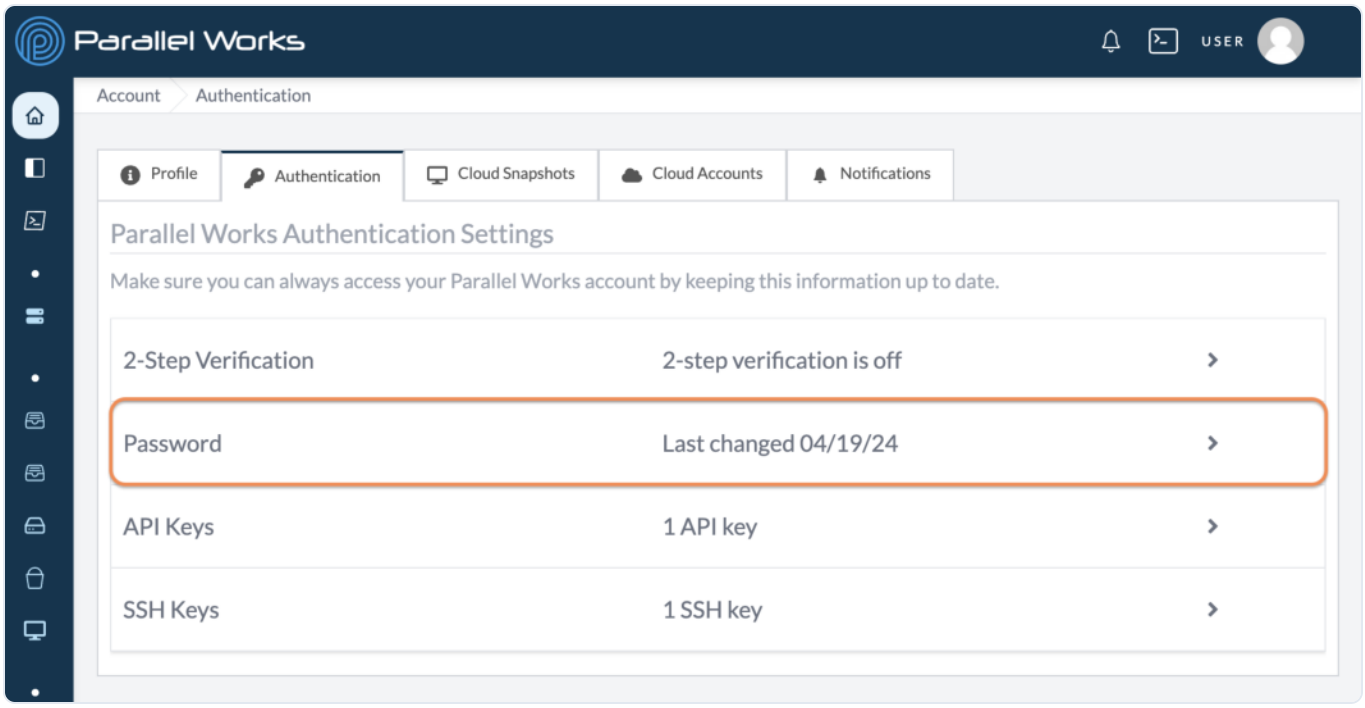
Navigate to your account settings (**Username** > **Account**).



On the next page, click **Authentication**.



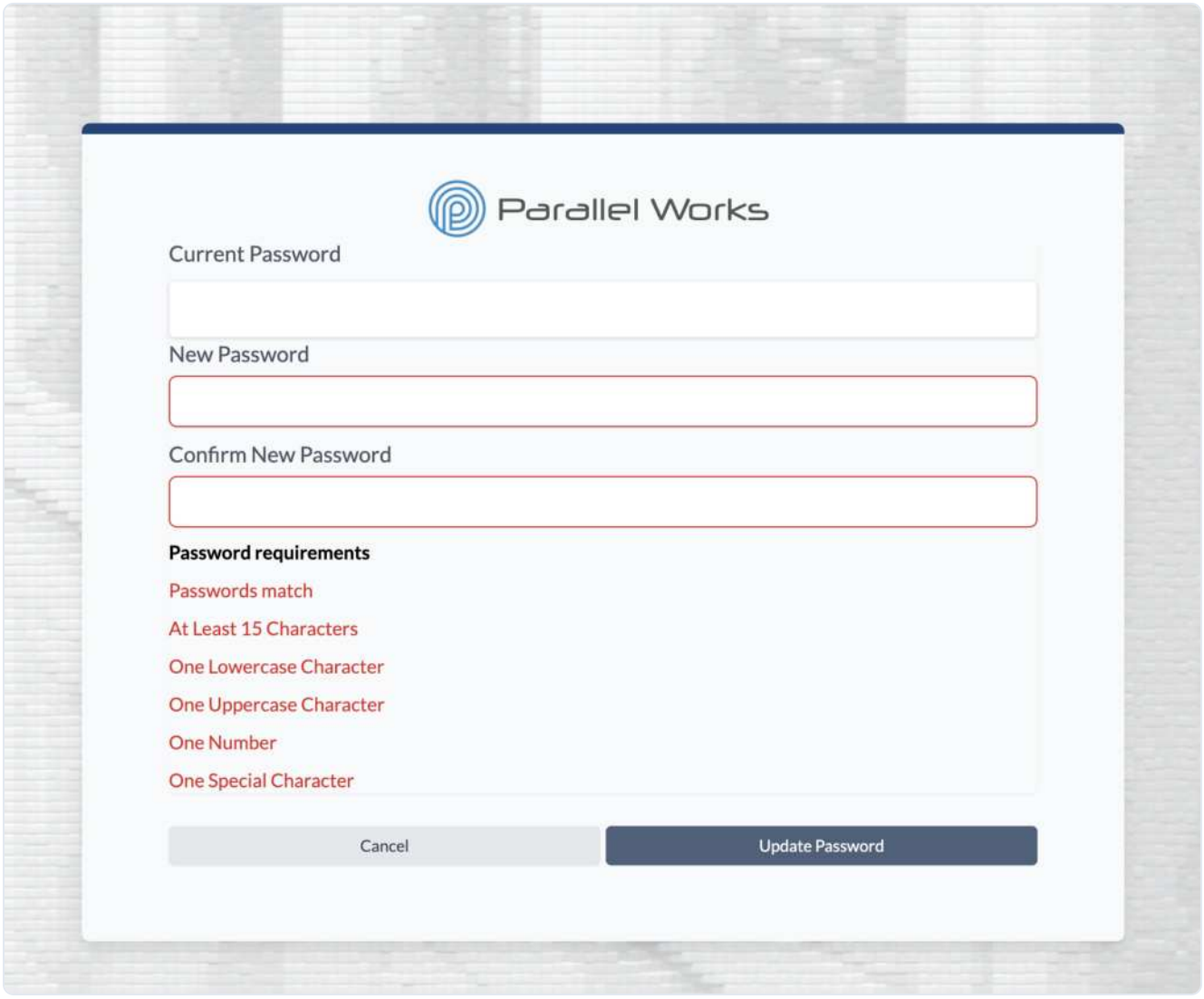
Click **Password**.



On the next page, enter your current password and your new password.

Your new password must contain:

- at least 15 characters
- one lowercase character
- one uppercase character
- one number
- one special character



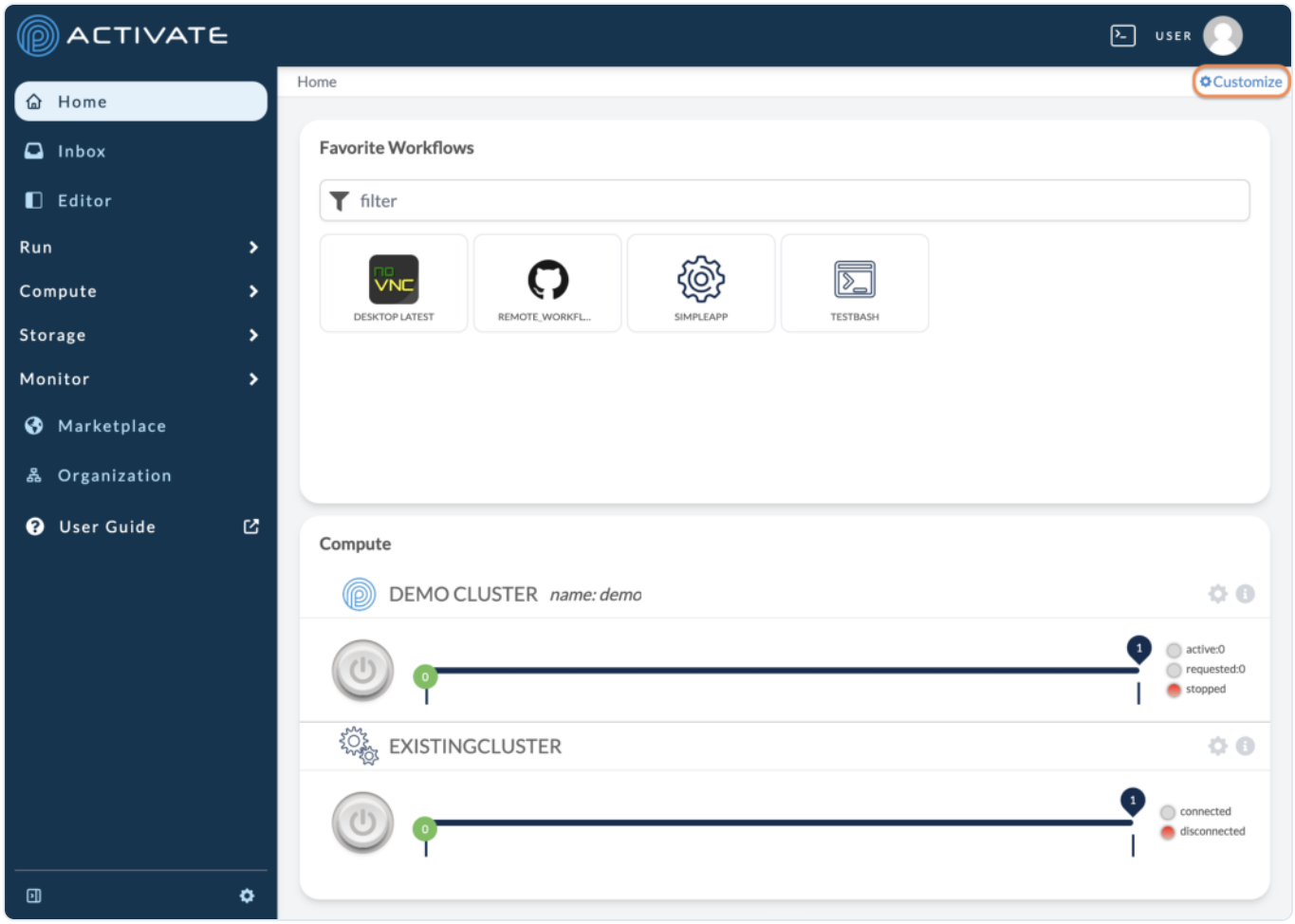
When you're done, click **Update Password**.

Customizing Layouts

You can customize both your Dashboard and your sidebar layouts by adding, removing, and reordering items.

Customizing the Dashboard

On the **Dashboard**, click ** Customize**.



Move Modules

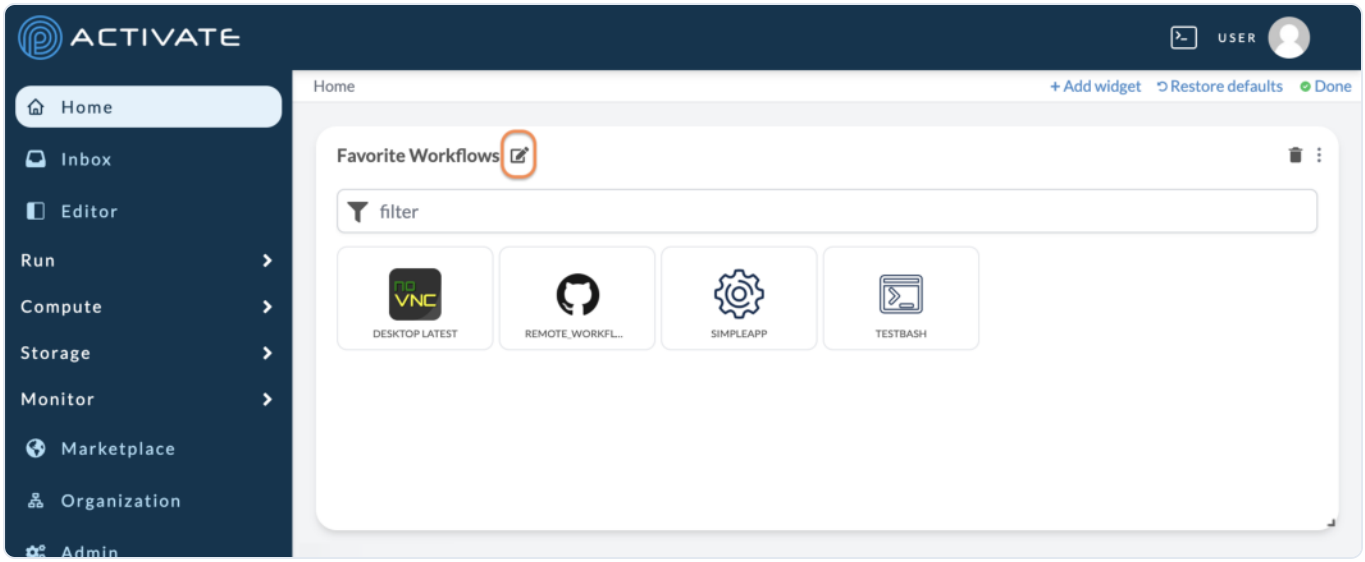
Move your cursor over a module until it becomes the directional arrows icon. Click to drag and drop a module anywhere in the grid.

Resize Modules

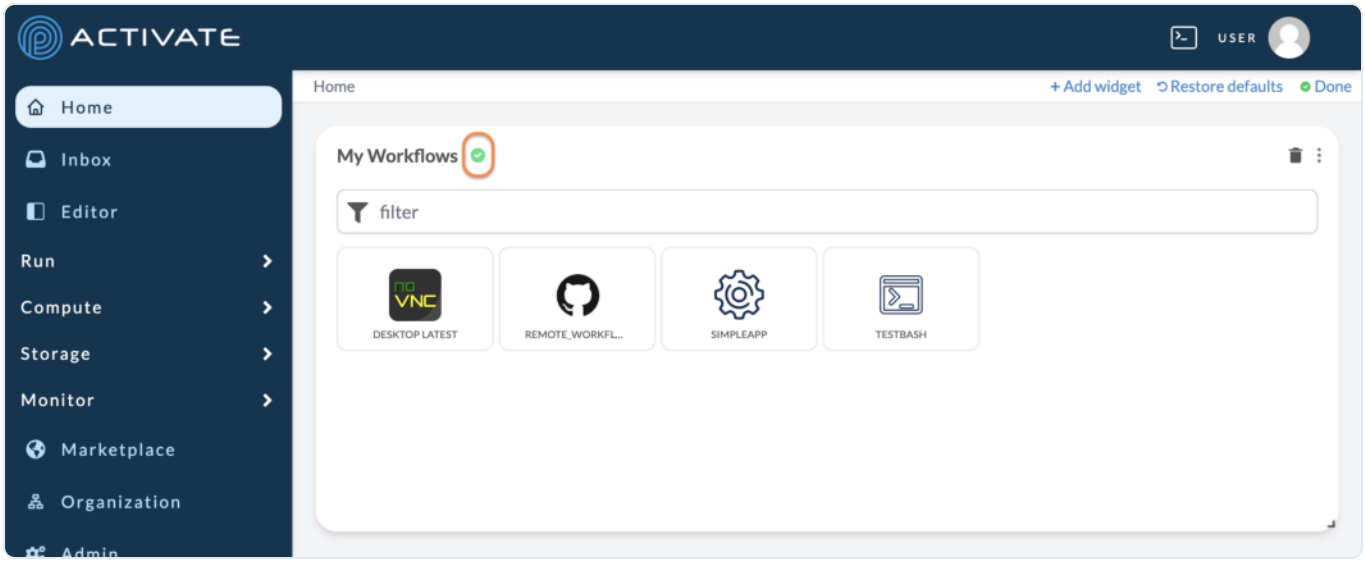
Move your cursor to the bottom-right corner of a module until it becomes the diagonal arrow icon. Click, drag, and drop the corner to resize the module.

Rename Modules

Click the edit icon.

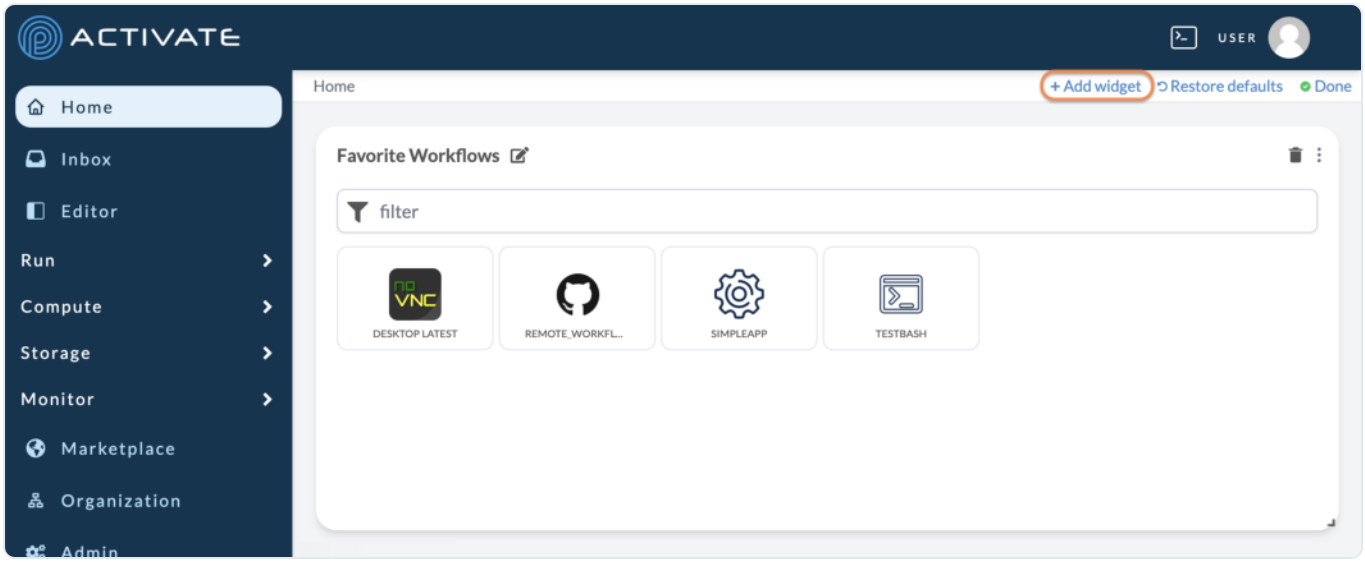


Enter a new name for the module. Click the checkmark icon.



Add Modules

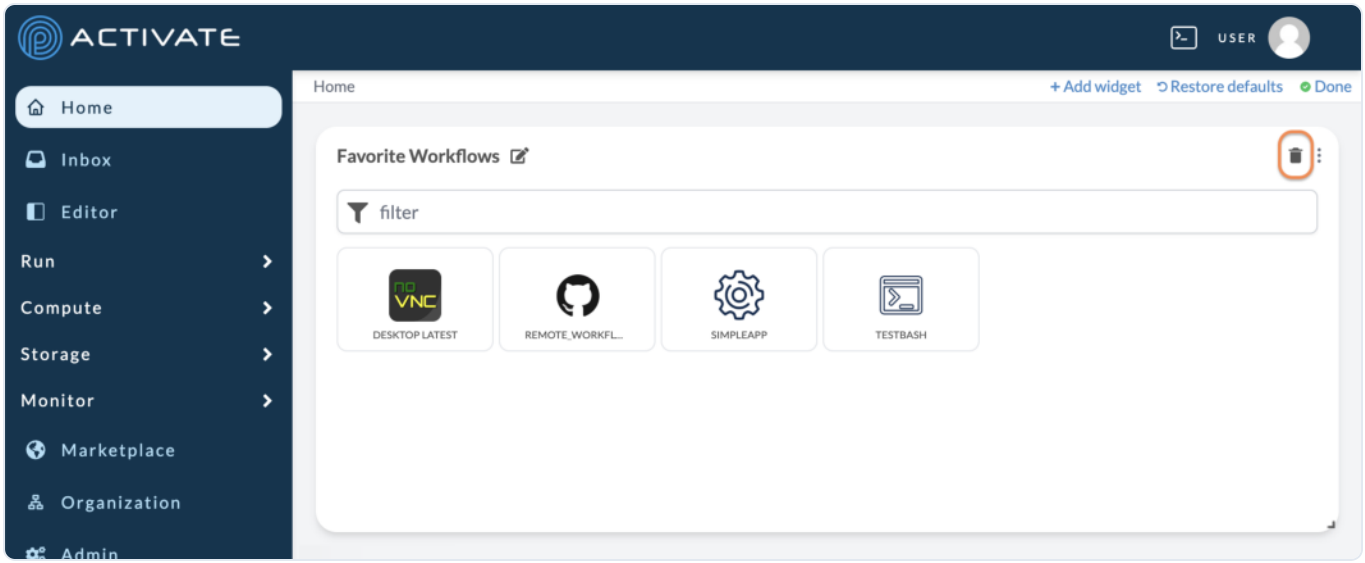
Click + **Add Widget**.



A panel will appear with the available modules. Click, drag, and drop a module into the grid to add it.

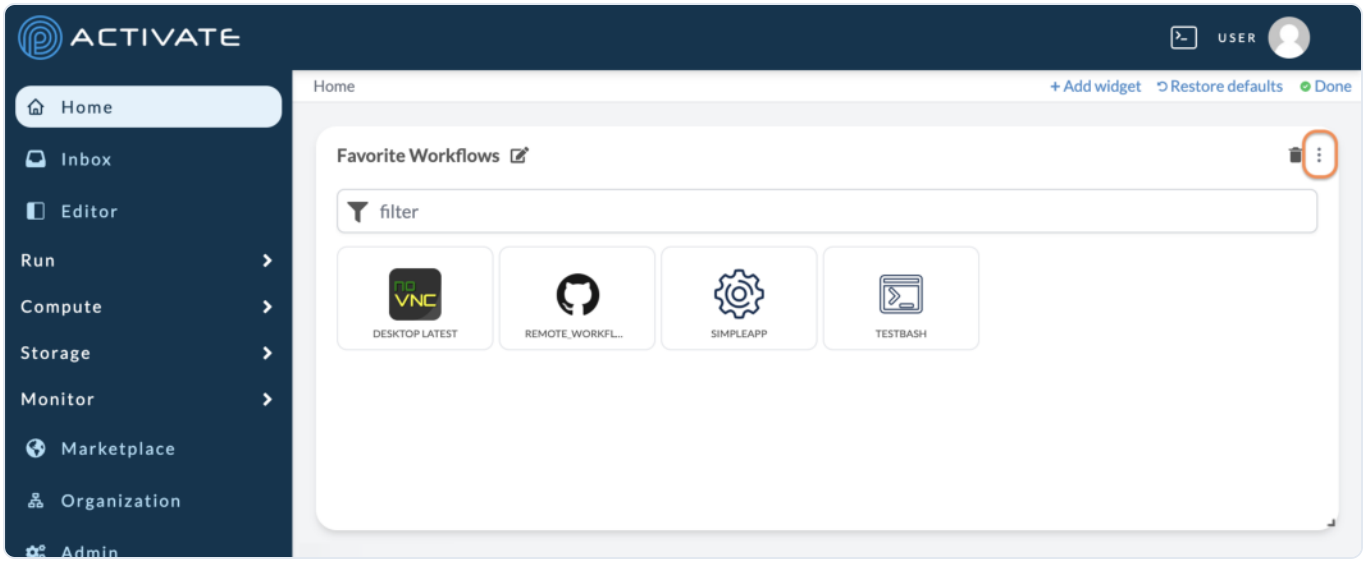
Delete Modules

Click the delete icon.



Filter Data

Some modules allow for further customization. Click the menu icon.

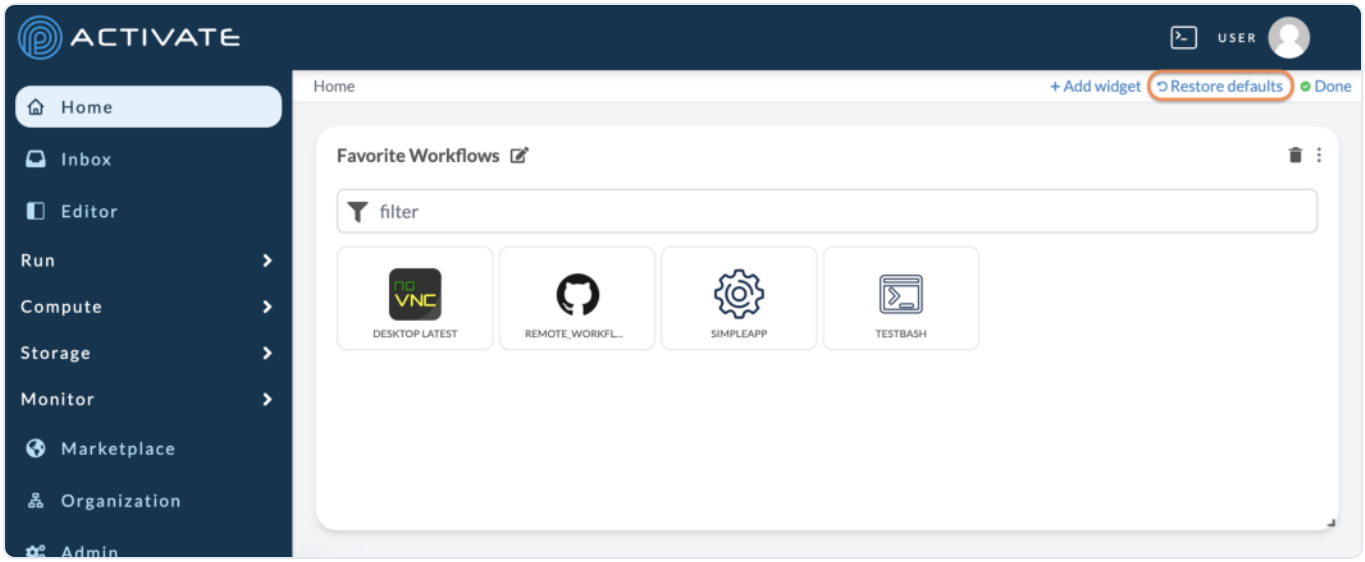


The available options will depend on the module. For example, some modules allow filtering the data they display.

After making your changes, click the menu icon again.

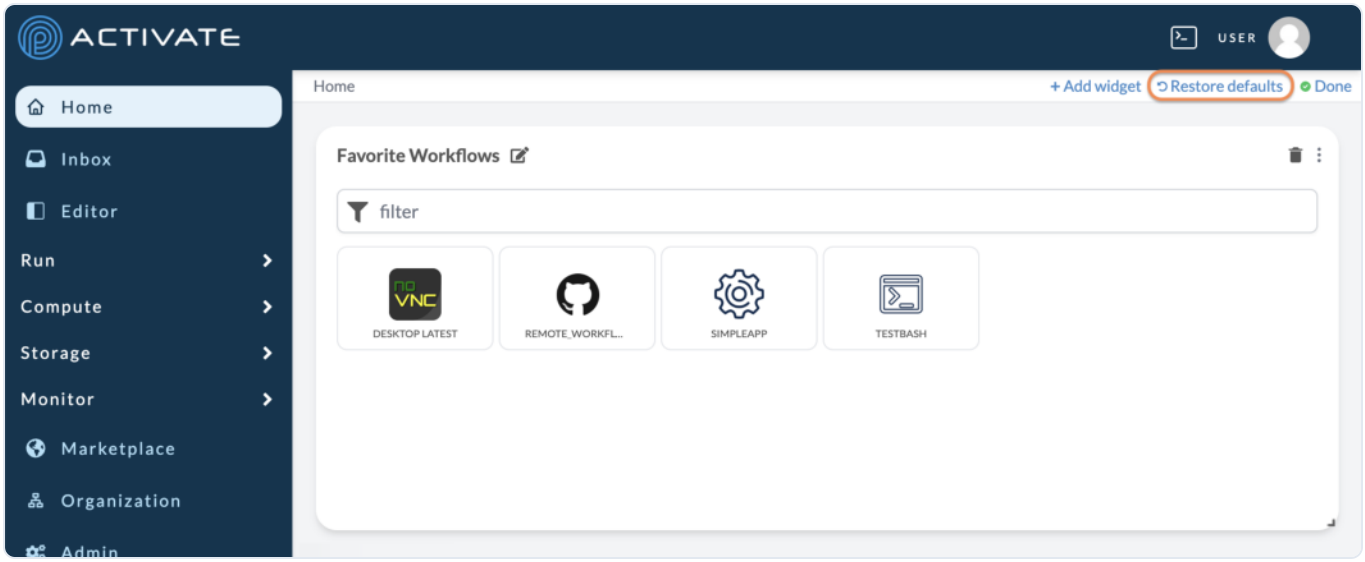
Restore Defaults

If you want to revert your changes, click ** Restore defaults**.



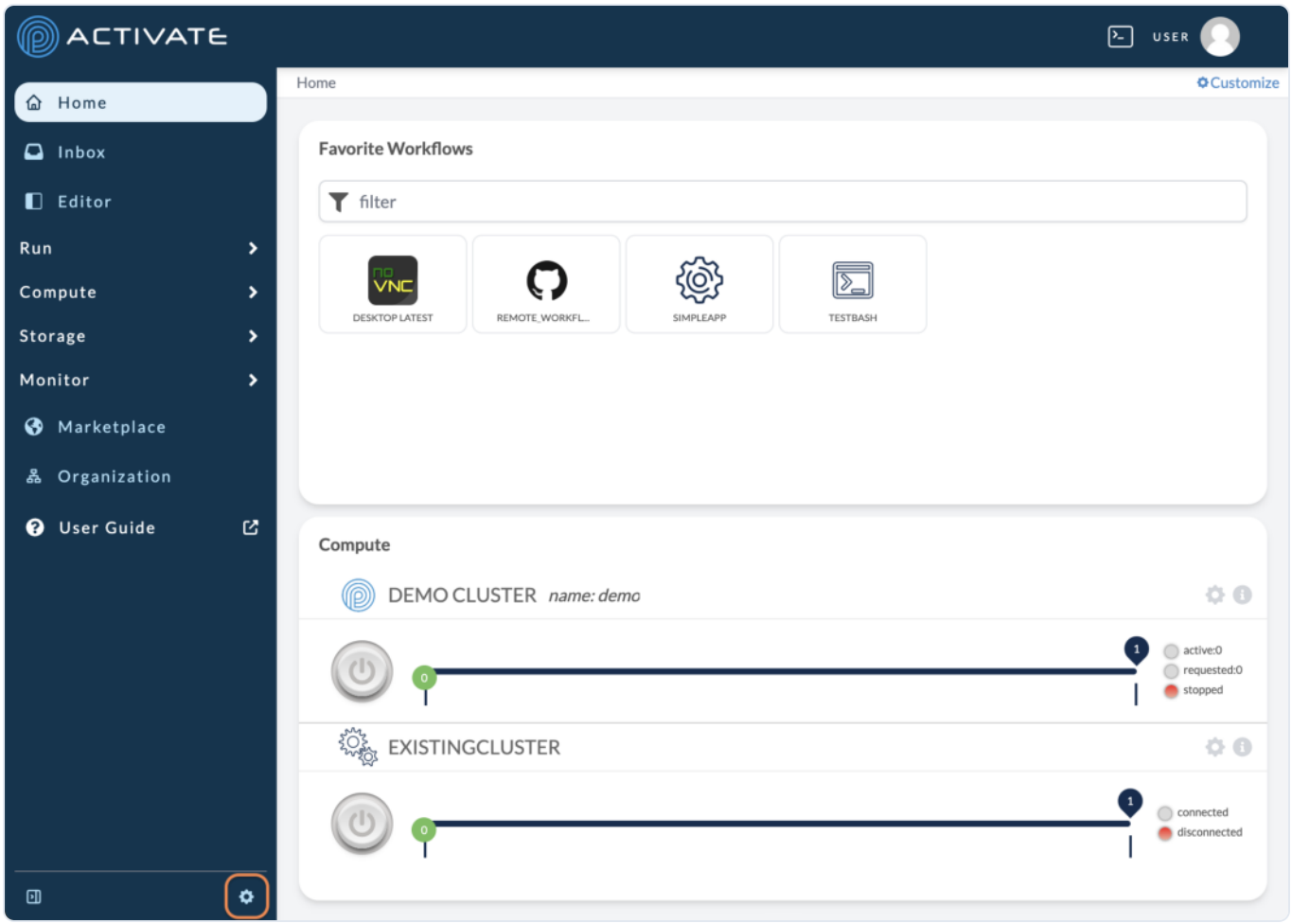
Save Changes

When you're done making changes, click ** Done**.

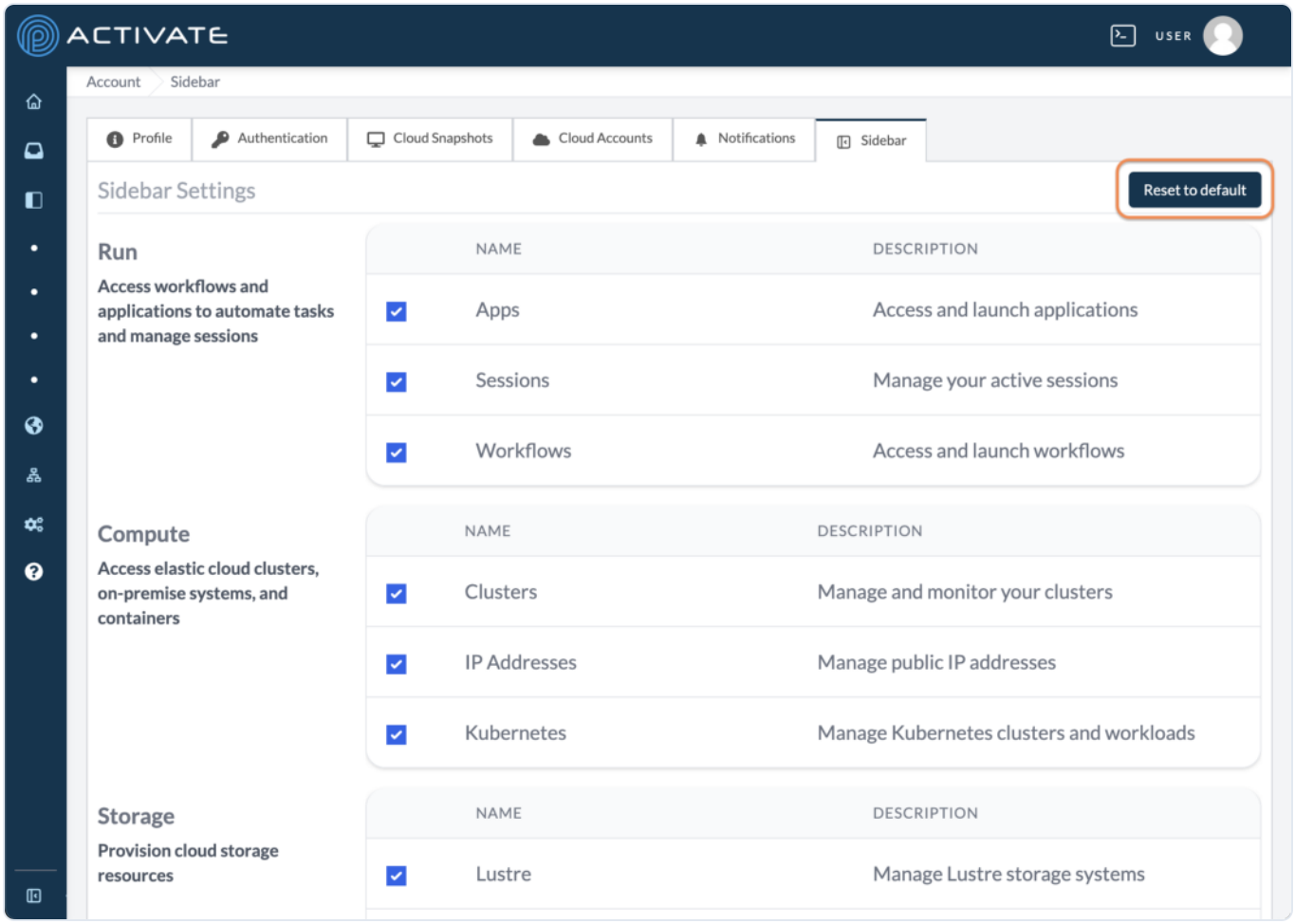


Customizing the Sidebar

At the bottom of the sidebar, click the gear icon. Alternatively, navigate to **Account** > **Sidebar**.



In **Sidebar Settings**, you can uncheck boxes to remove items. Click **Reset to default** to undo your changes.



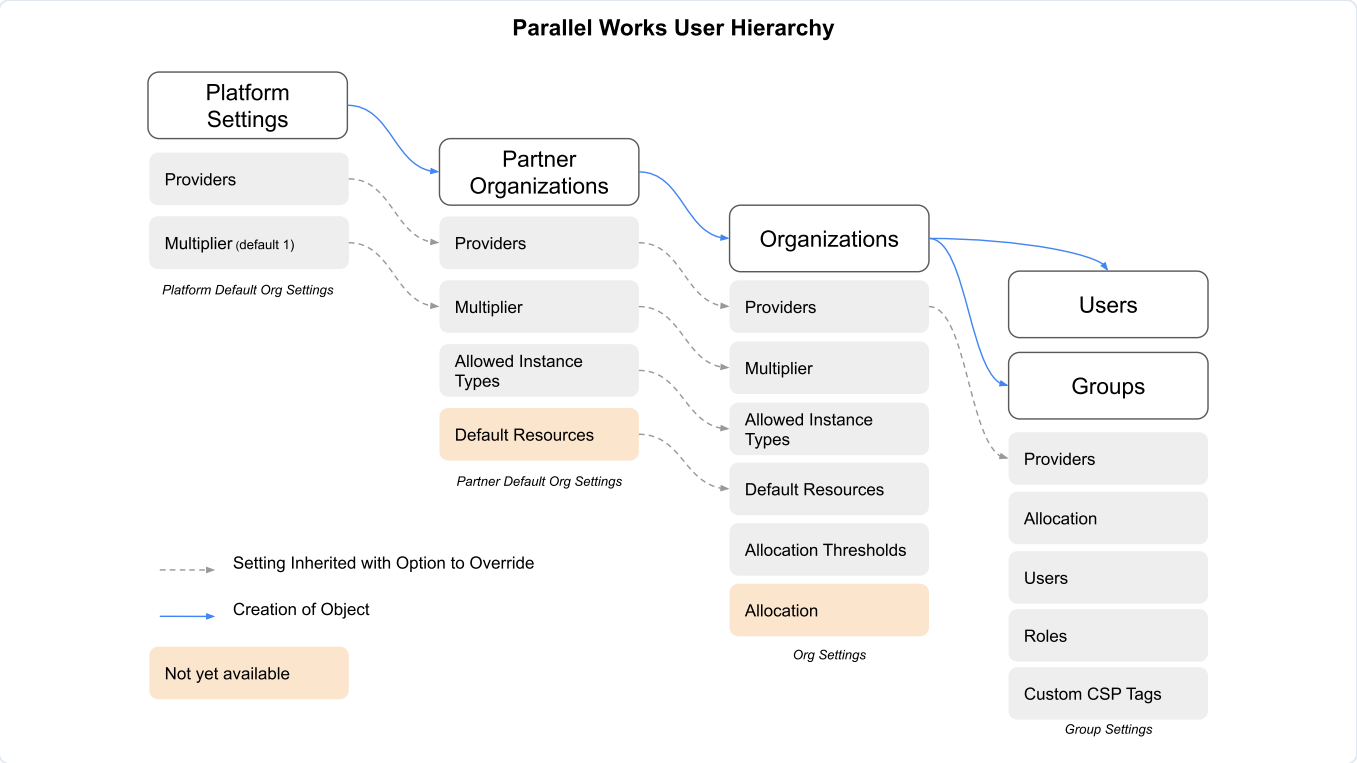
NOTE

Please note that your administrator can remove items from the default sidebar's options. You may not have access to all of the items in the screenshot above.

For Organization Admins

About Partners, Organizations, and Groups

On the ACTIVATE platform, we use partners, organizations, and groups to organize users and manage access permissions. This user hierarchy is shown in the diagram below with a detailed explanation afterward.



Platform Settings are the highest level from which the platform can be configured. Only ACTIVATE administrators can manage these settings. Currently, the settings that can be configured at the platform level are cloud service providers (CSPs) and multipliers.

The next management level is **Partner Organizations**, which have the ability to create other organizations. Organizations created by a partner organization are considered to be *managed by* the partner organization. Any organization that manages another organization can:

- access all of the organization's settings, users, groups, and CSP billing data.
- set custom billing multipliers.
- configure allowed instance types.

All organizations can:

- configure allowed instance types.
- configure default resources, which are created for new users.
- customize the organization theme, sidebar, and login banner.

Groups are used to manage user permissions that affect the administration of users, groups, and group settings. Note that to configure many of these settings, you must be in a group with the `org:admin` role, or a role specific to the setting you're configuring. Partner organizations and ACTIVATE administrators are automatically granted necessary roles to configure organizations. For more specific information, please see **About Group Roles** below.

Users are intended to be named users (i.e. one real person per username), and our system is designed around this idea.

These elements create a user hierarchy with a waterfall structure, where each tier inherits the settings above it but can be overridden at the lower level.

Navigating Organization Settings

When you select an organization, you'll see a sidebar on the left with the following sections:

- **Users** and **Groups** — Manage the people in your organization.
- **Settings** — Configure Authentication, Policies, Webhooks, and Variables.
- **Customize** — Adjust the Theme, Sidebar, User Defaults, and Login Banner.
- **Cloud** — Manage Cloud Accounts, Instance Types, Bootstrap Script, and Reservations.
- **Billing** — View the Cost Dashboard and configure Allocation Thresholds.
- **Partner** — For partner organizations: configure Billing Multiplier and Instance Limits.
- **Admin** — Platform-level configuration, visible to platform administrators only.

Organizations for Partners and ACTIVATE Administrators

The **Organizations** list will look different for you than it will for most users. Partners and administrators see a list that includes their own organization and all organizations that they manage.

This page displays each organization's name and which organization it's managed by (if any). Click on any organization to open its settings.

There are two features that are important for partner organizations: configuring billing multipliers and creating organizations.

About Group Roles

We currently have four ACTIVATE roles to customize user permissions within your organization.

- Users with the `org:admin` permission have full control. They can make any changes to any settings in the organization.
- Users with the `org:settings` permission can manage organization settings.
- Users with the `org:users` permission can create and manage users.
- Users with the `org:groups` permission can create and manage groups. However, they cannot assign or remove roles for groups; only users with the `org:admin` permission have access to make those changes.

These roles are controlled through groups. All users within a group have the same permissions. Groups can also have multiple roles assigned to them.

At this time, you cannot assign roles to individual users.

Creating Organizations

This page explains how to create and delete organizations on ACTIVATE.

Requires: `platform:admin`

Creating an Organization

Navigate to the **Organizations** list.

Click **Create Organization**.

Enter a name for your organization, then click **Create Organization**.

A success message will appear, and the new organization will be listed on the **Organizations** page.

Deleting Organizations

> **WARNING**
>
> Deleting an organization is a permanent action that cannot be undone. All users, groups, cloud accounts, and associated cloud resources will be removed.

Navigate to the organization you want to delete. In the sidebar, click **Admin**.

At the bottom of the Admin page, click **Delete this organization**. A confirmation modal will appear requiring you to type the organization name to confirm the deletion.

For more details on the Admin page, see Platform Administration.

Managing Users

This page explains how to list, create, import, edit, and delete users on ACTIVATE.

Requires: `org:admin` or `org:users`

Listing Users

From the **Organizations** list, select your organization. The **Users** page is the default view.

The users table displays the following columns:

- **User** — Avatar, display name, and email address. A green dot on the avatar indicates the user is currently online (active within the last 60 seconds).
- **Status** — Shows **Active** (green badge) for licensed users or **No license** (amber badge) for users without a license.
- **Last Login** — The date and time of the user's most recent login.
- **Actions** — Available actions such as **Impersonate user** (platform administrators only).

Searching and Filtering

Use the **search field** above the table to filter users by username or email. Results update automatically as you type.

Filter pills below the search field let you narrow the list:

- **All** — Show all users.
- **Active** — Show only users with an active license.
- **No license** — Show only users without a license.

Sorting

Click the **Username** or **Last Login** column headers to sort the table by that column. Click again to reverse the sort order.

Pagination

The table displays 50 users per page. Use the **Previous** and **Next** buttons at the bottom of the table to navigate between pages.

The table auto-refreshes every 30 seconds to keep the data current.

Creating New Users

To create a new user individually:

1. On the **Users** page, click the **Add user** button in the top-right corner.
2. Fill in the form fields, which are arranged in a 2x2 grid:
    - **Name** — The user's display name.
    - **Username** — Must contain only alphanumeric characters, periods, and hyphens.
    - **Email** — The user's email address.
    - **Password** — Must meet complexity requirements (see below).
3. Click **Create user** to submit.

Password Requirements

Passwords must meet all of the following complexity requirements:

- Minimum length
- At least one uppercase letter
- At least one lowercase letter
- At least one number
- At least one special character

INFO

> We do not send any emails regarding the creation of new users. You will need to share the login credentials with the user directly.

Importing Users

The import feature lets you create multiple users at once using a CSV file. It can also be used for mass-updating existing users.

1. On the **Users** page, click the **Import** link next to the **Add user** button.
2. Follow the 3-step process:

Step 1: Download Template

Click **Download template** to download a CSV file. Open the file in any spreadsheet editor (Excel, Numbers, Google Sheets, etc.).

Step 2: Fill In the CSV

The CSV template includes the following fields:

Field	Description
`username`	Required. The user's login name.
`password`	The user's password.
`name`	The user's display name.
`email`	The user's email address.
`uid`	Optional. A custom user identifier.
`groups`	Optional. Group names separated by semicolons (e.g., `engineering;design`).
`active`	Whether the user account is active.

Step 3: Upload and Import

1. Click **Upload template** and select your completed CSV file.
2. A preview table displays the uploaded data. Review it for accuracy.
3. Click **Import** to create the users.

After the import completes, an **Import Log** appears showing per-user results, including any errors that occurred.

> **TROUBLESHOOTING**
>
> On macOS, spreadsheet applications may attempt to save a CSV file in a different format (e.g., `.numbers` or `.xlsx`). Ensure you save the file as a `.csv` before uploading, as the import feature only accepts CSV files.

Editing Users

Click a username in the users table to open the user detail page.

The user detail page uses a two-column layout:

- **Left column — User Details panel**
  - **Username** — Read-only, cannot be changed after creation.
  - **Name** — Editable display name.
  - **Email** — Editable email address.
  - **Status** — Current account status.
  - **Created** — Date the account was created.
  - **Organization** — The organization the user belongs to.
- **Right column — User Activity** (platform administrators only)
  - Displays a graph of the user's activity over the past year.

Click **Update User** to save any changes to the user details.

Feature Previews

Organization administrators can toggle feature flags for individual users in the **Feature Previews** section of the user detail page. This is useful for enabling upcoming features for testing.

Platform Admin Settings

Platform administrators have access to additional settings:

- **No seat** — When enabled, the user does not consume one of the organization's allotted seats. Useful for service accounts or test accounts.
- **Admin** — Grants the user platform administrator privileges.

User Detail Sub-Pages

The user detail page includes a navigation bar at the top with links to the following sub-pages:

Groups

View and manage the user's group memberships.

- Use the dropdown to add the user to a group.
- Click the trash icon next to a group to remove the user from it.

Workspace

Configure the user's workspace settings:

Setting	Description
Type	Workspace type (Docker or Kubernetes).

User Host	The host where the workspace runs.
Image	The container image for the workspace.
CPU Min / Max	Minimum and maximum CPU allocation.
Memory Min / Max	Minimum and maximum memory allocation.
Ephemeral Storage Max	Maximum ephemeral storage capacity.
Debug Mode	Enable debug mode for troubleshooting.

External Auth

View and edit external authentication mappings for each configured authentication method. Available fields depend on your organization's auth configuration and may include:

- **CAC ID**
- **LDAP UID**
- **OIDC Sub**

Delete

A delete confirmation page that displays the resources associated with the user that will be cleaned up upon deletion. See Deleting Users for details.

User Actions

The user detail page includes an action bar with the following options. Available actions depend on your role.

Impersonate User

Platform administrators only.

Click **Impersonate User** to log in as the selected user. This is useful for troubleshooting issues from the user's perspective.

Restart Workspace

Click **Restart Workspace** in the action bar. A confirmation dialog will appear asking: *Are you sure you want to restart [user's] workspace?*

Click **Restart** to confirm. This is useful if the user is experiencing platform errors.

> **TROUBLESHOOTING**
>
> Please contact us if restarting the workspace does not resolve the user's errors.

Disable / Enable User

Click **Disable User** to deactivate the account. Disabled users:

- Cannot log in to ACTIVATE.
- Do not consume one of the organization's allotted seats.

To re-enable the account, click **Enable User** from the same action bar.

> **CAUTION**
>
> A warning is displayed if the user has active resources when you attempt to disable their account.

Change Password

Click **Change Password** to open a modal with two fields:

- **New Password** — Enter the new password (must meet password complexity requirements).
- **Confirm Password** — Re-enter the new password.

Remove MFA

Platform administrators and organization administrators only.

Click **Remove MFA** to remove multi-factor authentication from the user's account. The user will need to set up MFA again on their next login if MFA is required by the organization's policies.

Revoke Login Sessions

Click **Revoke Login Sessions** to force the user out of all active sessions. The user will need to log in again.

Delete User

Click **Delete User** to navigate to the delete confirmation page. See Deleting Users for details.

Deleting Users

To delete a user:

1. Navigate to the user detail page by clicking their username in the users table.
2. Click **Delete User** in the action bar, or navigate to the **Delete** sub-page.
3. The delete confirmation page displays a list of resources associated with the user, including each resource's name, type, and status.
4. Click the **Delete** button.
5. A confirmation modal appears. Confirm the deletion.

> **CAUTION**
>
> If the user has active resources, an amber warning explains that those resources will be deprovisioned as part of the deletion process.

Monitoring User Status

In the users table, users who have been online within the past minute display a green status dot on their avatar. This provides a quick way to see which users are currently active on the platform.

Managing Groups

This page explains how to create, edit, and delete groups on ACTIVATE. Groups are used to assign user permissions, designate cost allocations, and share resources between users.

Requires: `org:admin` or `org:groups`

Navigating to the Groups Page

From the **Organizations** list, select your organization. In the sidebar, click **Groups**.

The **Groups** page displays the section header "Groups" with the description "Manage groups and their resource allocations."

Listing Groups

All groups in your organization are listed in a table with the following columns:

- **Group Name** -- clickable link that opens the group edit page
- **Description** -- the group's description
- **Allocation Used** -- the amount of allocation currently consumed by the group
- **Allocation Total** -- the total allocation assigned to the group
- **Members** -- the number of users in the group
- **Created** -- the date the group was created (formatted as MM/DD/YY)
- **Delete** -- a trash icon to delete the group

Allocation values are formatted with thousand separators for readability.

Use the **Search groups...** field to filter groups by name. The search is case-insensitive.

Creating a Group

1. On the **Groups** page, click **Add Group** in the top right corner.
2. Enter a **Name** for your group. The name input automatically removes special characters and only allows alphanumeric characters, dashes, and underscores.
3. Optionally, enter a **Description**.
4. Click **Create Group** to submit.

You will be redirected to the groups list on success, where your new group will appear in the table.

Editing Allocations

You can update allocation totals for all groups at once using batch editing.

1. On the **Groups** page, click **Edit Allocations** in the groups list header.
2. The **Allocation Total** column becomes editable for all groups. Modify values as needed.
3. Click **Save Allocations** to save all changes in a single batch update.

Success or error messages are shown per group to indicate the result of each update.

Editing a Group

Click a group name in the table to open the group edit page. The edit page has three sections: **Manage Description**, **Manage Roles**, and **Manage Members**.

Manage Description

Use the text input to update the group's description. Click **Save Description** to save your changes.

Manage Roles

This section displays four role checkboxes that control what permissions members of the group have within your organization:

- `org:admin` -- Full control over an organization
- `org:settings` -- Can manage organization settings
- `org:users` -- Can create and manage users
- `org:groups` -- Can create and manage groups, except groups with the `org:admin` role

Check or uncheck the roles as needed, then click **Save Roles** to apply your changes immediately.

Manage Members

Use the dropdown search field to find and add organization users to the group. Select a user from the dropdown to add them as a member.

Current group members are displayed in a table with the following columns:

- **Members** -- the member's username
- **Email** -- the member's email address
- **Delete** -- a trash icon to remove the member from the group

To remove a member, click the trash icon next to their name. The user will be removed from the group immediately.

> **ABOUT REMOVING MEMBERS**
>
> Removing a user from a group only affects their group-based permissions and resource allocations. Their account and their access to ACTIVATE will not be affected.

Deleting a Group

1. In the groups table, click the trash icon on the row of the group you want to delete.
2. A confirmation modal will appear: "Are you sure you want to delete this group?"
3. Click **Confirm** to delete the group.

SCIM Provisioning

Requires: `org:admin`

ACTIVATE exposes a **read-only SCIM 2.0 API** that lets external services synchronize your organization's users and groups. This is how systems outside ACTIVATE discover who your users are, what groups they belong to, and the POSIX identity and SSH keys associated with each account.

> **DIRECTION OF SYNC**
>
> ACTIVATE is the **source** of identity, not a destination. Consuming services *read* users and groups from ACTIVATE; they cannot create, modify, or delete them through this API. Every write request (`POST`, `PUT`, `PATCH`, `DELETE`) returns `501 Not Implemented`.

Enabling SCIM

1. From the **Organizations** list, select your organization.
2. In the left sidebar, under **Settings**, click **SCIM Provisioning**.
3. Toggle **Enable SCIM provisioning**.

Once enabled, the page displays your organization's **SCIM endpoint URL**. Provide this URL to the consuming service:

```
https://<platform-host>/api/organizations/<organization>/scim/v2
```

The endpoint is scoped to a single organization - only that organization's users and groups are returned.

Bearer Tokens

SCIM requests are authenticated with a bearer token. Tokens are scoped to the organization and are independent of any single user account.

To create a token:

1. On the **SCIM Provisioning** page, find the **Bearer Tokens** section.
2. Click **Create SCIM token**.
3. Enter a descriptive **name**. Optionally set an **expiration** (in days); leave it blank for a non-expiring token.
4. Copy the generated token.

> **TOKENS ARE SHOWN ONCE**
>
> The full token value is displayed only at creation time and is never shown again. Store it securely (for example in the consuming service's secret manager) before closing the dialog. If you lose it, revoke the token and create a new one.

The consuming service sends the token as a standard bearer credential on every request:

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

To revoke access, delete the token from the **Bearer Tokens** list. Any service using that token immediately loses access to the SCIM API.

> **TOKEN CREATION REQUIRES A BROWSER SESSION**
>
> For security, SCIM tokens can only be minted from a logged-in browser session by an organization admin - not via the API itself.

What the API Exposes

The API implements the SCIM 2.0 discovery and resource endpoints:

Endpoint	Purpose
`GET /ServiceProviderConfig`	Advertises supported capabilities (filtering, bearer auth).
`GET /ResourceTypes`	Lists the `User` and `Group` resource types.
`GET /Schemas`	Returns the core User/Group schemas plus the CoreWeave extensions.
`GET /Users` , `GET /Users/{id}`	List or fetch users.
`GET /Groups` , `GET /Groups/{id}`	List or fetch groups.

Filtering and pagination

- **Users** can be filtered by `userName` , `externalId` , or `displayName` using the `eq` operator, e.g. `?filter=userName eq "jdoe@example.com"` .
- **Groups** can be filtered by `displayName eq` .
- Results are paginated with `startIndex` (1-based) and `count` (default 50, maximum 200).

CoreWeave extension attributes

ACTIVATE groups map directly to SCIM groups, and ACTIVATE users to SCIM users. In addition to the standard SCIM fields, each resource carries a CoreWeave extension that exposes the POSIX identity consumers need to provision Linux accounts.

The extension blocks are **omitted by default** and are only returned when explicitly requested via the `attributes` query parameter:

```
?attributes=urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser
```

User extension (`urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveUser`):

Attribute	Description
`sunkPosixUsername`	The user's POSIX (Linux) username.
`sunkPosixUserId`	The user's POSIX UID.

`sunkPosixGroupId`	The user's primary POSIX GID.
`sunkLoginShell`	Login shell (defaults to `/bin/bash`).
`sunkPreferredHomeDirectory`	Home directory (defaults to `/home/<username>`).
`sunkSshKeys`	The user's registered authorized SSH public keys.

Group extension

(`urn:coreweave:params:scim:schemas:extension:coreweave:2.0:CoreWeaveGroup`):

Attribute	Description
`sunkPosixGroupId`	The group's POSIX GID.
`sunkPosixGroupName`	The group's POSIX name.

> **EXCLUDING INACTIVE USERS**
>
> When listing groups, pass `?excludeInactiveUsers=true` to drop disabled users from each group's member list. CoreWeave's identity cache uses this so that deactivated ACTIVATE accounts stop resolving on the cluster.

Next Steps

- Connecting a CoreWeave Slurm Cluster - the most common consumer of this API, which uses SCIM to populate Linux users, groups, and SSH keys on the cluster.

Settings

Organization Policies

Requires: `org:admin` or `org:settings`

Navigation

From the Organizations list, select your organization. In the sidebar, under **Settings**, click **Policies**.

Overview

Organization policies are scoped to your organization, but some platform-level policies take precedence and cannot be changed.

Each policy has three states controlled by a toggle:

- **Enabled** -- The policy is actively enforced for your organization.
- **Disabled** -- The policy is explicitly turned off for your organization.
- **Not Set** -- The policy inherits the platform default behavior.

Policies

No Root Access

Disable root access to cloud compute resources for all users, including resource owners, in your organization. Defaults to "Enable root access" if no policy is set.

Nitro Instance Types Only

Restrict compute resources to AWS Nitro instance types only, in your organization. Defaults to "Allow all AWS instance types" if no policy is set.

Archive Cost Data

Automatically summarize and then archive cost data after a specified number of months to optimize database performance. This will not delete any data, it will only summarize older data.

Enforce YubiKey MFA

> **FEATURE PREVIEW**
>
> This policy is feature-flagged and may not be visible in your organization.

Require all users in your organization to set up and use a YubiKey for multi-factor authentication when logging in with a password. Users without a registered security key will be prompted to register one before accessing the platform.

Webhooks

This page explains how to add webhooks to your organization. Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide.

Requires: `org:admin`

Adding a Webhook

From the **Organizations** list, select your organization. In the sidebar, under **Settings**, click **Webhooks**.

Click + **Add Webhook**.

On the next page, select a **Type**. Currently, you can choose from these webhook types:

- `USER_CREATED` flags when a new user logs in for the first time.
- `USER_DISABLED` flags when a user has been disabled.
- `USER_REMOVED_FROM_GROUP` flags when a user has been removed from a group in your organization.

Add a **Webhook name** and paste the **URL** that your webhook will be sent to. **Description** is an optional field.

Choose whether to **Enable** this webhook.

Click **Create webhook**.

You'll be taken back to the **Webhooks** page. A dialog box with the message *Webhook created successfully* will appear in the bottom right corner of your screen.

Deleting a Webhook

From the **Organizations** list, select your organization. In the sidebar, under **Settings**, click **Webhooks**. Select the webhook you'd like to delete, then click **Delete webhooks**.

A pop-up module will appear. Click **Delete**.

You'll be taken back to the **Webhooks** page. A dialog box with the message *Webhook(s) deleted successfully* will appear in the bottom right corner of your screen.

Responding to Events

When your application receives webhook events, they'll take the shapes below. All `webhook` shapes are identical, while the `payload` shape changes according to the type of event.

Please also note that `organization` has a different meaning, depending on its location:

- `organization` inside the `webhook` object indicates the organization that owns the webhook.
- `organization` inside the `payload` object indicates the organization that the user is in.

`USER_CREATED`

```
{
    "type": "USER_CREATED",
    "webhook": {
        "id": "66ec434e84b4103f670cb617",
        "organization": "parallelworks",
        "name": "demo"
    },
    "payload": {
        "username": "demo",
        "email": "demo@parallelworks.com",
        "userId": "66ec445884b4103f670d0c86",
        "organization": "parallelworks",
        "remainingSeats": 5
    }
}
```

`USER_DISABLED`

```
{
    "type": "USER_DISABLED",
    "webhook": {
        "id": "66ec41b884b4103f670c5ee5",
        "organization": "parallelworks",
        "name": "test"
    },
    "payload": {
        "username": "demo",
        "email": "demo@parallelworks.com",
        "userId": "653c787c8e048caa7cb4e554",
```

```
        "organization": "parallelworks"    }}
```

`USER_REMOVED_FROM_GROUP`

```
{
    "type": "USER_REMOVED_FROM_GROUP",
    "webhook": {
        "id": "66ec439584b4103f670cd3ae",
        "organization": "parallelworks",
        "name": "test"
    },
    "payload": {
        "username": "demo",
        "email": "demo@parallelworks.com",
        "userId": "653c787c8e048caa7cb4e554",
        "organization": "parallelworks",
        "groupName": "Demo-Group"
    }
}
```

Workflow Variables

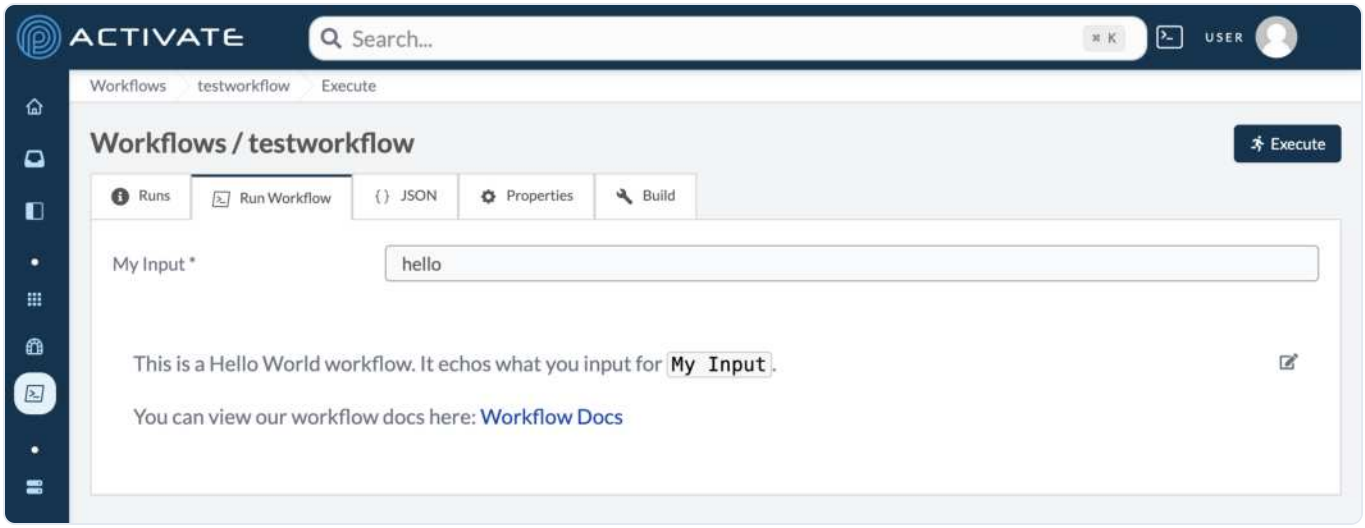
About Workflow Variables

Requires: `org:admin` or `org:settings`

Contexts can be a useful feature when you are building workflows on ACTIVATE. Contexts have many uses, such as setting the default values for input fields in the workflow's configuration form. The `org` context calls custom workflow variables. These variables are configured in your organization's settings.

In the example below, you will see that the form for this workflow contains only one input field. Any text you enter will be output along with the value of `orgVariable`. If you run the workflow without entering an input, only the value of `orgVariable` will be output.

```
jobs:
  main:
    steps:
      - name: Run
        run: echo ${{ inputs.hello }} ${{ org.orgVariable }}
'on':
  execute:
    inputs:
      hello:
        label: My Input
        type: string
        default: '${{ org.orgVariable }}'
```



The workflow's input

The workflow's output

Configuring Workflow Variables

From the **Organizations** list, select your organization. In the sidebar, under **Settings**, click **Variables**.

In **Workflow Variables**, click **Add Key Value Pair**.

Enter a **Key**, then enter a **Value**. Click **Save**. Users will immediately be able to use the new workflow variable.

Adding Authentication Options

ACTIVATE allows you to configure how users authenticate to your organization. You can set up multiple authentication methods, including passwords, LDAP, and OpenID Connect (OIDC), to match your organization's security requirements.

Requires: `org:admin`

Navigating to Authentication Settings

From the **Organizations** list, select your organization. In the sidebar, under **Settings**, click **Authentication**.

The Authentication page displays a table of all configured authentication methods with the following columns:

- **Name** — The display name of the method (clickable for LDAP and OIDC methods)
- **Type** — The method type (`cac` , `ldap` , `oidc` , or `password`)
- **Delete** — A trash icon to remove the method

If no methods have been configured, the page displays: "No authentication options configured."

Adding Authentication Methods

Click the **Add method** dropdown button to see the available options:

- **Password** — Creates a password-based login immediately with no additional configuration. This option is only shown if no password method already exists.
- **LDAP** — Opens the LDAP configuration page.
- **OIDC** — Opens the OIDC configuration page.

Password Authentication

Password authentication requires no configuration. Select **Password** from the **Add method** dropdown and the method is created immediately.

Only one password method is allowed per organization. Once a password method exists, the **Password** option is hidden from the dropdown. Password methods can be deleted like any other method.

LDAP Authentication

To add a new LDAP method, select **LDAP** from the **Add method** dropdown. To edit an existing LDAP method, click its name in the authentication methods table.

Basic Connection Settings

Field	Required	Description

Name	Yes	Display name shown on the login page.
Domain	Yes	LDAP server domain (e.g., `example.com`).
Filter	Yes	User lookup filter. Use `__USERNAME__` as a placeholder for the authenticating user's name (e.g., `(uid=__USERNAME__)`).
Base DN	No	Starting point for LDAP searches (e.g., `ou=orgUnit,dc=example,dc=com`).

Service Account

Toggle **Use Service Account** to enable or disable service account authentication.

- **When enabled:** Two additional fields appear:
  - **Service Account Bind** — The full distinguished name (DN) of the service account.
  - **Service Account Password** — The password for the service account.
- **When disabled:** The system attempts an anonymous bind.

TLS / Security

Toggle **Use TLS** to enable or disable Transport Layer Security when connecting to the LDAP server.

- **When enabled:** Two additional fields appear:
  - **Client Certificate** — Paste the PEM-encoded client certificate (textarea).
  - **Client Key** — Paste the PEM-encoded client key (textarea). Required when creating a new LDAP method.

User Attribute Mapping

These fields control how LDAP attributes map to ACTIVATE user properties.

Field	Default	Description
Unique Identifier	`cn`	Attribute that uniquely identifies users.
Username	`uid`	Attribute used as the username.
Full Name	`cn`	Attribute used as the display name.
Email	`mail`	Attribute used as the email address.
UID Number	*(none)*	Unix UID used during account creation. If not set, a UID is auto-generated.

Test Connection

Before saving, you can verify the LDAP configuration by testing a connection.

1. Enter a **Username** and **Password** for a known LDAP account.
2. Click **Test Connection**.
3. The system will attempt to authenticate using the current configuration and report the result.

OIDC Authentication

To add a new OIDC method, select **OIDC** from the **Add method** dropdown. To edit an existing OIDC method, click its name in the authentication methods table.

Basic Information

Field	Default	Description
Name	`oidc`	Internal identifier. Must be lowercase alphanumeric characters and hyphens only.
Display Name	`OIDC`	The label shown on the login page.

Endpoint Discovery

Toggle **Discover Endpoints?** to control how OIDC endpoints are configured. This toggle is enabled by default.

When enabled: Enter only the **Issuer URL**. All other endpoints (user info, token, authorization, and end session) are discovered automatically from the provider's well-known configuration.

When disabled: Enter each endpoint manually:

Field	Required	Description
User Info Endpoint	Yes	URL for retrieving user profile information.
Token Endpoint	Yes	URL for exchanging authorization codes for tokens.
Authorization Endpoint	Yes	URL where users are redirected to authenticate.
End Session Endpoint	No	URL for logging users out of the identity provider.

Client Configuration

Field	Default	Description
Client ID	*(none)*	The client ID from your identity provider. Required.
Scopes	`openid profile email`	Space-separated list of OAuth scopes to request. Required.

Authentication Method

The **Token Endpoint Auth Method** dropdown determines how the ACTIVATE platform authenticates with your identity provider's token endpoint. The available options are:

- `client_secret_post` — The client secret is sent in the POST body of token requests.
- `client_secret_basic` — The client secret is sent in the Authorization header as a Base64-encoded string.
- `private_key_jwt` — A JWT signed with a private key is used for authentication.

Credentials change based on the selected auth method:

- For `client_secret_post` or `client_secret_basic`: A **Client Secret** text field is displayed.
- For `private_key_jwt`: A **Private Key PEM** textarea is displayed where you paste the PEM-encoded private key.

Registration Options

Option	Default	Description
Title Case	Off	Converts usernames to title case for normalization.
Skip Platform MFA Verification	Off	Bypasses the additional MFA step after OIDC login.
Create account on first login	On	Automatically creates ACTIVATE accounts for new OIDC users. When disabled, only users with existing ACTIVATE accounts can log in via OIDC.

Redirect URI

A read-only field at the bottom of the configuration page displays the callback URL:

```
https://your-domain/api/sso/oidc/callback
```

Copy this value and configure it as an allowed redirect URI in your identity provider.

Deleting Authentication Methods

To delete any authentication method:

1. Click the trash icon next to the method in the authentication methods table.
2. A confirmation modal appears: *"Are you sure you want to delete this authentication option? Users will no longer be able to login with this option."*
3. Confirm the deletion.

Supported Identity Providers

OIDC authentication supports any standards-compliant OpenID Connect identity provider, including:

- Azure Active Directory (Entra ID)
- Google Workspace
- Okta
- Keycloak
- Auth0
- Login.gov

Customize

Customizing Organization Theme

Here you can change how the page looks for users in the organization. All users in the organization will see these changes. Users can opt-out of organization-specific themes in their account settings.

Requires: `org:admin` or `org:settings`

Navigation

From the **Organizations** list, select your organization. In the sidebar, under **Customize**, click **Theme**.

Options

At the top of the Theme page, you will find two options that control how you interact with the theme editor:

- **Preview Settings** — When checked, changes are previewed live on the page as you edit. This allows you to see how your theme will look before saving.
- **Calculate Complementary Colors** — When enabled, selecting a color for certain settings will automatically calculate and set complementary colors for related settings. For example, setting a sidebar hover background color will auto-suggest a matching sidebar hover text color.

Color Presets

A row of predefined color presets is displayed near the top of the page. Selecting a preset populates all color fields at once, giving you a quick starting point for your theme.

Select **default** to reset to the platform default theme.

Top Banner Settings

Configure a banner that appears at the top of every page for users in your organization.

- **Top Banner Message** — Text displayed in the banner at the top of every page.
- **Top Banner Text Color** — Color of the banner text.
- **Top Banner Background Color** — Background color of the banner.

Logo Settings

Upload and configure organization logos.

- **Show Powered By** — Checkbox to display "Powered by Parallel Works" next to the logo. This option is only available to platform administrators.
- **Standard Logo** — Upload or remove the organization logo displayed on light backgrounds.
- **Dark Mode Logo** — Upload or remove the logo variant displayed on dark backgrounds.
- **Invert Logo on Dark Backgrounds** — Toggle to invert logo colors on dark backgrounds.

Sidebar Settings

Customize the colors of the sidebar navigation.

- **Banner Color** — The primary color for the sidebar header/banner area.

- **Sidebar Text Color** — Text color in the sidebar.
- **Sidebar Hover Background** — Background color when hovering over sidebar items.
- **Sidebar Hover Text** — Text color when hovering over sidebar items.
- **Sidebar Active Background** — Background color of the currently active sidebar item.
- **Sidebar Active Text** — Text color of the currently active sidebar item.

Element Settings

Configure the appearance of primary interactive elements such as buttons.

- **Element Color** — Primary button/element background color.
- **Element Text Color** — Text color on primary buttons/elements.

App & Panel Settings

Control the colors of the main application layout and content panels.

- **App Background** — Main application background color.
- **App Text Color** — Main application text color.
- **Panel Background** — Panel/card background color.
- **Panel Text Color** — Panel/card text color.
- **Muted Panel Background** — Background for muted/secondary panels.
- **Muted Panel Text Color** — Text color in muted panels.
- **Muted Text Color** — Color for secondary/helper text.
- **Link Color** — Color for hyperlinks.

Border & Input Settings

Customize borders, dividers, and form input appearances.

- **Border Color** — Color for borders and dividers.
- **Input Background** — Background color for form inputs.
- **Disabled Input Background** — Background for disabled inputs.
- **Disabled Input Text** — Text color for disabled inputs.

Interactive Elements

Configure hover state colors for interactive elements throughout the interface.

- **Hover Color** — Background color for hover states on interactive elements.
- **Hover Text Color** — Text color for hover states.

Saving

Each color field has a reset button (refresh icon) that reverts that individual color to its last saved value.

- **Reset** — Reverts all changes to the last saved state.
- **Save Theme** — Saves all changes permanently. All users in the organization will see the updated theme.

Customizing Default Sidebar

Requires: `org:admin` or `org:settings`

Navigation

From the Organizations list, select your organization. In the sidebar, under **Customize**, click **Sidebar**.

Overview

Customize the default sidebar items shown to users in this organization. Users can override these settings from their personal sidebar settings.

Sidebar Modes

The page provides two modes via radio buttons:

Default Settings

Uses the platform default sidebar configuration. All navigation items are shown.

Custom Settings

Allows you to toggle visibility of individual sidebar navigation items. When selected, a list of all navigation items appears with checkboxes. Some items are marked as "unhideable" and cannot be removed.

How It Works

1. Select the **Custom Settings** radio button to reveal the list of sidebar navigation items.
2. Use the checkboxes to toggle visibility of each item.
3. Changes are applied immediately when toggling items.

> **NOTE**
>
> Individual users can override these organization defaults from their personal account settings.

User Defaults

About User Defaults

When new users log in to ACTIVATE for the first time, they are guided through an onboarding flow that helps them set up their account. During onboarding, users are presented with recommended compute resources and workflows to add to their account. The resources and workflows you configure on this page are what appear as **"Recommended by [Your Organization]"** during that onboarding flow.

Setting up default compute resources ensures new users start with the resource configurations your organization expects them to use. A copy of each selected resource is made for the new user's account — these copies are independent, so changes to one user's resource won't affect others.

Setting up default workflows is useful if there's a particular workflow that you expect all of your users to need, such as running a Jupyter Notebook.

Requires: `org:admin` or `org:settings`

Configuring Default Compute Resources

From the **Organizations** list, select your organization. In the sidebar, under **Customize**, click **User Defaults**.

Under **New User Compute Resources**, use the dropdown to browse compute resources published to the marketplace for your organization. Select a resource to add it to the defaults list.

Selected resources appear in a table below the dropdown with columns for **Resource**, **Type**, and **Description**. To remove a default resource, click the trash icon in its row.

Configuring Default Workflows

On the same page, scroll down to the **New User Workflows** section.

Use the dropdown to browse workflows published to the marketplace for your organization. Select a workflow to add it to the defaults list.

Selected workflows appear in a table below the dropdown with columns for **Workflow**, **Type**, and **Description**. To remove a default workflow, click the trash icon in its row.

Saving Changes

Click **Save** at the top right of the page to apply your changes. The selected resources and workflows will appear as recommendations during the onboarding flow for all new users created after saving.

Configuring Login Banner

Requires: `org:admin` or `org:settings`

Navigation

From the Organizations list, select your organization. In the sidebar, under **Customize**, click **Login Banner**.

Overview

Configure a notice banner that will be displayed on the organization login page. When set, the login page will show a split layout with this notice displayed on the left side.

The page has two panels side by side: a form on the left and a live preview on the right.

Fields

Banner Title

The title shown in the banner header (e.g., "System Access Notice").

Banner Body

The body content of the banner. Supports markdown formatting:

- **Bold text** using `**bold**`

- *Italic text* using `*italic*`
- Bullet lists
- Links using `[text](url)`

Saving Changes

- **Discard Changes** -- Resets the form to the last saved state.
- **Save Banner** -- Saves the banner configuration. This button is disabled until changes are made.

After saving, you are redirected back to the organization Users page.

Default Behavior

When no banner is configured, the login page uses the standard centered layout.

Cloud

Instance Types

This page explains how to choose which instance types users will have access to when they configure clusters on the **Compute** page. This feature is useful if, for example, your users only need a certain amount of power for their work; in that case, you can limit instances to only lower CPU types.

Requires: `org:admin` or `org:settings`

From the **Organizations** list, select your organization. In the sidebar, under **Cloud**, click **Instance Types**.

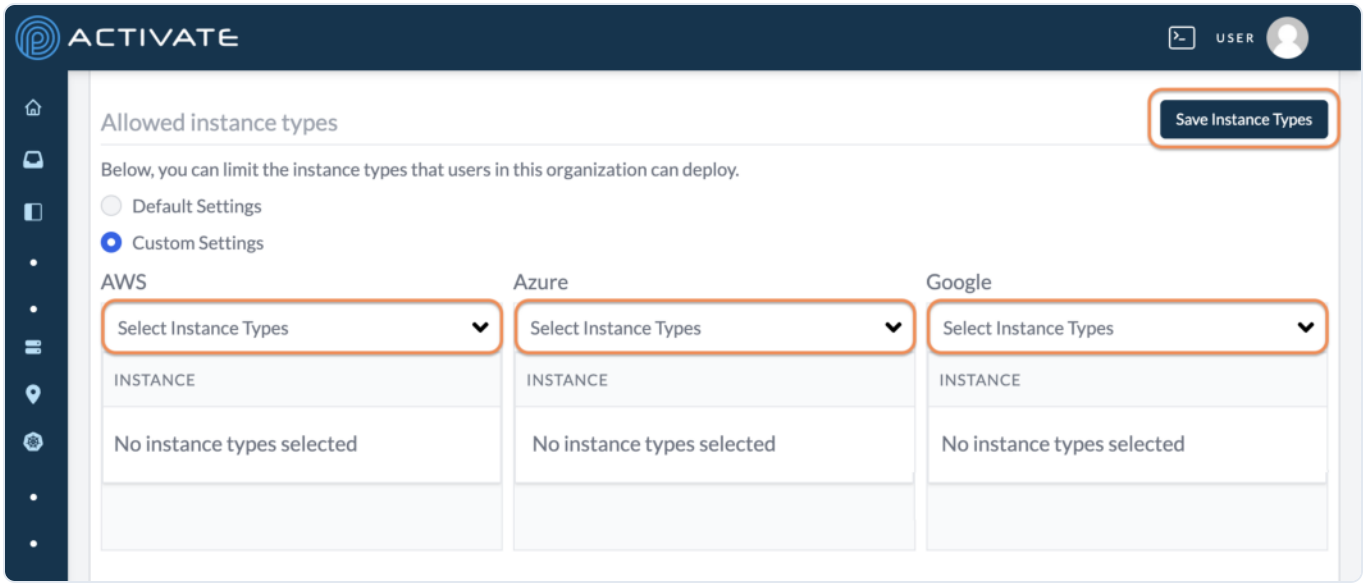
In **Allowed instance types**, if **Default Settings** is selected, users will have access to all instance types.

Select **Custom Settings** to make changes.

You can customize instance types on AWS, Azure, and Google resources. Use the arrows for each CSP to open their dropdown menus.

Click the instance type(s) you want to add. The instance type(s) you select will appear in the list below the dropdown menus.

When you're done, click **Save Instance Types**. Your changes will be applied immediately.



Bootstrap Script

About Organization Bootstrap Scripts

This page explains how to add a bootstrap script for your organization. The script will run whenever a user in your organization starts a cluster in their account. This feature is useful for automating tasks, such as sending ACTIVATE data to a specific location.

Requires: `org:admin`

Navigating to Organization Bootstrap Settings

From the **Organizations** list, select your organization. In the sidebar, under **Cloud**, click **Bootstrap Script**.

Enter your bootstrap script in the text box. This text will run as a bash script during cluster provisioning.

When you're done, click **Save**. Your changes will be applied immediately.

> **NOTE**
>
> If a user adds their own bootstrap script in a cluster's configuration settings, the script you set with the instructions will run before the user's added script.

Testing a Sample Bootstrap Script

You can test this feature with a simple script that echoes text in a designated file.

Enter the following command in the bootstrap script box:

```
echo "hello world" > /tmp/org-boot.out

exit 0
```

Click **Save**.

Navigate to the **Compute** page and start one of your clusters.

When the cluster is active, log in to the controller with `ssh`. For detailed instructions on that process, please see Logging In to the Controller.

After logging in, navigate to `/tmp`.

Read `org-boot.out` to see that it matches the `echo` command.

> **NOTE**
>
> If you encounter errors during this test, please contact us.

Capacity Reservations

Requires: `org:admin` or `org:settings`

Navigation

From the Organizations list, select your organization. In the sidebar, under **Cloud**, click **Reservations**.

Overview

Manually map capacity reservation IDs to groups for billing data.

Prerequisites

Before setting a capacity reservation on ACTIVATE, you must first create the reservation on your CSP's management console:

- AWS documentation
- Azure documentation
- GCP documentation

Adding a Reservation

1. Click **Add Reservation**.
2. Enter the **Reservation ID** from your CSP's management console.
3. Select a **Group** from the dropdown. Note: only one group can be assigned per capacity reservation, and access will be limited to that group's members.
4. Click **Add**.

Managing Reservations

Reservations appear in a table and can be deleted individually.

Cloud Accounts

Requires: `org:admin`

What Is a Cloud Account?

A cloud account is ACTIVATE's unified concept for connecting to and managing infrastructure on a cloud service provider (CSP). Each cloud account represents a set of credentials and configuration that links your CSP environment to ACTIVATE, allowing you to provision networks, manage billing, control access, and run compute workloads -- all from a single interface.

Cloud accounts are managed at the organization level. Once a cloud account is added, it becomes available for network provisioning, billing configuration, and access sharing with groups within your organization.

Supported Cloud Service Providers

ACTIVATE supports the following cloud service providers:

- **Amazon Web Services (AWS)** -- authenticated via access keys
- **Microsoft Azure** -- authenticated via service principal client secrets
- **Google Cloud** -- authenticated via service account keys
- **OpenStack** -- authenticated via application credentials

> **FEATURE FLAG**
>
> **Oracle Cloud Infrastructure (OCI)** is available as a feature-flagged provider. Contact your ACTIVATE administrator to enable Oracle Cloud support for your organization.

Cloud Account Lifecycle

Setting up a cloud account follows these general steps:

1. **Prepare your CSP** -- Configure the necessary roles, permissions, and credentials in your cloud provider. See the CSP-specific preparation guides in Managing Cloud Accounts for details.
2. **Add the cloud account** -- Register the cloud account in ACTIVATE by providing your CSP credentials. ACTIVATE will verify that the credentials have sufficient permissions.
3. **Create networks** -- Provision the networking infrastructure (VPCs, subnets, route tables, etc.) that clusters will be deployed on. See Networks for details.
4. **Provision billing** -- Set up billing infrastructure so that ACTIVATE can track and report on cloud spend. See Billing for details.
5. **Configure access** -- Share the cloud account with specific groups in your organization. See Access Control for details.

GovCloud and Assured Workloads

ACTIVATE supports government-specific cloud environments for the following providers:

- **AWS GovCloud** -- isolated AWS regions designed to host sensitive data and regulated workloads
- **Azure Government** -- Microsoft's cloud environment for US government agencies and their partners
- **Google Assured Workloads** -- compliance-focused workload configurations for regulated industries

When adding a cloud account for one of these providers, you can enable the GovCloud option to ensure ACTIVATE provisions resources in the appropriate government regions.

Naming Convention

Cloud account names must use **lowercase letters and numbers only**. Special characters, spaces, and uppercase letters are not permitted. Choose a descriptive name that identifies the CSP and purpose of the account, as the name cannot be changed after creation.

AWS

This page explains how to set up an AWS account so that ACTIVATE can manage your AWS infrastructure, clusters, billing, storage, and usage data.

> **INFO**
>
> The steps on this page should be completed by a **cloud engineer** in your organization.

AWS Account

We recommend creating a new AWS account for the ACTIVATE platform, which will allow you to keep your existing AWS account separate from the platform and make it easier to manage billing and usage data. This will also ensure the principle of least privilege, as ACTIVATE will only have access to the resources it needs to manage.

If you use AWS Organizations, you can create a new account within your organization. Otherwise, you can make a non-organization account.

Setting Up AWS Credentials

To get started quickly, you can create a new IAM User and assign the `AdministratorAccess` AWS-managed policy to it.

Alternatively, you can create the policies listed in **AWS Policies** below, then attach those policies to the IAM user.

Create an Access Key

ACTIVATE requires the use of an AWS access key to authenticate with AWS. If you don't have an access key, you can create one.

> **SECURITY BEST PRACTICES**
>
> The ACTIVATE platform will immediately rotate the secret access key after it's entered into the system. The platform will then use the rotated secret access key to generate short-term credentials, which will be used by all ACTIVATE services.
>
> For more information about AWS keys and security best practices, see this FAQ on the AWS website.

AWS Policies

This section includes the policies you'll need to attach to the IAM user you create for ACTIVATE. You can create these policies in the IAM console, or you can create them in the AWS CLI by entering the JSON files listed under each policy.

pw-ec2-mgmt

This policy allows ACTIVATE to manage EC2 resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "pwec2mgmt",
      "Effect": "Allow",
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AssociateDhcpOptions",
        "ec2:AssociateRouteTable",
        "ec2:AttachInternetGateway",
        "ec2:AttachVolume",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateDhcpOptions",
        "ec2:CreateImage",
        "ec2:CreateInternetGateway",
        "ec2:CreateKeyPair",
        "ec2:CreateNatGateway",
        "ec2:CreateNetworkInterface",
        "ec2:CreatePlacementGroup",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateSubnet",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:CreateVpc",
        "ec2:DeleteDhcpOptions",
        "ec2:DeleteInternetGateway",
        "ec2:DeleteKeyPair",
        "ec2:DeleteNatGateway",
        "ec2:DeletePlacementGroup",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DeleteSecurityGroup",
        "ec2:DeleteSnapshot",
        "ec2:DeleteSubnet",
        "ec2:DeleteVolume",
        "ec2:DeleteVpc",
        "ec2:DeregisterImage",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
```

```
        "ec2:DescribeAvailabilityZones",        "ec2:DescribeDhcpOptions",
"ec2:DescribeImages",        "ec2:DescribeInstanceAttribute",        "ec2:DescribeInstanceTypes",
"ec2:DescribeInstances",        "ec2:DescribeInternetGateways",        "ec2:DescribeKeyPairs",
"ec2:DescribeNatGateways",        "ec2:DescribeNetworkAcls",
"ec2:DescribeNetworkInterfaces",        "ec2:DescribePlacementGroups",
"ec2:DescribeRegions",        "ec2:DescribeRouteTables",        "ec2:DescribeSecurityGroups",
"ec2:DescribeSnapshots",        "ec2:DescribeSubnets",        "ec2:DescribeTags",
"ec2:DescribeVolumes",        "ec2:DescribeVpcAttribute",        "ec2:DescribeVpcClassicLink",
"ec2:DescribeVpcClassicLinkDnsSupport",        "ec2:DescribeVpcs",
"ec2:DetachInternetGateway",        "ec2:DetachVolume",        "ec2:DisassociateAddress",
"ec2:DisassociateRouteTable",        "ec2:GetPasswordData",        "ec2:ImportKeyPair",
"ec2:ModifyImageAttribute",        "ec2:ModifyInstanceAttribute",
"ec2:ModifyNetworkInterfaceAttribute",        "ec2:ModifySubnetAttribute",
"ec2:ModifyVpcAttribute",        "ec2:RegisterImage",        "ec2:ReleaseAddress",
"ec2:RevokeSecurityGroupEgress",        "ec2:RunInstances",        "ec2:StartInstances",
"ec2:StopInstances",        "ec2:TerminateInstances"      ],      "Resource": "*"    }  ]}
```

pw-fsx-mgmt

This policy allows ACTIVATE to manage FSx resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "pwfsxmgmt",
      "Effect": "Allow",
      "Action": [
        "fsx:CreateFilesystem",
        "fsx:DeleteFilesystem",
        "fsx:DescribeFilesystems",
        "fsx:ListTagsForResource",
        "fsx:TagResource",
        "fsx:UntagResource",
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

> **NOTE**
>
> `iam:CreateServiceLinkedRole` and `iam:PutRolePolicy` are required for using S3 import/export with FSx. If S3 import/export is not needed, these permissions can be removed.

pw-route53-mgmt

This policy allows ACTIVATE to manage Route 53 resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {      "Sid": "pwroute53mgmt",      "Effect": "Allow",      "Action": [
"route53:ChangeResourceRecordSets",        "route53:ChangeTagsForResource",
"route53:CreateHostedZone",        "route53:DeleteHostedZone",        "route53:GetChange",
"route53:GetDNSSEC",        "route53:GetHostedZone",        "route53:ListHostedZones",
"route53:ListResourceRecordSets",        "route53:ListTagsForResource"      ],      "Resource":
"*"    }  ]}
```

pw-sts-mgmt

This policy allows ACTIVATE to create federated users used for short term tokens.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "pwstsmgmt",
      "Effect": "Allow",
      "Action": ["sts:GetCallerIdentity", "sts:GetFederationToken"],
      "Resource": "*"
    }
  ]
}
```

pw-tgw-mgmt

If you're using a Transit Gateway account, these additional permissions are necessary to attach created VPCs to the Transit Gateway.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "pwfsxmgmt",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTransitGatewayVpcAttachment",
        "ec2:DescribeTransitGatewayVpcAttachments",
        "ec2:DescribeTransitGateways",
        "ec2:DeleteTransitGatewayVpcAttachment"
      ],
      "Resource": "*"
    }
  ]
}
```

pw-buckets

This policy allows ACTIVATE to access billing information and Create and Mange S3 Buckets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [        "cur:*",        "s3:CreateBucket",        "s3:DeleteBucket",
"s3:DeleteBucketPolicy",        "s3:DeleteObject",        "s3:DeleteObjectVersion",
"s3:GetAccelerateConfiguration",        "s3:GetBucketAcl",        "s3:GetBucketCORS",
"s3:GetBucketLocation",        "s3:GetBucketLogging",
"s3:GetBucketObjectLockConfiguration",        "s3:GetBucketPolicy",
"s3:GetBucketPolicyStatus",        "s3:GetBucketRequestPayment",        "s3:GetBucketTagging",
"s3:GetBucketVersioning",        "s3:GetBucketWebsite",        "s3:GetEncryptionConfiguration",
"s3:GetLifecycleConfiguration",        "s3:GetObject",        "s3:GetObjectAcl",
"s3:GetObjectVersion",        "s3:GetReplicationConfiguration",        "s3:ListAllMyBuckets",
"s3:ListBucket",        "s3:ListBucketVersions"        "s3:ListBucketVersions",
"s3:PutAccelerateConfiguration",        "s3:PutBucketAcl",        "s3:PutBucketCORS",
"s3:PutBucketLogging",        "s3:PutBucketObjectLockConfiguration",        "s3:PutBucketPolicy",
"s3:PutBucketRequestPayment",        "s3:PutBucketTagging",        "s3:PutBucketVersioning",
"s3:PutBucketWebsite",        "s3:PutEncryptionConfiguration",
"s3:PutLifecycleConfiguration",        "s3:PutObject",        "s3:PutObjectAcl",
"s3:PutReplicationConfiguration"      ],      "Resource": "*"    }  ]}
```

pw-storage

This policy allows ACTIVATE to create and manage EFS storages.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "pwstoragemgmt",
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:CreateMountTarget",
        "elasticfilesystem:DeleteFileSystem",
        "elasticfilesystem:DeleteMountTarget",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeLifecycleConfiguration",
        "elasticfilesystem:DescribeMountTargetSecurityGroups",
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:TagResource"
      ],
      "Resource": "*"
    }
  ]
}
```

Azure

This page explains how to set up an Azure subscription so that ACTIVATE can manage your Azure infrastructure, clusters, billing, storage, and usage data.

> **INFO**
>
> The steps on this page should be completed by a **cloud engineer** in your organization.

Azure Subscription

We recommend creating a new Azure subscription for the ACTIVATE platform, which will allow you to keep your existing Azure subscription separate from the platform and make it easier to manage billing and usage data. This will also ensure the principle of least privilege, as ACTIVATE will only have access to the resources it needs to manage.

To learn more about Azure subscriptions, please see the Azure documentation.

Setting Up Azure Credentials

To get started quickly, you can create a new service principal and add the `Owner` role. This will allow ACTIVATE to manage all resources in your subscription. If you want to limit the scope of the service principal, you can create a custom role and assign it to the service principal.

Creating a Client Secret

ACTIVATE uses Azure client secrets to authenticate with Azure. You can create a new client secret by following the steps in the Azure documentation.

Azure Access Policies

pw-billing

Assign the following predefined roles to your application in order for ACTIVATE to provision billing infrastructure and access true cost data.

```
Owner
Storage Blob Data Owner
```

Google

This page explains how to set up a Google account so that ACTIVATE can manage your Google infrastructure, clusters, billing, storage, and usage data.

> **INFO**
>
> The steps on this page should be completed by a **cloud engineer** in your organization.

Google Account

We recommend creating a new Google project for the ACTIVATE platform, which will allow you to keep your existing Google project separate from the platform and make it easier to manage billing and usage data. This will also ensure the principle of least privilege, as ACTIVATE will only have access to the resources it needs to manage.

Ensure the following APIs are enabled for the project in the API & Services page:

- Cloud Resource Manager API
- IAM API

- Compute Engine API
- Filestore API

Setting Up Google Credentials

To get started quickly, you can create a new service account and add the `Owner` role. This will allow ACTIVATE to manage all resources in your project. If you want to limit the scope of the service account, you can create a custom role and assign it to the service account.

Creating a Service Account Key

ACTIVATE uses service account keys to authenticate with Google Cloud. You can create a new service account key by following the steps on the Google documentation.

Google Permissions

This section includes the permissions or roles you'll need to assign to the Google service account you create for ACTIVATE. You can create a custom role with all the necessary permissions in the IAM console.

pw-billing

These permissions are used to provision and access billing infrastructure. You can also assign the existing Google IAM roles `BigQuery User` and `Service Usage Admin` to your Google service account.

```
serviceusage.operations.get
serviceusage.services.disable
serviceusage.services.enable
serviceusage.services.get
serviceusage.services.list
monitoring.timeSeries.list
serviceusage.operations.cancel
serviceusage.operations.delete
serviceusage.operations.list
serviceusage.quotas.get
serviceusage.quotas.update
serviceusage.services.use
bigquery.datasets.create
bigquery.datasets.get
bigquery.jobs.create
bigquery.tables.list
resourcemanager.projects.get
bigquery.bireservations.get
bigquery.capacityCommitments.get
bigquery.capacityCommitments.list
bigquery.config.get
bigquery.datasets.getIamPolicy
bigquery.jobs.list
bigquery.models.list
bigquery.readsessions.create
bigquery.readsessions.getData
bigquery.readsessions.update
bigquery.reservationAssignments.list
bigquery.reservationAssignments.search
bigquery.reservations.get
```

```
bigquery.reservations.listbigquery.routines.listbigquery.savedqueries.getbigquery.savedqueries.lis
tbigquery.transfers.getbigquerymigration.translation.translateresourcemanager.projects.list
```

OpenStack

This page explains how to set up an OpenStack environment so that ACTIVATE can manage your OpenStack infrastructure, clusters, and usage data.

> **INFO**
>
> The steps on this page should be completed by a **cloud engineer** in your organization.

OpenStack Project

We recommend creating a dedicated OpenStack project for the ACTIVATE platform, which will allow you to keep your existing projects separate from the platform and make it easier to manage usage data. This will also ensure the principle of least privilege, as ACTIVATE will only have access to the resources it needs to manage.

Setting Up OpenStack Credentials

ACTIVATE authenticates with OpenStack using a username and password. You will need to provide the following information when adding your OpenStack cloud account in ACTIVATE:

Project Details

Field	Description
Project ID	The unique identifier of your OpenStack project.
Project Name	The name of your OpenStack project. Some OpenStack endpoints require a project ID and others require a project name, so both are needed.
Project Domain	The domain the project belongs to. Defaults to `default` if not specified.

You can find your Project ID and Project Name in the OpenStack dashboard under **Identity > Projects**, or by running `openstack project list` from the CLI.

Endpoint URLs

ACTIVATE requires the following OpenStack service endpoint URLs:

Field	Description	Example
Identity Endpoint	The Keystone identity service URL.	`http://example.openstack.com:5000/v3/`
Compute Endpoint	The Nova compute service URL.	`http://example.openstack.com:8774/v2.1/`
Network Endpoint	The Neutron network service URL.	`http://example.openstack.com:9696/`

You can find your endpoint URLs in the OpenStack dashboard under **Identity > Application Credentials**, or by running `openstack endpoint list` from the CLI.

User Credentials

Field	Description
Username	The OpenStack username for authentication.
Password	The password for the OpenStack user.
User Domain	The domain the user belongs to. Defaults to `default` if not specified.

We recommend creating a dedicated user for ACTIVATE with the appropriate roles assigned in your project. At a minimum, the user should have the `member` and `admin` roles on the project to allow ACTIVATE to manage compute, network, and storage resources.

> **AUTOMATIC METADATA SYNC**
>
> After you add your OpenStack cloud account, ACTIVATE will automatically sync metadata from your OpenStack environment, including available regions, instance flavors, and networks. This information is used when deploying infrastructure through ACTIVATE.

Oracle

This page explains how to set up an Oracle Cloud Infrastructure (OCI) account so that ACTIVATE can manage your Oracle Cloud infrastructure, clusters, and usage data.

> **FEATURE AVAILABILITY**
>
> Oracle Cloud support is currently feature-flagged and may not be available in all ACTIVATE deployments. Contact your ACTIVATE administrator to enable Oracle Cloud support.

> **INFO**
>
> The steps on this page should be completed by a **cloud engineer** in your organization.

Oracle Cloud Account

We recommend creating a dedicated compartment for the ACTIVATE platform, which will allow you to keep your existing resources separate from the platform and make it easier to manage usage data. You can create a compartment in the Oracle Cloud Console under **Identity > Compartments**.

Setting Up Oracle Cloud Credentials

ACTIVATE authenticates with Oracle Cloud using API key-based authentication. You will need to provide the following information when adding your Oracle Cloud account in ACTIVATE.

Tenancy and Compartment

Field	Description	Format

Tenancy OCID	The OCID of your Oracle Cloud tenancy. Found in the Oracle Cloud Console under **Administration > Tenancy Details**.	`ocid1.tenancy.oc1...<unique_ID>`
Compartment OCID	The OCID of the compartment to use for resources. Found under **Identity > Compartments**.	`ocid1.compartment.oc1...<unique_ID>`

API Key Credentials

Oracle Cloud uses API signing keys for authentication. To set up API key credentials:

1. Navigate to **Identity > Users** in the Oracle Cloud Console and select the user account that ACTIVATE will use.
2. Under **Resources**, select **API Keys**.
3. Click **Add API Key**.
4. Select **Generate API Key Pair** to create a new key pair, or **Paste a Public Key** if you already have one.
5. Download the private key file (PEM format) and store it securely.
6. After adding the key, Oracle Cloud will display a configuration file preview that contains the required values.

For more details, see the Oracle Cloud documentation on API signing keys.

You will need the following credentials:

Field	Description	Format
User OCID	The OCID of the user. Found under **Identity > Users**.	`ocid1.user.oc1...<unique_ID>`
API Key Fingerprint	The fingerprint of the API signing key. Generated when you add an API key to your user.	`aa:bb:cc:dd:ee:ff:00:11:22:33:44:55:66:77:88:99`
API Private Key (PEM)	The private key in PEM format used for API authentication. This is the private key you downloaded when creating the API key pair.	PEM-encoded RSA private key

Required Policies

Ensure the user has sufficient permissions to manage resources in the target compartment. At a minimum, you can assign the user to a group with the following policy:

```
Allow group <group_name> to manage all-resources in compartment <compartment_name>
```

For more restrictive access, create policies that grant permissions for compute, networking, and storage resources. See the Oracle Cloud policy reference for details.

Managing Cloud Accounts

This page explains how to add, view, update, and delete cloud service provider accounts in your ACTIVATE organization. Cloud accounts store the credentials that ACTIVATE uses to provision infrastructure and start clusters on your behalf.

Requires: `org:admin`

Adding a Cloud Account

1. Navigate to **Organization > Cloud Accounts**.
2. Click **Add Account**.
3. Enter a **Name** for the cloud account.
4. Select your **Cloud Service Provider** from the dropdown menu.
5. Fill in the provider-specific fields described below.
6. Click **+ Add Account**.

AWS

Field	Description
Account ID	A 12-digit number that uniquely identifies your AWS account. You can find this in the upper-right corner of the AWS Management Console.
GovCloud	Enable this toggle if the account belongs to an AWS GovCloud region.
AWS Access Key ID	The access key ID for an IAM user or role with the required permissions.
AWS Secret Access Key	The corresponding secret access key.

If you do not have an access key, see Managing access keys for IAM users in the AWS documentation.

Azure

Field	Description
Subscription ID	The ID of the Azure subscription where resources will be provisioned.
GovCloud	Enable this toggle if the subscription belongs to an Azure Government region.
Azure Client ID	The Application (client) ID of the service principal.
Azure Client Secret	A client secret generated for the service principal.
Azure Tenant ID	The Directory (tenant) ID of your Azure Active Directory.

If you do not have a service principal, see Create a service principal in the Azure documentation.

Google Cloud

Field	Description
Project ID	The Google Cloud project ID where resources will be provisioned.
GovCloud	Enable this toggle if the project uses Assured Workloads. When enabled, you must also provide a **CMEK Crypto Key ID**.
Google Service Account JSON Credentials	The full JSON key file for a service account with the required permissions.
Role IDs	A comma-separated list of role IDs granted to the service account (for example, `roles/owner,roles/compute.admin`).

If you do not have a service account key, see Create and delete service account keys in the Google Cloud documentation.

OpenStack

Project

Some OpenStack endpoints require a project ID and others require a project name, so both values are required.

Field	Description
Project ID	The ID of the project in OpenStack.
Project Name	The name of the project in OpenStack.
Project Domain	The domain the project belongs to. Defaults to `default`.

Endpoints

Field	Description
Identity Endpoint	The Keystone identity endpoint (for example, `http://example.openstack.com:5000/v3/`).
Compute Endpoint	The Nova compute endpoint (for example, `http://example.openstack.com:8774/v2.1/`).
Network Endpoint	The Neutron network endpoint (for example, `http://example.openstack.com:9696/`).

Credentials

Field	Description
Username	The OpenStack username.
Password	The corresponding password.
Domain	The domain the user belongs to. Defaults to `default`.

For more information, see Application Credentials in the OpenStack documentation.

Oracle Cloud

Oracle Cloud Infrastructure

Field	Description
Tenancy OCID	The OCID of your Oracle Cloud tenancy. Found in the Oracle Cloud Console under **Administration > Tenancy Details**.
Compartment OCID	The OCID of the compartment where resources will be provisioned. Found under **Identity > Compartments**.

API Key Credentials

Field	Description
User OCID	The OCID of the user. Found in the Oracle Cloud Console under **Identity > Users**.
API Key Fingerprint	The fingerprint of the API signing key, generated when you add an API key to your user.
API Private Key (PEM)	The private key in PEM format used for API authentication. This is the private key you downloaded when creating the API key.

After the account is created, ACTIVATE automatically validates that the provided credentials have sufficient permissions. You can check the validation status on the cloud account detail page.

Viewing Cloud Account Details

To view details for a cloud account, navigate to **Organization > Cloud Accounts** and click the account name.

The detail page displays the following information:

- **Principal** -- The primary credential identifier (for example, the AWS Access Key ID or the Google service account email).
- **Billing** -- Whether billing has been provisioned for this account, along with when billing data was last refreshed.
- **Permissions** -- Whether the credential has sufficient permissions. If permissions are insufficient, a link to the relevant preparation guide is provided.
- **Credentials last updated** -- When the credentials were last changed.
- **Shared with** -- Which groups or the entire organization have access to use this cloud account for network provisioning.
- **Networks** -- A table of all networks provisioned under this cloud account.

From the detail page you can also manage access permissions, provision or deprovision billing, and add networks.

Updating Credentials

To rotate or update the credentials on an existing cloud account:

1. Navigate to **Organization > Cloud Accounts** and click the account name.
2. Click **Update Credentials** in the action bar.
3. Update the credential fields as needed. The account name and cloud service provider cannot be changed.
4. Click **Update Credentials** to save.

After updating, ACTIVATE re-validates the credential permissions automatically.

Deleting a Cloud Account

> **NETWORK DEPROVISIONING**
>
> Deleting a cloud account will deprovision all networks associated with the account. Ensure that no active clusters are running on those networks before proceeding.

To delete a cloud account:

1. Navigate to **Organization > Cloud Accounts**.
2. Click the delete icon next to the account you want to remove.
3. Confirm the deletion in the dialog.

If billing has been provisioned for the account, deprovision billing before deleting the account.

Networks

In order to provision cloud clusters, you must first deploy a network into your cloud account. A network is a networking stack deployed into your cloud service provider (CSP) that clusters run on top of. Networks are managed per cloud account, and you can deploy multiple networks within a single account.

Below, you can find specific details about the implementation of networks for each CSP.

AWS Networks

Deploying a network provisions the following resources on AWS:

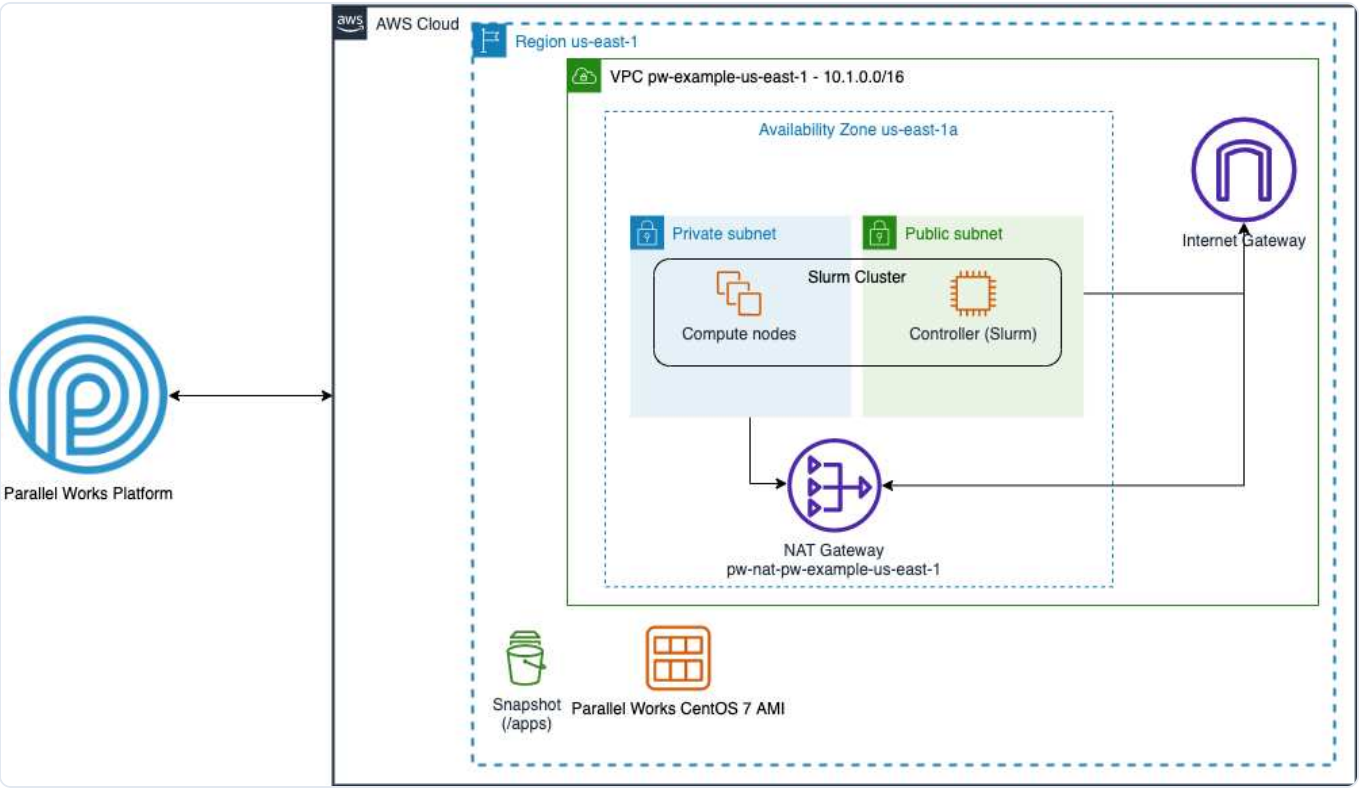
- a Virtual Private Cloud (VPC)
- an internet gateway
- public and private subnets
- public and private route tables and routes
- an Elastic IP address for the NAT gateway
- a Route 53 Hosted Zone
- a security group

The schematic below shows an ACTIVATE-deployed cluster in AWS Cloud.

The controller and compute nodes are shown in Availability Zone us-east-1a inside the us-east-1 Region. When using the default networking mode, compute nodes in a region use a single NAT gateway for outbound internet connectivity. This NAT gateway is pre-provisioned as part of the network. The AMI used for the controller and compute nodes is inside the Parallel Works AWS account.

EXPENSE INFORMATION

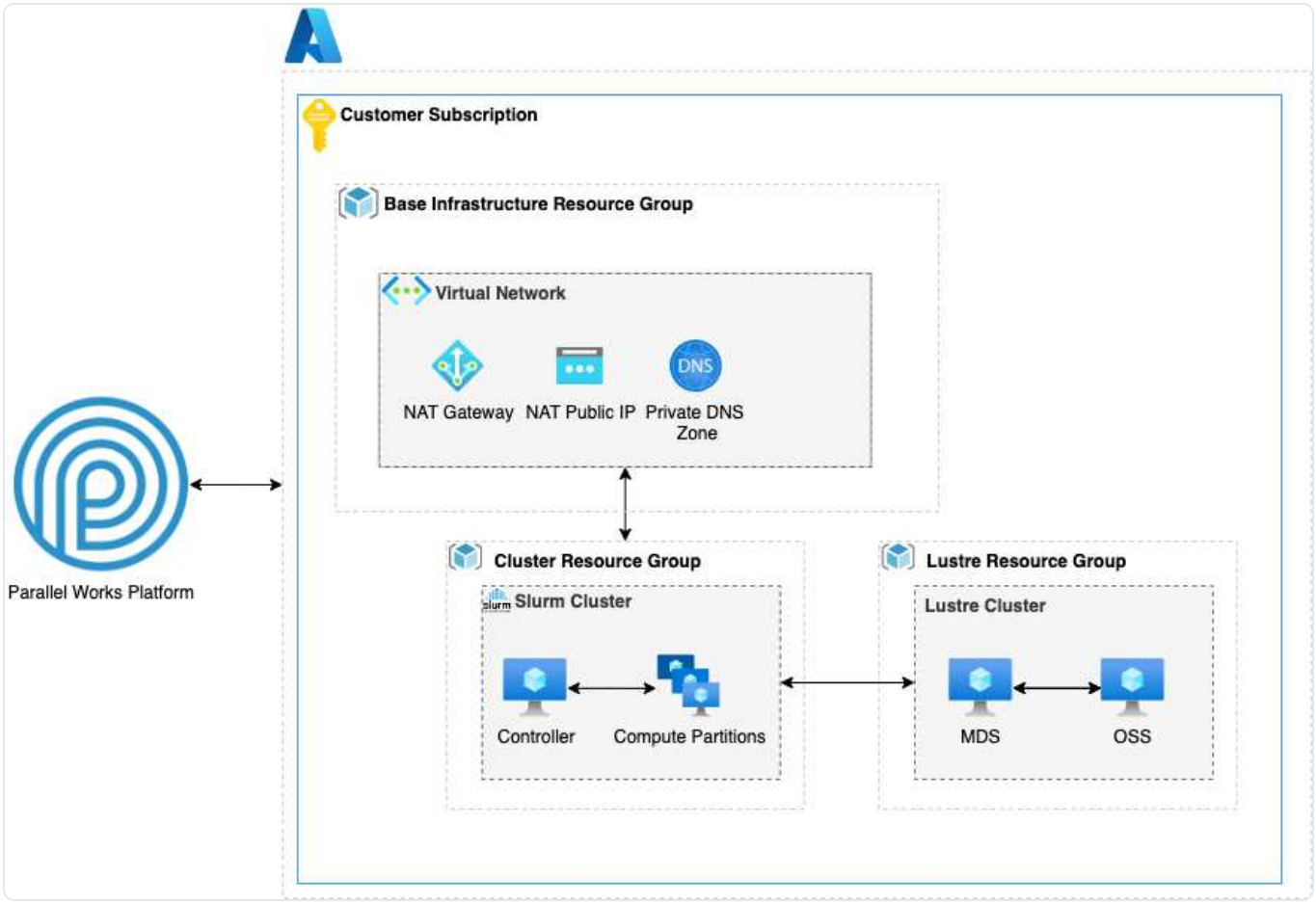
Depending on the networking mode, a NAT gateway will be created resulting in an additional monthly charge in your AWS account. For more information, please see this AWS pricing list.



A diagram of ACTIVATE clusters in AWS Cloud

Azure Networks

Deploying a network on Azure provisions a similar set of networking resources, including a virtual network, subnets, NAT gateway, and associated security rules.



A diagram of ACTIVATE clusters in Azure Cloud

Google Networks

Deploying a network on Google Cloud provisions a VPC network, subnets, firewall rules, and Cloud DNS resources in the specified project and region.

OpenStack Networks

For OpenStack environments, network configuration is specific to your OpenStack deployment. Networks are created using your provider's available networking resources, regions, and flavors.

Creating Networks

This page explains how to create a network in your cloud account, which will allow members of your organization to provision clusters.

Requires: `org:admin`

Steps

1. Navigate to **Cloud Accounts** in the sidebar.
2. Select the cloud account you want to add a network to.

3. Click **Add Network**.
4. Enter a **Name** for the network. The name must contain only lowercase letters and numbers.
5. If using an AWS cloud account, select a **Networking Mode**. For more information about networking modes, see Configuring Networks.
6. Configure the **Regions** and **CIDR blocks** for your network. Each region must have a unique CIDR block, and CIDR blocks cannot overlap.
7. Submit the form to begin provisioning. The network will start provisioning automatically, and you can track its progress on the network detail page.

> **ABOUT REGIONS**
>
> You can configure multiple regions for a single network. Each region requires its own CIDR block, and regions must be unique within a network.

Configuring Networks

This page explains how to configure networks according to CSP-specific parameters.

Requires: `org:admin`

AWS

AWS Configuration

Region

Select the **Region** that you want your network to be deployed in. For more information about Regions, see the AWS documentation.

CIDR Block

Enter a **CIDR block** for the network in the selected region. CIDR blocks must be valid and cannot overlap with other regions in the same network.

Networking Mode

ACTIVATE provides different networking modes for compute nodes on a cluster. The networking mode determines how compute nodes connect to the internet.

Networking Mode	Description	Tracks Egress Cost?	Notes
Private Subnets	This networking mode will provision a NAT gateway. The NAT gateway will be used for outbound internet connectivity for all compute nodes in the region.	No	The NAT Gateway will incur a charge for every hour it is running.
Controller as NAT Gateway	All compute nodes will route their outbound internet traffic through the controller node. Compute nodes will		

	have the same IP address as the controller node.	Yes	Partitions must be in the same availability zone as the controller.
Nodes With Public IPs	This networking mode will assign a public IP address to each compute node. The nodes' public IPs will not be accessible from the internet.	Yes	Each node will have its own IP address rather than having the same one as other nodes.
Private	This networking mode creates a fully private network with no direct internet access. Requires a Transit Gateway for connectivity when peered to the platform.	No	A Transit Gateway ID is required when peering to the platform.

Transit Gateway Peering (Private Mode)

When using the **Private** networking mode on AWS, you can peer the network to the ACTIVATE platform via a Transit Gateway. If you enable peering, you must provide a **Transit Gateway ID**.

Enable NAT Gateway (Private Mode)

When using the **Private** networking mode on AWS without peering to the platform, an **Enable NAT Gateway** toggle becomes available. When enabled, ACTIVATE provisions a NAT gateway that allows resources in the private network to make outbound internet connections (for example, downloading packages or accessing external APIs) while remaining unreachable from the internet.

This option is not available when using Transit Gateway peering, as traffic is expected to route through a central networking account.

> **NAT GATEWAY COSTS**
>
> The NAT gateway incurs a charge for every hour it is running. For more information, see the AWS documentation on NAT gateway pricing.

Azure

Azure Configuration

Region

Select the **Region** that you want your network to be deployed in. For more information about Regions, see the Azure documentation.

VM Image Resource Group ID

Enter the Resource ID of the resource group you want this network to be associated with. If you don't have a resource group, you can create one. You can find the Resource ID in the Azure portal by navigating to **Home** > **Pay-As-You-Go | Resource groups** > **GroupName** > **Properties** > **Resource ID**.

VM Image Resource Group Name

Enter the name of the resource group you want this network to be associated with.

NAT IP Availability Zone

Enter the Availability Zone that you want your network's IP address to be deployed in. We suggest entering `Zone-Redundant` in this field. For more information about Availability Zones, see the Azure documentation.

Google

Google Configuration

Project ID

Enter the **Project ID** of the Google Cloud project that you want this network to be deployed in. If you don't have a project, please see Preparing Google.

Region

Select the **Region** that you want your network to be deployed in. For more information about Regions, see the Google documentation.

Availability Zone

Select the **Availability Zone** for the network within the chosen region.

OpenStack

OpenStack Configuration

Network configuration for OpenStack is specific to your deployment. Available regions, networks, and flavors are synced from your OpenStack provider. Contact your OpenStack administrator for details about available networking resources.

Managing Networks

This page explains how to view, edit, and delete networks associated with your cloud accounts.

Requires: `org:admin`

Viewing Networks

To view the networks for a cloud account, navigate to **Cloud Accounts** and select the account. The networks associated with that account will be listed on the account detail page.

Click on a network to view its detail page, which includes:

- **Provision status** tracking for the network's cloud resources (e.g., VPCs, subnets, security groups, DNS zones)
- **Region and CIDR configuration** for the network
- **Cloud resource details** provisioned by the network

Provision Status

Each network tracks its provisioning progress through a series of status indicators. A network can be in one of the following states:

Status	Description
Pending	The provisioning step has not yet started.
Provisioning	The step is currently being provisioned.
Provisioned	The step has completed successfully.
Failed	The step encountered an error during provisioning.
Deleting	The resource is being deprovisioned.

Editing Networks

You can modify a network's regions and CIDR blocks after it has been created. To edit a network:

1. Navigate to the network detail page.
2. Update the **Regions** and **CIDR blocks** as needed.
3. Save your changes. The network will be re-provisioned with the updated configuration.

> **CIDR VALIDATION**
>
> When editing a network, CIDR blocks must remain valid and cannot overlap with other regions in the same network.

Deleting Networks

To delete a network:

1. Navigate to the network detail page.
2. Click **Delete Network**.

> **ABOUT DELETION**
>
> When you delete a network, the infrastructure will be deprovisioned from your CSP account. This process removes all cloud resources associated with the network, including VPCs, subnets, and security groups. Ensure that no clusters are running on the network before deleting it.

Troubleshooting

After initiating network provisioning, you won't be able to edit the configuration while the network is being built. If the provisioning process encounters errors, you can delete the network and create a new one.

If errors persist, please contact support.

Billing

Cloud service providers (CSPs) provide resource usage and cost reports at set intervals. ACTIVATE can provision the necessary infrastructure to enable CSP billing exports, allowing the platform to access, process, and display cost data on the cost dashboard.

Billing is managed directly from the cloud account detail page. Navigate to **Cloud Accounts**, select an account, and use the billing options available on that page.

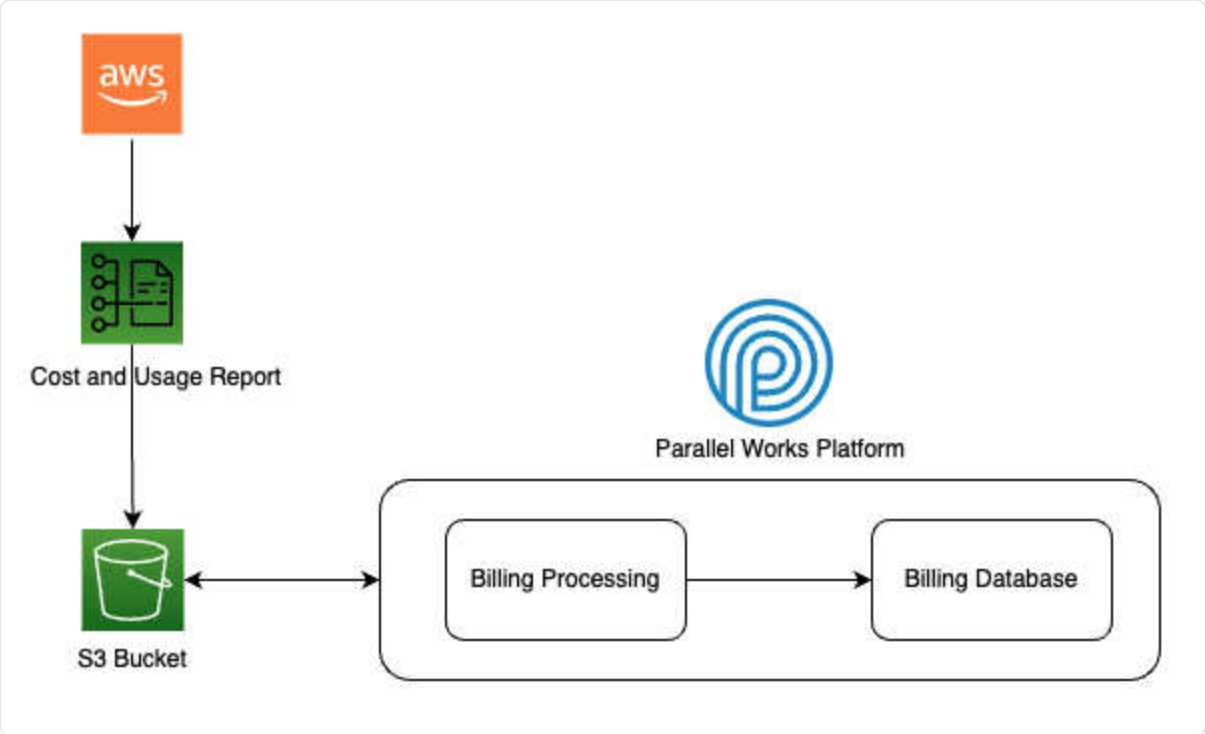
> **ONE-TIME SETUP PER CSP**
>
> Billing only needs to be provisioned **once per CSP per organization**. After setting up billing on one cloud account, all other cloud accounts using the same CSP automatically detect billing data within an hour. You do not need to provision billing separately for each cloud account.

AWS Billing Infrastructure

Provisioning billing infrastructure on AWS creates the following resources:

- **Cost Usage Report**: exports AWS cost data periodically
- **S3 Bucket**: stores exported AWS cost data

The diagram below shows how ACTIVATE communicates with AWS after billing infrastructure is provisioned.

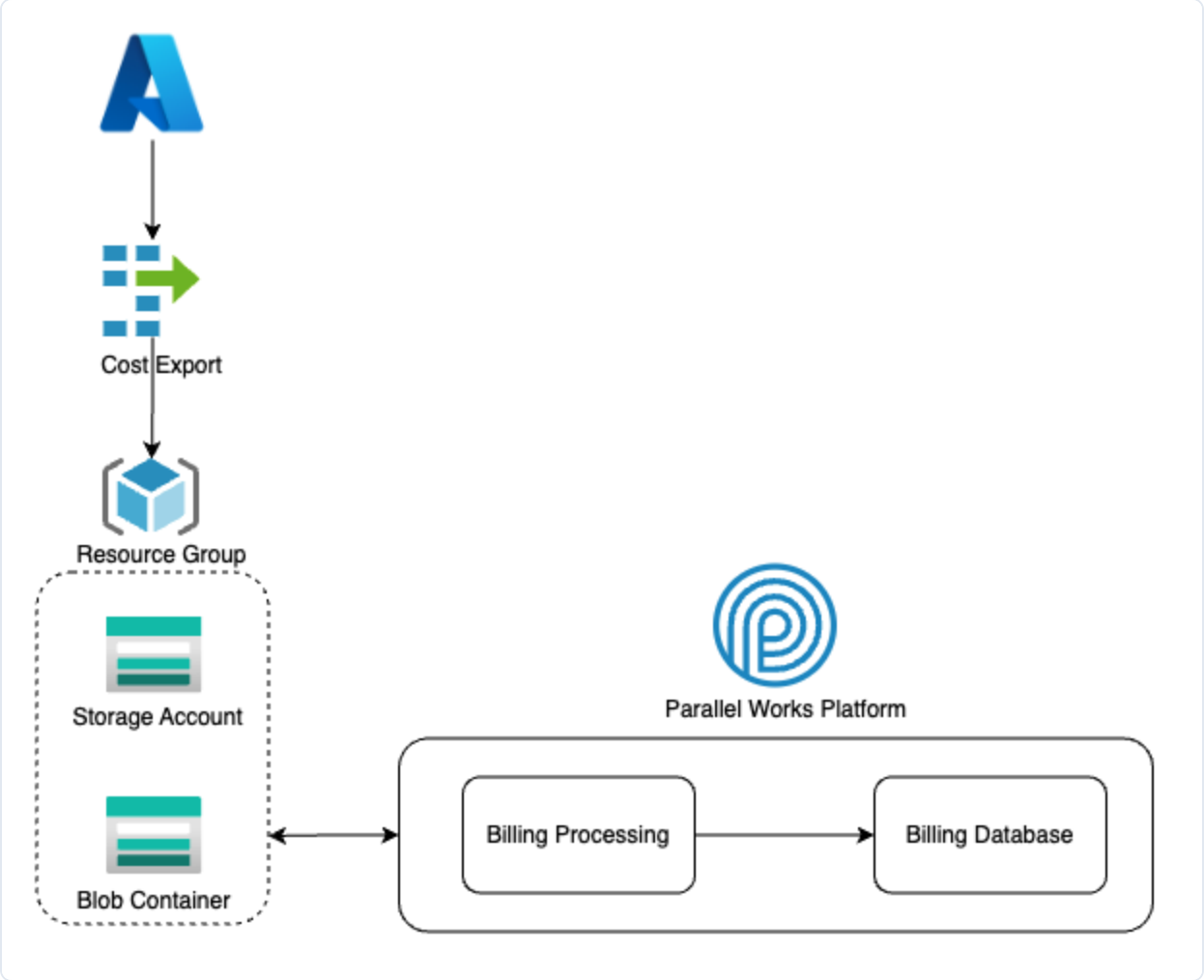


Azure Billing Infrastructure

Provisioning billing infrastructure on Azure creates the following resources:

- **Cost Export**: exports Azure cost data daily
- **Billing Resource Group**: groups the storage account and container
- **Storage Account**: an account to contain the blob storage container
- **Blob Container**: stores exported Azure cost data
- **Cost Management Export Provider**: registers the necessary provider for cost export (not included in the diagram)

The diagram below shows how ACTIVATE communicates with Azure after billing infrastructure is provisioned.

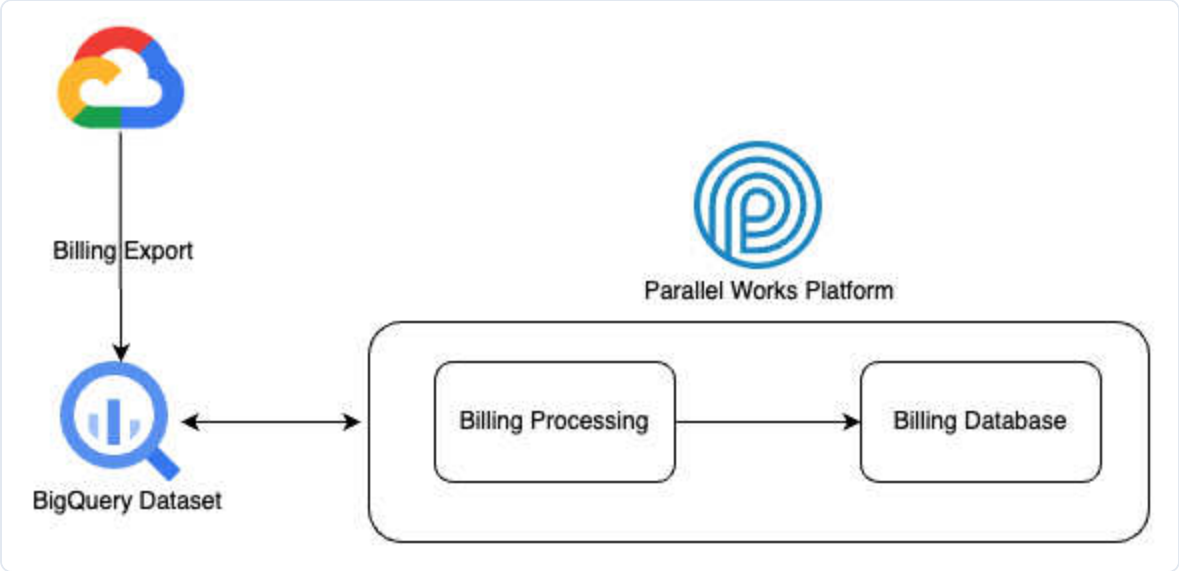


Google Billing Infrastructure

Provisioning billing infrastructure on Google creates the following resources:

- **BigQuery Dataset**: stores exported Google cost data
- **BigQuery Data Transfer API**: enables the necessary API to export data to BigQuery (not included in the diagram)

The diagram below shows how ACTIVATE communicates with Google after billing infrastructure is provisioned.



Need Help?

If you need assistance with billing setup or have questions about billing infrastructure, please contact support.

Provisioning Billing

This page explains how to provision billing infrastructure for a cloud account, enabling ACTIVATE to access, process, and display cost data on the cost dashboard.

Requires: `org:admin`

> **ONE-TIME SETUP**
>
> Billing only needs to be provisioned **once per CSP per organization**. After provisioning billing on one cloud account, all other cloud accounts using the same CSP automatically detect billing data within an hour.

Provisioning Billing

1. Navigate to **Cloud Accounts**.
2. Select the cloud account you want to provision billing for.
3. Click **Provision Billing**.

The provisioning process and required fields vary by CSP.

AWS

Select the **Region** you want your billing infrastructure to be deployed in. For more information about regions, see the AWS documentation.

AWS Cost Allocation Tags

ACTIVATE manages billing data by attaching tags to provisioned cloud resources. To enable the necessary tags in your AWS billing data, you need to activate custom cost allocation tags in your AWS account.

See the AWS documentation for the required permissions to complete the steps below.

1. Log in to the **AWS Management Console**.
2. Navigate to the **Billing** page.
3. Click **Cost allocation tags**.
4. In the tab for **User-defined allocation tags**, activate the following tags:
   - `Date`
   - `GroupID`
   - `Name`
   - `OrganizationID`
   - `Pool`
   - `Project`
   - `Session`
   - `SessionID`
   - `User`

Azure

Select the **Region** you want your billing infrastructure to be deployed in. For more information about regions, see the Azure documentation.

Toggle the **Cost Management Export** option to **Yes** if you have not yet registered the `Microsoft.CostManagementExports` provider in your Azure subscription.

If you toggle the option to **Yes** but have already registered the `Microsoft.CostManagementExports` provider, the provisioner will fail because Azure denies duplicate registration attempts.

For more details about providers, see the Azure documentation.

Google

Select the **Region** you want your billing infrastructure to be deployed in. For more information about regions, see the Google documentation.

In the **Project** field, enter the Project ID of the project where the billing infrastructure will be deployed. Your Google credential will need certain permissions enabled in this project. For more information about required permissions, see Preparing Google.

Manual BigQuery Export Setup

Google does not have an automated option to export billing data to the provisioned BigQuery dataset. Please contact support for assistance with completing this configuration.

Deprovisioning Billing

To remove billing infrastructure from a cloud account:

1. Navigate to **Cloud Accounts**.
2. Select the cloud account with provisioned billing.
3. Click **Deprovision Billing**.

> **WARNING**
> Deprovisioning billing removes the billing infrastructure from your CSP account. If this is the only cloud account with provisioned billing for this CSP, all cloud accounts using that CSP will lose access to billing data.

Billing Status

Each cloud account displays a billing status that indicates the current state of billing data availability:

- **Checking**: ACTIVATE is currently checking for billing data.
- **Provisioned**: Billing data is available for this cloud account. This status is set automatically when ACTIVATE detects billing data from the CSP, which is checked hourly.
- **Unprovisioned**: No billing data is available for this cloud account.

Troubleshooting

If the provisioning process encounters errors, you can retry by first deprovisioning the infrastructure and then provisioning again.

If errors persist after retrying, please contact support.

Access Control

Cloud account access control determines which groups in your organization can use a cloud account to provision networks and start clusters. By default, a newly created cloud account is not shared with any group.

Requires: `org:admin`

How Access Control Works

Access is managed through a single permission type called **network**. When a group is granted the network permission for a cloud account, members of that group can create and manage networks under that account and use those networks when starting clusters.

There are two ways to share a cloud account:

- **Organization-wide sharing** -- Share the cloud account with every group in the organization. When this option is enabled, all groups automatically receive the network permission and the individual group checkboxes are locked.
- **Group-based access** -- Restrict access to specific groups. Only the selected groups will be able to provision networks and use the cloud account for clusters.

> **RESOURCE TYPE ACCESS**

> As of v7.0.0, available resource types are automatically determined by cloud account access. If a group has the network permission on at least one cloud account for a given cloud service provider, that CSP's resource types are available to the group.

Managing Access

To manage which groups can use a cloud account:

1. Navigate to **Organization > Cloud Accounts** and click the cloud account name to open the detail page.
2. Click **Manage access** in the action bar. This opens the **Group Permissions Manager** panel.
3. In the permissions panel you will see a table listing every group in your organization along with an **Organization** row at the top.

Sharing with the entire organization

To grant access to all groups at once, check the **network** checkbox in the **Organization** row. When organization-wide access is enabled, all individual group checkboxes are automatically checked and disabled because access is inherited from the organization setting.

Sharing with specific groups

To grant access to individual groups, leave the Organization row unchecked and check the **network** checkbox next to each group that should have access.

Saving changes

After adjusting the checkboxes, click **Save Access** to apply the changes. A confirmation message will appear once the update is saved.

Viewing Current Access

You can see the current sharing status of a cloud account on its detail page under the **Shared with** field in the Account Details section. This field shows one of the following:

- **Entire Organization** -- The cloud account is shared with all groups.
- **N groups** -- The cloud account is shared with a specific number of groups. Click the link to open the permissions panel and see which groups have access.
- **Not shared** -- No groups have access to the cloud account.

Instance Types

This page explains how to manage the allowed instance types for cloud clusters at the organization level. Restricting instance types is useful when your users only need certain machine sizes for their work, or when you want to control costs by limiting access to expensive instance types.

Requires: `org:admin`

Configuring Allowed Instance Types

1. Navigate to **Organization > Cloud Accounts**.
2. Click the **Instance Types** button in the action bar.

3. The instance types page displays options for each cloud service provider. By default, all instance types are available.
4. To restrict the available instance types, deselect the types you do not want users to access. Only the selected instance types will appear when users configure cloud clusters.
5. Click **Save** to apply your changes. The new settings take effect immediately.

Instance type restrictions apply across the entire organization. Any user creating a cloud cluster will only be able to select from the allowed instance types.

OpenStack Flavor Pricing

For OpenStack cloud accounts, ACTIVATE syncs the available flavors (instance types) from your OpenStack environment. Administrators can view and edit the cost per hour for each flavor. This pricing information is used for cost tracking and reporting within ACTIVATE.

To manage OpenStack flavor pricing:

1. Navigate to **Organization > Cloud Accounts** and click the name of an OpenStack cloud account.
2. On the cloud account detail page, scroll down to the **OpenStack Flavors** section. This table lists all flavors synced from your OpenStack environment, including their vCPUs, memory, disk size, and current cost per hour.
3. To update pricing, edit the **Cost/hr** value for any flavor.
4. Click **Save Costs** to apply the changes.

> **STALE FLAVORS**
>
> OpenStack flavors are periodically synced from your OpenStack environment. If a flavor appears as stale, the flavor may have been removed or modified in OpenStack since the last sync.

Organization Bootstrap Script

Administrators can configure a bootstrap script that runs on all cloud clusters before any user bootstrap or health check scripts execute. This script is accessible from the **Bootstrap Script** button on the Cloud Accounts page.

For more information, see Organization Bootstrap Script.

Billing

Cost Dashboard

The Cost Dashboard provides an interactive view of your organization's cloud spending. Use it to monitor costs across groups, time periods, and cloud providers.

Requires: `org:admin`

Navigation

From the **Organizations** list, select your organization. In the sidebar, under **Billing**, click **Cost**.

Filters

The dashboard includes the following filter controls at the top of the page:

- **Time Period** — Select the time range for cost data (e.g., "Month to Date", "Last 30 Days", "Last 90 Days", etc.).
- **Group** — Filter cost data by a specific group within the organization, or view all groups.
- **Realtime** — Toggle between realtime (estimated) data and true cost data from your cloud service providers. Realtime data is available immediately but may differ slightly from final CSP billing. True cost data may be delayed by up to 24-48 hours depending on the provider.

Dashboard Modules

The dashboard displays cost data through visualization modules showing spending breakdowns by group, time period, and cloud provider.

Understanding Cost Data

- **Realtime data** is estimated based on active compute usage and is available immediately. It provides a close approximation of actual costs but may not account for all billing adjustments from your CSP.
- **True cost data** is imported from your cloud service provider's billing exports and represents the actual billed amounts. This data is typically delayed by 24-48 hours.

> **NOTE**
>
> The Cost Dashboard requires that billing infrastructure has been configured for at least one cloud account. See Cloud Account Billing for setup instructions.

Allocation Thresholds

This page explains how to set and monitor allocations for your organization's groups. When a group reaches its allocation total, its users will no longer be able to start clusters.

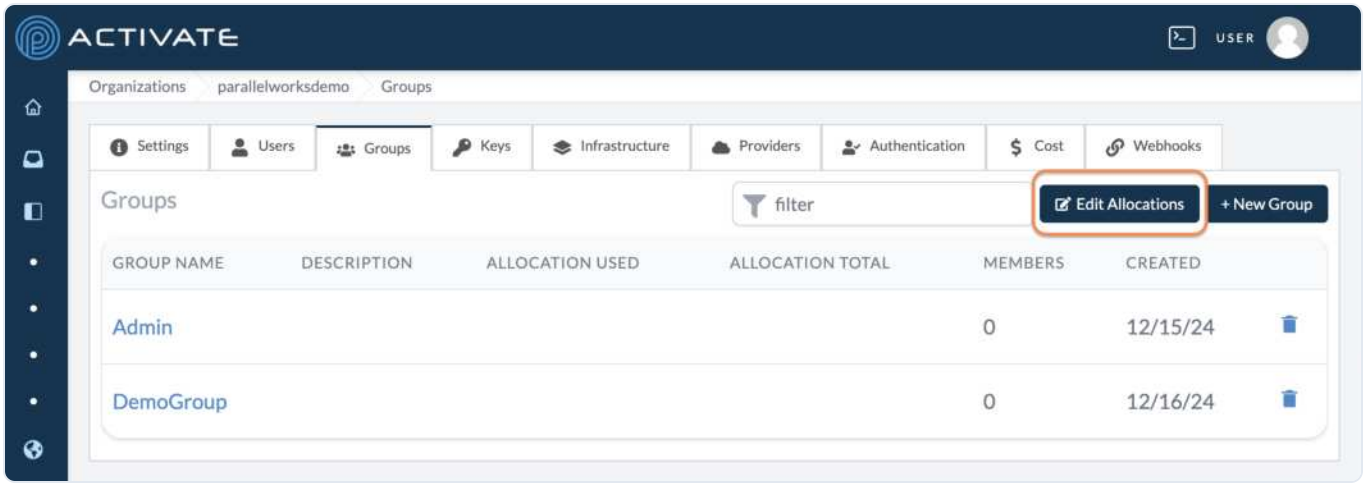
You can also set custom allocation thresholds for your organization. This feature can alert users when their group approaches its allocation total as well as freeze or shut down clusters when an allocation total is reached.

> **NOTE**
>
> The steps included on this page can only be completed by **ACTIVATE platform administrators** and users with the `org:admin` role. For more information, please see **About Group Roles**.

Setting Allocations

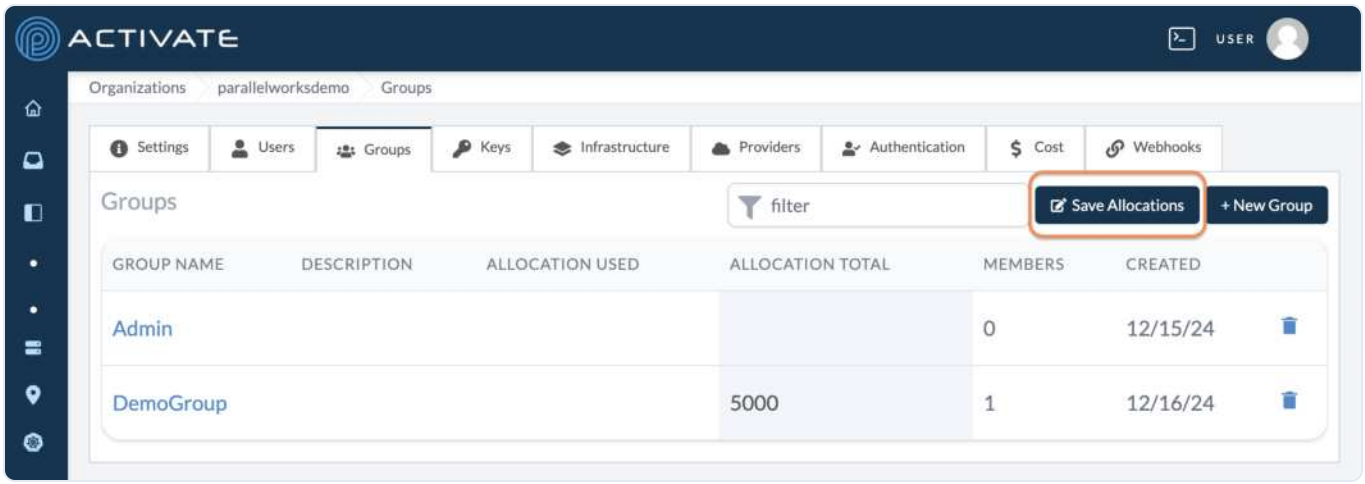
From the **Organizations** list, select your organization. In the sidebar, under **Managing Groups**, click **Groups**.

In the **Groups** tab, click **Edit Allocations**.

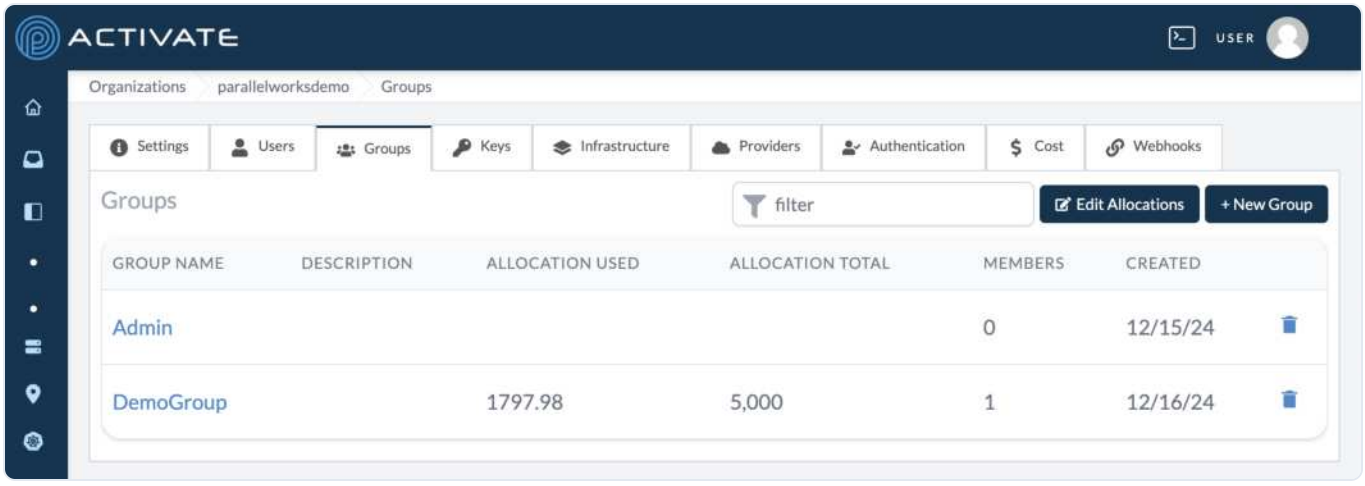


Enter a value in the **Allocation Total** field. This number is measured in U.S. dollars.

Click **Save Allocations**.



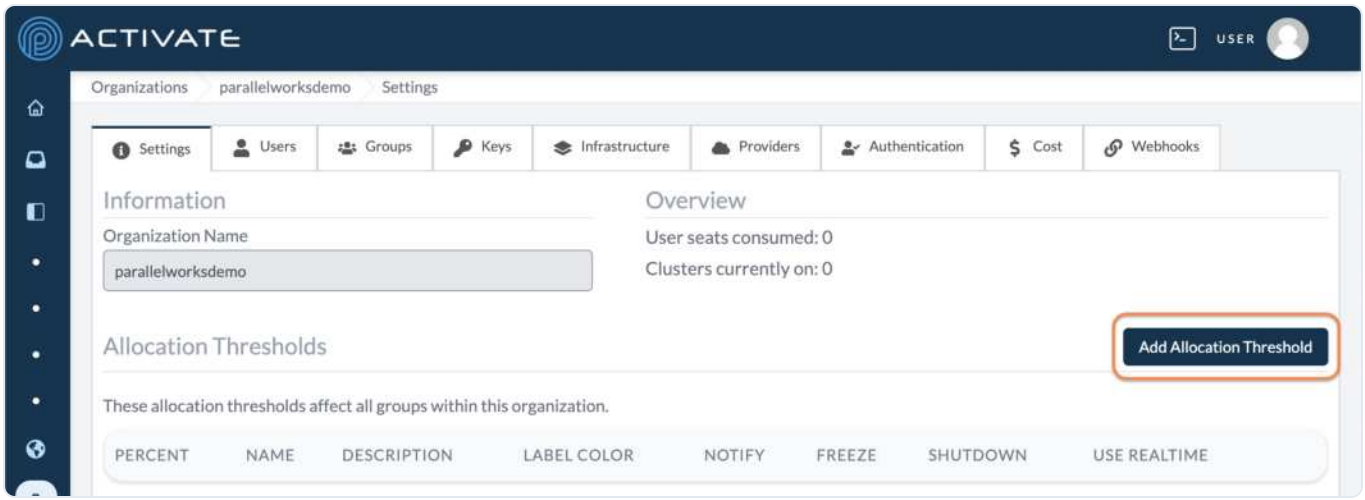
Once you've set an allocation total, a group's accumulated costs will appear in the **Allocation Used** column.



Setting Allocation Thresholds

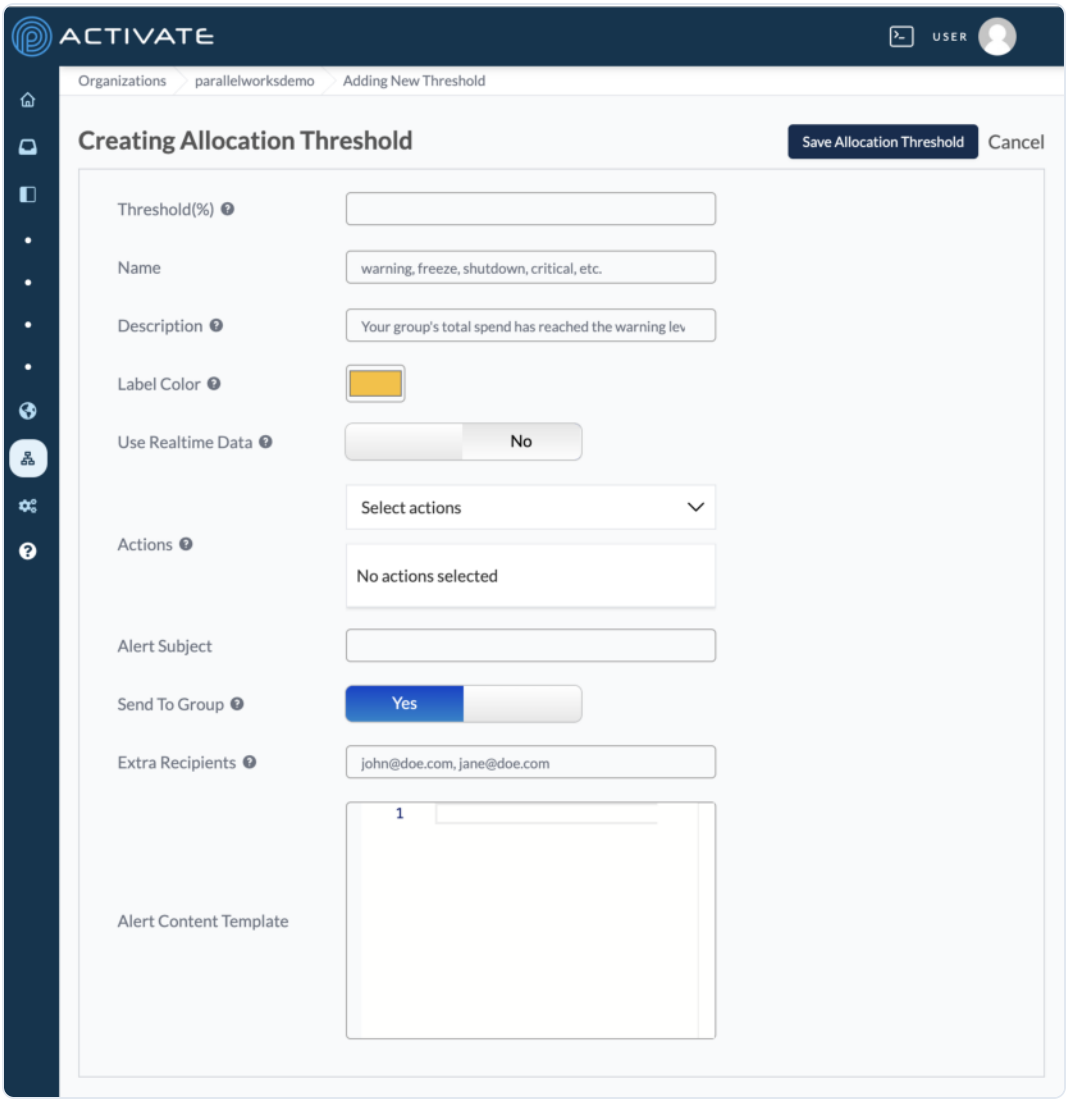
From the **Organizations** list, select your organization. In the sidebar, under **Billing**, click **Thresholds**.

Click **Add Allocation Threshold**.



Configure Allocation Thresholds

You can configure your allocation threshold with the settings outlined below.



Threshold(%)

Use this field to set the allocation threshold as a percentage from `0` to `100`. For example, if a group's allocation total is $5,000, an `85%` threshold will affect users when their total spend reaches $4,250.

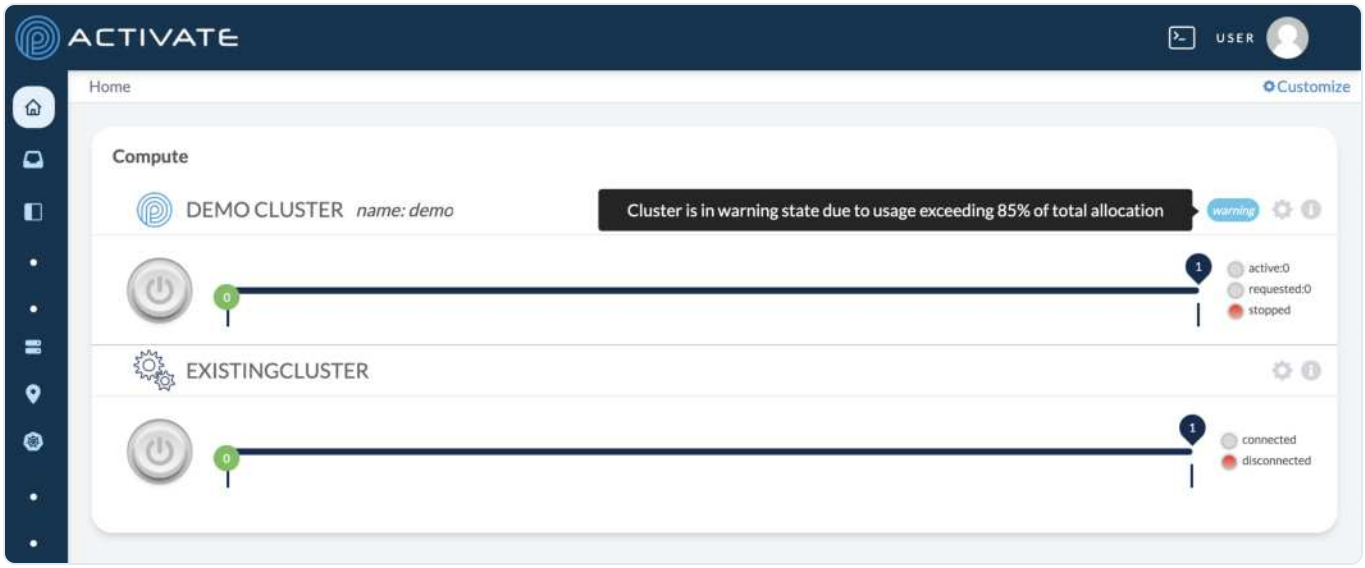
Name

Use this field to name your threshold. Common names for thresholds include `warning` and `shutdown`.

Please note that this field must be one word, lower-cased, and contain no numbers or special characters.

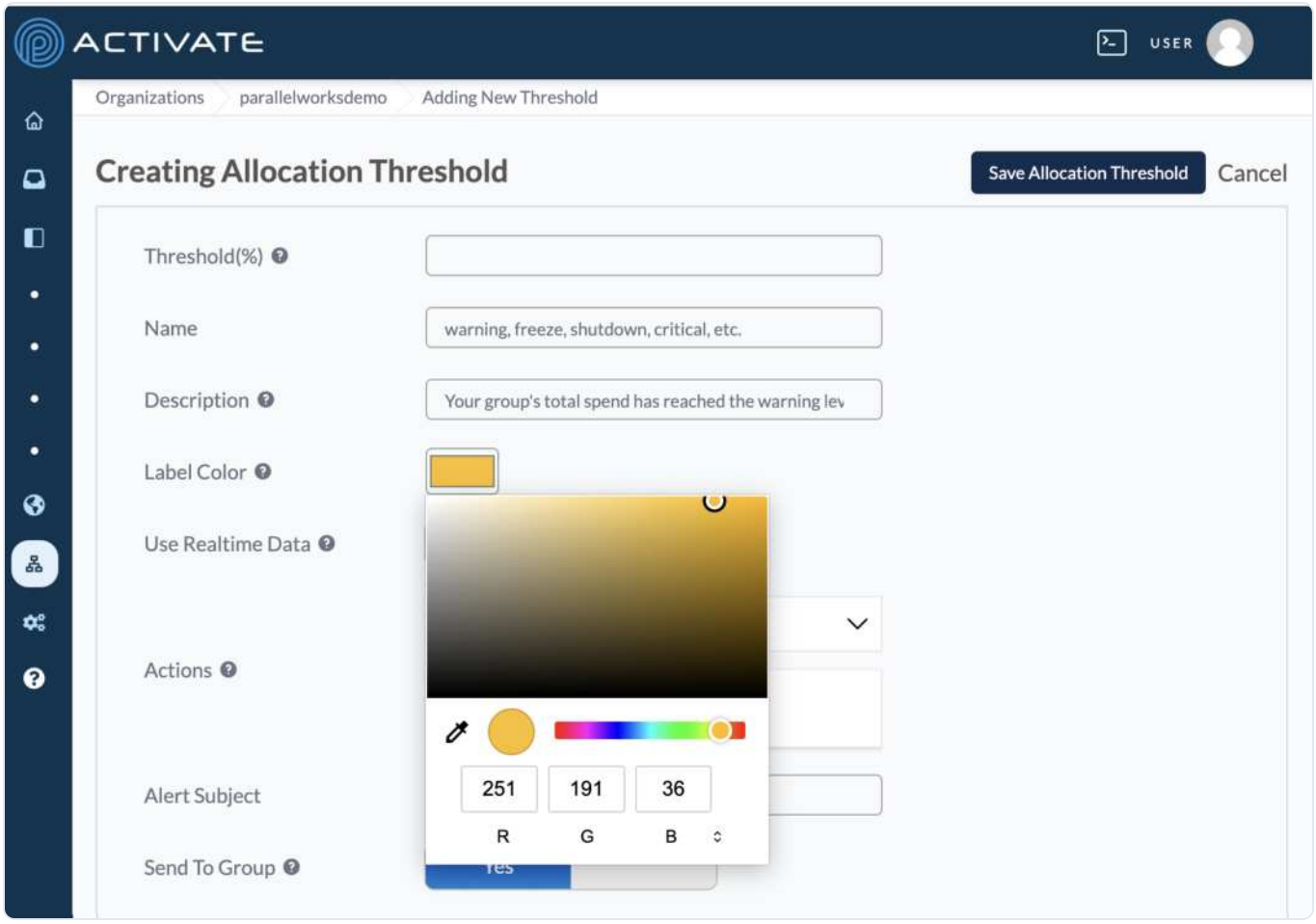
Description

Use this field to enter a description for the alert icon that appears on clusters. When users mouse over the icon, this **Description** appears as a tool tip.



Label Color

Use this selection tool to choose the color of the alert icon that appears on clusters. Click on the colored box to expand it into a color dropper.



Click the arrows to cycle the color fields between RGB, HSL, and Hex values.

Use Realtime Data

Use this toggle button to enable real-time cost tracking for your threshold.

By default, this option is set to `No`. If you don't enable real-time cost tracking, the cost tracking will be calculated using the true cost data from CSPs.

Actions

Use this dropdown menu to select the action for your threshold. Options include:

- **Notify**: Users receive an alert via email.
- **Freeze**: Users will not be able to start new clusters.
- **Shutdown**: All clusters will be shut down.

You can select more than one action at a time for your threshold alert, especially if you want users to receive a notification for `freeze` or `shutdown`. For more info, please see **Alert Content Template** below.

Alert Subject

Use this field to enter a subject for your alert email.

Send To Group

Use this toggle button to send the alert email to all users in a group whenever it reaches its allocation threshold.

By default, this option is set to `Yes`.

Extra Recipients

Use this field to enter email addresses for additional users who will receive the threshold alert email.

Alert Content Template

Use this field to enter the text of the threshold alert email.

Each action has a default email template that populates this field. You can change the text or replace it completely.

If you write your own alert template, you can use the variables below in the body of your message. The variables will automatically be filled in with the correct information.

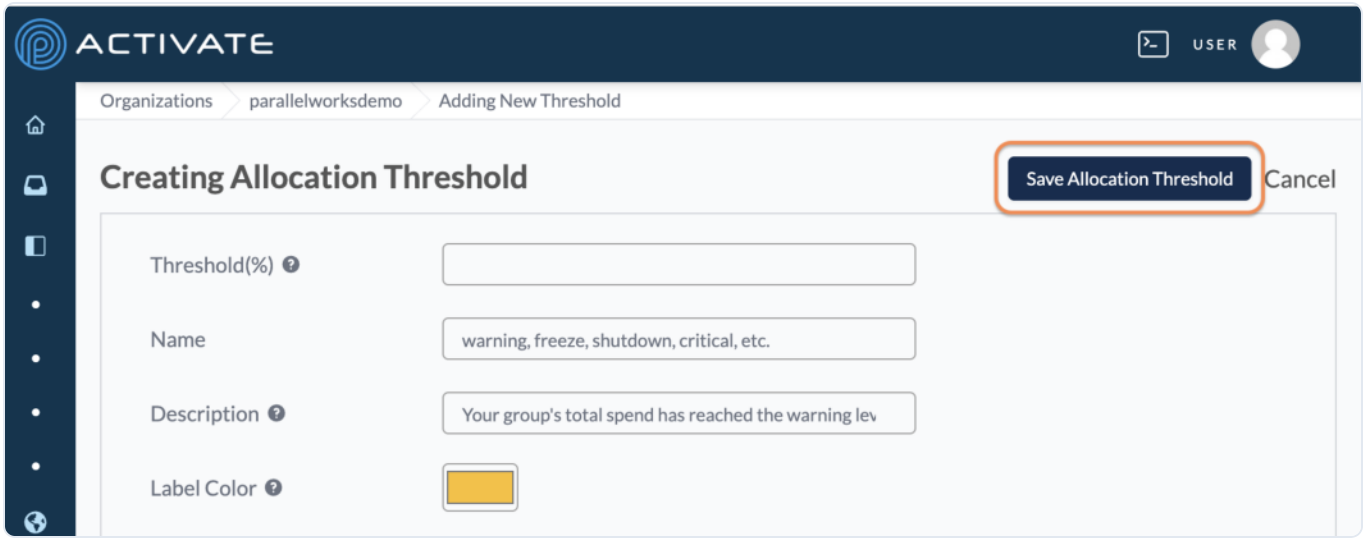
- `{allocation}`: a group's total budget allocation
- `{allocation_used}`: a group's current total spend
- `{allocation_used_percent}`: the percentage of allocation a group has used
- `{estimated_allocation_used}`: a group's current estimated (real-time) total spend
- `{estimated_allocation_used_percent}`: the estimated (real-time) percentage of allocation a group has used
- `{group}`: a group's name
- `{name}`: the threshold's label you entered in the **Name** field

- `{organization}` : your organization's name
- `{threshold}` : the value you entered in the **Threshold(%)** field

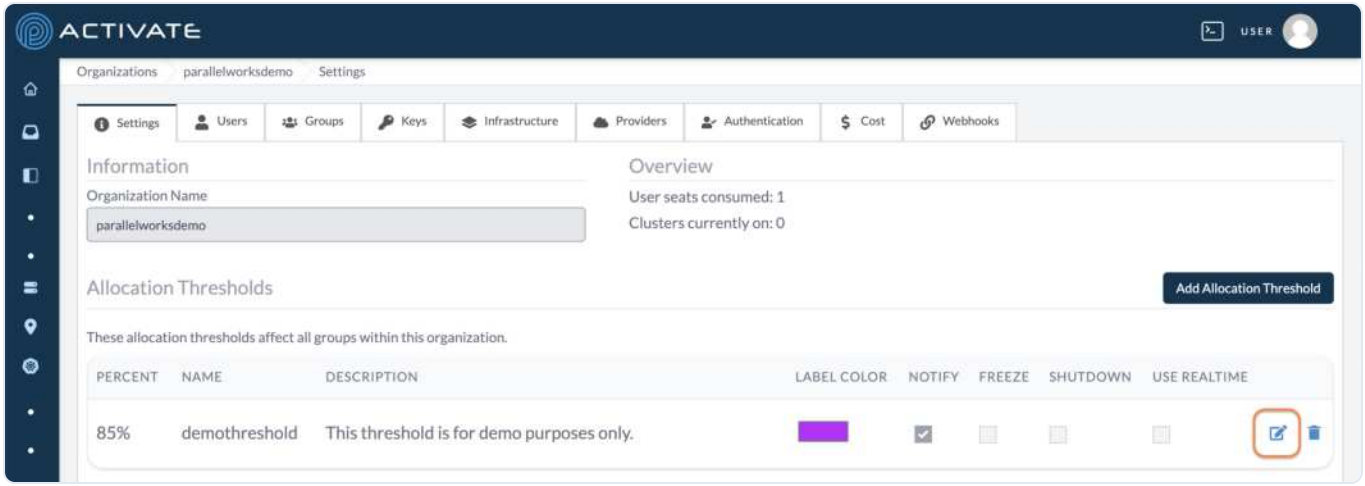
> **ABOUT ALERTS**
>
> If you want users to be notified about a threshold for the `freeze` or `shutdown` actions, you have to add the `notify` action too.

Save Allocation Thresholds

After you've customized the settings above, click **Save Allocation Threshold**.

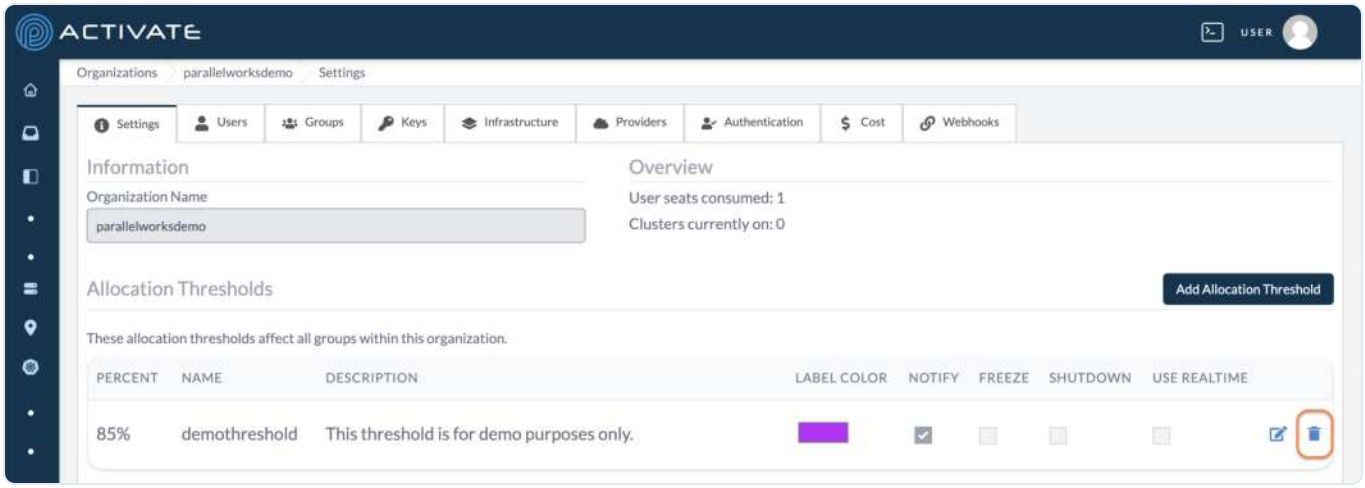


Your new allocation threshold will appear in your organization's settings.



Editing Allocation Thresholds

In your organization's settings, click the edit icon to change an allocation threshold.



Next, you'll see the threshold's configuration page, where you can edit any of the settings in **Threshold Configuration** above.

Click **Save Allocation Threshold** when you're done making changes.

Allocations (Preview)

Allocations are named budgets — denominated in U.S. dollars or in a custom unit such as core-hours, GPU-hours, or a pool of service credits — that let organization admins track and cap usage independently of groups. Custom units are the typical use case: awarded HPC compute, grant-funded resources, and other non-dollar accounting that the Cost Dashboard alone cannot represent. You can share an allocation with one or more groups, nest it under a parent allocation, and feed it usage from the platform or from external sources like HPC clusters.

Requires: `org:admin`

> **FEATURE PREVIEW**
>
> Allocations is currently in **Feature Preview**. To use it, enable **both** of the following features for your own account:
>
> - `allocations` — exposes the Allocations UI and APIs.
> - `cost` — enables the updated cost dashboard that allocation usage feeds into.
>
> Click your username, select **Feature Preview**, then click **Enable** next to each feature. See Feature Preview for the full walkthrough. Until both are enabled on your account, the **Allocations** page will not appear in your sidebar.

Concepts

Allocations

An allocation is a named budget owned by your organization. Each allocation tracks three values:

- **Total** — the budgeted amount.

- **Used** — confirmed spend recorded against the allocation.
- **Estimated used** — pending or in-flight spend that has not yet been finalized.

Allocations can optionally be nested one level under a parent allocation, which is useful when you want to carve a department-level budget into project-level sub-budgets.

Units

A **unit** is the dimension an allocation is measured in. By default, allocations are denominated in U.S. dollars. Organizations can also define custom units — for example, `core-hours` or `gpu-hours` — and attach time-varying rate rules so that pricing can change over time without rewriting historical usage.

SKUs

A **SKU** (Stock Keeping Unit) is a billable resource type attached to a custom unit. When you set up a custom unit, you define the SKUs that describe what it charges for — for example, `SLURM_NODE_HOUR` for a per-node-hour HPC rate, or distinct codes for individual license tiers. Each usage event references a SKU code so the platform can rate it correctly against the unit's rules.

Usage events

A **usage event** is a record of consumption against an allocation. Each event carries a quantity, a start/end time, and a SKU code. The platform rates the event using the SKU's unit rules and rolls the result into the allocation's **Used** and **Estimated used** totals. Usage events can be deleted if a charge needs to be reversed.

Creating an allocation

From the **Organizations** list, select your organization. In the sidebar, under **Billing**, click **Allocations**, then click **New Allocation**.

Name

A unique name for the allocation within your organization. This name is shown in the dashboard and is used by integrations that record usage against the allocation.

Total

The budgeted amount, expressed in the allocation's unit.

Unit

The billing dimension. Leave this set to **USD** for a dollar-denominated budget, or select a custom unit your organization has defined.

Parent allocation

Optional. Select an existing allocation to nest this one beneath. Child allocations consume from the parent's pool. Only one level of nesting is supported.

Click **Save** when you're done.

Sharing an allocation

By default, only organization admins can see or use a new allocation. To let other users record or consume against it, find the allocation in the table and click the **permissions** icon in the **Actions** column to open its permissions panel.

You can grant three permission levels, either to specific groups or organization-wide:

- **Read** — view the allocation and its usage.
- **Use** — call the usage-event API to record consumption against the allocation.
- **Admin** — edit the allocation's total and manage its permissions.

Recording usage

> **API ONLY**
>
> In this preview, usage events can only be recorded through the platform's REST API — there is no in-platform UI for adding usage, and launching a cluster does not yet automatically bill to an allocation. To populate an allocation, call the usage-event endpoint directly or run an integration that does so on your behalf.

The endpoint accepts a quantity, a start and end time, and a SKU code. The platform rates the event using the SKU's unit rules and updates the allocation's totals.

Viewing usage

The allocations table shows each allocation's **Total**, **Used**, **Estimated**, **Remaining**, and **% Used** at a glance. To drill into the individual usage events recorded against an allocation, click the **usage** ($) icon in the **Actions** column. As with the Cost Dashboard:

- **Used** reflects confirmed, finalized spend (for cloud usage, this is the true-cost data imported from your cloud service provider, which may lag by 24-48 hours).
- **Estimated used** reflects real-time, in-flight consumption that has not yet been confirmed.

The sum of the two gives you a near real-time view of how much of the allocation remains available.

Editing and deleting

To change an allocation's total, click the **edit** icon in the **Actions** column and update the **Total** field. Other fields (name, unit, parent) are fixed at creation time.

To delete an allocation, click the **trash** icon in the **Actions** column. Deletion removes the allocation along with all of its recorded usage events, so make sure you have exported any data you need first.

Tracking Slurm usage

If your organization runs Slurm-managed HPC clusters, the `slurm-tracker` CLI can post usage from those clusters directly into the allocations system. It runs on a Slurm login or controller node (typically as a cron job

every few minutes), maps each Slurm account to an allocation and each Slurm partition to a SKU, and records core-hour consumption as usage events.

See the slurm-tracker README for installation, configuration, and the account-to-allocation mapping format.

Partner

Billing Multiplier

This page explains how to set a billing multiplier for your organization. This number is multiplied by the true cost of the instance to determine the displayed cost of the instance in the organization's billing dashboard. This feature is useful for integrating the total billing cost of both platform charges and partner charges for your customers.

Please note that multipliers are a nested feature, which means that lower levels inherit the settings of higher levels. If you make changes to the multiplier at the partner level, any organizations that have their multiplier set to **Default Settings** will inherit the partner settings.

You can further customize billing multipliers on the organization level by navigating to **Organization** > **OrganizationName** > **Partner**.

Requires: `org:admin`

From the **Organizations** list, select your organization. In the sidebar, under **Partner**, click **Billing**.

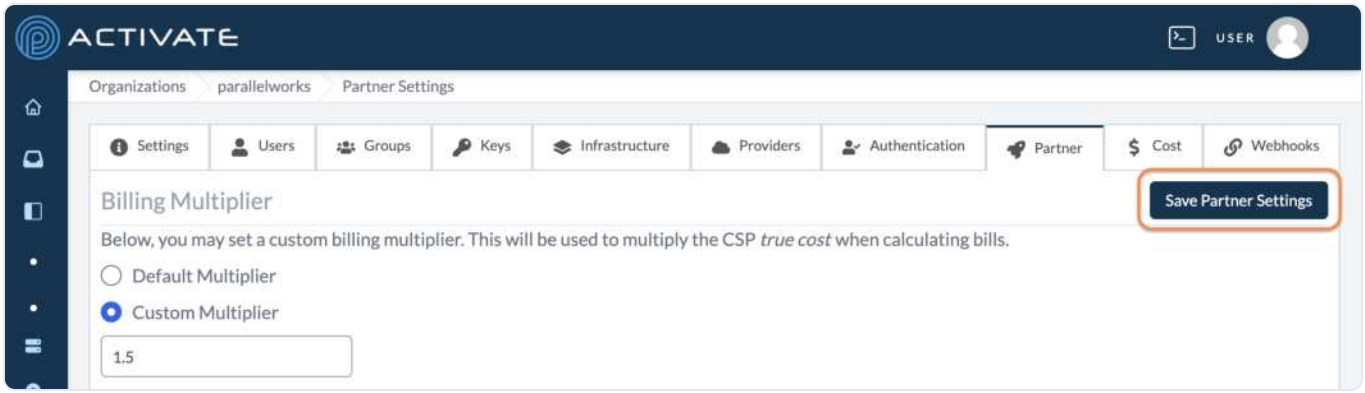
Click the partner icon for the organization you're editing.

In **Billing Multiplier**, if **Default Multiplier** is selected, users will be charged at the default cost. The checkboxes for providers are locked while **Default Multiplier** is selected.

Select **Custom Multiplier** to make changes.

Enter a number in the field below **Custom Multiplier**.

When you're done, click **Save Partner Settings**. Your changes will be applied immediately.



Partner Instance Limits

Restrict which instance types are available to organizations managed by this partner.

Requires: `org:admin`

> **NOTE**
>
> This page is only visible for organizations that have been designated as a **Partner Organization** by a platform administrator. See Platform Administration for more information.

Navigation

From the **Organizations** list, select your organization. In the sidebar, under **Partner**, click **Instance Types**.

About Partner Instance Limits

Partner instance limits allow you to control which cloud instance types are available across all organizations that your partner organization manages. This is useful for:

- **Enforcing cost controls** by limiting access to expensive instance types
- **Ensuring compliance** by restricting to specific instance families (e.g., Nitro instances on AWS)
- **Standardizing infrastructure** across managed organizations

Configuring Instance Limits

The page displays available instance types organized by cloud service provider. You can:

1. Select **Default Settings** to allow all instance types (no restrictions).
2. Select **Custom Settings** to manually choose which instance types are available.

When **Custom Settings** is selected:

- All available instance types are listed, organized by provider.
- Check or uncheck instance types to enable or disable them.
- Changes affect all organizations managed by this partner.

Click **Save** to apply your changes.

Platform Administration

Platform-level configuration for this organization. Only visible to platform administrators.

Requires: `platform:admin`

Navigation

From the **Organizations** list, select your organization. In the sidebar, click **Admin**.

Organization Flags

Configure high-level flags that control organization behavior.

- **Partner Organization** — Toggle whether this organization can create and manage other organizations.
- **Enforce MFA** — Require all users in this organization to use multi-factor authentication.
- **Receive Active User Report** — When enabled, shows an **Email Recipients** field where you can enter a comma-separated list of email addresses that will receive periodic active user reports.

Domain & Access

Configure the organization's platform access and domain settings.

- **Platform Domain** — The URL of the platform instance (e.g., `activate.parallel.works`).
- **Seats** — The number of user seats allocated to this organization. Set to `0` for unlimited.
- **Default User Host** — The default hostname for user workspaces (e.g., `user-1.example.com`).
- **Custom Login Domain** — A custom domain for the organization's login page (e.g., `example.parallel.works`).

Workspace Settings

A JSON editor for configuring workspace-level settings that apply to all users in the organization. The following fields are available:

- `k8sMounts` — Kubernetes volume mounts
- `mounts` — Additional filesystem mounts
- `envs` — Environment variables
- `dns` — Custom DNS servers
- `homeDirPrefix` — Home directory prefix path
- `dockerNetwork` — Docker network configuration

Click **Save Configuration** to apply changes.

Deleting an Organization

> **WARNING**
>
> Deleting an organization is a permanent, destructive action that cannot be undone.

At the bottom of the Admin page, click **Delete this organization**. A confirmation modal will appear requiring you to type the organization name to confirm. This action will:

- Permanently delete the organization and all of its users, groups, and cloud accounts
- Terminate all associated cloud resources

For Platform Admins

About the Admin Panel

This page explains the features in the **Admin Panel**, which you can access by clicking your username and selecting **Admin**.

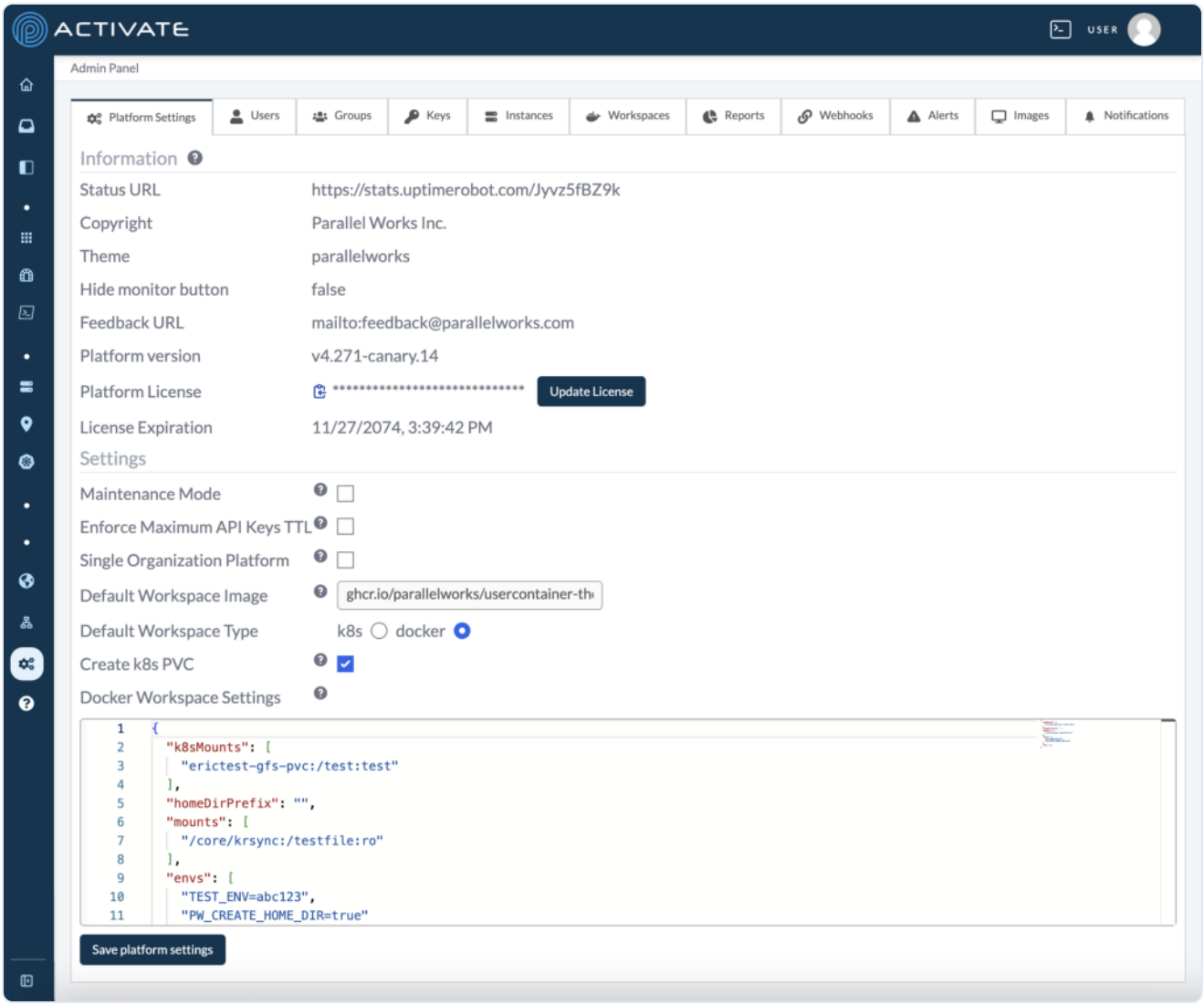
> **PERSONA**
>
> You'll only be able to access these features if you're an ACTIVATE platform administrator.

> **ABOUT CONTAINERS**
>
> In much of the user guide, we say "user workspaces" to talk about a user's account on the ACTIVATE platform. Please note that this term is synonymous with "user container," which we use on this page.

Platform Settings

When you navigate to the **Admin Panel**, you'll see **Platform Settings** by default. This tab displays essential information about your organization's version of the platform.

When making changes on this page, click **Save platform settings**. Your changes will be applied immediately.



Maintenance Mode

While **Maintenance Mode** is enabled, only admins will be able to start and stop resources.

To enable **Maintenance Mode**, click the checkbox next to it.

The platform will remain in maintenance mode until you deactivate it.

API Key Expiration

By default, users can customize the expiration date for their created API keys with no expiration or a date in 7, 30, 60, or 90 days.

You can change the maximum number of days for API keys to expire. For example, if you set this date to 30 days, users may not create a key that expires after 60 days.

To make this change, check the box for **Enforce Maximum API Keys TTL** (Time to Live) and enter a number in the **Days** field that appears.

Single Organization Platform

If you check this box, other users will only see login options for the organization your account belong to.

Default Workspace Image

This field contains the URL for the default workspace image that will be assigned to new users' accounts.

Default Workspace Type

This option lets you select how user containers are hosted. Currently, the available workspace types are Kubernetes (k8s) and Docker.

Create k8s PVC

This option creates a PersistentVolumeClaim (PVC) in users' home directories. For more information about PVCs, please see this page of Kubernetes's documentation.

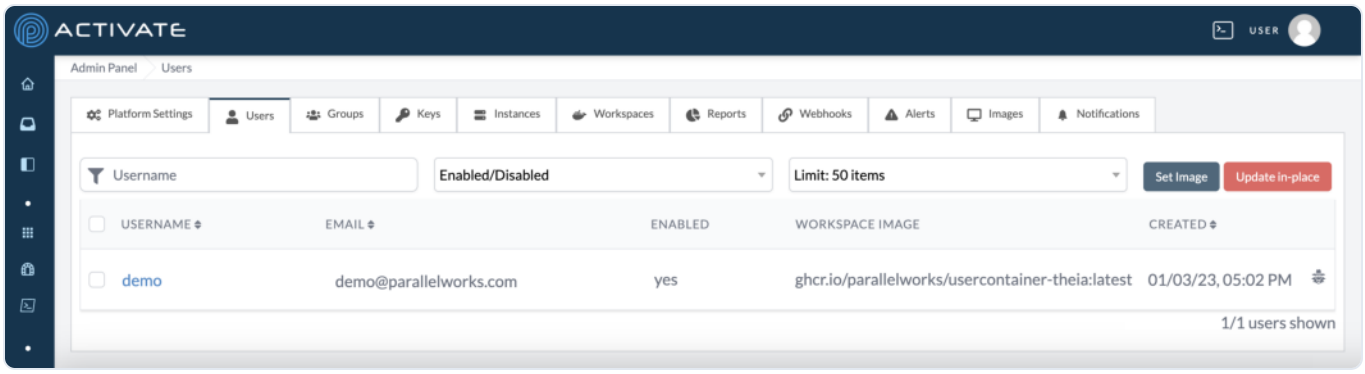
Docker Workspace Settings

Use this field to change Docker workspace settings, such as additional mounts, variables, and Domain Name System (DNS) settings.

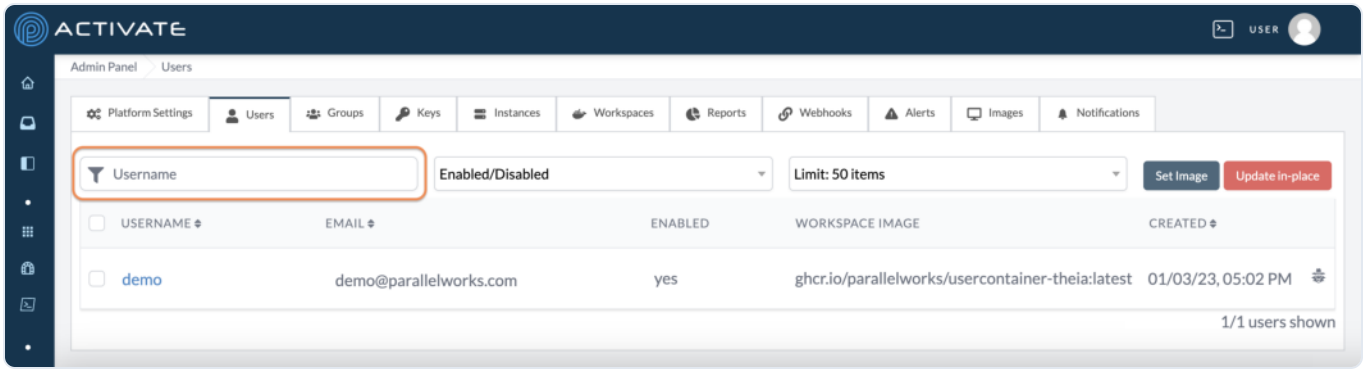
Users

This tab displays all of your users' information, including:

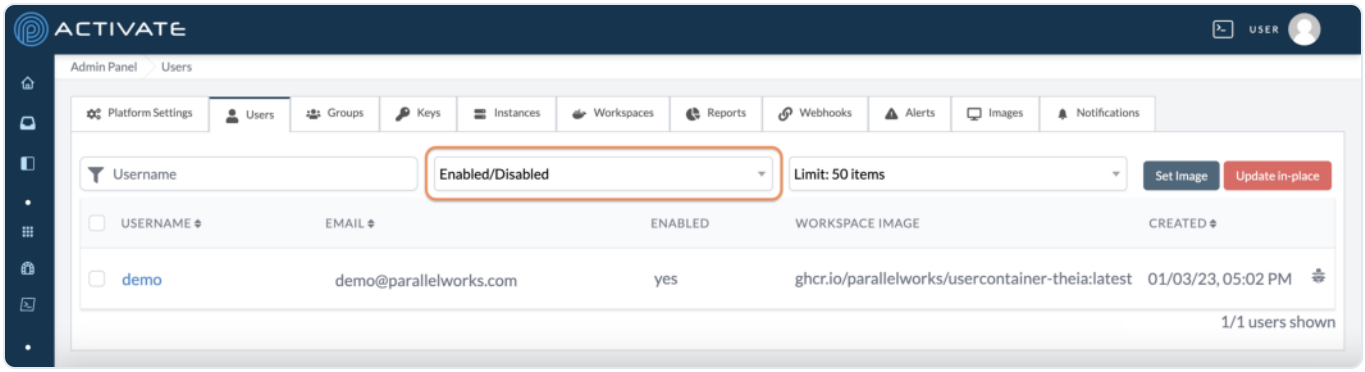
- username
- email address
- enabled status
- workspace image
- account creation date



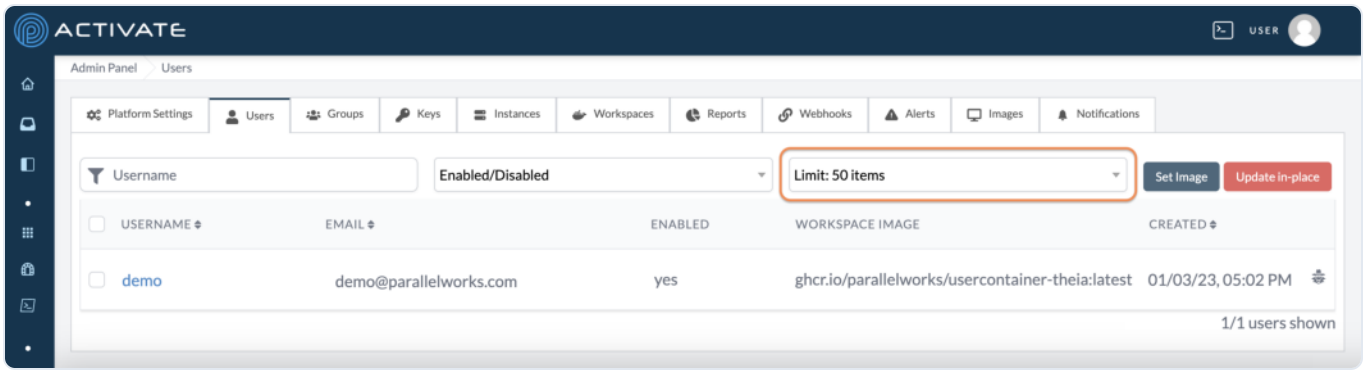
Use the **Username** field to search for users.



Use the **Enabled/Disabled** dropdown menu to display users that are enabled, disabled, or either. This option is set to **Enabled/Disabled** by default.

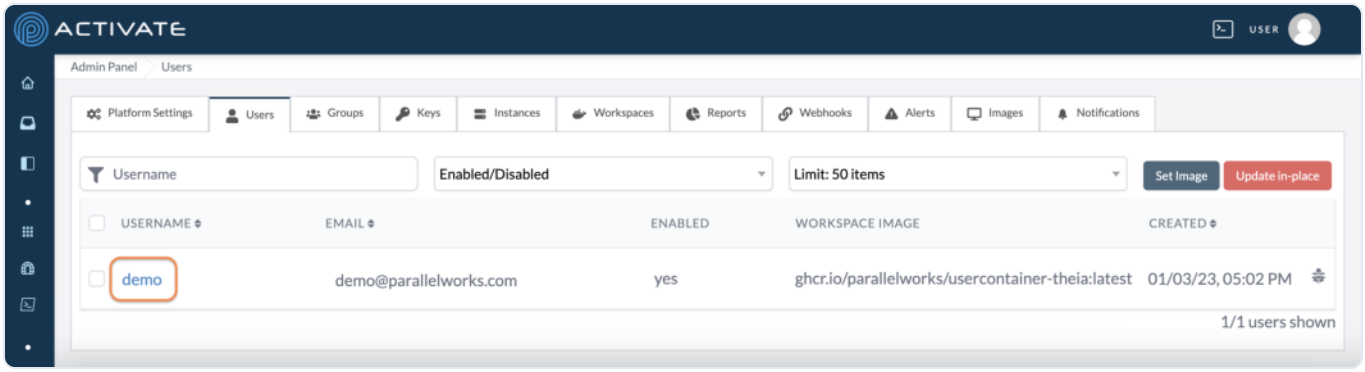


Use the **Limit** dropdown menu to narrow the listed users to 10, 20, 50, 100, or All. This option is set to **50** by default.



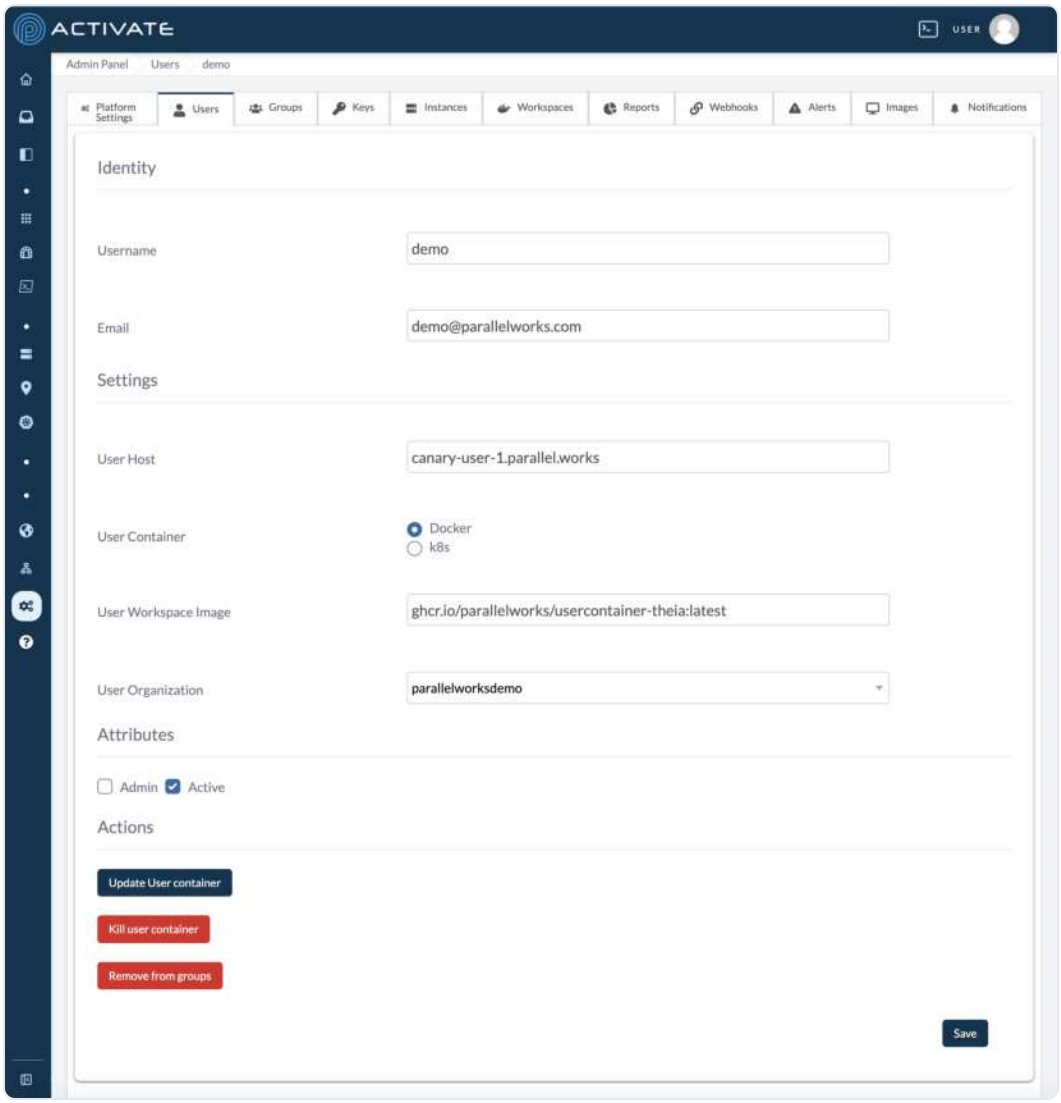
Editing User Information

In the **Users** tab, click on the username you'd like to edit.



On the next page, you can edit several of the user's parameters.

The first section contains information about the user's account, container, and access to the platform.



Identity

Here, you can edit a user's **Username** or **Email**. Altering either of these parameters will change the user's login credentials.

Settings

Here, you can control a user's access to the platform.

User Host

This field identifies the user host, which is the part of the platform where account resources are provisioned.

We recommend contacting us before you make changes to this parameter.

User Container

This option identifies which service will be used to deploy a user's container.

Use the bubbles to select **Docker** or **k8s** (Kubernetes).

We recommend contacting us before you make changes to this parameter.

MFA Enabled

Use the bubbles to select whether the user will use multi-factor authentication.

Select **No** if a user doesn't need multi-factor authentication.

Select **Duo** if you work with a company or organization that is not affiliated with the government. Duo is a Cisco service that is more popular with other types of institutions.

Select **MyProxy** if you work with a governmental agency, such as a public university or a research organization. Generally, MyProxy is preferred for these types of institutions.

We recommend contacting us before you make changes to this parameter.

User Workspace Image

This field identifies which image version a user's workspace is running.

User Organization

Use this dropdown menu to assign the user to an organization.

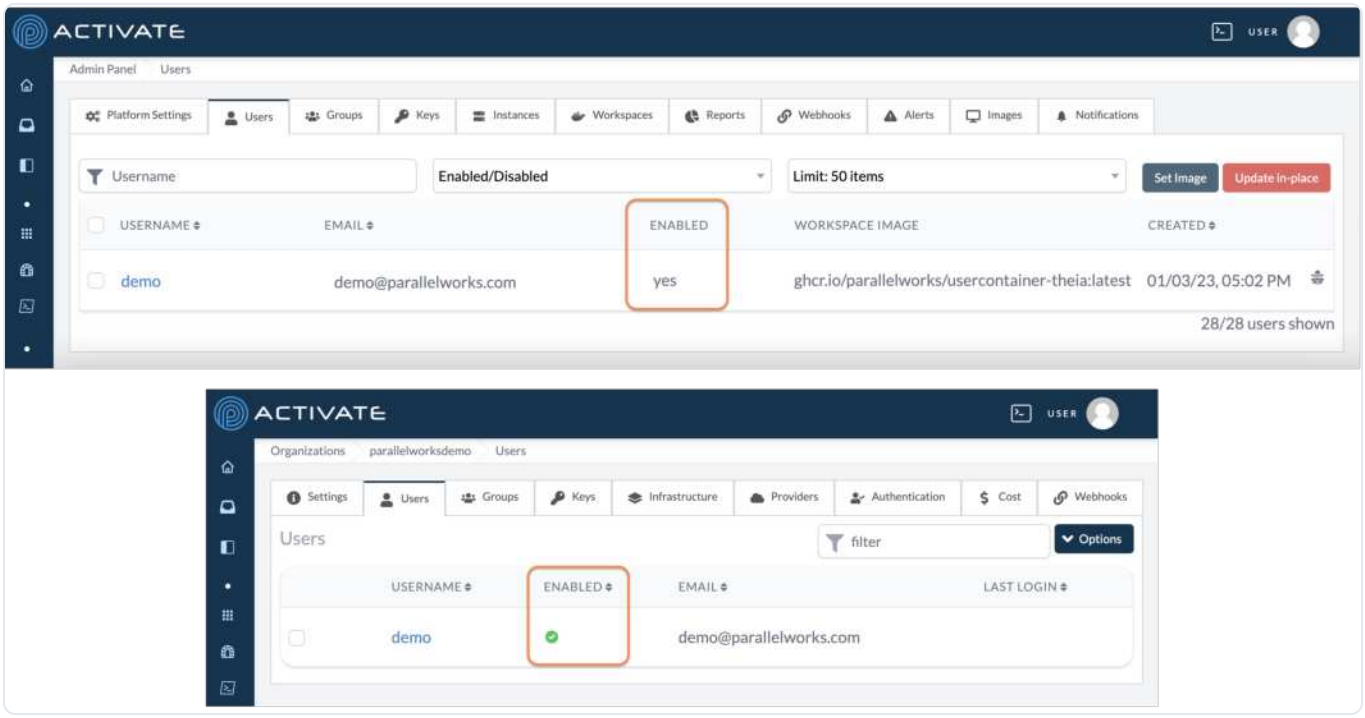
Attributes

Here, you can control whether a user is an **Admin** or **Active** by selecting their respective checkboxes.

A user with **Admin** enabled will be able to access the **Admin Panel** and all of its settings. This option does not affect the group setting `org:admin`. For more information, please see **About Roles**.

A user with **Active** enabled will be able to log in to the platform with their username and password. Unchecking **Active** will disable a user's access to the platform.

Users who are **Active** are marked as **Enabled** in **Admin** > **Users** and **Organization** > **Users**.



Actions

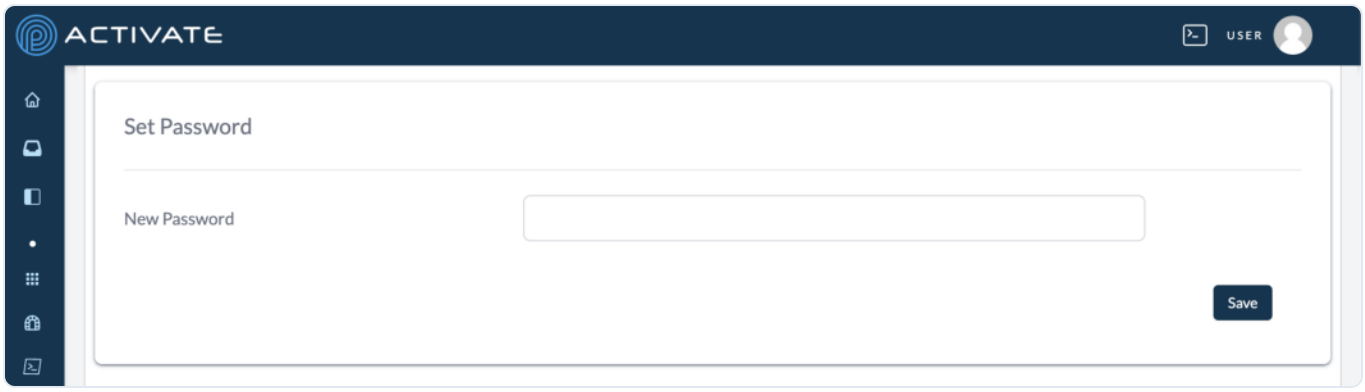
Click the **Update User container** button to migrate a user's container to the latest image.

Click the **Kill user container** button to restart a user's container.

Click the **Remove from teams** button to remove a user from all of their assigned groups.

> **NOTE**
>
> If you change any of the parameters above, click **Save** before leaving the page. Your changes will be applied immediately.

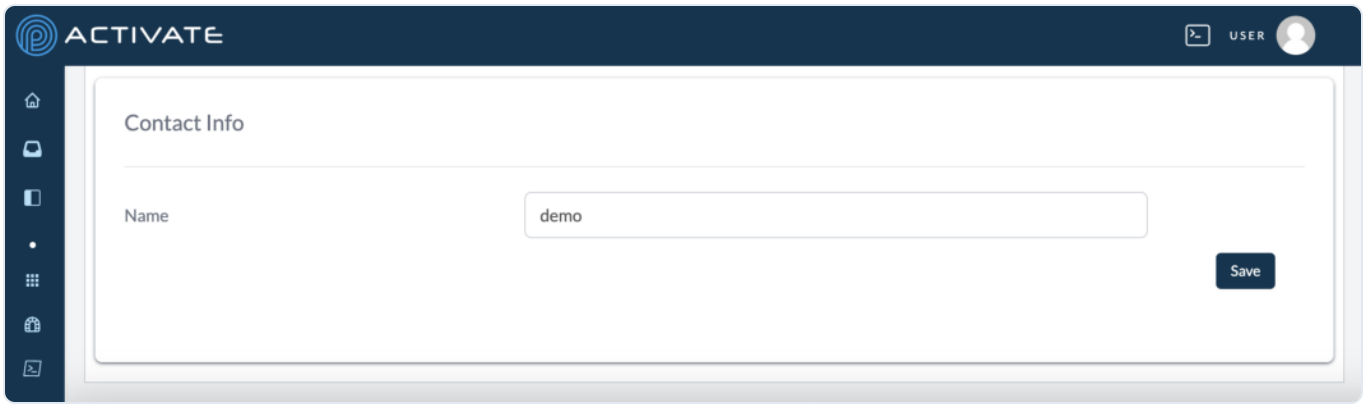
Set Password



Here, you can reset a user's login credentials.

If you change this parameter, click **Save** before leaving the page. Your change will be applied immediately.

Contact Info



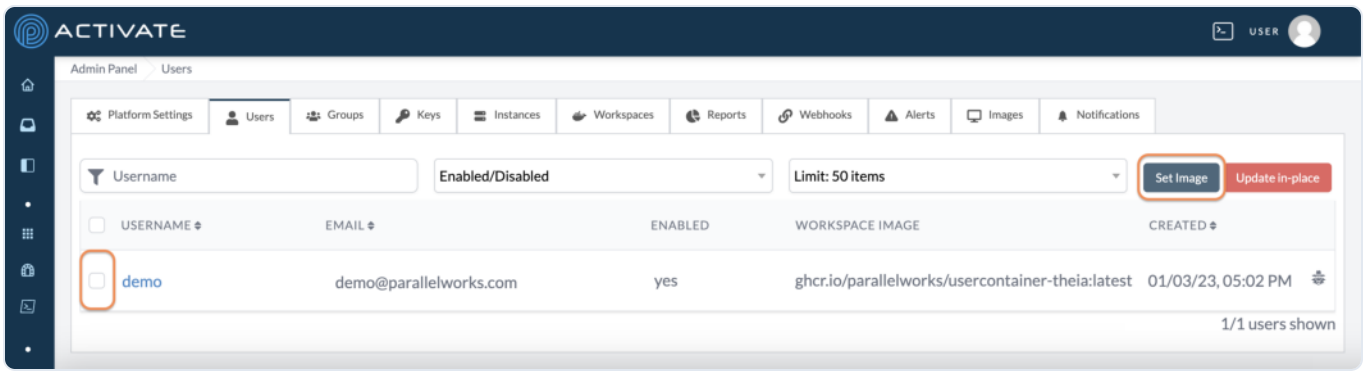
Here, you can change a user's contact information, including their display name, platform URL, and geographical location.

This information is only visible to platform admins and does not affect a user's login credentials.

If you change these parameters, click **Save** before leaving the page. Your changes will be applied immediately.

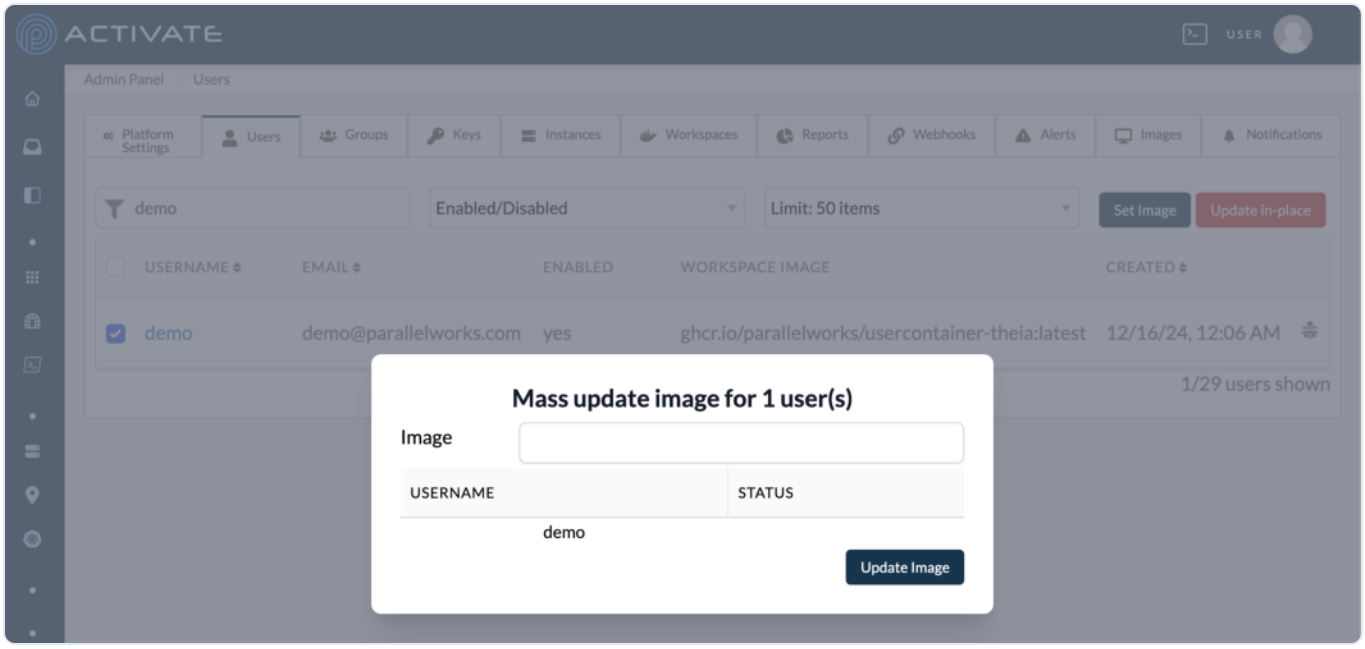
Setting User Images

In the **Users** tab, select users with their checkboxes. Click **Set Image**.



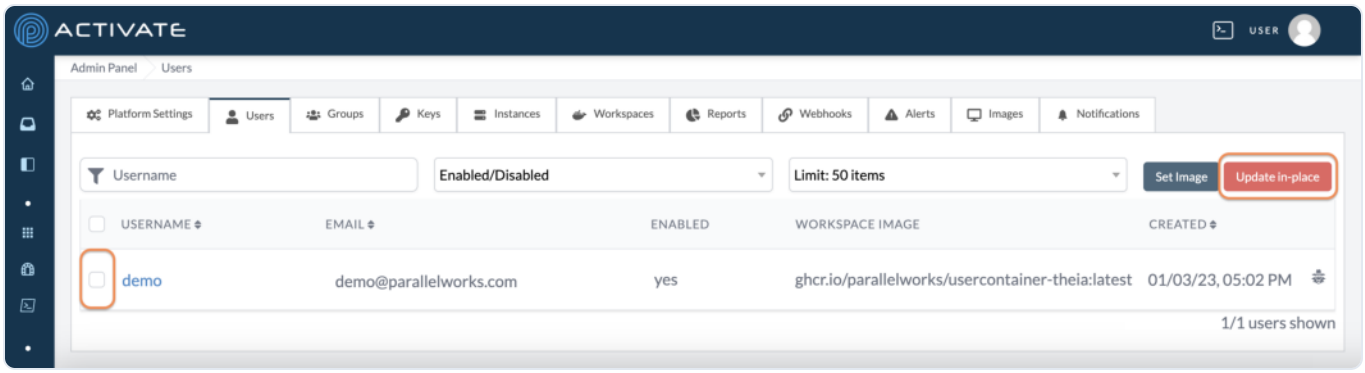
A dialog box will appear. Enter the **Image Name** and the image **Version**.

Click **Update Image**. Your changes will be applied immediately.



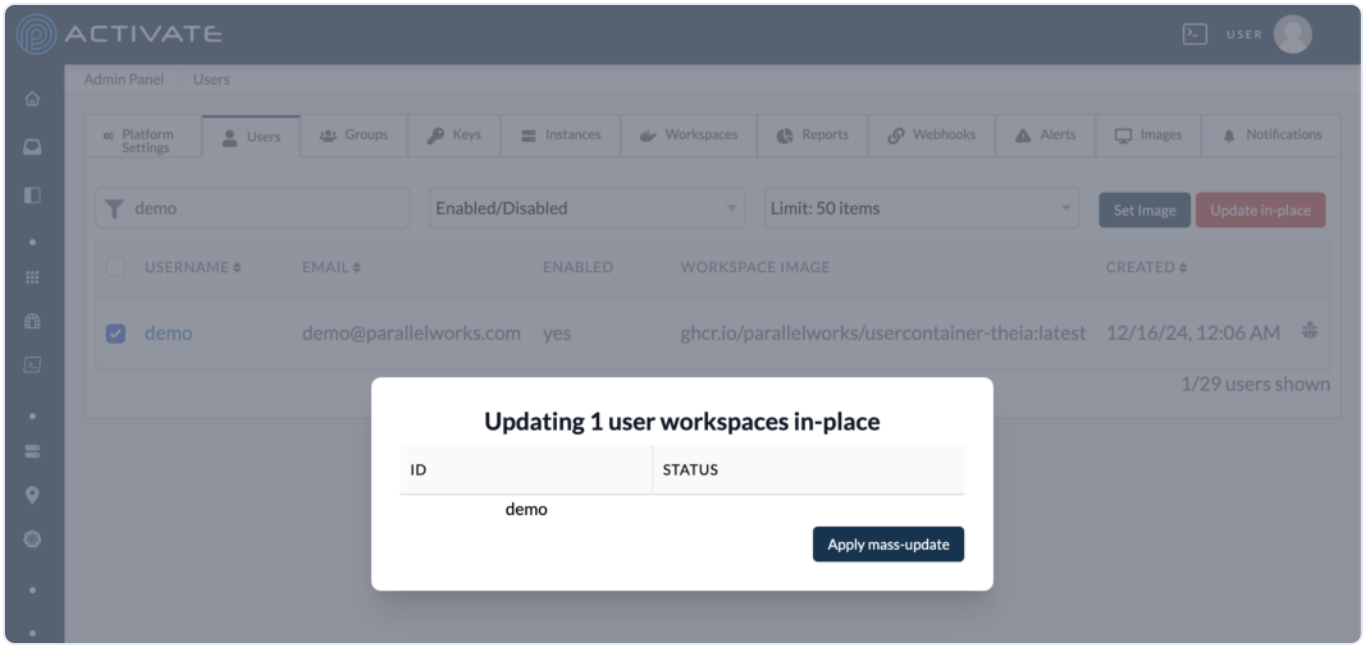
Updating User Images

In the **Users** tab, select users with their checkboxes. Click **Update in-place**.



A dialog box will appear. Confirm that you've selected the users you want to update.

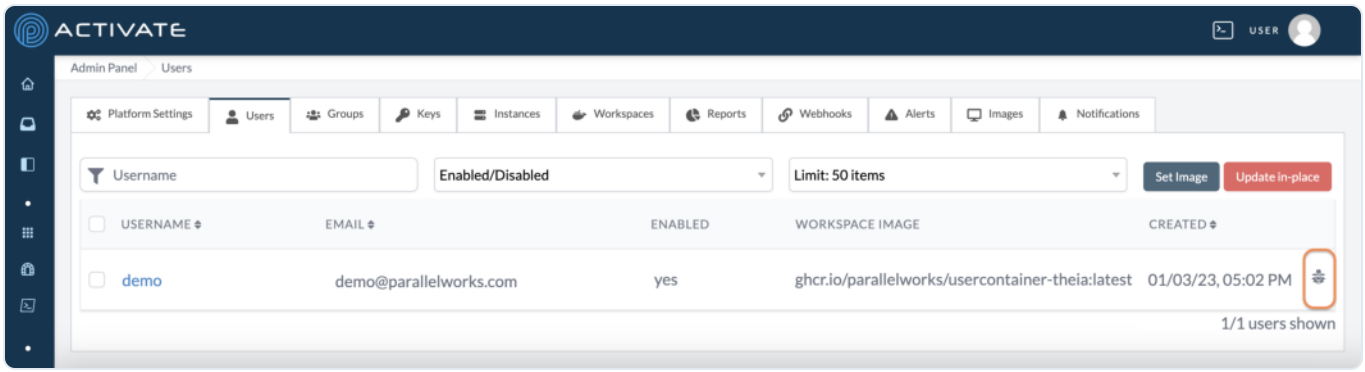
Click **Apply mass-update**. Your changes will be applied immediately.



Using Impersonate

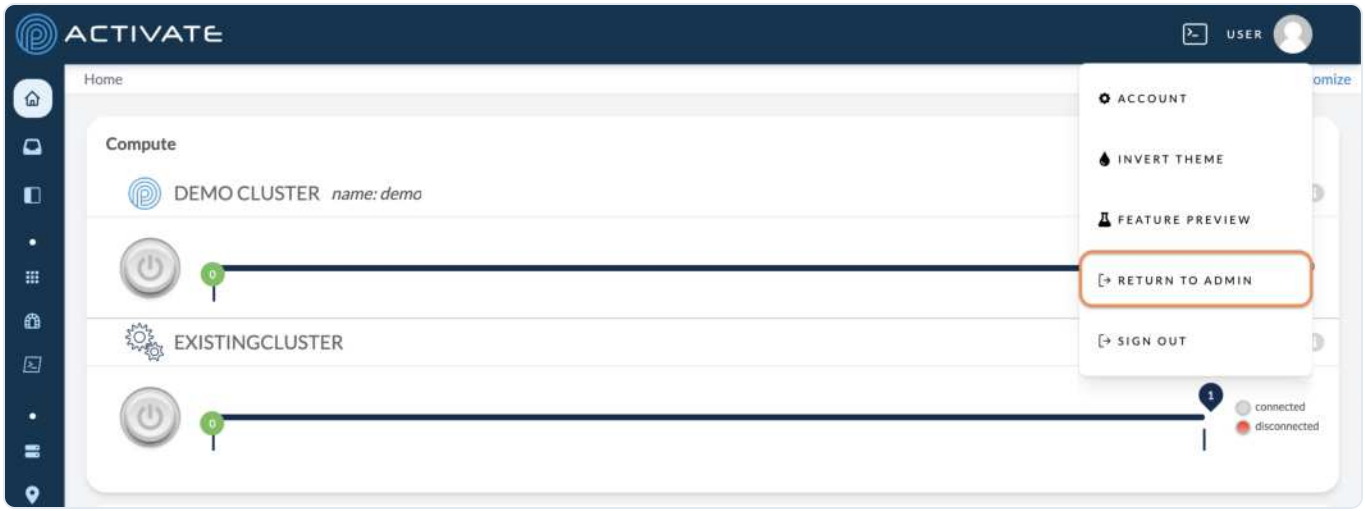
If you want to quickly access another user's account, you can use the impersonate feature to bypass the login/logout process.

In the **Users** tab, click the impersonate icon for the account you want to access.



You'll be redirected to the user's **Home** page. You can navigate and control resources, storage resources, and workflows as if you were logged in to the user's account.

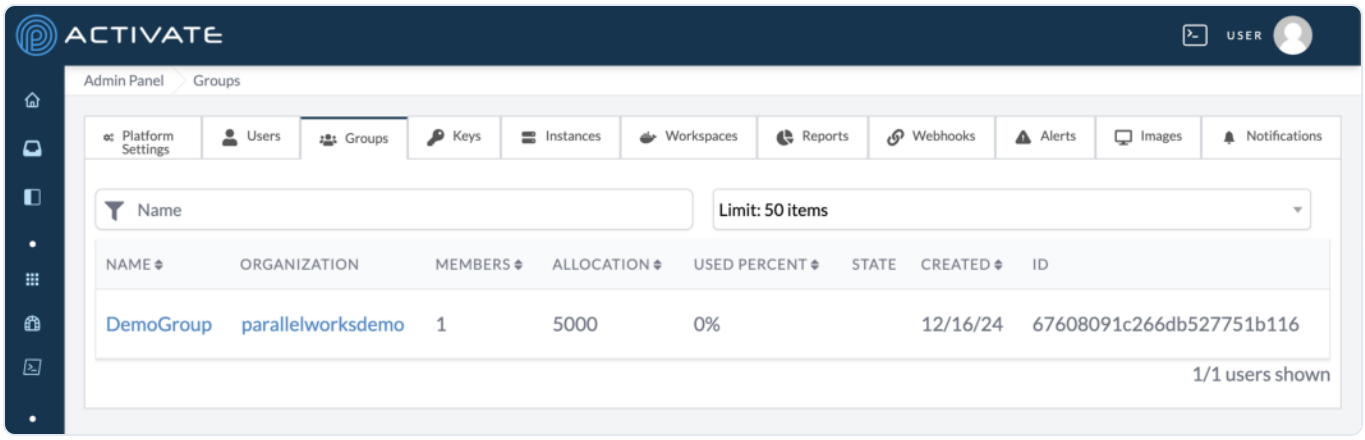
When you're done, click the user's name, then **Return to [Your Account]**.



Groups

This tab displays information for all of your groups, including:

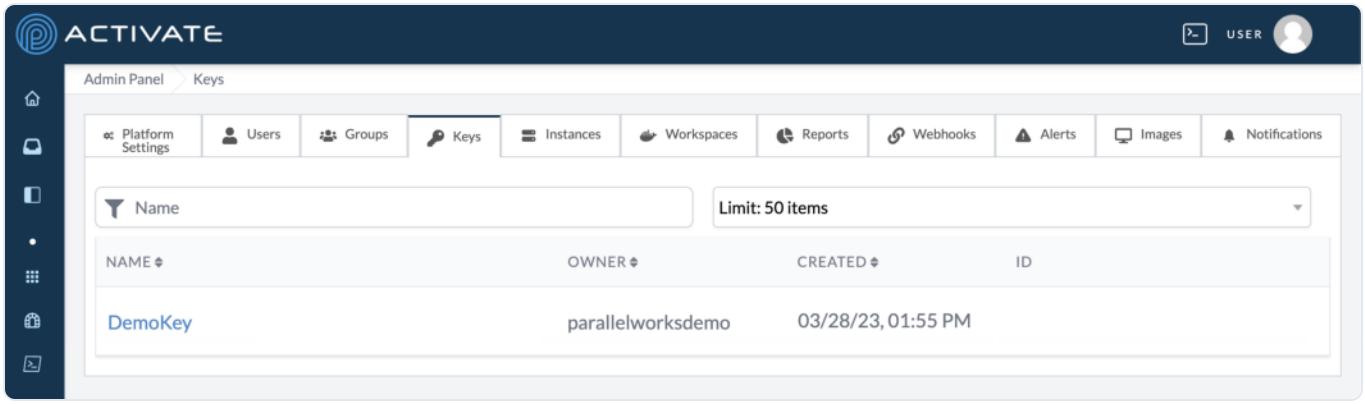
- name
- organization
- number of members
- allocation amount
- percentage of allocation used
- state
- creation date
- ID number



All the functions in **Managing Groups** can be completed from this tab as well.

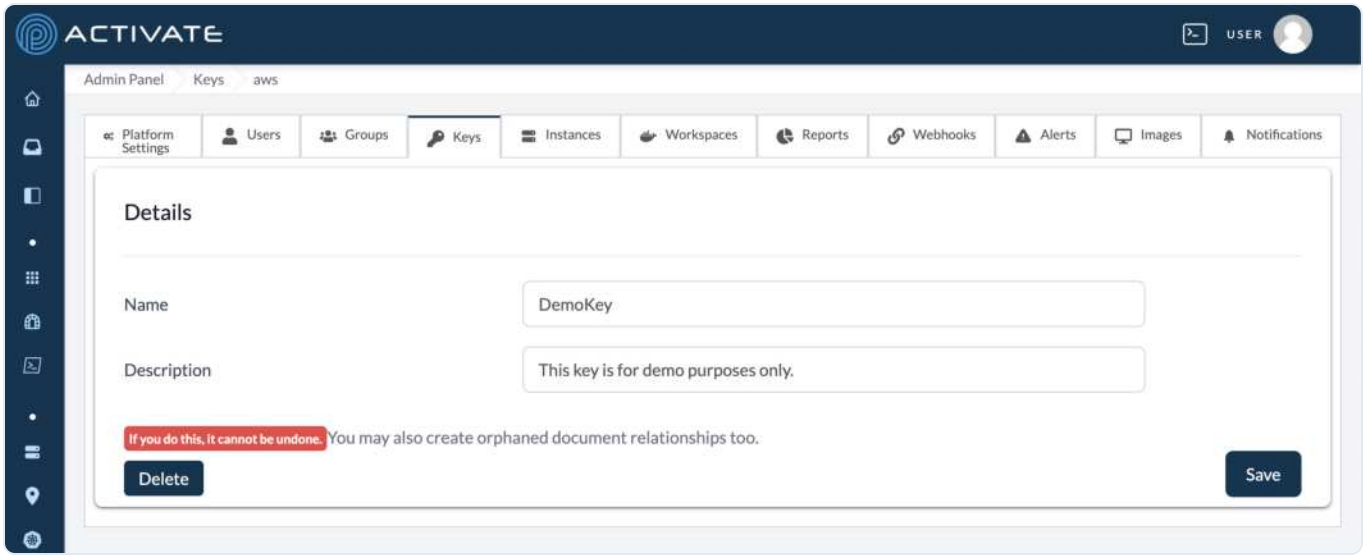
Keys

This tab shows information for all of your cloud account keys, including their name, which organization owns the key, the creation date, and the key ID.



You can manage your cloud account keys in this tab. If you click the name of a key, you can edit its **Name** and **Description** on the next page. Click **Save** after making any changes.

You can also permanently delete the key from the platform by clicking the **Delete** button. Please note that if you've used a key to provision infrastructure, you won't be able to delete that key until you've deprovisioned the infrastructure.



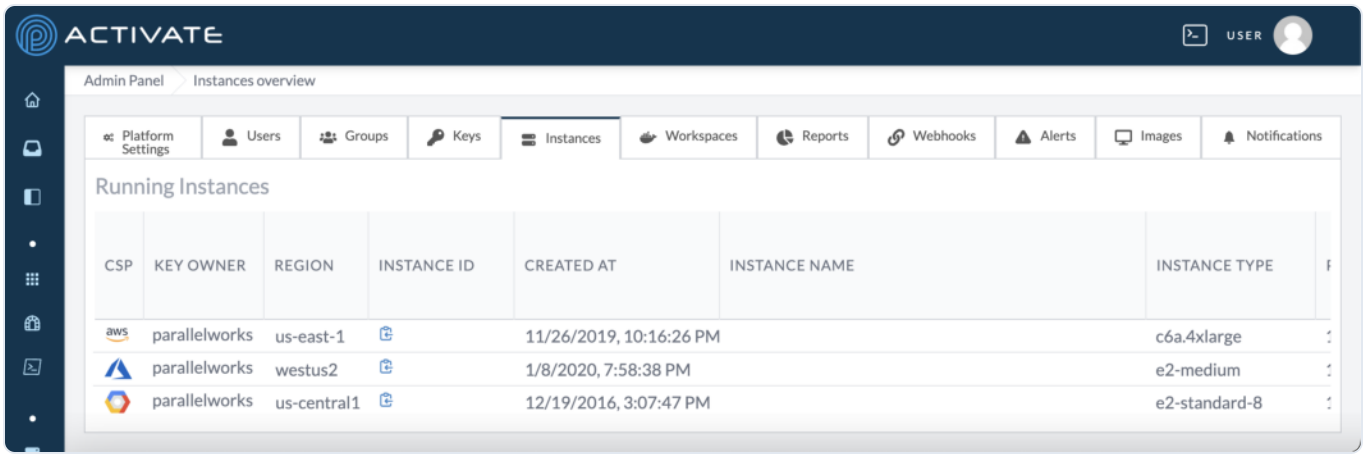
> **NOTE**
>
> If you need to add a cloud account to the platform, please see **Managing Cloud Accounts**.

Instances

This tab displays information for both running instances and other instances, including:

- CSP
- organization that owns the key
- region
- instance ID
- creation date

- instance name, if any
- instance type
- private IP
- public IP
- cloud state
- estimated cloud state time
- ACTIVATE state



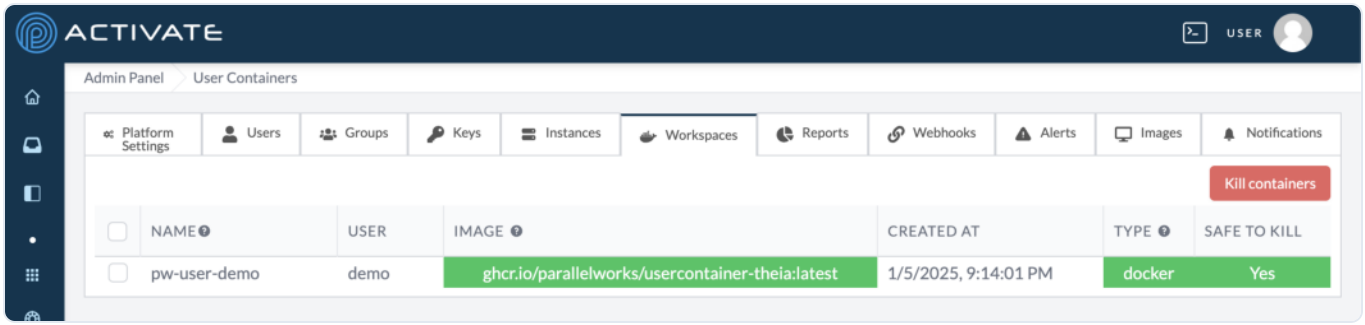
Please note that the instance list is wider than most monitors, so the included screenshot does not show all of the fields listed above. To see more information about your instances, scroll to the bottom of the page and use the horizontal scroll bar to show more columns.

You can configure your instance types on the platform by organization. For more information, please see **Configuring Instance Types**.

Workspaces

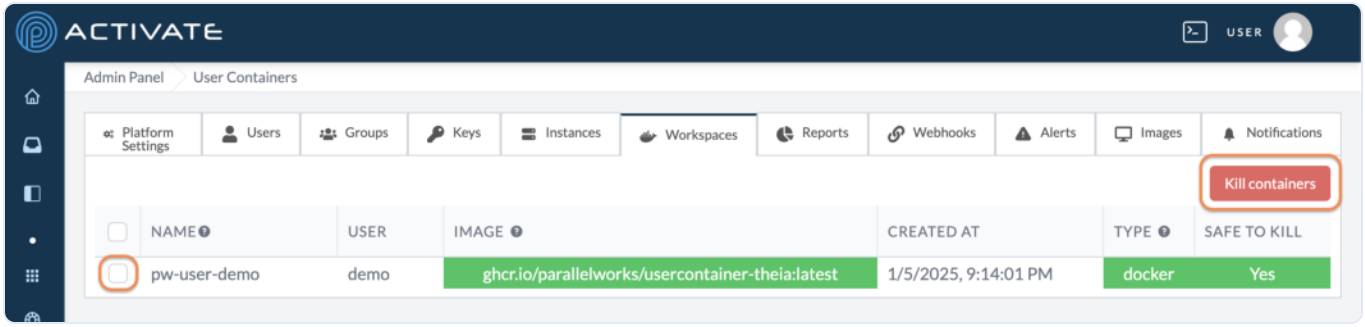
This tab displays information for all of your user containers, including:

- container name
- user
- container image
- creation date
- type (Docker or Kubernetes)
- safe to kill



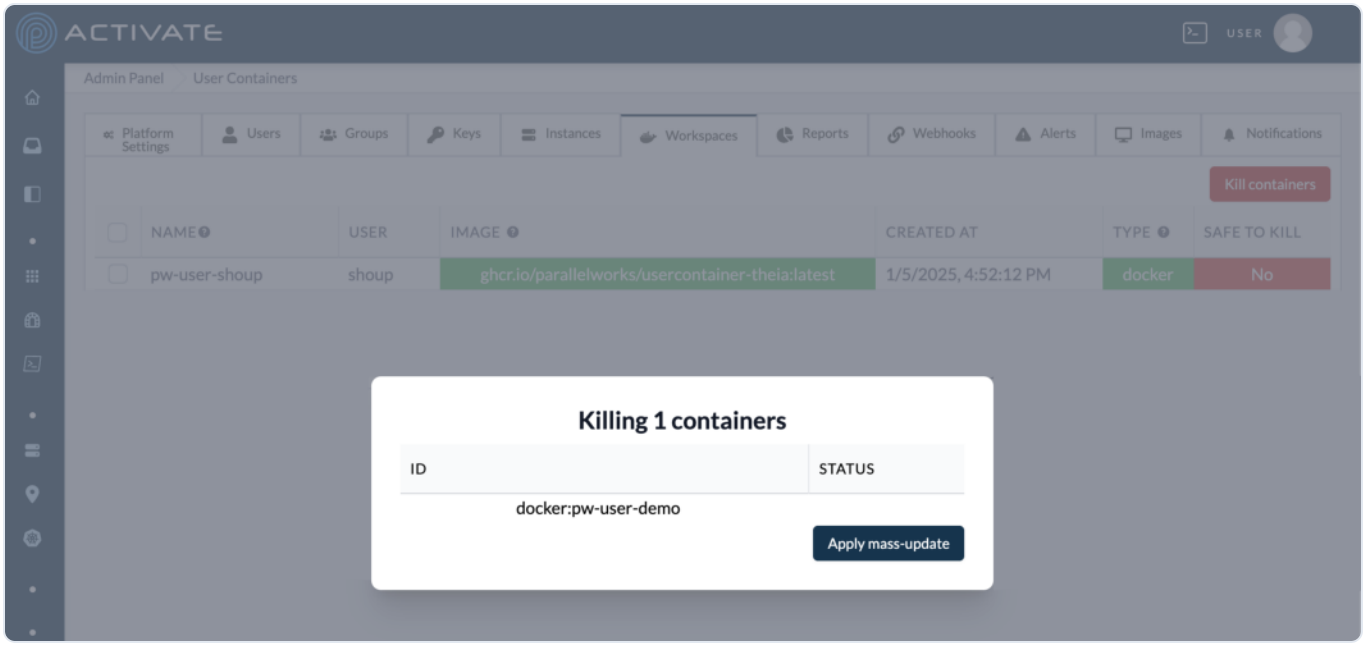
Killing User Containers

In the **Containers** tab, select users with their checkboxes. Click **Kill containers**.



A dialog box will appear. Confirm that you've selected the containers you want kill.

Click **Apply mass-update**. Your changes will be applied immediately.

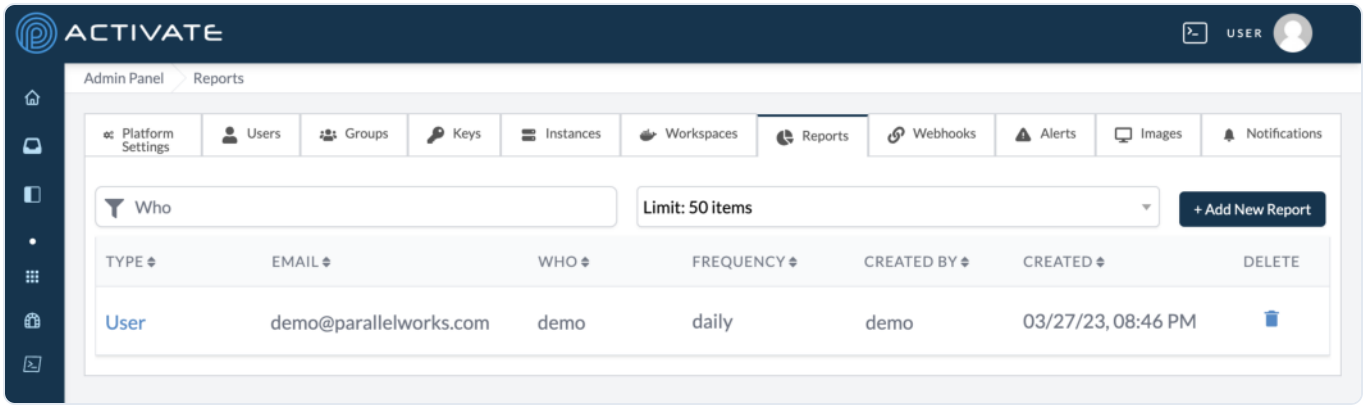


Reports

This tab displays information for your reports, including:

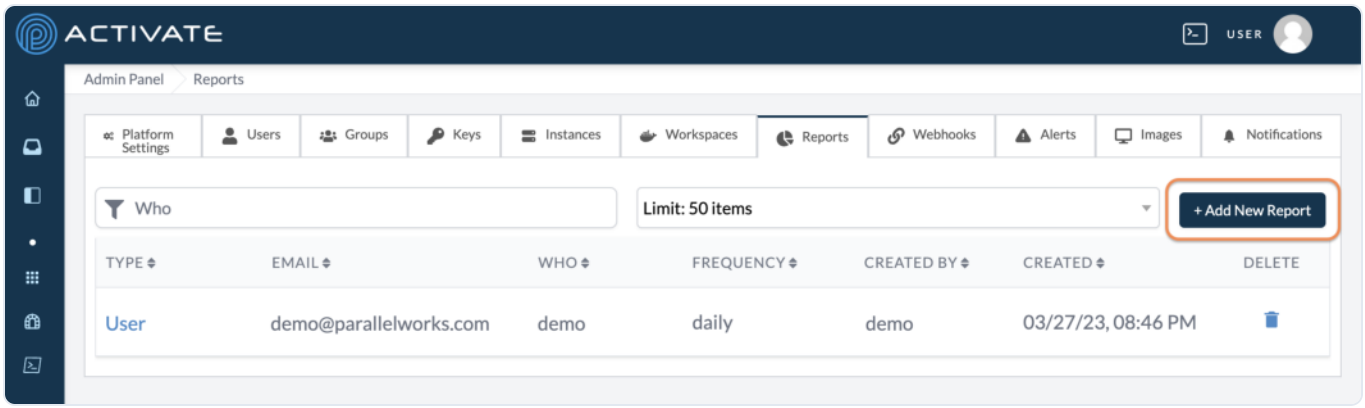
- type of report

- email address that receives reports
- who the report monitors
- frequency
- creator
- creation date

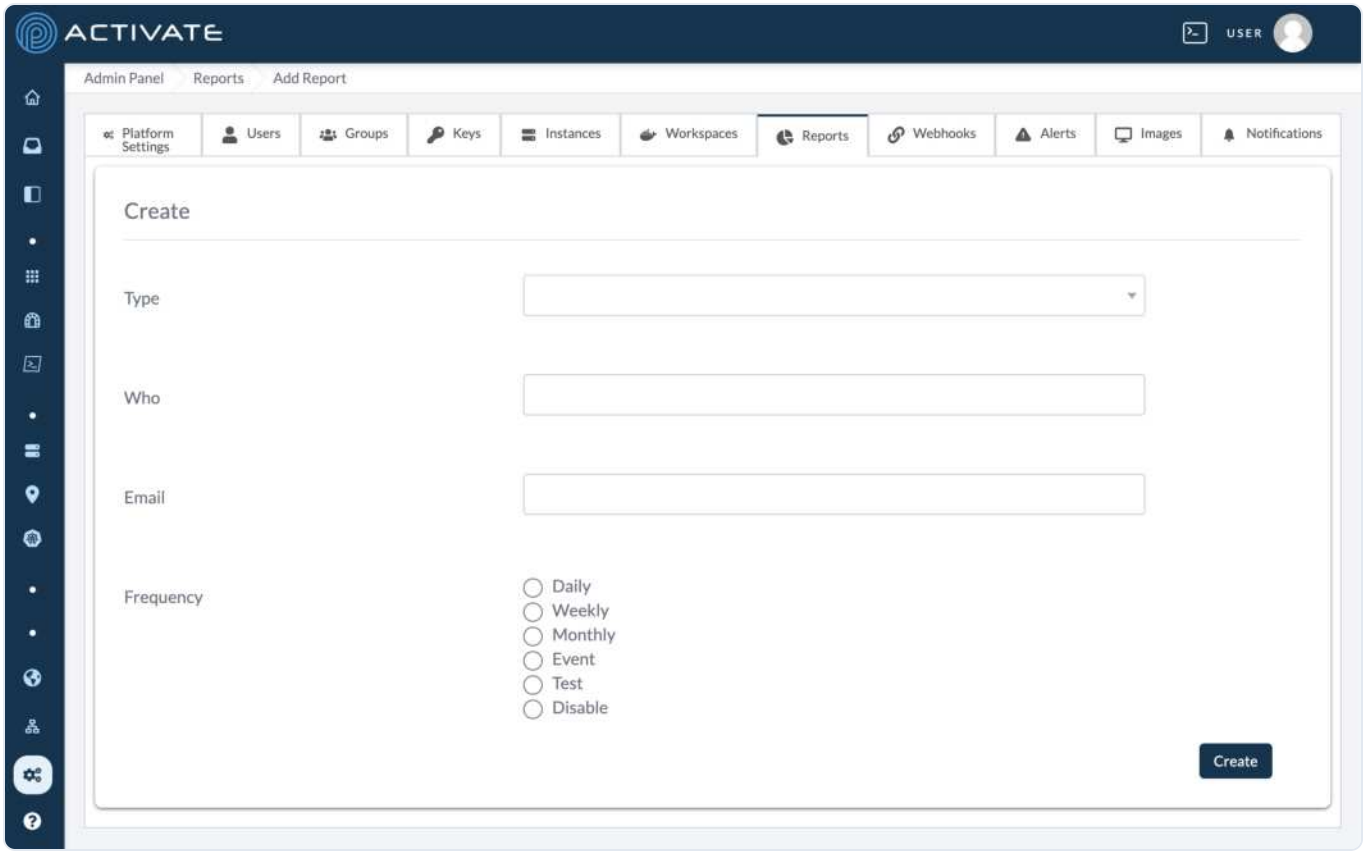


Creating Reports

In the **Reports** tab, click **+ Add New Report**.



On the next page, you can edit the parameters listed below.



Use the **Type** dropdown menu to select the type of report you want to generate:

- summary
- project
- monthly
- user
- application

Use the **Who** field to enter the username who the report will monitor.

Use the **Email** field to enter the report recipient's email address.

Use the **Frequency** bubbles to select how frequently the report will be generated:

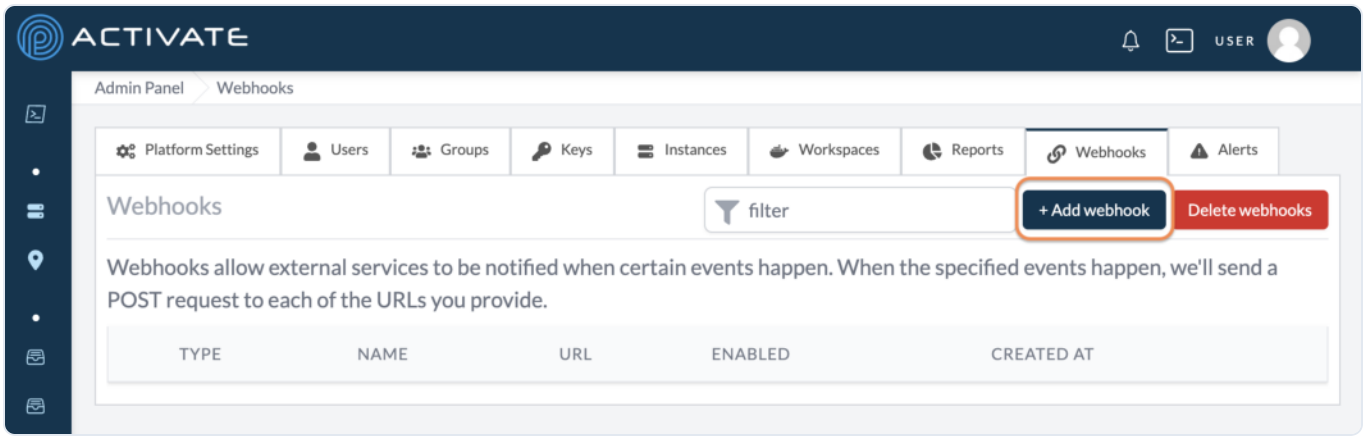
- daily
- weekly
- monthly
- event (once, at the time of creation)
- test (once, at the time of creation)
- disable (end reports for the time being)

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide.

Adding a Webhook

Click + **Add Webhook**.



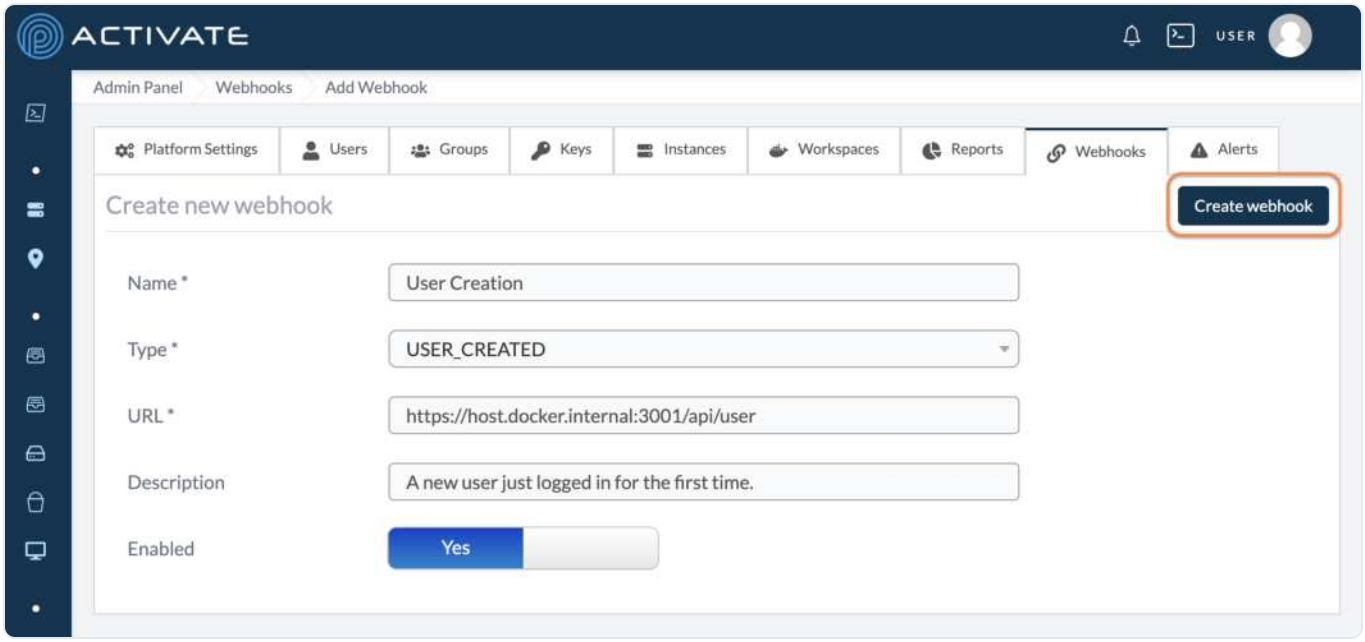
On the next page, select a **Type**. Currently, you can choose from these webhook types:

- `USER_CREATED` flags when a new user logs in for the first time.
- `USER_DISABLED` flags when a user has been disabled.
- `USER_REMOVED_FROM_GROUP` flags when a user has been removed from a group in your organization.

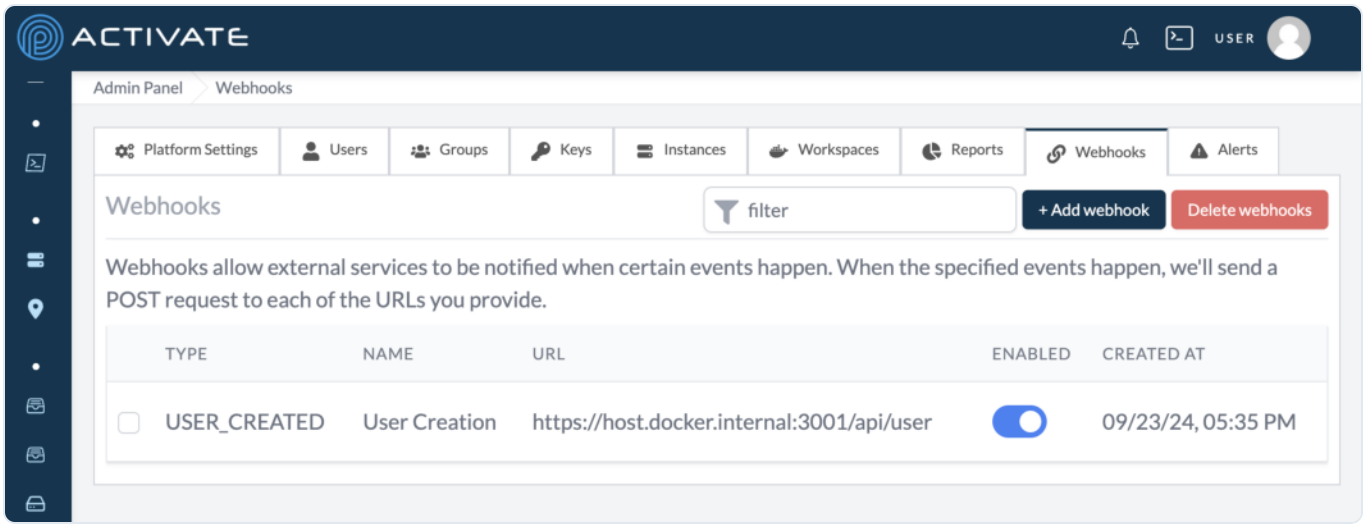
Add a **Webhook name** and paste the **URL** that your webhook will be sent to. **Description** is an optional field.

Choose whether to **Enable** this webhook.

Click **Create webhook**.

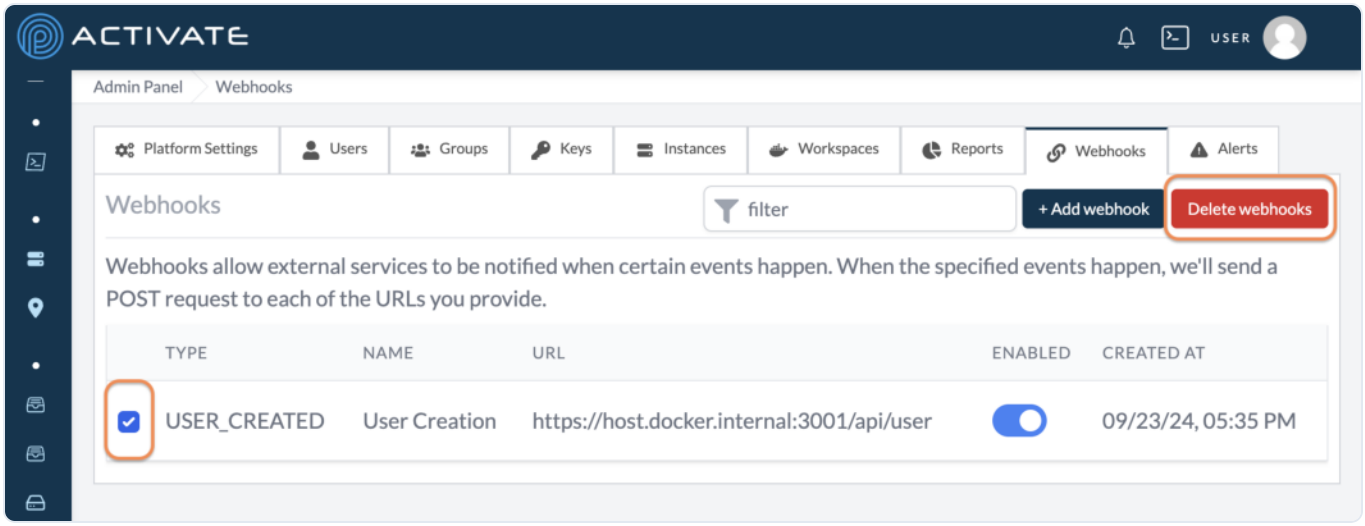


You'll be taken back to the **Webhooks** tab. A dialog box with the message *Webhook created successfully* will appear in the bottom right corner of your screen.

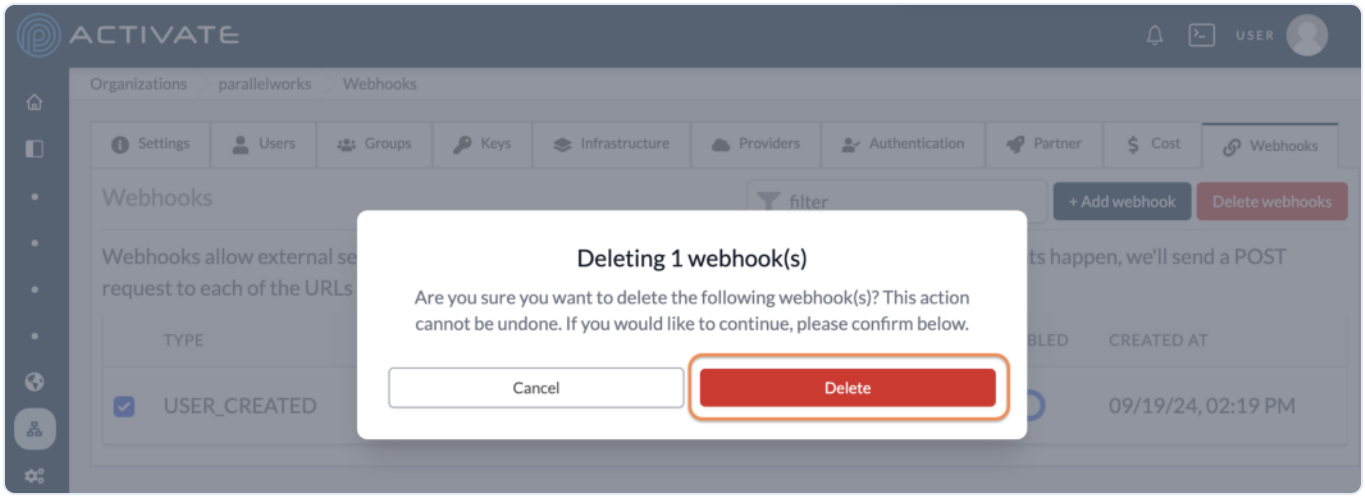


Deleting a Webhook

Select the webhook you'd like to delete, then click **Delete webhooks**.



A pop-up module will appear. Click **Delete**.



You'll be taken back to the **Webhooks** tab. A dialog box with the message *Webhook(s) deleted successfully* will appear in the bottom right corner of your screen.

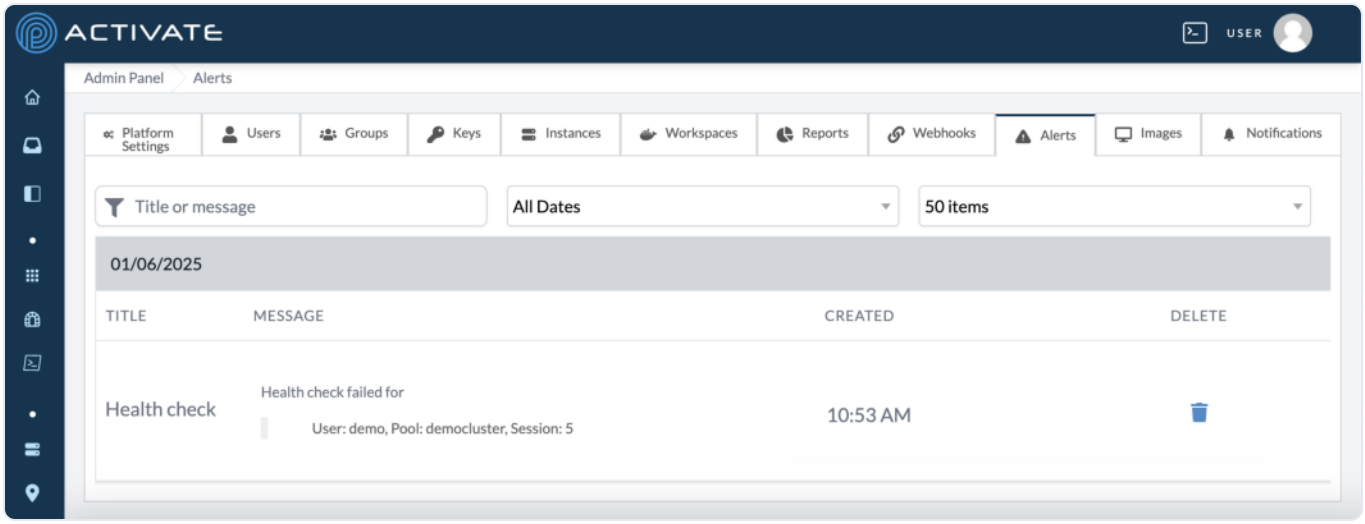
Responding to Events

To see the shape of webhooks, please see this section. Admin-level webhooks take the same shape as organizaiton-level webhooks, with one exception:

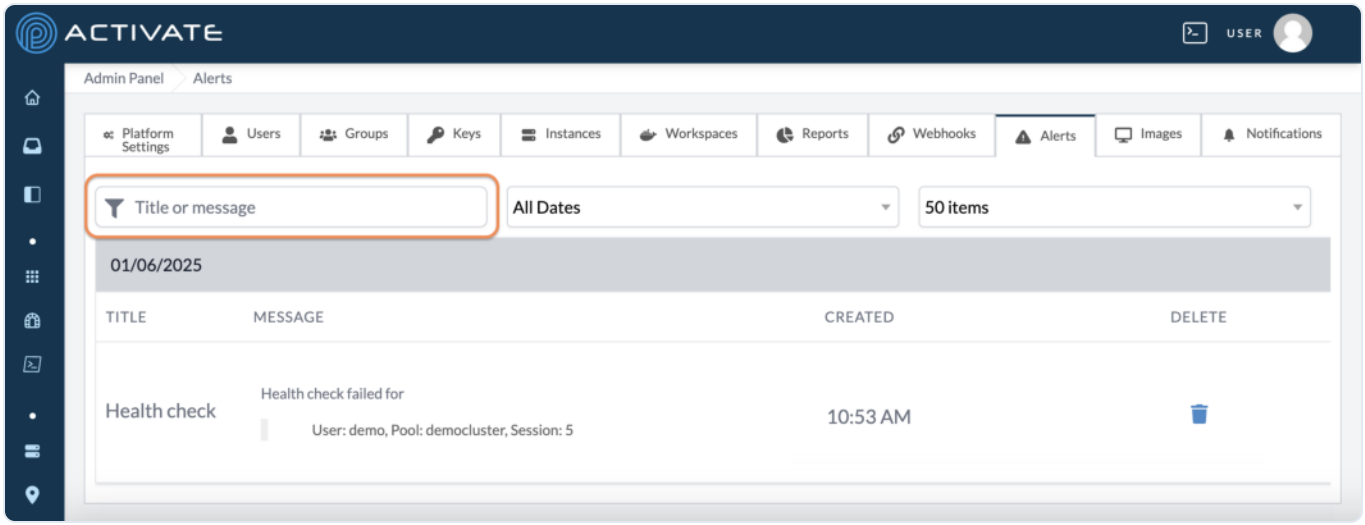
- You will not see `organization` inside the `webhook` object.

Alerts

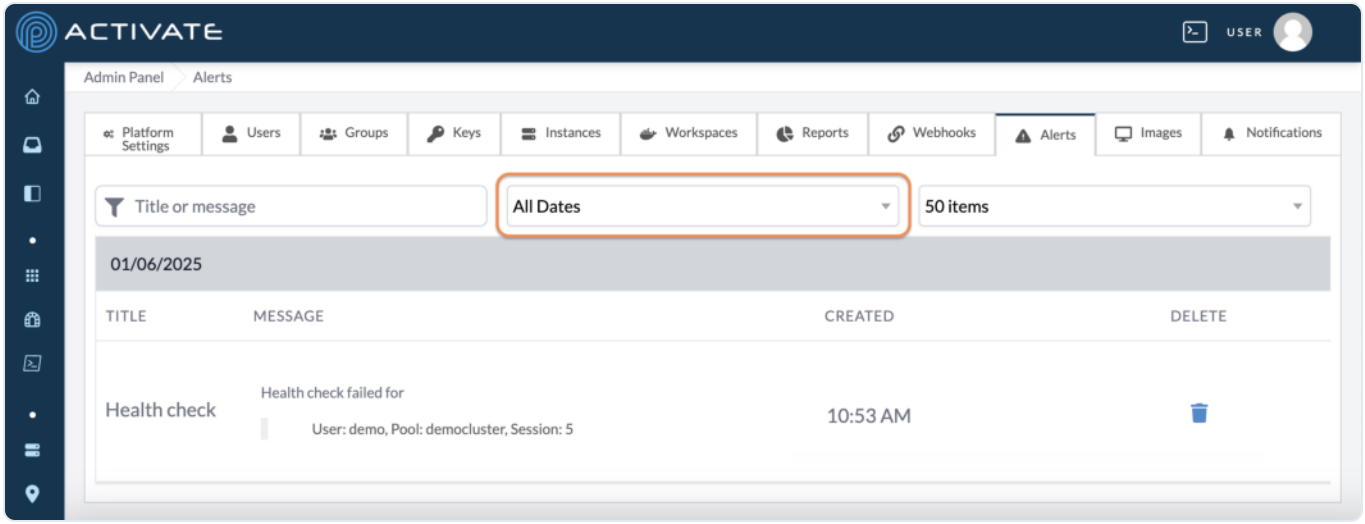
This tab collects error messages from all users in your organization. These messages are shown for 7 days before they're automatically deleted.



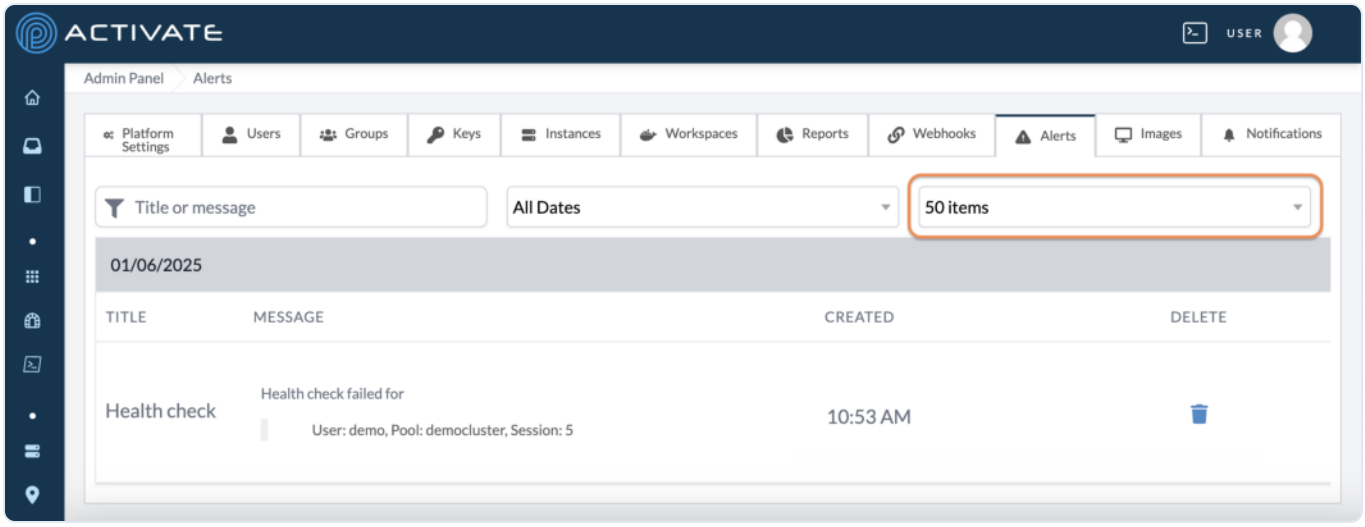
Use the **Title or message** field to search for alerts by keywords.



Use the **All Dates** dropdown menu to display alerts from a specific date.



Use the **max item** dropdown menu to narrow the listed users to 10, 20, 50, 100, or max items. This option is set to **50** by default.



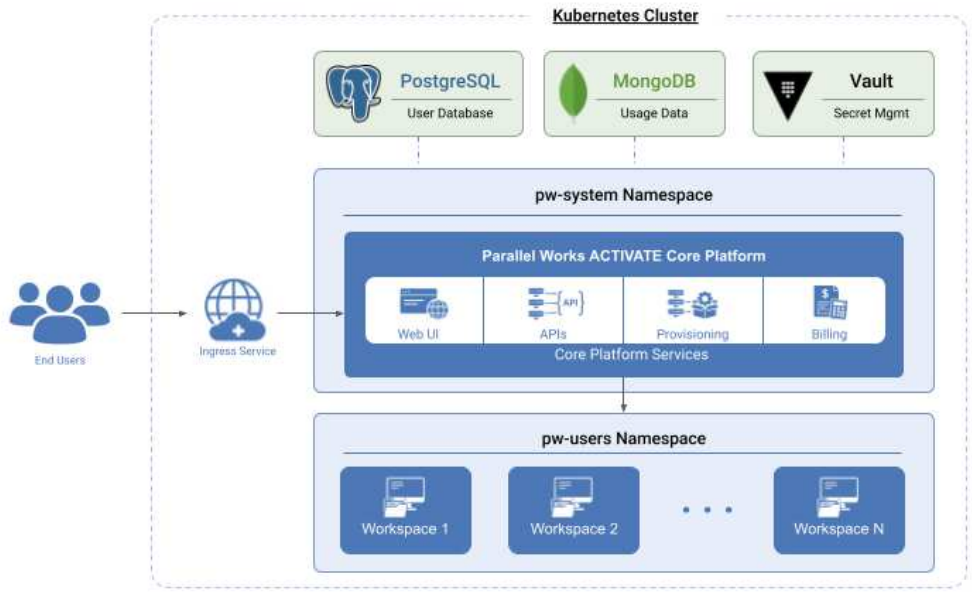
Self-Hosting

Kubernetes Deployment

This document describes the recommended technical process for deploying the Parallel Works ACTIVATE control plane into a customer-managed, self-hosted Kubernetes environment such as Red Hat OpenShift. The guide reflects a production-oriented deployment model and preserves flexibility for customer-specific infrastructure and security requirements. Parallel Works will provide validated Helm charts and Kubernetes manifests as part of the deployment engagement.

Deployment Topology

The following diagram illustrates the deployment architecture for ACTIVATE on Kubernetes:



Minimum Node Requirements

The following tables outline the minimum resource requirements for deploying ACTIVATE.

Cluster Sizing

Deployment Size	Nodes	CPU per Node	RAM per Node	Total CPU	Total RAM
Minimum	3	4 cores	16 GiB	12 cores	48 GiB
Recommended	5	8 cores	32 GiB	40 cores	160 GiB

Equivalent instance types: AWS `m5.xlarge` (minimum) / `m5.2xlarge` (recommended), GCP `n2-standard-4` / `n2-standard-8`.

Platform Resource Requests (Default Scale)

At `defaultScale: 1`, the ACTIVATE platform services require approximately:

Category	CPU	Memory	Persistent Storage
Core platform services	6 cores	6 GiB	—
Background workers	7 cores	11 GiB	11 GiB
Supporting services (message queue, secrets)	2 cores	4 GiB	3 GiB
Billing and usage	2.5 cores	4 GiB	—
Total	**17.5 cores**	**25 GiB**	**14 GiB**

Additional headroom is required for Kubernetes system pods (CoreDNS, kube-proxy, CNI) and the database services below.

Database Requirements

Database	CPU	Memory	Storage
PostgreSQL	2 cores	6 GiB	50 GiB
MongoDB (3-node replica set)	2 cores per node	4 GiB per node	50 GiB

Shared Storage

A shared filesystem with `ReadWriteMany` access is required for platform file storage:

Purpose	Minimum Size	Access Mode
User uploads, media, and workflow assets	10 GiB	ReadWriteMany

Additional `ReadWriteOnce` volumes (~10 GiB total) are provisioned automatically for internal caching and message queue data.

Supported shared storage backends include AWS EFS, GCP Filestore, or any other NFS or ReadWriteMany-capable storage class.

Prerequisites and Assumptions

Before beginning the deployment, ensure the following requirements are met:

- **Kubernetes Environment**: A dedicated Kubernetes or OpenShift cluster for ACTIVATE (see Minimum Node Requirements above)
- **Cluster Access**: Cluster-admin privileges during installation
- **Storage**: Persistent storage class available for PVCs and a shared filesystem (EFS, Filestore, or any other NFS or ReadWriteMany-capable storage class)

- **DNS**: External DNS entry for the ACTIVATE ingress endpoint
- **TLS Certificates**: TLS certificates issued for the chosen domain
- **Registry Access**: Outbound access to the Parallel Works container registry

Supporting Services Provisioning

ACTIVATE relies on several supporting services for metadata, billing, and secret management. These services are deployed into the cluster using Helm charts and Kubernetes manifests provided by Parallel Works.

PostgreSQL (Usage and Billing Data)

- **Version**: PostgreSQL 18
- **Deployment**: Kubernetes-managed service or managed database (e.g. AWS Aurora, GCP Cloud SQL)
- **Baseline Resources**: 2 vCPU, 6 GiB RAM, 50 GiB storage
- **Purpose**: Usage metrics, billing, and activity tracking

MongoDB (User and Platform Metadata)

- **Version**: MongoDB 8.x
- **Deployment**: Replica set mode with 3 nodes (required for sessions and transactions)
- **Baseline Resources**: 2 vCPU, 4 GiB RAM per node
- **Purpose**: User accounts, platform configuration, and metadata

HashiCorp Vault (Secrets Management)

- **Edition**: Vault Community Edition
- **Deployment**: In-cluster
- **Baseline Resources**: 1 vCPU, 2 GiB RAM
- **Requirements**:
  - Vault must be initialized and unsealed
  - Deployment engineers must have access to create a Vault token or have the root token available
- **Purpose**: Secure storage of credentials and sensitive configuration

Core Platform Deployment

The ACTIVATE control plane is deployed via Helm and a custom Kubernetes resource managed by the Parallel Works operator.

Namespace and Persistent Storage

- Create a dedicated namespace for the core platform, for example `pw-system`
- Provision a PersistentVolumeClaim of approximately 10 GiB with `ReadWriteMany` access mode
- This volume stores user-uploaded images and marketplace workflow assets

Kubernetes Secrets

Create Kubernetes secrets in the core platform namespace for the following:

- Container registry pull credentials (provided by Parallel Works)
- PostgreSQL credentials
- MongoDB credentials
- Vault token

Operator Installation

Add the Parallel Works Helm repository and install the operator:

```
helm repo add corerepo https://parallelworks.github.io/helm
helm upgrade --install parallelworks-operator corerepo/operator \
  --namespace pw-operator-system \
  --create-namespace
```

ACTIVATE Platform Custom Resource

Deploy the ACTIVATE platform by applying a `ParallelWorks` custom resource. This resource defines image versions, scaling defaults, ingress configuration, TLS, and database connectivity. Key configuration elements include:

- Platform image registry credentials
- Ingress hostname and TLS secret
- PostgreSQL endpoint and credential secret
- MongoDB replica set endpoints

Parallel Works will provide a validated YAML manifest customized for the customer environment. See Sample Configuration below for a YAML example.

Resulting Platform Namespace

Once applied, the operator provisions all ACTIVATE services into the designated core namespace (for example `pw-system`). This namespace hosts the control plane components and ingress services.

User Workspace Configuration

User workspaces are isolated into a separate namespace. Each user is assigned a dedicated Kubernetes StatefulSet that manages a single pod. Pods are scaled up when the user is active and scaled down when idle.

Resource Defaults

Each user workspace is provisioned with the following default resource allocation:

	CPU	Memory
Requests	500m	512 MiB
Limits	1 core	2 GiB

These values are configurable per user through the ACTIVATE platform.

Home Directory Storage

Each user is provisioned a 5 GiB home directory. Storage can be configured in one of two modes through the platform:

- **Per-user PersistentVolumeClaim** — Each user receives a dedicated PV for their home directory
- **Shared storage mount** — Users share a ReadWriteMany volume (e.g. NFS, EFS, or Filestore)

Ingress and External Access

An ingress controller or external load balancer must be configured to expose port 443 to the ACTIVATE ingress service running in the core platform namespace.

- DNS should resolve the chosen domain to the ingress endpoint
- TLS termination is handled using the configured Kubernetes TLS secret

Initial Platform Access

After ingress is operational, access the ACTIVATE web interface via HTTPS on the configured domain.

The initial setup workflow allows you to:

1. Enter the ACTIVATE license key
2. Create the first platform administrator account

Once completed, the platform is ready for user onboarding and workspace provisioning.

Sample Configuration

Below is a sample `ParallelWorks` custom resource YAML configuration:

```
---
apiVersion: v1
kind: Namespace
metadata:
  name: <namespace>
---
apiVersion: platform.parallelworks.com/v1alpha1
kind: ParallelWorks
metadata:
  name: <deployment-name>
  namespace: <namespace>
spec:
  image:
    registry: <registry-url>
    repositoryPrefix: <repository-prefix>
    tag: <version>
    pullPolicy: IfNotPresent
    pullSecrets:
      - name: <image-pull-secret-name>

  # Default scale for all services that don't specify their own replicas
  defaultScale: 1
```

```
  # Ingress hostname for the platform  host: <your-domain>  # TLS certificate secret
tlsSecretName: <tls-secret-name>  # Database configuration  database:    postgres:      endpoint:
'<postgres-host>:<postgres-port>'      database: '<database-name>'      username: '<username>'
urlSecretRef:        name: '<postgres-secret-name>'        key: '<password-key>'    mongodb:
replicaSetName: '<replica-set-name>'      endpoints:        - '<mongodb-node-0-host>:27017'
- '<mongodb-node-1-host>:27017'        - '<mongodb-node-2-host>:27017'
```

CUSTOMIZATION

This is a sample configuration. Parallel Works will provide a validated and customized manifest tailored to your specific environment during the deployment engagement.

Next Steps

After completing the deployment:

1. Configure authentication providers (e.g. OAuth/LDAP)
2. Set up user groups and permissions
3. Configure compute resources and cloud providers
4. Import or create workflows in the marketplace

For additional assistance, please contact Parallel Works support or your deployment engineer.

Port Reference

This document provides a comprehensive reference for all network ports required when self-hosting the ACTIVATE platform. Use this guide to configure firewalls, network security groups, and load balancers for your deployment.

Architecture Overview

The ACTIVATE platform uses a hub-and-spoke network model:

- The **control plane** runs in a central Kubernetes cluster and exposes a small set of external ports.
- **Compute cluster agents** (on cloud, managed, or existing clusters) establish outbound WebSocket connections to the control plane. No inbound ports need to be opened on compute clusters.
- All inter-service communication within the control plane cluster uses internal Kubernetes networking.

External Ports {#external-ports}

These ports must be exposed on the control plane cluster's load balancer or ingress for the platform to function.

Port	Protocol	Required	Purpose
443	TCP (HTTPS)	Yes	Platform web UI, REST API, and agent WebSocket tunnels
80	TCP (HTTP)	Recommended	HTTP-to-HTTPS redirect
8443	TCP (HTTPS)	Optional	CAC/Smart Card authentication (mTLS)

Port 443 (HTTPS)

This is the primary entry point for all platform traffic. A single port serves:

- **Web interface**: the ACTIVATE UI
- **REST API**: all platform API requests (`/api/...`)
- **Agent tunnel connections**: compute cluster agents connect via `wss://<platform-host>/ws/ssh/tunnel` using persistent multiplexed WebSocket connections
- **SSH-over-WebSocket**: browser-based terminal sessions to user workspaces
- **SSE streams**: real-time event delivery to the UI

TLS termination can be handled by the ACTIVATE ingress service itself (using the configured TLS secret) or by an external load balancer / reverse proxy.

Port 80 (HTTP)

Serves only as an HTTP-to-HTTPS redirect. Can be omitted if your load balancer handles redirection externally.

Port 8443 (CAC Authentication)

Required only for deployments using DoD CAC / smart card authentication. This port runs a separate TLS listener that requires mutual TLS (client certificate verification) against the DoD PKE certificate chain. If your deployment uses CAC authentication, **TLS must be terminated by ACTIVATE**. An external load balancer or reverse proxy cannot terminate TLS on this port, as the platform needs direct access to the client certificate. If your deployment does not use CAC authentication, this port does not need to be exposed.

Internal Cluster Ports {#internal-ports}

These ports are used for communication between services inside the control plane Kubernetes cluster. They should **not** be exposed externally. Kubernetes internal networking handles routing.

Databases

Port	Protocol	Service	Notes
5432	TCP	PostgreSQL	Usage metrics, billing, and activity tracking
27017	TCP	MongoDB (replica set)	User accounts, platform configuration, and metadata

If using managed database services outside the cluster, ensure the control plane can reach the database endpoints on these ports. This may require VNet peering, private endpoints, or firewall rules on the database service.

Message Queue

Port	Protocol	Service	Notes
5672	TCP (AMQP)	RabbitMQ	Inter-service messaging for job orchestration and cluster operations

15672	TCP (HTTP)	RabbitMQ Management	Optional management UI for debugging. Do not expose externally.

Secrets Management

Port	Protocol	Service	Notes
8200	TCP (HTTP)	HashiCorp Vault	Credential storage. Must be initialized and unsealed.

Platform Services

These are internal HTTP services that communicate behind the ingress. They do not need external exposure or special firewall rules. Kubernetes service discovery handles routing.

Port	Service	Purpose
3000	Cloud cacher	Cloud resource caching service
4006	Legacy API	REST API (Node.js). Scheduled for removal in v8.
8888	Internal API	Workspace scaling, cost routes (admin, not externally exposed)

Compute Cluster Agent Connectivity {#agent-connectivity}

Compute cluster agents running on cloud clusters, managed clusters, or existing on-premises clusters communicate with the control plane over a single outbound connection. **No inbound ports need to be opened on the compute cluster side.**

Connection Architecture

<MermaidDiagram chart={` %%{init: {"theme": "neutral", "flowchart": {"padding": 20, "nodeSpacing": 30, "rankSpacing": 60}, "themeVariables": {"edgeLabelBackground": "#f8fafc", "clusterBkg": "transparent"}} }%% flowchart LR user@{ icon: "lucide:user", form: "circle", label: "User", pos: "b" }

```
subgraph controlplane["Control Plane"]
    direction TB
    ingress@{ icon: "lucide:shield", form: "circle", label: "Ingress", pos: "b" }
    api@{ icon: "lucide:server", form: "circle", label: "Platform Services", pos: "b" }
end

subgraph workspace["User Workspaces"]
    ws@{ icon: "lucide:monitor", form: "circle", label: "Workspace", pos: "b" }
end

subgraph compute["Compute Cluster"]
    direction TB
    agent@{ icon: "lucide:satellite-dish", form: "circle", label: "Agent", pos: "b" }
    sshd@{ icon: "lucide:lock", form: "circle", label: "SSH", pos: "b" }
end

user -->|"port 443"| ingress
ingress -->|"port 443"| ws
ingress -->|"port 443 · reverse tunnel"| agent
```

```
classDef cpNode fill:#dbeafe,stroke:#3b82f6,stroke-width:2px,color:#1e3a5f
classDef wsNode fill:#fef3c7,stroke:#f59e0b,stroke-width:2px,color:#78350f
classDef clusterNode fill:#d1fae5,stroke:#10b981,stroke-width:2px,color:#064e3b
classDef userNode fill:#ede9fe,stroke:#8b5cf6,stroke-width:2px,color:#4c1d95

class ingress,api cpNode
class ws wsNode
class agent,sshd clusterNode
class user userNode

style controlplane stroke:#3b82f6,stroke-width:2px
style workspace stroke:#f59e0b,stroke-width:2px
style compute stroke:#10b981,stroke-width:2px
```

`} />

How It Works

1. The agent on the compute cluster initiates an **outbound WebSocket connection** to `wss://<platform-host>/ws/ssh/tunnel` (port 443).
2. The connection is upgraded to a **multiplexed session**, allowing multiple logical streams over the single TCP connection.
3. The control plane can open new streams on this connection to reach the agent (effectively a reverse tunnel) without the agent needing any inbound ports.
4. The agent sends periodic heartbeats with resource metrics over the same HTTPS endpoint.

SSH Access to Clusters (Port 22)

User workspaces and the platform connect to compute clusters over SSH (port 22) for running jobs and interactive sessions. The `pw ssh` command intelligently selects the connection method based on network configuration:

- **Direct (public IP)**: If the cluster node has a public IP, the platform connects directly to port 22.
- **Direct (peered network)**: If the cluster's network is peered with the control plane network, the platform connects directly to the private IP on port 22.
- **Via agent tunnel**: If the cluster is in an isolated network (not peered), SSH traffic is routed through the agent's existing WebSocket tunnel. No additional ports need to be opened since port 22 is reached internally through the tunnel.

This means clusters should have **port 22 open for SSH** from the control plane, unless all SSH traffic is routed through the agent tunnel.

Cloud Clusters

Cloud clusters are provisioned by the platform through a configured cloud account (e.g., AWS, Azure, GCP). The platform provisions compute nodes in the cloud provider, and each node's agent automatically establishes a tunnel connection back to the control plane.

Network requirements from the cloud cluster:

Direction	Destination	Port	Protocol	Purpose
Outbound	Control plane host	443	TCP (WSS/HTTPS)	Agent tunnel, API calls, heartbeats
Inbound	Controller node	22	TCP (SSH)	SSH access from workspaces (if not using tunnel)

Managed Clusters

Managed clusters are batch scheduler clusters (e.g., PBS, Slurm, LSF) registered with the platform. Node agents are installed using a one-time registration token and maintain a persistent tunnel connection to the control plane.

Network requirements from the managed cluster:

Direction	Destination	Port	Protocol	Purpose
Outbound	Control plane host	443	TCP (WSS/HTTPS)	Agent tunnel, API calls, heartbeats
Inbound	Controller node	22	TCP (SSH)	SSH access from workspaces (if not using tunnel)

Existing Clusters (On-Premises)

Existing clusters are user-provided infrastructure that you register with the platform. They use the same outbound-only agent tunnel as cloud and managed clusters.

Network requirements from the existing cluster:

Direction	Destination	Port	Protocol	Purpose
Outbound	Control plane host	443	TCP (WSS/HTTPS)	Agent tunnel, API calls, heartbeats
Inbound	Controller node	22	TCP (SSH)	SSH access from workspaces (if not using tunnel)

> **ON-PREMISES FIREWALL CONFIGURATION**
>
> Since agents only make **outbound** connections to port 443 on the control plane, on-premises clusters behind restrictive firewalls only need a single egress rule allowing HTTPS traffic to the platform hostname. No inbound firewall rules or port forwarding are required on the on-premises network.

Cloud Cluster Internal Ports {#cloud-cluster-ports}

Cloud clusters provisioned by the platform use additional ports for internal communication between the controller node and compute nodes. These ports do **not** need to be exposed to the control plane. They are used within the cluster's own network.

Shared Storage

Port	Protocol	Purpose
2049	TCP (NFS)	NFS exports from controller to compute nodes

Slurm Scheduler

Port	Protocol	Purpose
6817	TCP	Slurmctld (Slurm controller daemon)
6819	TCP	Slurmdbd (Slurm database daemon)
3306	TCP	MySQL for Slurm accounting database
8192–60000	TCP	Slurm srun interactive job step communication

General

Port	Protocol	Purpose
1024–65535	TCP	User workflows and application traffic between controller and compute nodes

> NOTE
>
> These ports are for communication **within** the cloud cluster network. They do not need to be reachable from the control plane or the internet.

User Workspace Ports {#workspace-ports}

User workspaces can run on a dedicated user host, in a separate Kubernetes cluster, or in a dedicated namespace. The following ports are used by workspace containers and accessed by platform services.

Port	Protocol	Purpose
22	TCP (SSH)	SSH access to the workspace
3000	TCP (HTTP)	IDE (code editor)
3001	TCP (HTTP)	Workspace API
3003	TCP (HTTP)	Workspace API
3004	TCP (HTTP)	Workspace API (also used by connected clusters)
1024–65535	TCP	User-started services and application traffic

These ports are internal and do not need external exposure.

Shared Storage (NFS) {#shared-storage}

If using NFS or a shared filesystem for platform storage (see Kubernetes Deployment Guide), the control plane services need access to the NFS endpoint.

Port	Protocol	Direction	Purpose
2049	TCP (NFS)	Control plane → Storage	Shared filesystem for user uploads and workflow assets

Email (SMTP)

If the platform is configured to send email notifications (billing alerts, user invitations), the control plane needs outbound access to your SMTP server.

Direction	Destination	Port	Protocol	Purpose
Outbound	SMTP server	587	TCP (SMTP/STARTTLS)	Email delivery

Summary {#summary}

Minimum Required External Ports

For a standard deployment, only **one port** needs external exposure on the control plane:

Port	Protocol	Purpose
443	TCP	All platform traffic (UI, API, agent tunnels)

Complete Port Reference

Port	Protocol	Location	Direction	Required	Purpose
443	HTTPS/WSS	Control plane LB	Inbound	Yes	UI, API, agent tunnels
80	HTTP	Control plane LB	Inbound	Optional	HTTP redirect
8443	HTTPS (mTLS)	Control plane LB	Inbound	Optional	CAC/Smart Card auth
2049	NFS	Control plane → Storage	Outbound	Yes	Shared filesystem
3000	HTTP	Internal	—	Yes	Cloud cacher
4006	HTTP	Internal	—	Yes	Legacy API (removal planned in v8)
5432	TCP	Internal	—	Yes	PostgreSQL
5672	AMQP	Internal	—	Yes	RabbitMQ
8200	HTTP	Internal	—	Yes	Vault
8888	HTTP	Internal	—	Yes	Internal API
15672	HTTP	Internal	—	Optional	RabbitMQ Management
27017	TCP	Internal	—	Yes	MongoDB
587	SMTP	Outbound	Outbound	Optional	Email notifications
443	WSS/HTTPS	Agent → Control plane	Outbound (from agent)	Yes	Agent tunnel and heartbeats

22	SSH	Compute clusters	Inbound (from control plane)	Conditional	SSH access from workspaces (not needed if using tunnel)
2049	NFS	Cloud cluster internal	—	Yes	NFS exports (controller → compute)
6817	TCP	Cloud cluster internal	—	Conditional	Slurmctld (Slurm clusters)
6819	TCP	Cloud cluster internal	—	Conditional	Slurmdbd (Slurm clusters)
3306	TCP	Cloud cluster internal	—	Conditional	MySQL for Slurm accounting
8192–60000	TCP	Cloud cluster internal	—	Conditional	Slurm srun ports

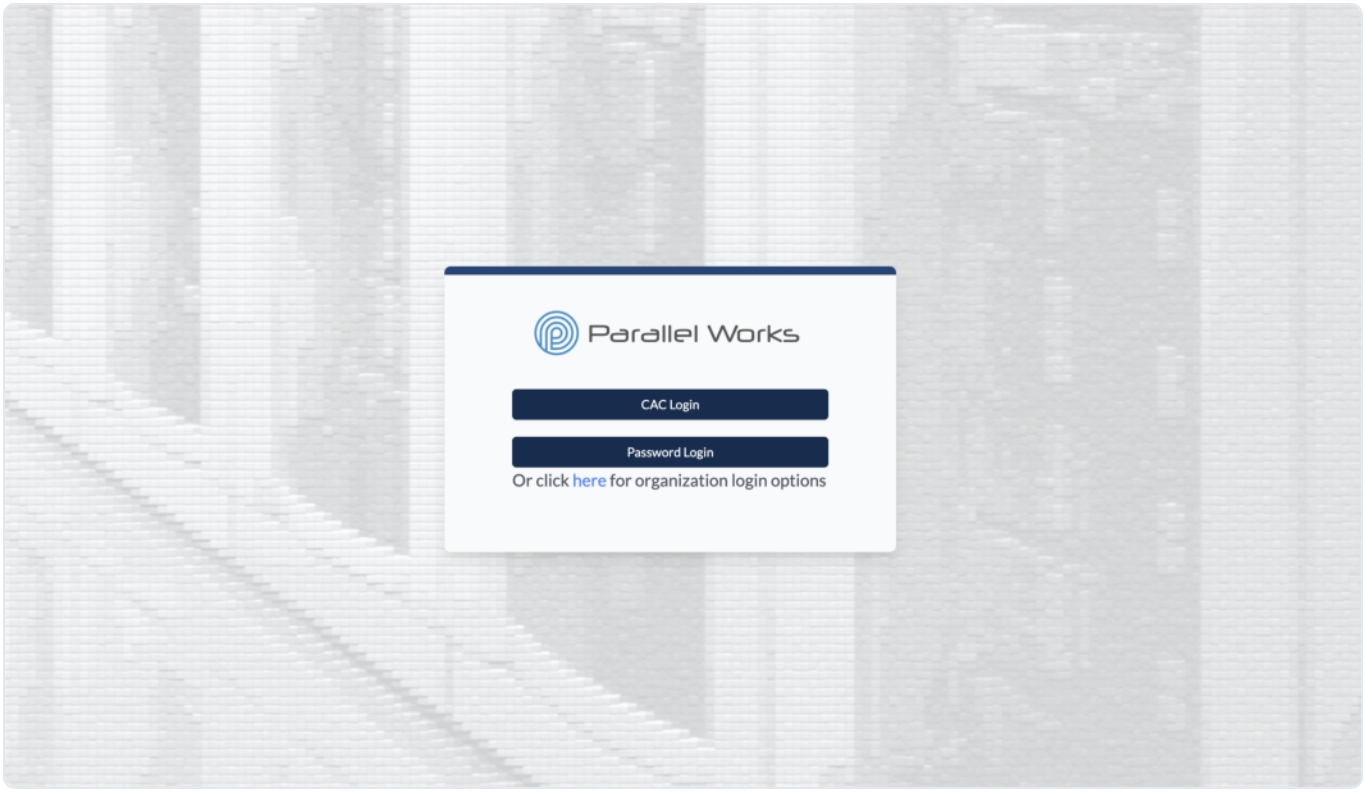
Reference

This section contains reference materials, support information, and compliance documentation.

Contents

Logging In

When you navigate to activate.parallel.works*, you'll see the login page.



Login Options

If your organization requires a Common Access Card (CAC) to verify your identity, click **CAC Login**. Your browser will display a certificate dialog box; select your certificate and click **OK**. Next, your browser will

display a PIN or password dialog box; enter your PIN or password and click **OK**. You'll then be logged in to ACTIVATE.

If you already have an ACTIVATE account, click **Password Login** to enter your username and password.

If you don't have an ACTIVATE account but your organization supports Azure Active Directory (AD), click the link below **Password Login**. Enter the name of your ACTIVATE organization and click **Next**. You'll be redirected to a Microsoft sign-in screen. After you enter your organization email and password, you'll be taken back to ACTIVATE. Enter your **Full Name**, **Username**, and **Email**. Click **Register** to create your account and log in to ACTIVATE.

If you see the error message *Invalid username or password* when logging in, please reach out to us at support@parallelworks.com for assistance.

**Please note that, depending on the organization, some users log in to ACTIVATE at a different URL (e.g., org.parallel.works).*

Sudo Mode

To keep your account secure, ACTIVATE requires you to authenticate using multi-factor authentication (MFA) when performing certain sensitive actions. Once you authenticate, you will enter **sudo mode**.

In **sudo mode**, you'll be able to perform sensitive actions without having to authenticate again. You will remain in **sudo mode** for a set period of 1 hour, after which you will be required to authenticate again to perform sensitive actions.

We consider these actions to be sensitive:

- Resetting your password

IL5/FedRAMP Compliance

In addition to the ACTIVATE SaaS platform, we offer a solution for the US Government: the Parallel Works High-Security Platform (PWHSP). Hosted in AWS GovCloud, the PWHSP is compliant with Federal Risk and Authorization Management Program (FedRAMP) standards.

Features and Services

The PWHSP includes the same major features of ACTIVATE. Some features and services have been modified for security compliance, which are detailed in this section.

At present, the PWHSP only supports connecting to on-premises clusters with an Authority to Operate (ATO) and deploying compute resources on AWS. We will be adding support for Microsoft Azure and Google Cloud Platform compute resources in the future. All AWS options are available on the PWHSP, but screenshots and page sections that mention other cloud service providers can be disregarded until further notice.

SSH Access

The PWHSP does not support SSH access from your personal computer to cloud resources. All of your work with compute resources must be conducted in your user workspace on the platform. SSH can be federated

through the PWHSP using the PW CLI, see the PW CLI documentation for more information.

Authentication Methods

MFA Requirements

On the PWHSP, users are required to have MFA enabled on their account or use an authentication method which ensures MFA, e.g. CAC, an OIDC provider with MFA.

Password Authentication

For users using the password authentication method, we support adding a YubiKey 5 FIPS model. YubiKeys plug into your computer via USB-A or USB-C and require physical touch after you enter your password. These YubiKeys are provisioned and issued by Parallel Works.

CAC Authentication

Common Access Cards (CAC) are the standard form of identification for government employees, personnel, and service members. If you use a CAC, your administrator will register your CAC with your PWHSP account before you access the PWHSP for the first time. You'll need your CAC PIN in addition to your CAC.

OpenID Connect

OpenID Connect can use either a YubiKey or a CAC for second-step validation. With OpenID connect, the PWHSP login page is provided by the government instead of Parallel Works. The YubiKey in this case will be provisioned and issued by the government.

FIPS 140-2 Compliance

The Federal Information Processing Standards (FIPS) are globally recognized guidelines for information security. The PWHSP adheres to FIPS 140-2.

CSP FIPS Endpoints

When accessing Cloud Service Provider (CSP) services from within the PWHSP, you must utilize endpoints which utilize FIPS. Below are instructions for configuring common cloud SDKs to use FIPS endpoints.

AWS

For AWS SDKs, you can enforce the use of FIPS endpoints by setting the following environment variable:

- `AWS_USE_FIPS_ENDPOINT=true`

Azure

For Azure, FIPS compliance involves using the Azure Government cloud. Ensure your authentication uses the correct authority host:

- `AZURE_AUTHORITY_HOST=https://login.microsoftonline.us`

Google Cloud Platform

For Google Cloud, FIPS compliance is generally handled at the cryptographic library level rather than specific FIPS endpoints. Ensure your application is built and run using FIPS-validated cryptographic modules.

Ports

The PWHSP can only be accessed through port 443. Port 80 is open for redirect to port 443 only. Port 8443 is open for CAC authentication. All data is encrypted at rest and in transit using validated FIPS 140-2 cryptographic modules.

Getting Support

If you're experiencing problems with ACTIVATE, we're here to help!

You can check the platform status at any time by clicking **Status** on the login page or in the bottom-left corner after you log in.

You can submit a help ticket by emailing us at support@parallelworks.com.

CLI Reference

The complete command-line reference for the PW CLI. Each entry lists the command's synopsis, options, and examples.

About the PW CLI

The Parallel Works command-line interface (PW CLI) offers a streamlined way to interract with your compute and storage resources from a terminal.

Installation

By default, the PW CLI is pre-installed on user workspaces, cloud clusters, and existing clusters. You can open a terminal on the ACTIVATE platform and immediately use any PW CLI command.

For existing clusters, the PW CLI is installed in a folder named `pw` inside your home directory.

Install Script (Recommended)

The easiest way to install the PW CLI is with the install script, which automatically detects your operating system and architecture. This works on Linux and macOS:

```
curl -fsSL https://activate.parallel.works/cli/install.sh | bash
```

By default, the CLI is installed to `~/.local/bin`. You can specify a custom install location with the `--to` flag:

```
curl -fsSL https://activate.parallel.works/cli/install.sh | bash -s -- --to /usr/local/bin
```

> **PATH CONFIGURATION**
>
> If `~/.local/bin` is not already on your PATH, the installer will provide instructions for adding it.

Manual Installation

If you prefer to download the binary manually, or are on Windows, follow the instructions below for your operating system and device's chip type.

Linux, AMD

```
mkdir -p ~/bin
wget "https://activate.parallel.works/cli/pw-linux-amd64" -O ~/bin/pw
```

Linux, ARM

```
mkdir -p ~/bin
wget "https://activate.parallel.works/cli/pw-linux-arm64" -O ~/bin/pw
```

macOS, Apple Silicon

```
mkdir -p ~/bin
wget "https://activate.parallel.works/cli/pw-darwin-arm64" -O ~/bin/pw
echo 'export PATH="$PATH:$HOME/bin"' >> ~/.zshrc # adjust if not using zsh
```

macOS, Intel

```
mkdir -p ~/bin
wget "https://activate.parallel.works/cli/pw-darwin-amd64" -O ~/bin/pw
echo 'export PATH="$PATH:$HOME/bin"' >> ~/.zshrc # adjust if not using zsh
```

Windows, AMD

```
$dest = "$env:LOCALAPPDATA\Programs\pw"
New-Item -ItemType Directory -Force -Path $dest
curl.exe -L "https://activate.parallel.works/cli/pw-windows-amd64.exe" -o "$dest\pw.exe"
[Environment]::SetEnvironmentVariable("Path", $env:Path + ";$dest",
[EnvironmentVariableTarget]::User)
```

Windows, ARM

```
$dest = "$env:LOCALAPPDATA\Programs\pw"
New-Item -ItemType Directory -Force -Path $dest
curl.exe -L "https://activate.parallel.works/cli/pw-windows-arm64.exe" -o "$dest\pw.exe"
[Environment]::SetEnvironmentVariable("Path", $env:Path + ";$dest",
[EnvironmentVariableTarget]::User)
```

Authentication

User workspaces and existing clusters are pre-authenticated. You can access the PW CLI and immediately interact with your compute and storage resources.

For cloud clusters, you must first authenticate either with an API key or a token.

API Key

First, create an API key by following these instructions in our user guide.

Next, use the `pw auth apikey` command by following these instructions.

Token

First, create a token by following these instructions in our user guide.

Next, use the `pw auth token` command by following these instructions.

About URIs

When using the CLI to interact with buckets, you will see them identified by a Uniform Resource Identifier (URI).

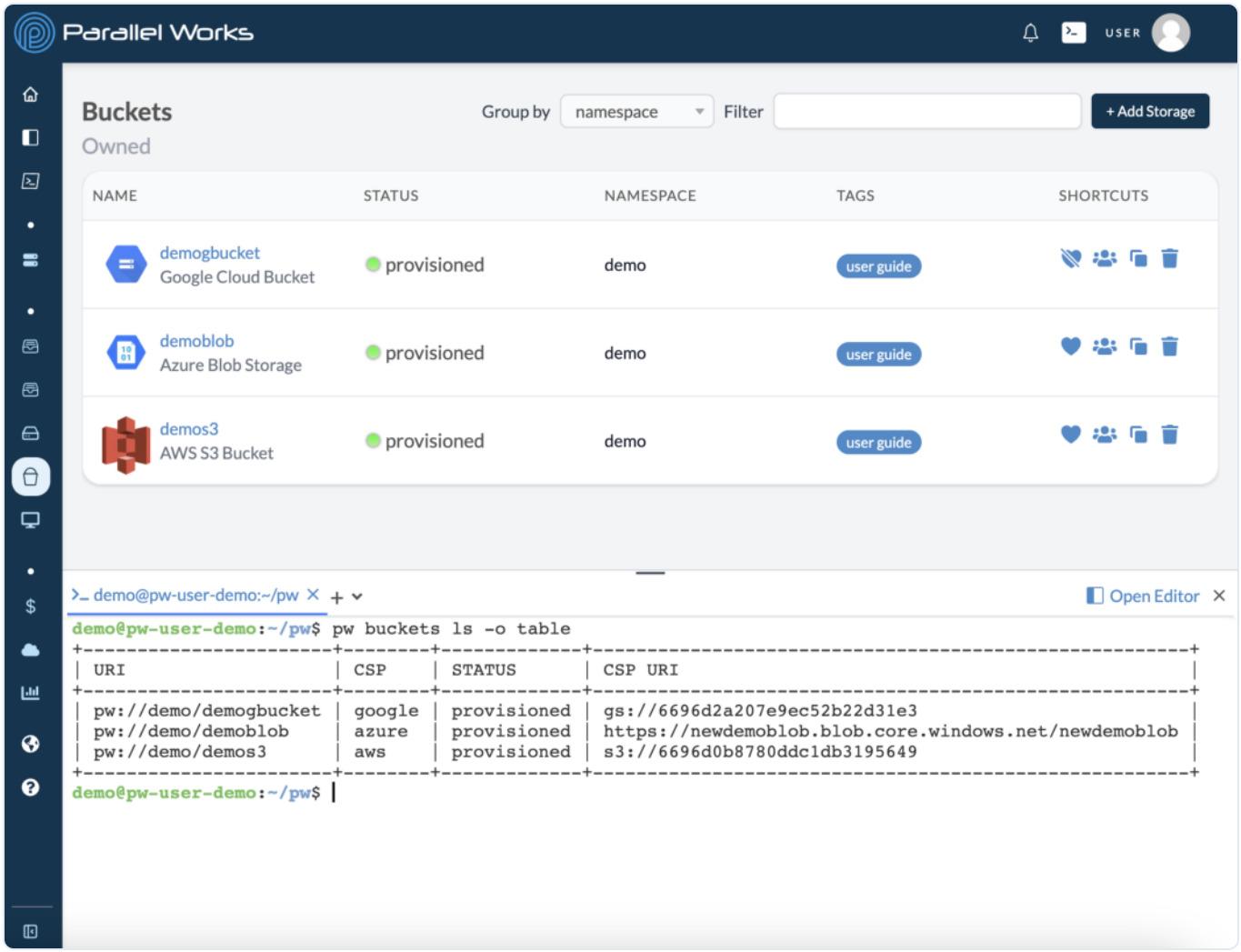
URIs function similarly to URLs. In fact, URLs are a type of URI. URIs, however, do not have to point to a location on a network; they only serve to identify a resource.

In the screenshot below, we used the PW CLI to list our buckets in a table format. Each bucket has a URI that's exclusive to the ACTIVATE platform, which takes the form of `pw://namespace/bucket`.

There is also a URI that's exclusive to a bucket's CSP, such as:

- `gs://google-bucket-name`
- `s3://aws-bucket-name`
- `https://azure-bucket-name.blob.core.windows.net/azure-bucket-name`

Buckets will only display a CSP URI if they have already been provisioned.



A screenshot of provisioned buckets, displayed in a terminal as a table

pw

pw

The Parallel Works ACTIVATE CLI

Synopsis

pw is the base for all other PW CLI commands.

Run **pw** alone to check that the PW CLI is installed and to list compatible commands.

Options

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
  -h, --help                   help for pw
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai - Manage AI providers, models, and chats
- pw auth - Authenticate the CLI
- pw buckets - Manage cloud buckets
- pw cluster - Manage clusters
- pw context - Manage CLI contexts
- pw forward - Forward remote ports to your local machine via SSH tunnels
- pw kube - Kubernetes commands
- pw lustre - Manage Lustre filesystems
- pw nfs - Manage NFS storages
- pw orgs - Manage organizations
- pw sessions - Manage sessions
- pw ssh - Initiate an SSH connection to a resource
- pw ssh-public-keys - Get the SSH public keys for a user
- pw upgrade - Update the version of the CLI
- pw users - Manage organization users
- pw vscode - Open VS Code connected to a remote resource
- pw workflows - Manage workflows

pw ai

pw ai

Manage AI providers, models, and chats

Synopsis

The **pw ai** command group has commands for managing AI providers, chatting with models, and viewing chats.

Options

```
  -h, --help   help for ai
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw ai chats - Manage AI chats
- pw ai models - Manage AI chat models
- pw ai providers - Manage AI chat providers

pw ai chats

pw ai chats

Manage AI chats

Synopsis

The **pw ai chats** command group has commands for starting, resuming, listing, viewing, and deleting AI chats.

Options

```
  -h, --help   help for chats
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai - Manage AI providers, models, and chats

- pw ai chats delete - Delete an AI chat
- pw ai chats get - Get an AI chat
- pw ai chats ls - List AI chats
- pw ai chats new - Start a new AI chat
- pw ai chats resume - Resume an existing AI chat

pw ai chats delete

pw ai chats delete

Delete an AI chat

Synopsis

The **pw ai chats delete** command deletes a chat.

```
pw ai chats delete <id> [flags]
```

Examples

```
  # Delete a chat
  pw ai chats delete abc-123
```

Options

```
  -h, --help   help for delete
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai chats - Manage AI chats

pw ai chats get

pw ai chats get

Get an AI chat

Synopsis

The **pw ai chats get** command retrieves a chat and displays its messages.

```
pw ai chats get <id> [flags]
```

Examples

```
  # View a chat
  pw ai chats get abc-123

  # View as JSON
  pw ai chats get -o json abc-123
```

Options

```
  -h, --help            help for get
  -o, --output string   Output format (text, json) (default "text")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai chats - Manage AI chats

pw ai chats ls

pw ai chats ls

List AI chats

Synopsis

The **pw ai chats ls** command lists your AI chats.

```
pw ai chats ls [flags]
```

Examples

```
  # List chats
  pw ai chats ls

  # List as JSON
  pw ai chats ls -o json

  # List with pagination
  pw ai chats ls --limit 10 --offset 20
```

Options

```
  -h, --help            help for ls
      --limit int       Maximum number of chats to return (default 20)
      --offset int      Number of chats to skip
  -o, --output string   Output format (table, json) (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai chats - Manage AI chats

pw ai chats new

pw ai chats new

Start a new AI chat

Synopsis

The **pw ai chats new** command starts a new chat with an AI model.

If no model is specified, you will be prompted to select one interactively. In non-interactive mode (with --prompt), a model must be specified.

```
pw ai chats new [model] [flags]
```

Examples

```
  # Interactive chat (select model interactively)
  pw ai chats new

  # Interactive chat with a specific model
  pw ai chats new owner:provider/model-name

  # Single prompt (non-interactive, model required)
  pw ai chats new -p "What is Go?" owner:provider/model-name

  # Use an org provider with a budget allocation
  pw ai chats new --allocation my-alloc org:provider/model-name

  # Non-interactive with org provider (allocation required)
  pw ai chats new --allocation my-alloc -p "Hello" org:provider/model-name
```

Options

```
      --allocation string   Budget allocation name (required for org provider models)
  -h, --help                help for new
  -p, --prompt string       Send a single prompt and exit
      --save                Persist the chat (only for non-interactive mode with --prompt)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai chats - Manage AI chats

pw ai chats resume

pw ai chats resume

Resume an existing AI chat

Synopsis

The **pw ai chats resume** command continues an existing chat.

The model is automatically detected from the conversation. You can override it with --model.

```
pw ai chats resume <id> [flags]
```

Examples

```
  # Resume a chat (model auto-detected)
  pw ai chats resume abc-123

  # Resume with a specific model
  pw ai chats resume -m owner:provider/model-name abc-123

  # Send a single prompt to an existing chat
  pw ai chats resume -p "Summarize our conversation" abc-123
```

Options

```
      --allocation string   Budget allocation name (required for org provider models)
  -h, --help                help for resume
  -m, --model string        Override the model (default: auto-detected from chat)
  -p, --prompt string       Send a single prompt and exit
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai chats - Manage AI chats

pw ai models

pw ai models

Manage AI chat models

Synopsis

The **pw ai models** command group has commands for listing available AI chat models.

Options

```
  -h, --help   help for models
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai - Manage AI providers, models, and chats
- pw ai models ls - List available AI chat models

pw ai models ls

pw ai models ls

List available AI chat models

Synopsis

The **pw ai models ls** command lists all AI chat models available across all configured providers.

```
pw ai models ls [flags]
```

Examples

```
  # List all models
  pw ai models ls

  # List as JSON
  pw ai models ls -o json
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (table, json) (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai models - Manage AI chat models

pw ai providers

pw ai providers

Manage AI chat providers

Synopsis

The **pw ai providers** command group has commands for managing AI chat providers.

Options

```
  -h, --help   help for providers
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai - Manage AI providers, models, and chats
- pw ai providers create - Create an AI provider

- pw ai providers delete - Delete an AI provider
- pw ai providers get - Get details of an AI provider
- pw ai providers ls - List AI providers
- pw ai providers models - List models available for a provider

pw ai providers create

pw ai providers create

Create an AI provider

Synopsis

The **pw ai providers create** command creates a new AI provider.

For **custom** providers, --endpoint and --api-key are required. For **azure** providers, --region, --model, --group, and --network are required.

```
pw ai providers create [flags]
```

Examples

```
  # Create a custom provider
  pw ai providers create --name my-provider --csp custom --endpoint https://api.example.com --api-
key sk-xxx

  # Create an Azure provider
  pw ai providers create --name my-azure --csp azure --region eastus --model gpt-4 --group my-
group --network my-network
```

Options

```
      --api-key string    API key (required for custom)
      --csp string        Cloud service provider (required, e.g. azure, custom)
      --endpoint string   API endpoint (required for custom)
      --group string      Resource group
  -h, --help              help for create
      --model string      Model name (required for azure)
      --name string       Provider name (required)
      --network string    Network
      --region string     Region (required for azure)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai providers - Manage AI chat providers

pw ai providers delete

pw ai providers delete

Delete an AI provider

Synopsis

The **pw ai providers delete** command deletes an AI provider.

You can specify a provider by name (defaults to your user) or by full URI (pw://user/name).

```
pw ai providers delete <provider> [flags]
```

Examples

```
  # Delete your own provider
  pw ai providers delete my-provider

  # Delete another user's provider
  pw ai providers delete pw://other-user/their-provider
```

Options

```
  -h, --help   help for delete
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai providers - Manage AI chat providers

pw ai providers get

pw ai providers get

Get details of an AI provider

Synopsis

The **pw ai providers get** command retrieves details of a specific AI provider.

You can specify a provider by name (defaults to your user) or by full URI (pw://user/name).

```
pw ai providers get <provider> [flags]
```

Examples

```
  # Get your own provider
  pw ai providers get my-provider

  # Get another user's provider
  pw ai providers get pw://other-user/their-provider

  # Get as JSON
  pw ai providers get -o json my-provider
```

Options

```
  -h, --help            help for get
  -o, --output string   Output format (table, json) (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai providers - Manage AI chat providers

pw ai providers ls

pw ai providers ls

List AI providers

Synopsis

The **pw ai providers ls** command lists all AI providers configured for the current user.

```
pw ai providers ls [flags]
```

Examples

```
  # List all providers
  pw ai providers ls

  # List as JSON
  pw ai providers ls -o json
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (table, json) (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai providers - Manage AI chat providers

pw ai providers models

pw ai providers models

List models available for a provider

Synopsis

The **pw ai providers models** command lists the models available for a specific AI provider.

You can specify a provider by name (defaults to your user) or by full URI (pw://user/name).

```
pw ai providers models <provider> [flags]
```

Examples

```
  # List models for your own provider
  pw ai providers models my-provider

  # List models for another user's provider
  pw ai providers models pw://other-user/their-provider
```

Options

```
  -h, --help   help for models
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw ai providers - Manage AI chat providers

pw auth

pw auth

Authenticate the CLI

Synopsis

The **pw auth** command group lets you grant authorization to the PW CLI to access your cloud resources.

On ACTIVATE, the PW CLI is pre-authenticated for user workspaces and existing clusters. You only have to authenticate if you're accessing a cloud cluster or running the PW CLI locally.

For more information about this process, please see Authentication.

Options

```
  -h, --help   help for auth
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw auth apikey - Authenticate the CLI with an API key
- pw auth identify-platform - Print effective platform host
- pw auth token - Authenticate the CLI with a short-term token
- pw auth whoami - Print effective user

pw auth apikey

pw auth apikey

Authenticate the CLI with an API key

Synopsis

After you enter **pw auth apikey**, the CLI will prompt you to paste your API key on a new line.

You can create a new API key on ACTIVATE by going to **Account** > **Authentication** > **API Keys**. For more information about these steps, please see this page.

When creating an API key, you can choose whether your key will expire in 7, 30, 60, or 90 days. Shorter expiration periods are more secure.

Alternatively, you can authenticate with a short-term token. Please see **pw auth token** for more information.

Use **--context-name** to specify a custom name for the context. This is useful if you want multiple contexts for the same user/server combination (e.g., different API keys for different purposes).

```
pw auth apikey [flags]
```

Options

```
      --context-name string   Name for the context (defaults to user@server)
  -h, --help                  help for apikey
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw auth - Authenticate the CLI

pw auth identify-platform

pw auth identify-platform

Print effective platform host

Synopsis

Use **pw auth identify-platform** to verify which platform CLI commands will be executed against.

Prints the currently targeted platform host.

```
pw auth identify-platform [flags]
```

Options

```
  -h, --help   help for identify-platform
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
```

```
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw auth - Authenticate the CLI

pw auth token

pw auth token

Authenticate the CLI with a short-term token

Synopsis

After you enter **pw auth token**, the CLI will prompt you to paste your short-term token on a new line.

You can create a new token on ACTIVATE by going to **Account** > **Authentication** > **API Keys**. For more information about these steps, please see this page.

These short-term tokens last a maximum of 24 hours, after which you will need to re-authenticate.

Alternatively, you can authenticate with an API key. Please see **pw auth apikey** for more information.

Use **--context-name** to specify a custom name for the context. This is useful if you want multiple contexts for the same user/server combination.

```
pw auth token [flags]
```

Options

```
      --context-name string   Name for the context (defaults to user@server)
  -h, --help                  help for token
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw auth - Authenticate the CLI

pw auth whoami

pw auth whoami

Print effective user

Synopsis

Use **pw auth whoami** to verify your authentication.

Prints the currently authenticated identity. Typically, this will be the username of the account used to authenticate. However, it can also be other values, such as `cluster-session:demo/democluster/1`.

```
pw auth whoami [flags]
```

Options

```
  -h, --help   help for whoami
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw auth - Authenticate the CLI

pw buckets

pw buckets

Manage cloud buckets

Synopsis

The **buckets** command group lets you create, delete, list, and copy files to your cloud buckets.

Options

```
  -h, --help   help for buckets
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw buckets cp - Copy files between local and buckets

- pw buckets get-token - Print temporary access credentials for a cloud bucket
- pw buckets ls - List buckets or files
- pw buckets rm - Remove files from buckets

pw buckets cp

pw buckets cp

Copy files between local and buckets

Synopsis

The **pw buckets cp** command must be followed by a source and a destination. The source and destination can be either a local file or a bucket URI.

Both PW URIs and CSP URIs are accepted. For more information about URIs, please see this section.

When copying to/from a bucket with a PW URI, you must specify at least the bucket name. However, paths can be longer, such as `pw://[namespace]/[bucket-name]/[folder]/[folder2]/[folder3]/[file-name]`.

```
pw buckets cp [source] [destination] [flags]
```

Examples

```
  # Upload a local file named sample.txt to a bucket named demos3bucket in the demo namespace
  pw buckets cp sample.txt pw://demo/demos3bucket

  # Download a file named sample.txt to a local file
  pw buckets cp pw://demo/demos3bucket/sample.txt sample2.txt

  # Download a file from Azure Blob Storage to a local folder, using the Azure URI
  pw buckets cp https://azuredemo.blob.core.windows.net/azuredemo/sample.txt sample2.txt

  # Recursive download
  # Result: ./sample-dir/a.txt, ./sample-dir/sub/b.txt, ...
  pw buckets cp -r pw://demo/demos3bucket/sample-dir ./sample-dir

  # Recursive download of the entire bucket root into a local folder
  # Result: ./downloads/<all objects from bucket root...>
  pw buckets cp -r pw://demo/demos3bucket ./downloads/
```

Options

```
  -h, --help        help for cp
  -r, --recursive   Copy directories/prefixes recursively
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
```

```
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw buckets - Manage cloud buckets

pw buckets get-token

pw buckets get-token

Print temporary access credentials for a cloud bucket

Synopsis

The **pw buckets get-token** command must be followed by a bucket URI.

Both PW URIs and CSP URIs are accepted. For more information about URIs, please see this section.

The short-term tokens that this command prints are the same tokens seen here.

```
pw buckets get-token [bucketURI] [flags]
```

Options

```
  -h, --help   help for get-token
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw buckets - Manage cloud buckets

pw buckets ls

pw buckets ls

List buckets or files

Synopsis

The **pw buckets ls** command can be used with or without arguments.

When no arguments are provided, all buckets are listed.

If you provide a namespace, all buckets in that namespace will be listed. If you provide a namespace and bucket name, the files in the bucket will be listed.

You can list individual directories in the bucket by providing the full path to the directory.

This command also supports using the same CSP URI format accepted by CSP CLIs. For more information about URIs, please see this section.

```
pw buckets ls [bucketURI] [flags]
```

Examples

```
  # List all buckets
  pw buckets ls

  # List all buckets in a namespace
  pw buckets ls pw://[namespace]

  # List files in a bucket
  pw buckets ls pw://[namespace]/[bucket]

  # List files in a bucket, using the AWS URI format
  pw buckets ls s3://[aws-bucket-name]

  # List files in a bucket, using the Google URI format
  pw buckets ls gs://[google-bucket-name]

  # List files in a bucket, using the Azure URI format
  pw buckets ls https://[azure-bucket-name].blob.core.windows.net/[azure-bucket-name]

  # List all buckets you have access to, displayed as a table
  # Tables display URI, CSP, provisioning status, and CSP URI
  pw buckets ls -o table

  # List currently provisioned buckets with csp URIs
  pw buckets ls --csp-uris
```

Options

```
      --csp-uris               Show CSP URIs
  -h, --help                   help for ls
  -o, --output OUTPUT_FORMAT   Output format. One of: list, json, table (default table)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw buckets - Manage cloud buckets

pw buckets rm

pw buckets rm

Remove files from buckets

Synopsis

The **pw buckets rm** command must be followed by a bucket URI and file name.

Both PW URIs and CSP URIs are accepted. For more information about URIs, please see this section.

```
pw buckets rm [bucket URI]/[file] [flags]
```

Examples

```
  # Delete a file named sample.txt from a bucket named demos3bucket
  pw buckets rm pw://demo/demos3bucket/sample.txt
```

Options

```
  -h, --help   help for rm
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw buckets - Manage cloud buckets

pw cluster

pw cluster

Manage clusters

Options

```
  -h, --help   help for cluster
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
```

```
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw cluster attach-storage - Attach storage to a cluster
- pw cluster ls - List clusters

pw cluster attach-storage

pw cluster attach-storage

Attach storage to a cluster

Synopsis

The **attach-storage** command attaches storage to a cluster at the specified mount point.

```
pw cluster attach-storage [flags]
```

Examples

```
  # Attach storage to a cluster
  pw cluster attach-storage --cluster-uri pw://namespace/cluster-name --storage-uri
pw://namespace/storage-name --mount-point=/mnt-point
```

Options

```
      --cluster-uri string   URI of the cluster (required)
  -h, --help                 help for attach-storage
      --mount-point string   Mount point for the storage (required)
      --storage-uri string   URI of the storage to attach (required)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw cluster - Manage clusters

pw cluster ls

pw cluster ls

List clusters

```
pw cluster ls [URI] [flags]
```

Examples

```
  # List all clusters
  pw cluster ls

  # List clusters owned by the user "Jake.Thayne"
  pw cluster ls pw://Jake.Thayne

  # List a specific cluster by URI
  pw cluster ls pw://Jake.Thayne/my-cluster

  # List only clusters you own
  pw cluster ls --owned

  # List only provisioned clusters
  pw cluster ls --status=active

  # List all clusters, displayed as a table
  # Tables display URI, Provision status, and Type
  pw cluster ls -o table
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (json, table)
      --owned           Show only clusters owned by the authenticated user
      --status string   Filter clusters by status (e.g., active, off, failed). 'on' is accepted as
an alias for 'active'.
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw cluster - Manage clusters

pw context

pw context

Manage CLI contexts

Synopsis

The **pw context** command group lets you manage multiple CLI identities and switch between them.

Contexts allow you to authenticate with multiple Parallel Works platforms or user accounts and easily switch between them without re-authenticating.

Each context stores the authentication credentials (API key or token), the platform server, and the associated organization.

Options

```
  -h, --help   help for context
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw context current - Show the current context
- pw context delete - Delete a context
- pw context list - List all available contexts
- pw context rename - Rename a context
- pw context use - Switch to a different context

pw context current

pw context current

Show the current context

Synopsis

The **pw context current** command displays the name of the current context.

```
pw context current [flags]
```

Examples

```
  # Show the current context
  pw context current

  # Using alias
  pw ctx current
```

Options

```
  -h, --help   help for current
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw context - Manage CLI contexts

pw context delete

pw context delete

Delete a context

Synopsis

The **pw context delete** command removes a context from the credentials file.

If the context being deleted is the current one, you must use the --force flag. When deleting the current context with --force, another available context will be selected as the new current context automatically.

```
pw context delete <context-name> [flags]
```

Examples

```
  # Delete a context
  pw context delete user@staging.parallel.works

  # Force delete the current context
  pw context delete --force user@activate.parallel.works

  # Using alias
  pw ctx delete old-context
```

Options

```
  -f, --force   Force delete even if this is the current context
  -h, --help    help for delete
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
```

```
--platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any other
platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw context - Manage CLI contexts

pw context list

pw context list

List all available contexts

Synopsis

The **pw context list** command displays all available contexts.

The current context is marked with an asterisk (*).

```
pw context list [flags]
```

Examples

```
  # List all contexts (simple list)
  pw context list

  # List all contexts as a table
  pw context list -o table

  # List all contexts as JSON
  pw context list -o json

  # Using alias
  pw ctx list
```

Options

```
  -h, --help                   help for list
  -o, --output OUTPUT_FORMAT   Output format. One of: list, json, table (default table)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw context - Manage CLI contexts

pw context rename

pw context rename

Rename a context

Synopsis

The **pw context rename** command renames an existing context.

If the renamed context is the current one, the current context reference is updated automatically.

```
pw context rename <old-name> <new-name> [flags]
```

Examples

```
  # Rename a context
  pw context rename user@activate.parallel.works production

  # Using alias
  pw ctx rename old-context new-context
```

Options

```
  -h, --help   help for rename
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw context - Manage CLI contexts

pw context use

pw context use

Switch to a different context

Synopsis

The **pw context use** command switches the current context to the specified one.

After switching, all subsequent commands will use the credentials from the selected context.

```
pw context use <context-name> [flags]
```

Examples

```
  # Switch to a different context
  pw context use user@activate.parallel.works

  # Using alias
  pw ctx use user@staging.parallel.works
```

Options

```
  -h, --help   help for use
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw context - Manage CLI contexts

pw forward

pw forward

Forward remote ports to your local machine via SSH tunnels

Synopsis

The **pw forward** command creates SSH tunnels that make remote ports accessible on your local machine. This uses the same mechanism as the -L option in OpenSSH.

Unlike **pw ssh -L**, this command automatically reconnects if the SSH connection is interrupted. Local listeners stay open across reconnects, so client programs only see a brief interruption rather than a closed port. This makes it ideal for long-lived tunnels.

Resource names can be specified in the following formats:

- **pw://username/name** - full format with explicit username
- **pw://name** - managed cluster by name in your organization
- **name** - short format where username defaults to your username

```
pw forward -L [bind_address:]port:host:hostport [-L ...] <resource> [flags]
```

Examples

```
  # Make remote port 8080 accessible at localhost:8080
  pw forward -L 8080:localhost:8080 my-cluster

  # Forward multiple remote ports
  pw forward -L 5900:localhost:5901 -L 3000:localhost:3000 my-cluster

  # Bind to all interfaces (accessible from other machines on your network)
  pw forward -L 0.0.0.0:8080:localhost:8080 my-cluster

  # Forward a port from your user workspace
  pw forward -L 8080:localhost:8080 workspace
```

Options

```
  -h, --help                help for forward
      --keepalive int       Interval in seconds between keepalive messages (0 to disable) (default
30)
  -L, --local stringArray   Forward local [bind_address:]port to remote host:hostport (required,
repeatable)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI

pw kube

pw kube

Kubernetes commands

Synopsis

The **pw kube** command group contains commands to interact with Kubernetes resources.

Options

```
  -h, --help   help for kube
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
```

```
--platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any other
platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw kube auth - Set up cluster configuration
- pw kube ls - List all kubernetes clusters
- pw kube token - Token to authenticate with the Kubernetes API

pw kube auth

pw kube auth

Set up cluster configuration

Synopsis

The **pw kube auth** command must be followed by a cluster name.

```
pw kube auth [cluster name] [flags]
```

Examples

```
  # Sets up kubectl context for  **demo** cluster
  pw kube auth demo
```

Options

```
  -h, --help                help for auth
      --no-context-switch   Skip switching to the new context after configuration
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw kube - Kubernetes commands

pw kube ls

pw kube ls

List all kubernetes clusters

```
pw kube ls [flags]
```

Examples

```
  # List all kubernetes clusters
  pw kube ls

  # List all kubernetes clusters, displayed as json
  pw kube ls -o json

  # List all kubernetes clusters, displayed as a table
  # Tables display Name, total vCPUs, total Memory
  pw kube ls -o table
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (json, table)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw kube - Kubernetes commands

pw kube token

pw kube token

Token to authenticate with the Kubernetes API

Synopsis

The **pw kube token** command must be followed by a cluster name. Use this command to get a token to authenticate with the Kubernetes API.

```
pw kube token [clientId] [flags]
```

Examples

```
  # Get a token with the cluster name **demo**
  pw kube token demo
```

Options

```
  -h, --help   help for token
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw kube - Kubernetes commands

pw lustre

pw lustre

Manage Lustre filesystems

Synopsis

The **Lustre** command group lets you create, delete, list, and copy files to your Lustre filesystems.

Options

```
  -h, --help   help for lustre
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw lustre ls - List all Lustre filesystems

pw lustre ls

pw lustre ls

List all Lustre filesystems

```
pw lustre ls [flags]
```

Examples

```
  # List all Lustre filesystems
  pw lustre ls

  # List all Lustre filesystems, displayed as a table
  # Tables display URI, CSP, provisioning status, and CSP URI
  pw lustre ls -o table
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format. One of: table, json, list (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw lustre - Manage Lustre filesystems

pw nfs

pw nfs

Manage NFS storages

Synopsis

The **NFS** command group lets you manage your NFS storages.

Options

```
  -h, --help   help for nfs
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI

- pw nfs ls - List all NFS Storages

pw nfs ls

pw nfs ls

List all NFS Storages

```
pw nfs ls [flags]
```

Examples

```
  # List all NFS Storages
  pw nfs ls

  # List all NFS Storages, displayed as a table
  # Tables display URI, CSP, provisioning status, type, region, and tags
  pw nfs ls -o table
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format. One of: table, json, list (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw nfs - Manage NFS storages

pw orgs

pw orgs

Manage organizations

Options

```
  -h, --help   help for orgs
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
```

```
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw orgs delete - Delete an organization
- pw orgs ls - List organizations

pw orgs delete

pw orgs delete

Delete an organization

Synopsis

The **pw orgs delete** command deletes an organization and all its users, groups, and settings.

This action is irreversible. You will be prompted to type the organization name to confirm. Use **-f** to skip the confirmation prompt.

```
pw orgs delete <organization> [flags]
```

Examples

```
  # Delete an organization
  pw orgs delete my-org

  # Skip confirmation
  pw orgs delete -f my-org
```

Options

```
  -f, --force   Skip confirmation prompt
  -h, --help    help for delete
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw orgs - Manage organizations

pw orgs ls

pw orgs ls

List organizations

Synopsis

The **pw orgs ls** command lists all organizations accessible to you.

```
pw orgs ls [flags]
```

Examples

```
  # List all organizations
  pw orgs ls

  # List as JSON
  pw orgs ls -o json

  # List names only
  pw orgs ls -o list
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (table, json, list) (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw orgs - Manage organizations

pw sessions

pw sessions

Manage sessions

Options

```
  -h, --help   help for sessions
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw sessions connect - Connect to a running session
- pw sessions create - Create a new session on a remote resource
- pw sessions ls - List all sessions
- pw sessions open - Open a session in the browser
- pw sessions stop - Stop and delete a session

pw sessions connect

pw sessions connect

Connect to a running session

Synopsis

Connect to a running session by name. The session must already be running.

For desktop sessions, this creates a local TCP listener that proxies traffic to the session's WebSocket endpoint.

For tunnel sessions, this creates an SSH port forward to the session's remote host and port.

VS Code sessions are not supported by connect. Use **pw session open** to open in the browser or **pw vscode** for local VS Code.

Press Ctrl+C to disconnect. This does not stop the session.

```
pw sessions connect <session-name> [flags]
```

Examples

```
  # Connect to a desktop session
  pw session connect my-desktop

  # Connect to a tunnel session on a specific local port
  pw session connect --port 8080 my-tunnel
```

Options

```
  -h, --help       help for connect
```

```
--port int   Local port to use (0 = auto-select)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw sessions - Manage sessions

pw sessions create

pw sessions create

Create a new session on a remote resource

Synopsis

The **pw session create** command creates a new session on a remote resource.

Supported session types:

- **desktop** - VNC desktop session
- **vscode** - VS Code server session
- **tunnel** - TCP port tunnel (requires --remote-port)

By default, the command returns immediately after creating the session. Use **--wait** to wait for the session to be running before exiting. **--open** and **--connect** imply **--wait**.

Resource names can be specified in the following formats:

- **workspace** or **user-workspace** - your user workspace
- **pw://username/name** - full format with explicit username
- **pw://name** - managed cluster by name in your organization
- **name** - short format where username defaults to your username

```
pw sessions create --type TYPE <resource> [flags]
```

Examples

```
  # Create a desktop session on your user workspace
  pw session create --type desktop workspace

  # Create a desktop session on a resource
  pw session create --type desktop my-cluster
```

```
pw session create --type vscode my-cluster

  # Create a VS Code session opening a specific directory
  pw session create --type vscode --directory '~/projects' my-cluster

  # Create a tunnel session forwarding a remote port
  pw session create --type tunnel --remote-port 8080 my-cluster

  # Create and immediately open in the browser
  pw session create --type desktop --open my-cluster

  # Create a session with a custom name
  pw session create --type tunnel --name my-tunnel --remote-port 8080 my-cluster

  # Create and immediately connect (port forward)
  pw session create --type tunnel --connect --port 8080 --remote-port 3000 my-cluster
```

Options

```
      --connect              Connect (port forward) after creation (desktop and tunnel only,
implies --wait)
      --directory string     Directory to open in VS Code
  -h, --help                 help for create
      --name string          Custom session name (default: <type>_<timestamp>)
      --open                 Open the session in the browser after creation (implies --wait)
      --port int             Local port for --connect (0 = auto-select)
      --remote-host string   Remote host for tunnel type (default "localhost")
      --remote-port int      Remote port (required for tunnel type)
      --type string          Session type: desktop, vscode, tunnel (required)
      --wait                 Wait for the session to be running before exiting
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw sessions - Manage sessions

pw sessions ls

pw sessions ls

List all sessions

```
pw sessions ls [flags]
```

Examples

```
  # List all sessions
  pw session ls

  # List all sessions, displayed as a table
  # Tables display URI, Status, Local Port, Remote Host, Remote Port, and Workflow Run
  pw sessions ls -o table

  # List only desktop sessions
  pw session ls --type desktop
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (json, table)
  -t, --type string     Filter by session type (desktop, vscode, tunnel)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw sessions - Manage sessions

pw sessions open

pw sessions open

Open a session in the browser

Synopsis

Open a running session in your default browser.

For desktop sessions, opens the VNC viewer in the browser. For VS Code sessions, opens the VS Code server in the browser. For tunnel sessions, opens the session page in the browser.

The session name can be found using **pw session ls**.

```
pw sessions open <session-name> [flags]
```

Examples

```
  # Open a desktop session in the browser
  pw session open my-desktop
```

```
# Open a vscode session in the browser
  pw session open my-vscode
```

Options

```
  -h, --help   help for open
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw sessions - Manage sessions

pw sessions stop

pw sessions stop

Stop and delete a session

Synopsis

The **pw session stop** command stops a running session and deletes it.

The session name can be found using **pw session ls**.

```
pw sessions stop <session-name> [flags]
```

Examples

```
  # Stop a desktop session
  pw session stop desktop_1771523513

  # Stop a tunnel session
  pw session stop tunnel_1771523513
```

Options

```
  -h, --help   help for stop
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
```

```
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw sessions - Manage sessions

pw ssh-public-keys

pw ssh-public-keys

Get the SSH public keys for a user

Synopsis

The **pw ssh-public-keys** command retrieves the SSH public keys for a specific user.

```
pw ssh-public-keys <username> [flags]
```

Options

```
  -h, --help   help for ssh-public-keys
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI

pw ssh

pw ssh

Initiate an SSH connection to a resource

Synopsis

The **pw ssh** command initiates an SSH connection to a specific compute resource. The resource has to be running, and you must have SSH access to it.

Resource names can be specified in the following formats:

- **pw://username/name** - full format with explicit username
- **pw://name** - managed cluster by name in your organization
- **name** - short format where username defaults to your username

```
pw ssh <resource> [command] [flags]
```

Examples

```
  # Interactively connect to a resource named 'my-resource' owned by you
  pw ssh my-resource

  # Interactively connect to a resource owned by another user
  pw ssh pw://Jake.Thayne/my-resource

  # Interactively connect to a managed cluster in your organization
  pw ssh my-managed-cluster
  pw ssh pw://my-managed-cluster

  # Interactively connect to your user workspace
  pw ssh workspace

  # Run the "hostname" command on a resource and exit
  pw ssh my-resource hostname

  # Run the "hostname" command on your user workspace and exit
  pw ssh workspace hostname

  # Connect to a node via a pw:// jump host (username defaults to resource username)
  pw ssh -J pw://Jake.Thayne/my-cluster 10.0.1.5

  # Connect to a node via a jump host with an explicit username
  pw ssh -J pw://Jake.Thayne/my-cluster alice@10.0.1.5

  # Run as a ProxyCommand for SSH
  ssh -i ~/.ssh/pwcli -o ProxyCommand="pw ssh --proxy-command %h" user@my-resource
  ssh -i ~/.ssh/pwcli -o ProxyCommand="pw ssh --proxy-command %h" user@pw://Jake.Thayne/my-
resource
  ssh -i ~/.ssh/pwcli -o ProxyCommand="pw ssh --proxy-command %h" user@workspace

  # Run as a ProxyCommand for SSH, with port forward
  ssh -i ~/.ssh/pwcli -L 8080:localhost:80 -o ProxyCommand="pw ssh --proxy-command %h" user@my-
resource
  ssh -i ~/.ssh/pwcli -R 80:localhost:8000 -o ProxyCommand="pw ssh --proxy-command %h"
user@workspace

  # Run as a ProxyCommand for SCP
  scp -i ~/.ssh/pwcli -O -o ProxyCommand="pw ssh --proxy-command %h" file.txt user@my-
resource:/home/user/file.txt
  scp -i ~/.ssh/pwcli -O -o ProxyCommand="pw ssh --proxy-command %h" file.txt
user@workspace:/home/user/file.txt
```

Options

```
  -h, --help                help for ssh
  -J, --jump string         Connect to the target via a pw:// jump host (similar to ssh -J)
      --keepalive int       Interval in seconds between keepalive messages (0 to disable)
(default 30)
```

```
--proxy-command        Run as a proxy command for SSH or SCP
  -R, --remote stringArray   Forward remote [bind_address:]port to local host:hostport
  -X, --x11                  Enable X11 forwarding
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI

pw upgrade

pw upgrade

Update the version of the CLI

Synopsis

The **pw upgrade** command updates the current version of the PW CLI to the latest from the platform that it is currently authenticated to.

```
pw upgrade [flags]
```

Options

```
  -h, --help   help for upgrade
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI

pw users

pw users

Manage organization users

Options

```
  -h, --help   help for users
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw users delete - Delete a user from the organization
- pw users ls - List organization users
- pw users reset-onboarding - Reset onboarding state for one or more users

pw users delete

pw users delete

Delete a user from the organization

Synopsis

The **pw users delete** command deletes a user and cleans up all their resources.

By default it uses the organization from your current context. Use **--org** to target a different organization.

You will be prompted to type the username to confirm deletion. Use **-f** to skip the confirmation prompt.

```
pw users delete <username> [flags]
```

Examples

```
  # Delete a user
  pw users delete jake.thayne

  # Delete a user from a specific organization
  pw users delete --org my-org jake.thayne

  # Skip confirmation
  pw users delete -f jake.thayne
```

Options

```
  -f, --force        Skip confirmation prompt
  -h, --help         help for delete
```

```
--org string   Target organization (defaults to current context)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw users - Manage organization users

pw users ls

pw users ls

List organization users

Synopsis

The **pw users ls** command lists all users in the organization.

By default it uses the organization from your current context. Use **--org** to target a different organization.

```
pw users ls [flags]
```

Examples

```
  # List all users
  pw users ls

  # List users in a specific organization
  pw users ls --org my-org

  # List as JSON
  pw users ls -o json

  # Search for a user
  pw users ls --search john

  # List only active users
  pw users ls --active true
```

Options

```
      --active string     Filter by active status (all, true, false)
  -h, --help              help for ls
      --limit int         Maximum number of users to return (default 50)
      --org string        Target organization (defaults to current context)
  -o, --output string     Output format (table, json, list) (default "table")
```

```
--sort-by string    Sort by field (username, email, lastLogin, createdAt)
      --sort-dir string   Sort direction (asc, desc)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw users - Manage organization users

pw users reset-onboarding

pw users reset-onboarding

Reset onboarding state for one or more users

Synopsis

The **pw users reset-onboarding** command marks the given users as needing to complete onboarding again. Requires the **org:users** role in the target organization (platform admins also qualify).

By default it uses the organization from your current context. Use **--org** to target a different organization.

```
pw users reset-onboarding <username> [username...] [flags]
```

Examples

```
  # Reset one user
  pw users reset-onboarding alice

  # Reset multiple users in a specific organization
  pw users reset-onboarding --org my-org alice bob

  # Skip confirmation
  pw users reset-onboarding -f alice

  # Output as JSON
  pw users reset-onboarding -o json alice bob
```

Options

```
  -f, --force           Skip confirmation prompt
  -h, --help            help for reset-onboarding
      --org string      Target organization (defaults to current context)
  -o, --output string   Output format (table, json) (default "table")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw users - Manage organization users

pw vscode

pw vscode

Open VS Code connected to a remote resource

Synopsis

Open Visual Studio Code connected to a remote resource via SSH.

This configures VS Code's Remote-SSH extension to connect through the Parallel Works platform and launches VS Code.

Requires the **code** CLI on your PATH and the Remote-SSH extension installed in VS Code.

```
pw vscode <resource> [path] [flags]
```

Examples

```
  # Open VS Code on a resource
  pw vscode my-cluster

  # Open VS Code in a specific directory
  pw vscode my-cluster /home/user/project

  # Open VS Code on another user's resource
  pw vscode otheruser/their-cluster
```

Options

```
  -h, --help   help for vscode
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI

pw workflows

pw workflows

Manage workflows

Options

```
  -h, --help   help for workflows
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw - The Parallel Works ACTIVATE CLI
- pw workflows create - Create a new workflow
- pw workflows delete - Delete a workflow
- pw workflows get - Get details of a workflow
- pw workflows ls - List all workflows
- pw workflows run - Run a saved workflow or a local YAML file
- pw workflows runs - Manage workflow runs
- pw workflows update - Update a workflow

pw workflows create

pw workflows create

Create a new workflow

```
pw workflows create <name> [flags]
```

Examples

```
  # Create a local workflow with default template
  pw workflows create my-workflow

  # Create a workflow from a YAML file
  pw workflows create --yaml workflow.yaml my-workflow
```

```
# Create a workflow with a display name
  pw workflows create --display-name "My Workflow" my-workflow
```

Options

```
      --description string    Description of the workflow
      --display-name string   Display name for the workflow
  -h, --help                  help for create
  -o, --output string         Output format (json)
      --yaml string           Path to YAML file for workflow definition
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows

pw workflows delete

pw workflows delete

Delete a workflow

```
pw workflows delete <name> [flags]
```

Examples

```
  # Delete a workflow
  pw workflows delete my-workflow
```

Options

```
  -h, --help   help for delete
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows

pw workflows get

pw workflows get

Get details of a workflow

```
pw workflows get <name> [flags]
```

Examples

```
  # Get workflow details
  pw workflows get my-workflow

  # Get workflow details as JSON
  pw workflows get -o json my-workflow

  # Get workflow YAML definition
  pw workflows get -o yaml my-workflow
```

Options

```
  -h, --help            help for get
  -o, --output string   Output format (json, yaml)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows

pw workflows ls

pw workflows ls

List all workflows

```
pw workflows ls [flags]
```

Examples

```
  # List all workflows
  pw workflows ls
```

```
pw workflows ls -o list

  # Output as JSON
  pw workflows ls -o json
```

Options

```
  -h, --help            help for ls
  -o, --output string   Output format (json, list)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows

pw workflows run

pw workflows run

Run a saved workflow or a local YAML file

```
pw workflows run <name-or-file> [flags]
```

Examples

```
  # Run a saved workflow by name
  pw workflows run my-workflow

  # Run a marketplace workflow by slug
  pw workflows run marketplace/my-marketplace-item

  # Run a workflow from a local YAML file
  pw workflows run ./my-workflow.yaml

  # Run with input parameters
  pw workflows run -i '{"param1": "value1"}' my-workflow

  # Run with inputs from a JSON file
  pw workflows run -i inputs.json my-workflow

  # Run with saved inputs
  pw workflows run --saved-inputs my-inputs my-workflow

  # Run with saved inputs, overriding individual values from --inputs
  pw workflows run --saved-inputs my-inputs -i '{"param1": "value1"}' my-workflow
```

```
# Validate workflow without executing (dry run)
  pw workflows run --dry-run my-workflow

  # Run with a custom display name
  pw workflows run --name "Test run with new params" my-workflow

  # Output as JSON
  pw workflows run -o json my-workflow
```

Options

```
      --dry-run              Validate only, don't execute
  -h, --help                 help for run
  -i, --inputs string        JSON string or file path with input values
      --name string          Custom display name for the run
  -o, --output string        Output format (interactive, json, text) (default "interactive")
      --saved-inputs string  Name of saved inputs on the workflow to load input values from
(overridden by --inputs)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows

pw workflows runs

pw workflows runs

Manage workflow runs

Options

```
  -h, --help   help for runs
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows

- pw workflows runs cancel - Cancel a running workflow run
- pw workflows runs clean - Delete workflow runs based on filters
- pw workflows runs errors - Show errors from a workflow run
- pw workflows runs list - List workflow runs
- pw workflows runs logs - Fetch workflow run logs
- pw workflows runs view - View a workflow run

pw workflows runs cancel

pw workflows runs cancel

Cancel a running workflow run

```
pw workflows runs cancel <slug> [flags]
```

Examples

```
  # Cancel a run by slug
  pw workflows runs cancel swift-falcon-17
```

Options

```
  -h, --help   help for cancel
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows runs - Manage workflow runs

pw workflows runs clean

pw workflows runs clean

Delete workflow runs based on filters

```
pw workflows runs clean [flags]
```

Examples

```
  # Delete runs older than 7 days
  pw workflows runs clean --older-than-days 7
```

```
# Delete all completed runs
  pw workflows runs clean --status completed

  # Delete specific runs by slug
  pw workflows runs clean --slugs swift-falcon-17,quiet-river-83

  # Delete completed runs older than 30 days
  pw workflows runs clean --older-than-days 30 --status completed
```

Options

```
  -h, --help                  help for clean
      --older-than-days int   Delete runs older than N days
      --slugs string          Comma-separated list of run slugs to delete
      --status string         Delete runs with status (completed, error, canceled)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows runs - Manage workflow runs

pw workflows runs errors

pw workflows runs errors

Show errors from a workflow run

Synopsis

Extract and display failed jobs and steps from a workflow run with log tails.

```
pw workflows runs errors <slug> [flags]
```

Examples

```
  # Show errors as JSON (default)
  pw workflows runs errors swift-falcon-17

  # Show errors as text
  pw workflows runs errors -o text swift-falcon-17

  # Show last 100 lines of logs per failed step
  pw workflows runs errors --tail 100 swift-falcon-17
```

Options

```
  -h, --help            help for errors
  -o, --output string   Output format (json or text) (default "json")
      --tail int        Lines of log tail per failed step (default 50)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows runs - Manage workflow runs

pw workflows runs list

pw workflows runs list

List workflow runs

```
pw workflows runs list [flags]
```

Examples

```
  # List all workflow runs
  pw workflows runs list

  # List runs with a specific status
  pw workflows runs list --status completed

  # List runs for a specific workflow
  pw workflows runs list --workflow my-workflow

  # Search runs by name or slug
  pw workflows runs list --search "my run"

  # Output as JSON
  pw workflows runs list -o json
```

Options

```
  -h, --help              help for list
      --limit int         Maximum number of runs to return (default 50)
      --offset int        Number of runs to skip
  -o, --output string     Output format (json, table)
      --search string     Search by name or slug
```

```
--status string     Filter by status (running, completed, error, canceled)
      --workflow string   Filter by workflow name
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows runs - Manage workflow runs

pw workflows runs logs

pw workflows runs logs

Fetch workflow run logs

Synopsis

Fetch step logs from a workflow run. By default shows all steps; use filters to narrow output.

```
pw workflows runs logs <slug> [flags]
```

Examples

```
  # Show all logs for a run
  pw workflows runs logs swift-falcon-17

  # Show only failed step logs
  pw workflows runs logs --failed swift-falcon-17

  # Show logs for a specific job and step
  pw workflows runs logs --job build --step run-tests swift-falcon-17

  # Show last 50 lines per step
  pw workflows runs logs --tail 50 swift-falcon-17

  # Follow logs until run completes
  pw workflows runs logs -f swift-falcon-17

  # Output as JSON
  pw workflows runs logs -o json swift-falcon-17
```

Options

```
      --failed          Only show logs from failed/errored steps
  -f, --follow          Poll and stream new lines until run completes
  -h, --help            help for logs
```

```
-o, --output string   Output format (json)
      --step string     Filter to a specific step by name or index number (requires --job)
      --tail int        Last N lines per step (0 = all)
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows runs - Manage workflow runs

pw workflows runs view

pw workflows runs view

View a workflow run

```
pw workflows runs view <slug> [flags]
```

Examples

```
  # View a run interactively
  pw workflows runs view swift-falcon-17

  # View a run by ID
  pw workflows runs view 507f1f77bcf86cd799439011

  # Print run details as text
  pw workflows runs view -o text swift-falcon-17

  # Output as JSON
  pw workflows runs view -o json swift-falcon-17
```

Options

```
  -h, --help            help for view
  -o, --output string   Output format (interactive, text, json) (default "interactive")
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows runs - Manage workflow runs

pw workflows update

pw workflows update

Update a workflow

```
pw workflows update <name> [flags]
```

Examples

```
  # Update workflow YAML from a file
  pw workflows update --yaml workflow.yaml my-workflow

  # Update workflow display name
  pw workflows update --display-name "My Workflow" my-workflow

  # Update workflow description
  pw workflows update --description "Runs batch jobs" my-workflow

  # Update multiple fields at once
  pw workflows update --yaml workflow.yaml --display-name "New Name" my-workflow
```

Options

```
      --description string    New description for the workflow
      --display-name string   New display name for the workflow
  -h, --help                  help for update
  -o, --output string         Output format (json)
      --yaml string           Path to YAML file to replace workflow definition
```

Options inherited from parent commands

```
      --context string         The context to use. Overrides PW_CONTEXT environment variable and
current context from config.
      --platform-host string   The Parallel Works ACTIVATE platform host to use. Will override any
other platform host settings.
  -v, --verbose                Enable verbose logging
```

SEE ALSO

- pw workflows - Manage workflows