

# Jetstream2 Documentation

---

**None**

*None*

## Table of contents

---

1. Jetstream2 Main	5
1.1 Jetstream2 system status and information	7
2. Getting Started	9
2.1 Getting Started Tutorial	9
2.2 Logging in to Jetstream2	10
2.3 Creating your First Instance	15
2.4 Accessing your Instance	22
2.5 Volume Management	27
2.6 Installing and Running Software	36
2.7 Instance Management And SU Conservation	39
2.8 Snapshots and Images	42
2.9 Tutorial Cleanup and Next Steps	47
3. Jetstream2 Info	49
3.1 Jetstream2 System Overview	49
3.2 Architecture and Capabilities of Jetstream2	51
3.3 Configuration and specifications	54
3.4 Jetstream2 Network Configuration and Policies	55
4. Frequently Asked Questions	62
4.1 Jetstream2 FAQ Home	62
4.2 Troubleshooting	68
4.3 Jetstream2 Allocations FAQ	71
4.4 GPU Related FAQ	76
4.5 Security FAQ	79
4.6 Software FAQ	83
4.7 Science Gateways and related services FAQ	84
5. General Usage Information	85
5.1 Jetstream2 Resources	85

5.2 ACCESS Credits and Jetstream2	86
5.3 Instance Flavors	88
5.4 Quotas	92
5.5 Instance Management Actions	94
5.6 Featured Images	98
5.7 Microsoft Windows on Jetstream2	99
<b>6. Allocations</b>	<b>107</b>
6.1 Jetstream2 Allocations Overview	107
6.2 Jetstream2 Trial Allocations (JTA)	110
6.3 Jetstream2 Allocations	111
6.4 Jetstream2 Education Allocations	113
6.5 Jetstream2 Research Allocations (Maximize ACCESS)	115
6.6 Jetstream2 Allocation Supplements	117
6.7 Jetstream2 Allocation Extensions and Renewals	119
6.8 Jetstream2 Allocations FAQ	121
6.9 Budgeting for Common Usage Scenarios	126
6.10 Usage Estimation Calculator	129
<b>7. Storage</b>	<b>130</b>
7.1 Jetstream2 Storage Overview	130
7.2 Volumes	132
7.3 Manila	134
7.4 Object Store	138
7.5 Transferring Files	143
<b>8. User Interfaces</b>	<b>144</b>
8.1 USER INTERFACES	144
8.2 Exosphere	145
8.3 Horizon	0
8.4 CLI	0
8.5 CACAO	0

9. General VM Operations	0
9.1 Security	0
9.2 Maintenance and Administration	0
9.3 Research Software	0
10. Software Collection	0
10.1 Jetstream2 Software Collection	0
10.2 Using the Jetstream2 Software Collection from the CLI	0
10.3 Using the Jetstream2 Software Collection from the Web Desktop	0
10.4 Software Licenses	0
11. Policies and Best Practices	0
11.1 Acceptable Use Policies and Research and Export Control Guidance	0
11.2 Jetstream2 Export Control Guidance	0
11.3 Acceptable Usage Policies for Jetstream2-hosted Gateways	0
11.4 Best Practices for Jetstream2	0
12. Advanced Capabilities	0
12.1 Containers on Jetstream2	0
12.2 Terraform	0
12.3 Virtual Clusters on Jetstream2	0
12.4 Federating a gateway or JupyterHub on Jetstream2	0
12.5 Load Balancing with OpenStack Octavia	0

## 1. Jetstream2 Main

---



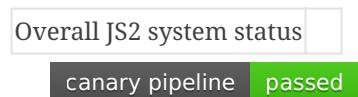
Jetstream2 is a user-friendly cloud computing environment for researchers and educators running on [OpenStack](#) and featuring [Exosphere](#) as the primary user interface. It is built on the successes of Jetstream1 and continues the main features of that system while extending to a broader range of hardware and services, including GPUs, large memory nodes, virtual clustering, programmable cyberinfrastructure with OpenStack Heat and Terraform, and many other features. It is designed to provide both infrastructure for gateways and other “always on” services as well as giving researchers access to interactive computing and data analysis resources on demand.

For a more in-depth description please see the [System Overview](#).

If you are interested in trying out our resources, consider applying for a [Jetstream2 Trial Allocation](#) and following our [Getting Started Tutorial](#).

---

### Jetstream2 Status



Please visit <https://jetstream.status.io/> for detailed system status information and planned maintenance announcements. Also see, [Jetstream2 system status and information](#) for additional information on our [outages and maintenance mailing list](#) and [community chat](#).

---

### Accessing Jetstream2

**Access to Jetstream2 is available solely through Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) allocations. You must be on a valid allocation or the PI of a valid allocation to have access to Jetstream2.**

**New users may wish to review:**

- [System Overview](#)
- [Acceptable Usage Policies](#)
- [Allocations Overview](#)
- [Exosphere Overview](#)
- [Troubleshooting and FAQ](#) may have helpful suggestions for common problems.

## 1.1 Jetstream2 system status and information

---

Overall JS2 system status

Jetstream2 maintains a system status and announcement site at <https://jetstream.status.io>. Please visit that site for detailed system status information and planned maintenance announcements.

### Canary

canary pipeline passed

The canary is a continuous integration pipeline that tests Jetstream2's primary (IU) region every few hours. If the status site is Operational but the badge above indicates a failing pipeline, this suggests a new systemic issue that needs staff attention.

### Mailing list for outages and maintenance

We also have a mailing list that receives system notices. All JS2 researchers, educators, and students get automatically added when they receive JS2 access. You may also subscribe by emailing *js2-users-l-subscribe AT list.iu.edu* or by contacting JS2 help to be added.

### Community chat

We are also introducing a new community discussion site on the Matrix communication network: <https://matrix.to/#/#jetstream-cloud:matrix.org>. This allows JS2 researchers and students to communicate easily with each other. There are many client choices, but we might suggest using [Element](#) via a web browser or app (available for iOS, Android, Mac, Windows, and Linux). While the discussion site is not an official support channel, Jetstream2 staff will participate during business hours and may answer questions found there.

The ticketing system is still the preferred mechanism for official support. That is available via [help@jetstream-cloud.org](mailto:help@jetstream-cloud.org) or the contact form <https://jetstream-cloud.org/contact/index.html> for any questions not covered in the JS2 documentation.

ACCESS issues (accounts, passwords, 2FA, allocations, etc) are best handled via the ACCESS support form: <https://support.access-ci.org/open-a-ticket>

Jetstream2 issues may be opened via the ACCESS form as well.

## ACCESS User News and Announcements

### What Is ACCESS?

The new NSF-funded program for advancing computational and data-intensive research – the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) – offers significant opportunities for community involvement. One of those is ensuring that requests for resource allocations meet the program's requirements in careful merit reviews by a diverse group of reviewers.

The ACCESS website is: <https://access-ci.org/>

ACCESS Infrastructure News and Outages may be found here [https://operations.access-ci.org/infrastructure\\_news](https://operations.access-ci.org/infrastructure_news)

## 2. Getting Started

---

### 2.1 Getting Started Tutorial

---

Welcome! If you are new to Jetstream2 or just want to brush up on the fundamentals, this tutorial will help guide you through the basics of using Jetstream2. Please follow along with the instructions presented here to get hands on practice interacting with Jetstream2. By the end of this tutorial, you will feel comfortable logging in, managing instances, and running software. Before you begin, please read the Prerequisites section below.

#### 2.1.1 Prerequisites

---

This tutorial is designed to be hands on, meaning we want you to do the steps yourself as you follow along. **In order to participate, you must first have an active allocation on Jetstream2.** For this tutorial, either a Jetstream2 Trial Allocation (JTA) or a full ACCESS allocation will work. If you are brand new, we recommend signing up for a Jetstream2 Trial Allocation.

Please visit [Allocations Overview](#) for details and instructions and return here after receiving an allocation.

#### 2.1.2 Let's Get Started

---

Once you have an active allocation, it's time to Log in. Click [Next](#) or following the link below to get started.

#### [Logging in to Jetstream2](#)

#### 2.1.3 Outline

---

- [Logging in to Jetstream2](#)
- [Creating your First Instance](#)
- [Accessing your Instance](#)
- [Volume Management](#)
- [Installing and Running Software](#)
- [Instance Management](#)
- [Snapshots and Images](#)
- [Next Steps](#)

## 2.2 Logging in to Jetstream2

There are many ways to interact with Jetstream2, and each interface has its own unique method of interacting with our system, with its own set of strengths and weaknesses. The entirety of this tutorial will focus on using our most beginner-friendly interface, Exosphere. After you have completed this tutorial, and if you are interested in learning about our other interfaces, please see [About our User Interfaces](#).

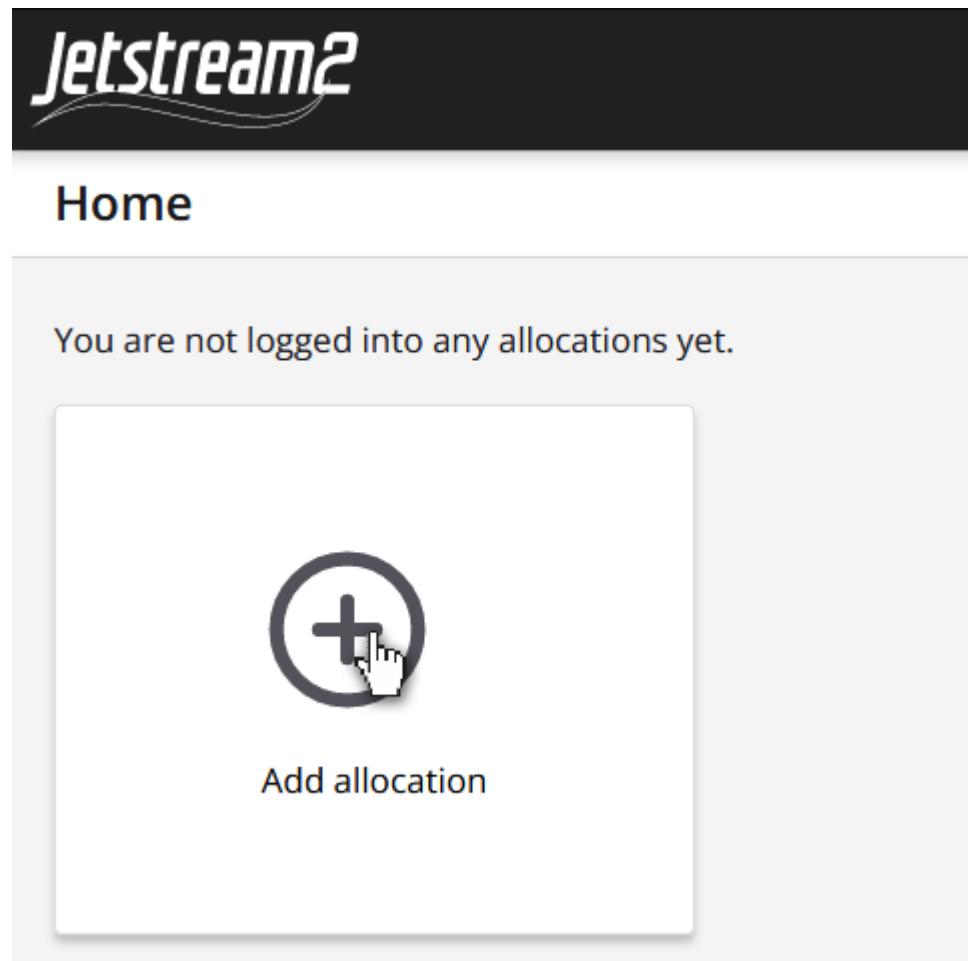
### 2.2.1 Enter Exosphere

[Click Here to Open Up Exosphere in a New Window](#)

### 2.2.2 Add your Allocation

You should be presented with a blank page saying you are not logged into any allocations yet. Follow the steps below to add your allocation(s).

Click **Add allocation**.



Click Add ACCESS Account

Home > Log in > Add ACCESS Account

## Add ACCESS Account

**ACCESS**

Add ACCESS Account

Recommended login method for Jetstream2

Select an Identity Provider -> Choose ACCESS CI

Select an Identity Provider

ACCESS CI (XSEDE) ?

Remember this selection ?

**LOG ON**

By selecting "Log On", you agree to the [privacy policy](#).

Log in using your ACCESS credentials.



If you had an XSEDE account, please enter your XSEDE username and password for ACCESS login.

**ACCESS Username**

**ACCESS Password**

**LOGIN**

Choose which allocations you want to use.

If you signed up for a Jetstream2 Trial Allocation, you will see an option like the second one, but with your username. If you belong to an ACCESS allocation, it will look like the top option, with your project code and title. You can select multiple allocations here and can always come back to this screen to add more later.

Home > Select Allocations for js2

## Choose Allocations for js2.jetstream-cloud.org

- TRA230023 — Jetstream2 Education and Training
- [REDACTED] — JTA User

**Choose**

### 2.2.3 The Updated Home Screen

Congrats, you have now added your allocation to Exosphere.

After adding an allocation, you should be returned to the home page, where you should see a card for each allocation you just added. From this screen, you can add more allocations, remove all your allocations (don't worry, if you do this then just repeat the steps on this page to readd your allocations), or click on a card to open the allocation.

Home Remove All Allocations

Allocation TRA230023 Jetstream2 Education and Training Jetstream2 IU  1 instance 1 public IP address	Allocation [REDACTED] JTA User Jetstream2 IU	 Add allocation
---	--	--------------------

## 2.2.4 Open your Allocation

---

Go ahead and click on the allocation you want to use and proceed to [Creating your First Instance](#)

## 2.3 Creating your First Instance

If you haven't done so already, go ahead and click on the allocation card you want to use. For this tutorial we will use a Jetstream2 Trial Allocation, but the steps here can be applied to any allocation with credits exchanged for the standard Jetstream2 compute resource.

### 2.3.1 Navigating the User Interface

The screenshot shows the Jetstream2 User Interface (JTA) Allocation Usage page. At the top, there is a navigation bar with links for Home, Allocation, Settings, Get Support, and About. The main content area is titled "Allocation Usage" and shows the following cards:

- Instances:** Shows 0 of 2 total instances used. Below it says "No instances to preview".
- Volumes:** Shows 0 of 10 total volumes used. Below it says "No volumes to preview".
- Public IP Addresses:** Shows 0 of 2 total public IP addresses used. Below it says "No public IP addresses to preview".
- SSH Public Keys:** Shows 5 of 100 total SSH public keys used. Below it lists "Windows-Server-2022-JS2-Beta" and "Featured-AlmaLinux8", both marked as "Public".
- Images:** Shows a list of images: "Windows-Server-2022-JS2-Beta" and "Featured-AlmaLinux8", both marked as "Public".

You should be presented with an overview page with several cards indicating instances, volumes, Public IP Addresses, etc. If you are using a brand new allocation, most of these cards will be empty. Go ahead and click around to familiarize yourself with the UI.

If you want to return back to the allocation overview page, simply click on your allocation in the nav bar.

The screenshot shows the Jetstream2 Instances page. At the top, there's a navigation bar with 'Home' and 'Allocation' (which has a hand cursor icon pointing at it), followed by 'Instances'. Below that is a header 'Jetstream2 IU - [REDACTED] (logged in as [REDACTED]@access-ci.org)'. The main content area has a title 'Instances' with a list icon. Below it are five status boxes: 'Instances used' (0 of 2 total), 'Cores used' (0 of 2 total), 'RAM used' (PB of 6 GB), all with progress bars. A message below says 'You don't have any instances yet, go create one!'

### 2.3.2 Creating an Instance

When you are ready, let's move on to creating your first instance. An instance is essentially an entire computer that exists virtually in “the cloud”, meaning you can only interact with it over the internet. Just like if you were to go shopping for a new computer, when you create an instance you can choose the operating system and how powerful you want the instance to be.

To begin, click on the **Create** button in the top right and select Instance.

The screenshot shows the Jetstream2 Instances page for Project TG-TRA160003. At the top, there's a navigation bar with 'Home' and 'Project TG-TRA160003' (logged in as cmart@xsede.org). There are also 'Remove Allocation' and 'Create ^' buttons. Below that is a summary box for 'Allocation usage' (0 of 1,000,000 SUUs) and a note 'Jetstream Staff Test Allocation'. To the right is a sidebar with three buttons: 'Instance' (with a server icon), 'Volume' (with a disk icon), and 'SSH Public Key' (with a key icon). The main content area has two sections: 'Instances' (showing 8 of 100 total used) and 'Volumes' (showing 0 of 10 total used).

### Choosing a Source/Image

Next you get to choose the instance ‘source’, or image. An image is essentially just a copy of an instance which includes the operating system and every installed software package. Images can be used as the base for future instances so you don’t have to reinstall everything

whenever you want a new instance. Jetstream2 staff maintains and distributes several featured images that include a few popular Linux operating systems and some basic software packages.

**For now, ensure you have [By Type](#) selected in the top bar and choose [Ubuntu 22.04](#).** This is by far our most popular featured image.

## Choose an Instance Source

[By Type](#) [By Image](#)



**Ubuntu**

- Wide compatibility with community software packages
- Good choice for new users

**22.04 (latest)** 

20.04



**Red Hat-like**

- Based on Red Hat Enterprise Linux (RHEL)
- Compatible with RPM-based software

**Rocky Linux 9**

Rocky Linux 8

AlmaLinux 9

AlmaLinux 8

### Give your Instance a Name

We recommend giving your instance a unique and descriptive name so that you or others will have some idea of what the instance is for at a later date. Alternatively, if you are just testing things out and plan to delete the instance later, we offer a button to generate a randomized name.

**Go ahead and give your instance a unique name now.**

Name *	<input type="text" value="happily-relaxed-gecko"/>		<input type="button" value="Random Name"/>
--------	--	--	--

### Choose a Flavor

The flavor determines the CPU count, the RAM available, and the size of the root disk. If you are using a Jetstream2 trial allocation, the maximum size available to you is an [m3.small](#). For a complete list of sizes, see [Instance Flavors](#).

For the purpose of this tutorial, an [m3.small](#) is more than enough, so choose that.

Flavor					
General-purpose					
	Name	CPU	RAM	Root Disk	Ephemeral Disk
<input type="radio"/>	m3.tiny	1	3 GB	20 GB	none
<input checked="" type="radio"/>	m3.small	2	6 GB	20 GB	none
X	m3.quad	4	15 GB	20 GB	none
X	m3.medium	8	30 GB	60 GB	none
X	m3.large	16	60 GB	60 GB	none
X	m3.xl	32	125 GB	60 GB	none
X	m3.2xl	64	250 GB	60 GB	none

Flavors marked 'X' are too large for your available quota

### Choose a Root Disk Size

We recommend you keep the default size selected here as this will not count against your total storage quota. If you need more space, you can always attach a volume for more space later. Also note that larger flavors have 60 GB of root disk instead of 20 GB.

**Leave as default (20 GB)**

### How Many Instances

**Leave this at 1**

### Enable Web Desktop

Select **Yes** to enable the Web Desktop. We will be using the Web Desktop later in this tutorial, so make sure to enable it here.

Enable web desktop?

No  Yes

### Choose an SSH Public Key

Leave as None for now.

### Create

Don't worry about Advanced Options. Go ahead and scroll all the way down and select [Create](#) to start building your instance.

**Note:** If you are interested in learning more about the options presented here, see [Creating an Instance with Exosphere](#).

### 2.3.3 Explore the Interface

Instances can take several minutes to build. After creating your instance you should be sent back to the home screen, where you should see the instance you just created with a [Building](#) sticker.

## Instances

Instances used      0 of 2 total



happily-relaxed-gecko

[Building](#)

### Instance Details Page

Take this time while your instance is building to explore the interface again. See if you can navigate to the instance details page, which looks like the following:

**Instance happily-relaxed-gecko**

**Info** 9d0f6098-1750-487d-9797-ef21912c5078

created 30 minutes ago by user dhabert@access-ci.org from image Featured-Ubuntu22 flavor m3.small Burn rate 2.00 SUs/hour

**Resource Usage**

CPU: of 2 total cores | RAM: of 6 total GB | Root Disk: of 20 total GB

**Interactions**

- Web Shell
- Web Desktop
- Native SSH : exouser@149.165.159.231
- Console

**Credentials**

Public IP Address: 149.165.159.231 [Unassign](#)

IP Details

Username: exouser

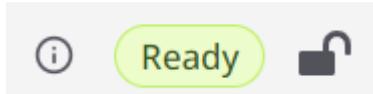
Passphrase: [Show](#)

SSH Public Key Name: dhavert\_key

From the Home Page, you can navigate to an instance's details page by clicking on the Instances card, then clicking on the name of the instance you want to inspect.

### Instance Ready

When the instance is ready, you will see the status sticker in the top right of the instance details page change from a yellow **Building** sticker to a green **Ready** sticker.



### What if the sticker changes to Error?

What if the status sticker for your instance changes to a red **Error** sticker instead?

Sometimes an instance will fail to initialize. There are any number of reasons this can occur. Don't worry though. The instance you created is brand new, so there is no harm in deleting it. The good news is that creating a new instance is incredibly easy, as you just learned.

### If your instance Errors out, delete it and make a new one

- You can delete an instance by selecting Actions (next to the status sticker), then select [Delete](#).
- Then create a new instance by repeating the steps in this guide

### Next

Once you have an instance that is [Ready](#), it's time to start interacting with it. Proceed to [Accessing Your Instance](#)

## 2.4 Accessing your Instance

In the last lesson we learned how to create an instance in the Exosphere user interface. Recall that an instance is essentially its own computer, except it's running virtually inside a much bigger computer. But you aren't next to the computer, and there is no physical monitor connecting to it, so how do you access your instance?

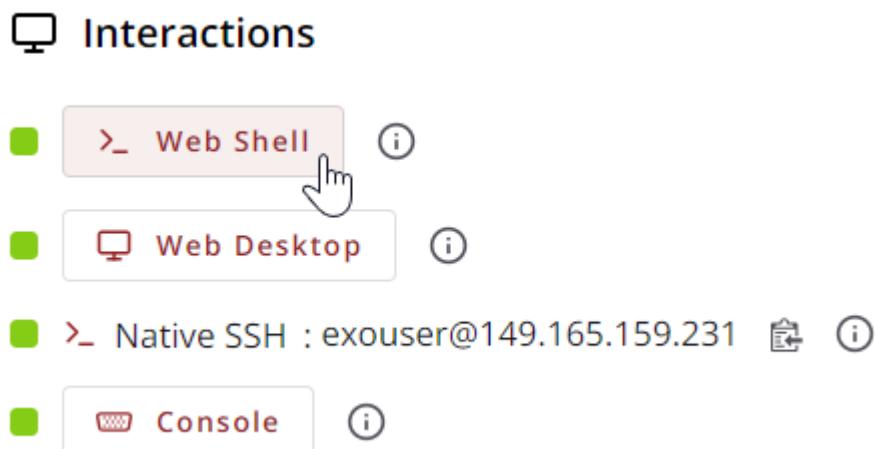
### 2.4.1 Web Shell

One of the reasons your instance took several minutes to build was that Exosphere was doing some neat stuff under the hood to create an account on your instance called `exouser` and set up the necessary software to allow you to remotely access this instance with the click of a button.

#### Opening the Web Shell

First, Navigate to the instance details page. If you don't remember what this looks like, go back to [Instance Details Page](#).

Then, look for the Interactions card and click on the `Web Shell` button



Alternatively, you can also connect to an instance from the Instance List page:

## Instances

Instances used: 1 of 2 total | Cores used: 2 of 2 total | RAM used: 6 of 6 GB

Filters: Created by me (dhavert@access-ci.org) [X](#) [+](#) [Clear filters](#) instance: 1

<input type="checkbox"/> <a href="#">happily-relaxed-gecko</a> ● m3.small · created 3 hours ago by dhavert@access-ci.org	<a href="#">Connect to ^</a>	<a href="#"></a>
---	------------------------------	------------------

1

### Using the Web Shell

The web shell opens a new tab in your browser and displays text that will look like the following:

```
System information as of Wed Mar 13 17:42:12 UTC 2024

System load:          0.0
Usage of /:           43.8% of 19.20GB
Memory usage:         25%
Swap usage:           0%
Processes:            256
Users logged in:     0
IPv4 address for br-381132f84b05: 172.18.0.1
IPv4 address for docker0:   172.17.0.1
IPv4 address for enp1s0:    10.2.129.14

=====
https://jetstream.status.io/=====

Overall Jetstream2 Status:  Operational

Scheduled Maintenance:
  (2024-04-14T12:00:00.000Z - 2024-04-15T00:00:00.000Z) April 14th Data Center Power Maintenance

=====
last login: Wed Mar 13 14:26:07 2024          I
to run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.

xouser@happily-relaxed-gecko:~$ █
```

When you first connect, notice you can see the system load and how much storage and memory you have used. Also, a brief window summarizing the current status will show whether there are any ongoing or future maintenance events scheduled.

As you may have noticed by now, Web Shell is a command line interface (CLI). If you are not comfortable using the command line, don't worry as we will move on to the Web Desktop option next. Even if you have no interest in learning command line, we recommend following along with this tutorial. The basics learned here will be useful to everyone.

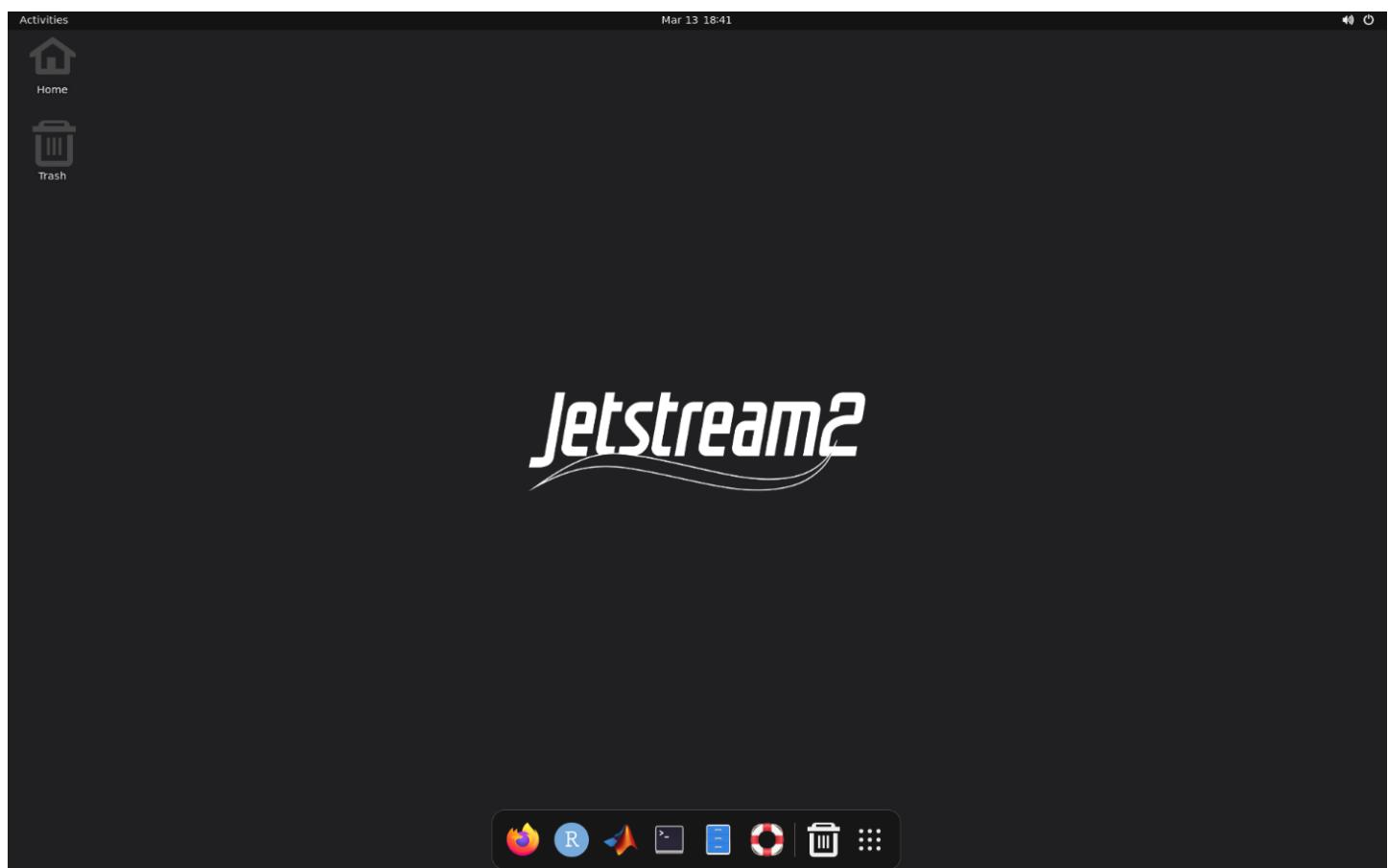
**Tip:** From the web shell, you can open/close a menu overlay with `ctrl + alt + shift` (on Windows) or `ctrl + command + shift` (on Mac). This menu allows you to change some settings such as the font size and color scheme, as well as transfer data between your local machine's clipboard and your instance's clipboard. It also facilitates uploading and downloading files to/from your instance (more on this later).

## 2.4.2 Web Desktop

The web desktop will not be available if you did not select to [enable web desktop](#) during instance creation. If you wish to use the web desktop, please delete this instance and create a new one with it enabled.

### Opening the Web Desktop

Close the Web Shell tab, return to the Instance Details page in Exosphere again, and **open the Desktop interface by clicking the [Web Desktop](#) button** instead of the Web Shell button. After clearing a few notifications, you should be presented with something that looks like the following:



### Using the Web Desktop

The default Linux desktop interface (called Gnome) might look a little different if you are coming from Windows or Mac, but it functions essentially the same.

Notice along the bottom bar that some software comes preinstalled on your Jetstream2 instance. These include, but are not limited to, Firefox, R Studio, and Matlab. Go ahead and explore these tools to familiarize yourself. Most noteworthy is the 3x3 grid of dots, which will show even more applications, and provide a search bar to search through all apps on available on the system.

**Tip:** From the web desktop, like the web shell, you can open/close a menu overlay with `ctrl + alt + shift` or `ctrl + command + shift`.

### 2.4.3 Next

---

Next we will learn about [Volume Management](#)

## 2.5 Volume Management

### 2.5.1 What are Volumes?

You can think of a volume as a flash drive. You can store data on it and move it between computers (instances). Also, you can attach multiple volumes to the same instance, but each volume can only attach to one instance at a time.

### 2.5.2 Creating a Volume

To create a volume, click **Create** like you did to create an instance, except this time select Volume.

The screenshot shows the Jetstream2 web interface. At the top, there is a navigation bar with links for Messages, Settings, Get Support, and About. Below the navigation bar, the main content area shows 'Allocation Usage' for CPU, GPU, and Large Memory instances. A 'Create' button is visible in the top right corner. A dropdown menu is open from the 'Create' button, listing 'Instance', 'Volume' (which is highlighted with a green border), and 'SSH Public Key'. Below the allocation usage, there is a section for 'Jetstream Staff Test Allocation' with 'Instances' and 'Volumes' buttons. The 'Volumes' button is also highlighted with a green border.

Next, name your volume something descriptive, and choose a size. For this tutorial, the default 10 GB is more than enough. Then click the Create button.

## Create Volume

Name \*

(Suggestion: choose a good name that describes what the volume will store.)

Size in GB

Create



After creating the volume, you will be brought to the Volume Lists page.

### 2.5.3 Attaching a Volume to an Instance

There are two ways to attach a volume to an instance.

1. From the Volume Lists page, which is where you currently are.

The screenshot shows the 'Volumes' page with the following details:

- Volumes used:** 1 of 10 total (represented by a progress bar)
- Storage used:** 10 of 1,000 GB (represented by a progress bar)
- Filters:** Created by me (with a clear filters button)
- Volume Details:**
  - Name:** my-very-important-data
  - Size:** 10 GB
  - Created:** 7 minutes ago by me
  - Status:** Attached to no instance
  - Actions:** Delete (trash icon), Attach (button with a hand cursor icon)

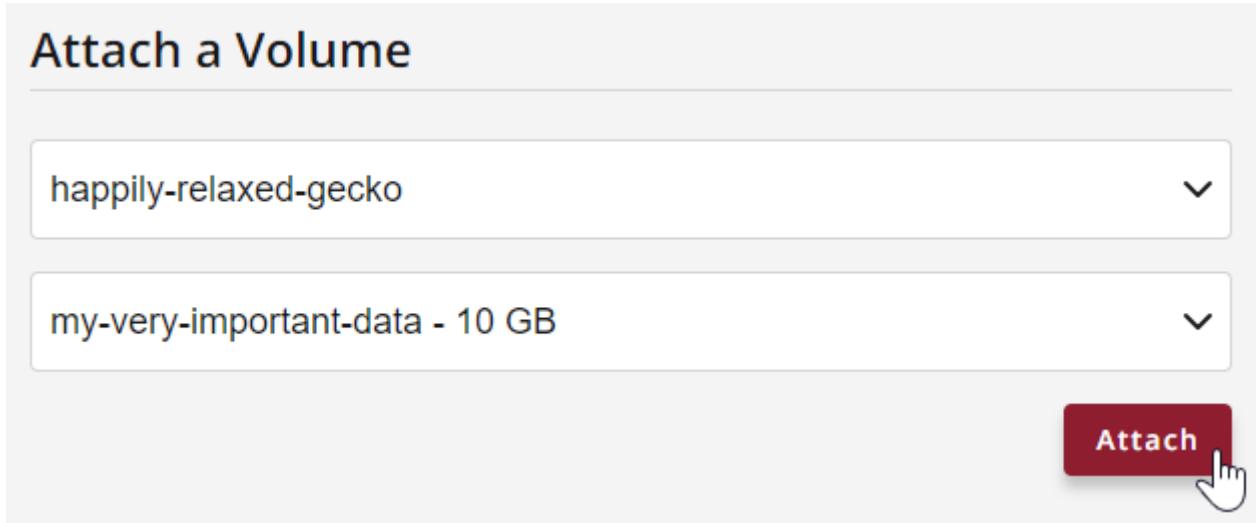
1. From the Instance Details page.

The screenshot shows the 'Instance painfully-relative-wasp' details page with the following information:

- Instance ID:** a6f96190-d9b1-43b4-8465-62253505a913
- Allocation Status:** Ready
- Created:** 2 minutes ago
- User:** @access-ci.org
- Image:** from image Featured-Ubuntu22
- Flavor:** flavor m3.tiny
- Metrics:**
  - CPU:** 100% usage of 1 total core at 3:07
  - RAM:** 100% usage of 3 total GB at 3:07
  - Root Disk:** 100% usage of 20 total GB at 3:07
- Actions:** Remove Allocation, Create
- Interactions:**
  - Web Shell (green icon)
  - Web Desktop (grey icon)
- Volumes:** (none) (button highlighted with a green border)

Both of these options will bring you to the volume attachment page, where you will select an instance and the volume you want to attach to it.

Select the instance you created earlier and the volume you just now created, then click Attach.



After clicking attach, a message will pop up telling you the device name given to the volume you just attached, as well as the location you can access the volume in the instance's filesystem.

A screenshot of a web-based interface titled "Volume Attached". It displays the following information:

- Volume: my-very-important-data
- Device: /dev/sdb
- Mount point: /media/volume/my-very-important-data
- The instance will mount your volume to /media/volume/my-very-important-data.
- For example, type `cd /media/volume/my-very-important-data` in a web shell on the instance.
- (This may not work on older operating systems, like CentOS 7. In that case, you may need to format and/or mount the volume manually.)

A red button labeled "Go to my instance" is visible at the bottom left.

Don't worry, this information will also be available on the Instance Details page. Click 'Go to my instance' to pull up that page now and look for the Volumes card to see the mount point.

## 2.5.4 Accessing the Volume

### Access Volume From Web Shell

First, open the Web Shell. If you are not sure how to do this, review [Accessing Your Instance](#).

Once the web shell is open, simply change directory to the location of the mount point. In our example, this looks like the following, but you will have to alter this to the name of the volume you created.

**Note:** The most sure-fire way is to copy/paste into the Web Shell is to use the menu overlay (`ctrl + alt + shift` / `ctrl + command + shift`). After copying something from this guide, paste it in the Clipboard section of the Web Shell menu overlay. Then close the overlay and paste it into the terminal. Depending on your browser, there may be shortcuts to paste such as right click or middle mouse, but this is not as reliable.

```
cd /media/volume/my-very-important-data
```

That's all there is to it!

If you are comfortable with using the command line, or are interested in learning some basics, continue with the next section. Otherwise, you can skip to [Access Volume From Web Desktop](#).

### Create A File in Web Shell

For demonstration purposes, let's create some data in this volume.

First, let's do a simple search to print out everything inside this volume:

```
find .
```

The output:

```
./lost+found find: './lost+found': Permission denied
```

You'll notice there is one folder called `lost+found` here that gives a permission denied error. Don't worry about this folder. Let's make our own folder, then create a file inside this folder with the contents of `Hello, World`.

```
mkdir my-folder echo "Hello, World" >> my-folder/hello.txt
```

Let's do another search to print out everything inside this volume. This time let's add `sudo` to the beginning of the find command. `sudo`, which stands for 'super user do', runs the rest of the line with admin level permissions. This will give the `find` command permission to search inside the `lost+found` folder and remove the permission denied message:

```
sudo find .
```

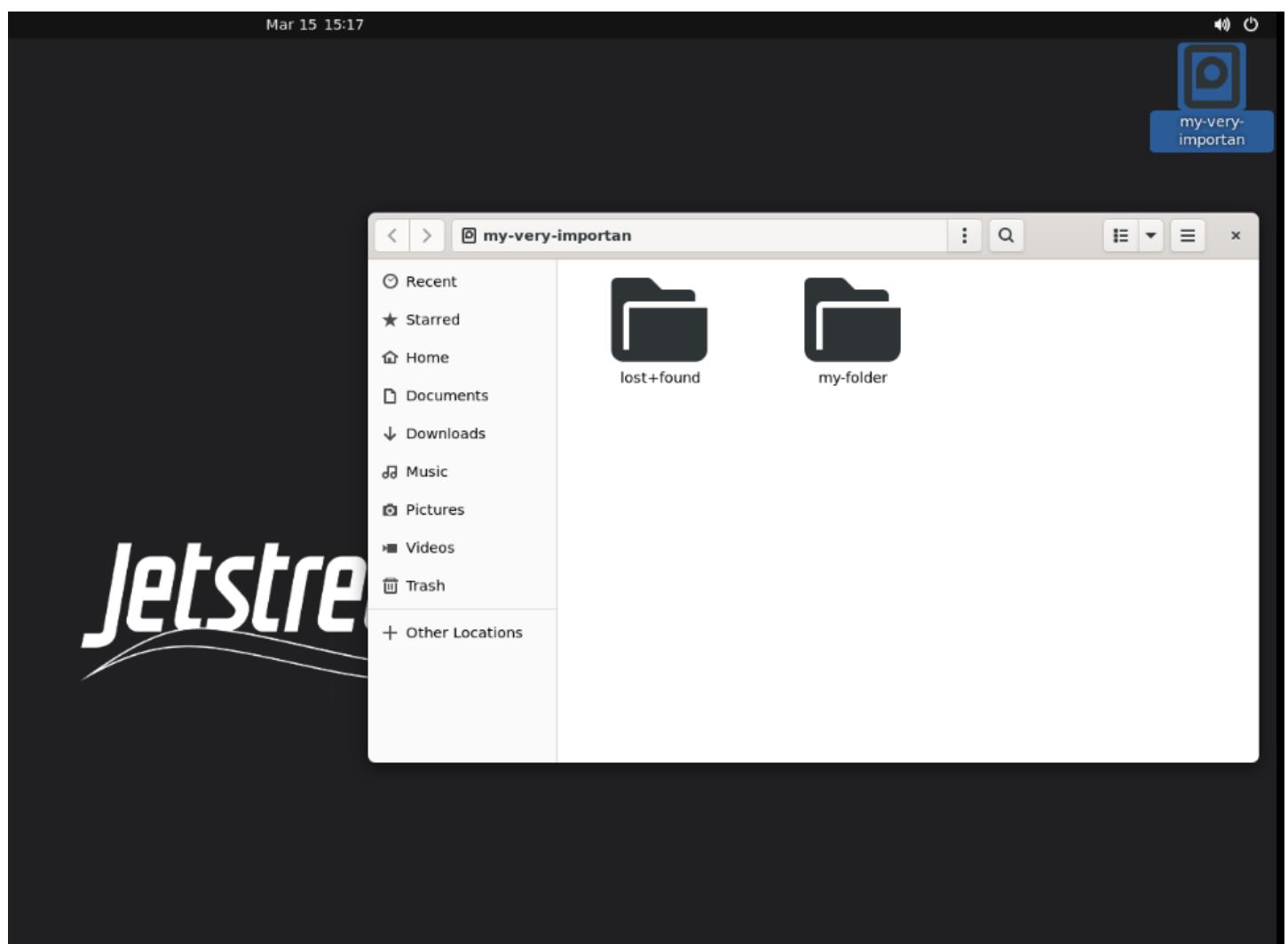
```
./lost+found ./my-folder ./my-folder/hello.txt
```

Sure enough, we can see that folder and file now exist.

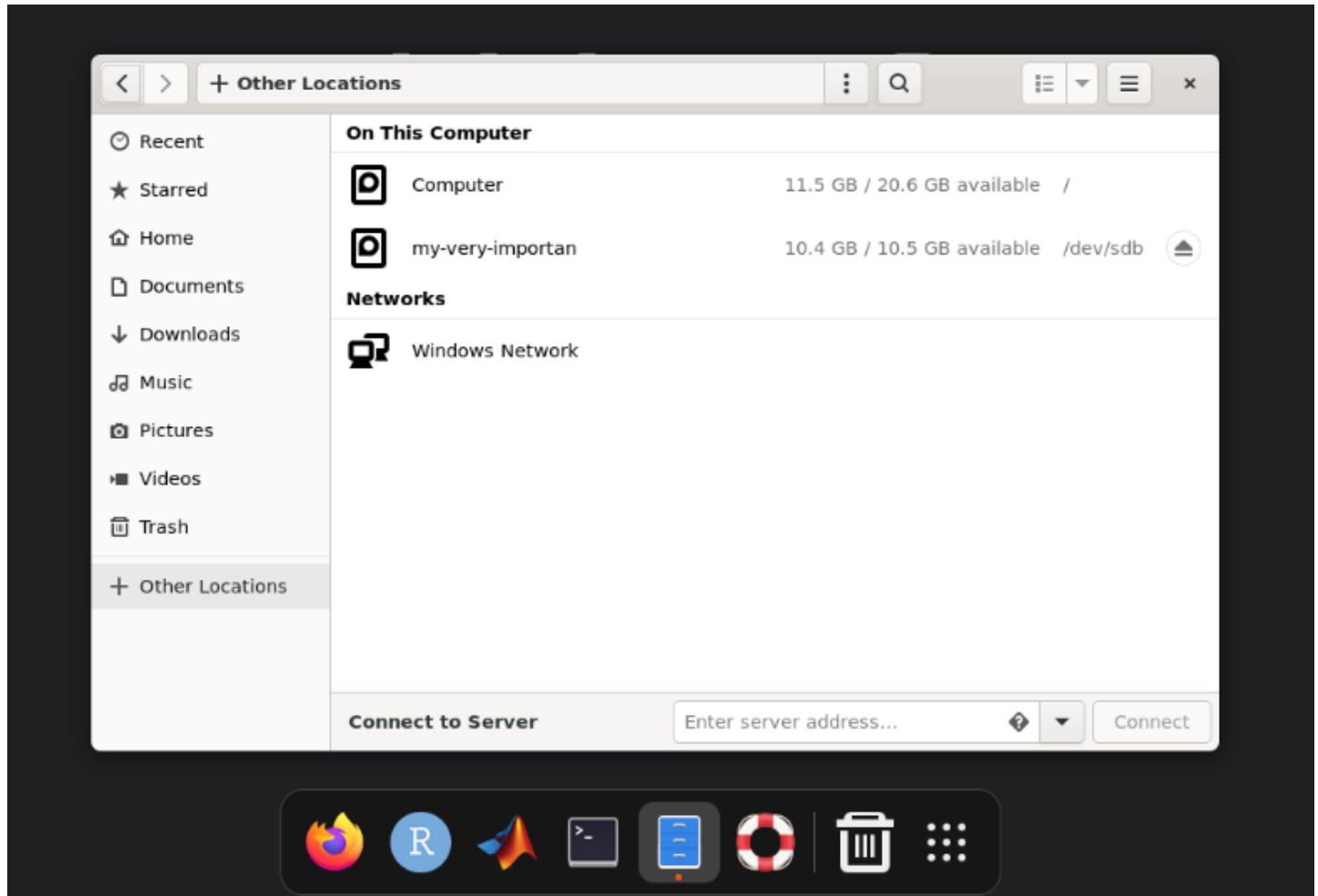
### Access Volume From Web Desktop

First, open the Web Desktop. If you are not sure how to do this, review [Accessing Your Instance](#).

With the volume attached, you should see an icon in the top right of the desktop with the name of your volume that you can double-click to open.



Alternatively, you can open the File Browser, click on 'Other Locations' on the left tab, then find your volume in the list.



If you followed the ‘Create a File Using Web Shell’ steps, you should be able to see the folder and file that you created.

## 2.5.5 Transferring Data

The easiest way to transfer data is to use the Web Shell or Web Desktops’s Menu Overlay. Recall you can open this overlay using `ctrl + alt + shift`. See [Transferring files with web shell and web desktop](#) for detailed instructions.

**See if you can transfer some files from your local machine into the volume that you just attached.**

If you would like to learn about other data transfer options, see [Transferring Files](#).

## 2.5.6 Removing the Volume

Before proceeding, make sure to close any Web Shell or Web Desktop tabs you have opened from the previous steps.

To Detach a volume from an instance, you must first navigate to the volume details page. As an exercise, see if you can find this page on your own. It should look like the following.

## Volume my-very-important-data

### Status

InUse

UUID: 7e557cff-34e5-41a7-8624-f49128916120 

### Attached to

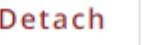
Instance: happily-relaxed-gecko

Device: /dev/sdb

Mount point\*: /media/volume/my-very-important-data

\* Volume will only be automatically formatted/mounted on operating systems which use systemd 236 or newer (e.g. Ubuntu 18.04 or newer, Rocky Linux, or AlmaLinux)

This volume must be detached before it can be deleted.

A quick way to get here is to find the Volumes card on the instance details page and click on the arrow next to the volume you want to inspect. Alternatively you can go back to your allocation home page and navigate through the volumes card there.

### Detach the Volume

Click the Detach button to detach the volume, then confirm the action. This will take a few seconds and then the volume will be available.

From here, you can attach the volume it to another (or the same) instance or delete the volume. For now, leave the volume alone. We will come back to it later.

### 2.5.7 Next

Move on to [Installing and Running Software](#) to learn about using our software collection and installing your own.

## 2.6 Installing and Running Software

### 2.6.1 Software Collection

The software collection is a shared network drive that includes some of the most popular research software, such as R Studio, Matlab, and Anaconda, so that you don't have to install these yourself.

#### Web Desktop

On the web desktop, all you have to do is find the software you want to use and open it. Matlab and R Studio are already on the toolbar, but you will have to search for others to find them.

For full instructions on accessing the software from the Web Desktop, see [Using the Jetstream2 Software Collection from the Web Desktop](#).

#### Web Shell

Things are a bit different on the command line interface. Software is available but not automatically loaded.

To see a full list of available software, enter

```
$ module avail
```

To load specific software, use the `module load` command. To test this out, let's load R.

```
$ module load R
```

Now we can start R by, you guessed it, entering `R` into the command line

```
$ R R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid" Copyright (C) 2022 The R Foundation for Statistical Computing Platform: x86_64-pc-linux-gnu (64-bit)
```

From here you can start entering R commands line by line. To quit, simply enter

```
> q()
```

For full instructions on accessing the software from the Web Shell, including how to save which modules you want loaded, see [Using the Jetstream2 Software Collection from the CLI](#).

## 2.6.2 Installing your own Software

There is a lot more software out there than we provide through the software collection. Every field of science has its own specialized software, plus there is a ton of additional non-science-specific software that you may need if you are running a server or more complicated setup. The good news is that most of this software is supported in Linux, and the developers usually provide guides for installing their software on Linux.

We can't go over every possible method of installing software here, so instead we will cover just the basics.

The first things to notes is what distribution of Linux your instance is running. If you've been following along with this tutorial, you may recall that when we created our instance we chose to install Ubuntu 22.04. If you selected something different, the instructions here will not be applicable.

### Advanced Package Tool (apt) - Command Line Install

Every Ubuntu install comes with the advanced package tool, or `apt`. To illustrate the general process of installing software on Ubuntu, we will use apt to install `duf` and `ncdu`, both disk usage analyzer tools.

Start by opening up a Web Shell interface.

First, it is generally good practice to update our package list. This can be done with

```
$ sudo apt update
```

**Note:** `sudo`, which stands for ‘super user do’, runs the rest of the line with admin level permissions, `apt` calls the advanced package tool, and `update` tells the advanced package tool to check for the latest updates for all installed packages.

The `update` command just looks for package updates. Next, we can actually install these updates with the `upgrade` command.

```
$ sudo apt upgrade
```

Now that our packages have all been updated, let's install a new package.

```
$ sudo apt install duf
```

Let's test our new package out. Simply enter the following into the terminal:

```
$ duf
```

This should display a chart that shows how much free space you have available on each device. Your root disk is usually labelled `/dev/sda1`, and its total size should be around 19.2 G.

**Excercise: Use what you just learned to install and run `ncdu`.**

If you installed it properly, running `ncdu` will open a menu that lists the size of every directory in your home folder. From here you can enter into a directory to see what is taking up space. Since we are using a relatively new instance, there is very little installed here now. However, this tool is extremely useful in tracking down what files are taking up space, and where they are located.

### 2.6.3 Next

---

Next we move on to [Instance Management](#), where you will learn how to stop and shelve instances, as well as best practices for conserving SUs.

## 2.7 Instance Management And SU Conservation

### 2.7.1 SU Consumption

As long as an instance is running, even if you are not actively using it, it is consuming Service Units (SUs). You can see this consumption in the top right of the Instance Details Page.

The screenshot shows the 'Instances' section of the Jetstream2 IU interface. At the top, there's a breadcrumb navigation: Home > Allocation dhavert > Instances > Instance happily-relaxed-gecko. Below that, it says 'Jetstream2 IU - dhavert (logged in as dhavert@access-ci.org)'. On the right, there are 'Remove Allocation' and 'Create' buttons. The main area displays the instance details for 'Instance happily-relaxed-gecko'. It includes a thumbnail, the instance name, its status as 'Ready', and an 'Actions' dropdown. Below this, the 'Info' section provides details: created 2 days ago by user dhavert@access-ci.org from image Featured-Ubuntu22 flavor m3.small. A green box highlights the 'Burn rate 2.00 SUs/hour' information.

If your instance is being used for a server or other service that requires it to be online 24/7, then leaving it running is the best course of action. However, if you use your instance as a virtual workstation, there are some actions you can do to save on usage consumption.

### 2.7.2 Shelving

The most efficient way to conserve your SUs is to shelf your instance. Shelving an instance automatically creates an image of your instance that will be used to create a new, albeit identical, instance at a later time when you unshelve the instance.

Shelving and unshelving can take several minutes to complete, but the SU consumption goes to Zero while the instance is shelved. Let's try this out.

**Shelf your instance by clicking on the Actions bar, then selecting Shelf.**

The screenshot shows the Jetstream2 web interface. At the top, there's a navigation bar with links for Home, Allocation dhavert, Instances, and Instance happily-relaxed-gecko. Below that, a header bar shows 'Jetstream2 IU - [REDACTED] (logged in as [REDACTED]@access-ci.org)'. On the right, there are buttons for Remove Allocation, Create, and Actions. The main content area shows an instance card for 'Instance happily-relaxed-gecko'. The 'Info' section displays the ID 9d0f6098-1750-487d-9797-ef21912c5078, creation date (2 days ago), user (dhavert@access-ci.org), and burn rate (2.00 SU/hour). Below it is a 'Resource Usage' section with CPU and RAM graphs. To the right is a context menu with several options: Lock, Suspend, Shelve (highlighted in green), Resize, Image, Reboot, and Delete.

You will see the status sticker change to **Shelving**. After several minutes, it will change to **Shelved**, at which point you should see the Burn rate change to 0.00 SUs/hour.

This screenshot shows the same instance card after shelving. The status is now 'Shelved' (highlighted in green). The burn rate is listed as 'Burn rate 0.00 SUs/hour'.

**Note:** It is good practice to detach your volumes before shelving an instance. If you have been following along with this tutorial, we did that in a previous section, [Volume Management - Removing the Volume](#).

## 2.7.3 Unshelving

When you want to use your instance again, you can unshelve the instance to bring it back up.

**Unshelve your instance by clicking on the Actions bar, then selecting Unshelve.**

After several minutes the status should be set back to Ready and you should see the burn rate change back to 2.00 SUs/hour.

#### 2.7.4 Other Actions

---

This tutorial only goes through shelving and unshelving actions as these are the most important actions for conserving your SUs. If you would like to learn more about the other Actions listed here, please see [Instance Management Actions](#).

#### 2.7.5 Next

---

Finally, we move on to the last topic for the tutorial, [Snapshots and Images](#).

## 2.8 Snapshots and Images

It can take a lot of time setting up your instances just how you like them and installing all the software you need. If you want to save this work so you don't have to repeat everything again, or if you want to share with your colleagues, you should take a snapshot of an instance.

Snapshots create an ‘image’ of your instance. This image can then be used as the source when booting a new instance, instead of using one of our featured images. You can also share images publicly or with specific other projects so other people can benefit from your hard work. This is especially useful if you are teaching a class and you don’t want your students to have to devote a bunch of time setting up their instances and installing the software they need to complete their assignments.

### 2.8.1 Create a Snapshot of an Instance

You can create a snapshot in the same instance action menu where you shelved your instance. This time choose Image.

The screenshot shows the Jetstream2 web interface. At the top, there's a navigation bar with the Jetstream2 logo, messages, settings, support, and about links. Below that is a header with 'Home > Allocation dhavert > Instances > Instance happily-relaxed-gecko', the user's name 'Jetstream2 IU - [REDACTED] (logged in as [REDACTED]@access-ci.org)', and buttons for 'Remove Allocation' and 'Create'. The main content area shows an instance card for 'Instance happily-relaxed-gecko' with details like creation date, user, and burn rate. To the right is an 'Actions' dropdown menu with several options: Lock (grey), Suspend (grey), Shelve (grey), Resize (grey), Image (highlighted in green), Reboot (yellow), and Delete (red). Below the instance card is a 'Resource Usage' section with CPU and RAM graphs.

#### Name your Image

Choose a name for your image. Then click Create.

## Create Image from Instance

Image name

**Create**

We will just leave the default image name for this tutorial, but we recommend giving the image a unique name so you remember what it's for.

### Delete your Original Instance

Note: You can skip this step if you **not** using a Jetstream2 Trial Allocation (JTA).

If you **are** using a Jetstream2 Trial Allocation, the quota that comes with the trial is too small to support two m3.small instances, so you will need to delete the original before creating a new instance from the image we just created.

To delete your instance,

- Navigate to the instance details page
- Open the Actions menu
- Choose Delete
- Confirm Deletion

**Note:** Normally we recommend only deleting your original instance after verifying the image has been successfully created and works as a source for a new image. Due to quota restrictions on JTA we simply cannot follow this guidance.

### 2.8.2 Create a New Instance From Custom Image

Now that we have an image of our instance, we are able to create a new instance from this image. To begin, start the process of creating a new instance.

## Create New Instance

The screenshot shows the Jetstream2 dashboard for Project TG-TRA160003. At the top, there are links for Messages, Settings, Get Support, and About. Below the header, it displays 'Allocation usage' (0 of 1,000,000 SUs) and 'Jetstream Staff Test Allocation'. On the left, there's a sidebar with 'Instances' (8 of 100 total) and 'Volumes' (0 of 10 total). On the right, there are buttons for 'Remove Allocation' and 'Create ^'. A dropdown menu under 'Create' includes options for 'Instance', 'Volume', and 'SSH Public Key'. The main area shows a summary of allocation usage.

## Choose Source

The screenshot shows the 'Choose an Instance Source' page. The 'By Image' tab is selected, highlighted with a green border. A search bar contains the text 'happi'. Below the search bar, there are filters: '+ Clear filters' and '1 image filtered from 29 total'. A single result is listed: 'happily-relaxed-gecko-image' (20 GB · private snapshot · belongs to this allocation), with 'Create Instance' and 'Set visibility' buttons next to it.

Make sure to change the tab at the top from ‘By Type’ to ‘By Image’. Instead of seeing only our featured images, you will instead see a list of all images available to boot from. This includes any images that have been shared publicly by other researchers.

Then search for the image you created above. It should show up in the list below.

Finally, select ‘Create Instance’. This will load the next page of the instance creation process. Here you should see the name of the image you selected:

**Note:** If the ‘Create Instance’ option is greyed out that means the image hasn’t finished building yet. Give it a few minutes and refresh your page. If the image is still not ready after several minutes, it’s possible it failed to create. Fortunately in this tutorial we are not losing any real work, so if this happens just create a new instance from our featured Ubuntu 22.04 image and try to create an image from that.

## Create Instance

Name \*  

Image:

### Give Your Instance a Name

You can name your new instance anything you want! Remember to keep it descriptive.

### Fill in Remaining Details

If you are unsure of what to select here, go back to [Creating your First Instance](#) to refresh your memory.

### Select Create

Now that you have given your instance a name and selected all the options you want, go ahead and click to create the instance. Remember that it will take several minutes to build.

### 2.8.3 Reattach Your Volume

As an exercise for practice, see if you can attach the volume that you created in [Volume Management](#) to this new instance.

Once you have it attached, navigate to its mount point to verify the data you uploaded to that volume is still there.

## 2.8.4 Verify Installed Software

Any changes you made to your old instance, before creating a snapshot of it, should be reflected in this new instance. Let's access the instance to check this.

If you followed along with the [Installing your own Software](#) section of the tutorial, then your new instance should already have `duf` and `ncdu` installed. Double check this by opening a Web Shell and running those commands again.

Since we have our volume attached now, running `duf` should display this volume in its list now. Below is a sample of the output from this command. As you can see, our volume, which is mounted at `media/volume/my-very-important-data`, has a size of 9.7G and is using 20.0K of space. Your disk usage may be different depending on how much data you uploaded.

## 2.8.5 Alternatives

Snapshots and images are not the only way of sharing your work between instances, and depending on your workflow may even be overkill. Some alternatives include creating your own [shell script](#) or containerizing your setup with something like [Docker](#) or [Apptainer](#).

## 2.8.6 Next

[Tutorial Cleanup and Next Steps](#)

## 2.9 Tutorial Cleanup and Next Steps

### 2.9.1 Cleanup

You are nearing the end of the getting started tutorial. To ensure you don't keep burning credits after completing the tutorial, let's clean up everything we've just done. This means we will need to remove all volumes, instances, and images we created during this tutorial. Follow these steps in order:

- Detaching all volumes
- Can be done from the Volume List page (Click on the Volumes card from the Allocation Overview page)
- Deleting all volumes
- Can also be done from the Volume List page
- Deleting all instance
- Can be done from the Instance List page
- Deleting any images *you* created
- Can be done from the Image List page. Filter by images belonging to 'this allocation'

The screenshot shows the 'Images' section of the Exosphere interface. At the top, there is a search bar with the placeholder 'try "Ubuntu"' and a 'Clear filters' button. Below the search bar, there are two filter buttons: 'Belongs to this allocation' (which is selected) and 'Clear filters'. A tooltip indicates that 2 images are filtered from a total of 30. A modal window titled 'Apply Filters' is open, containing the following settings:
 

- Visibility:** Options for 'Community', 'Private', and 'Public' are available, with 'Community' being selected.
- Belongs to:** Options for 'other allocations', 'this allocation' (which is selected and highlighted with a green border), and 'No choice' are available.
- Image type:** Options for 'image' and 'snapshot' are available, with 'image' being selected.

 Below the modal, a summary states '20 GB · private snapshot · belongs to this allocation'.

### 2.9.2 Next Steps

Congratulations! You have reached the end of the getting started tutorial. You now know how to use Exosphere, our most user-friendly interface, to create and manage instances, volumes, and images. These are the fundamentals of using cloud infrastructure.

## Get an ACCESS allocation

If you have been using a Jetstream2 Trial Allocation (JTA) for this tutorial and you are ready to get access to all of Jetstream2's features, you should begin the process of getting your own ACCESS allocation. See [ACCESS Overview](#) for guidance.

## Start Using Jetstream2

The best way to learn is by using. If you plan to use Jetstream2 to run your own virtual workstations, then you have all the tools you should need to create your instance, install and run your software, and conserve your SUs.

## Advanced Topics

Whether you want to use Jetstream2 to set up your own scaling infrastructure with Terraform, create a Science Gateway to provide for your community, or simply setup a simple static web server, we have a number of guides for getting started with these resources. Depending on what you plan to do, you may have to use one of our [other interfaces](#).

## 3. Jetstream2 Info

### 3.1 Jetstream2 System Overview

**Jetstream2** is a transformative update to the NSF's science and engineering cloud infrastructure and provides 8 petaFLOPS of supercomputing power to simplify data analysis, boost discovery, and increase availability of AI resources. It is an NSF-funded, user-friendly *cloud environment* designed to allow “always on” research infrastructure and to give researchers access to interactive computing and data analysis resources on demand, whenever and wherever they want to analyze their data.

#### jetstream is a cloud!

While it shares many common features and abilities with other research computing resources, it is **not** a traditional High Performance Computing (HPC) or High Throughput Computing (HTC) environment.

It provides a library of **virtual machines** and shared software designed to create research infrastructure or to perform discipline-specific scientific analysis. Software creators and researchers can create their own customized virtual machines (VM) or their own private computing system within Jetstream2.

Jetstream features multiple user interfaces, including the following **web-based user interfaces**:

- [Exosphere](#)
- [Horizon](#)
- [Cacao](#)

as well as the [OpenStack Command Line Interface \(CLI\)](#) and the [OpenStack Software Development Kit \(SDK\)](#).

The operational software environment is based on [OpenStack](#).

## Accessing Jetstream

Jetstream2 is primarily accessible through the [Exosphere](#) web interface using [ACCESS](#) credentials via [CILogon](#).

Newly created ACCESS accounts must be [added to a Jetstream2 specific allocation](#) by the PI or Resource manager in order to access Jetstream2.

Jetstream2 is meant primarily for **interactive** research, small scale processing **on demand**, or as the backend to science **gateways** to send research jobs to other HPC or HTC resources, or for general infrastructure for research or research-related development.

Jetstream2 may be used for **prototyping**, for creating tailored **workflows** to either use at smaller scale with a handful of CPUs or to port to larger environments after doing your **proof of concept** work at a smaller level.

## About Jetstream2

Consisting of five computational systems, Jetstream2's primary system will be located at Indiana University, with four smaller regional systems deployed nationwide at partners Arizona State University, Cornell University, the University of Hawai'i, and the Texas Advanced Computing Center. The 8 petaFLOPS cloud computing system will use next-generation AMD EPYC processors and NVIDIA A100 GPUs, and 17.2 petabytes of storage.

Within the Pervasive Technology Institute the project will be led by UITS Research Technologies with key contributions from the Cyberinfrastructure Integration Research Center and the Center for Applied Cybersecurity Research. Additional Jetstream2 partners include the University of Arizona, Johns Hopkins University, and the University Corporation for Atmospheric Research, with Dell Inc. as the primary supplier.

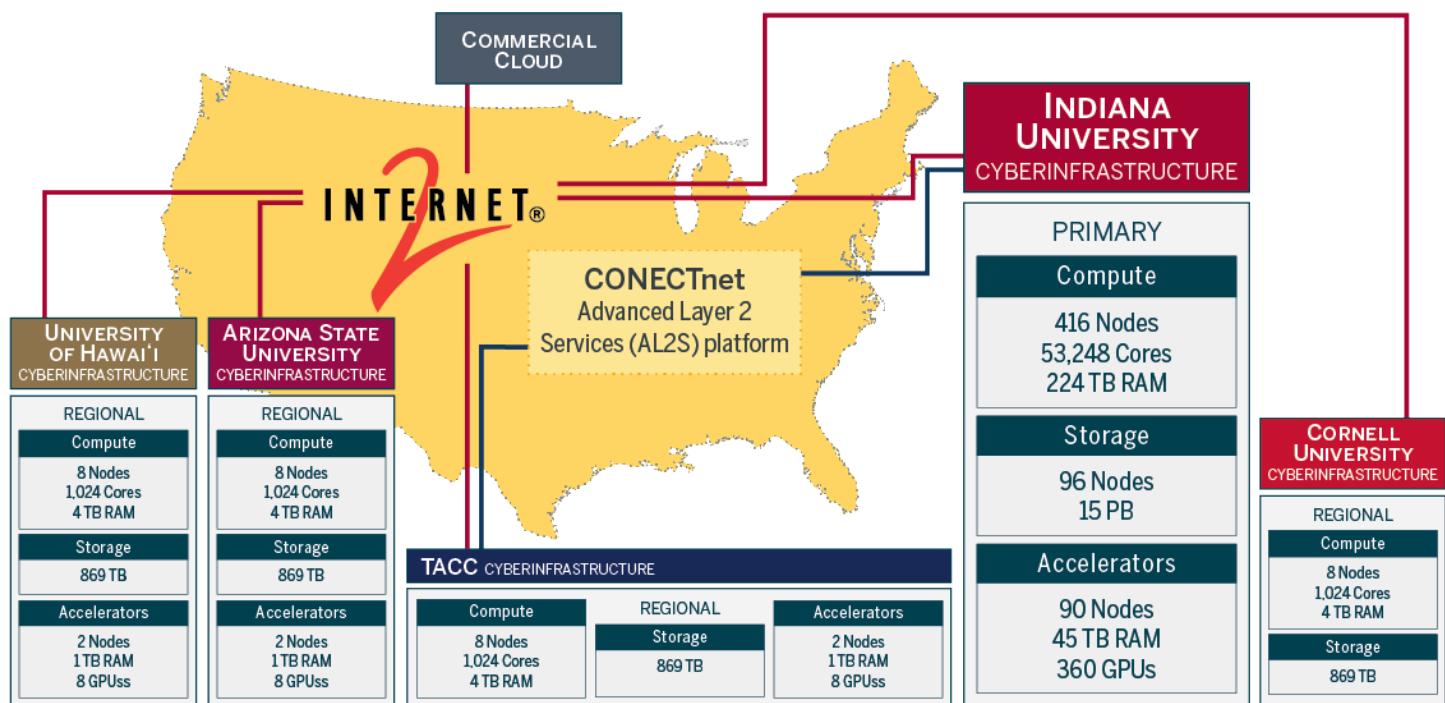
For more information, please see:

- [NSF Award 2005506 - Jetstream 2: Accelerating Science and Engineering On-Demand](#)
- [Jetstream System Specifications](#)

## 3.2 Architecture and Capabilities of Jetstream2

Jetstream2 consists of one primary cloud and multiple regional clouds. As you can see by the image below, the primary cloud is hosted at Indiana University in Bloomington, IN with regional clouds at various institutions across the United States.

*Another huge distinction is that instead of different domains – default (Atmosphere) and tacc (API) domains, there is one namespace for Jetstream2. You can change between the CLI, Horizon, Exosphere, and Cacao as you see fit to manage your resources*



Allocations will only be awarded on the primary cloud by default.

### Hardware

Jetstream2 brings **multiple classes of research computing hardware**. Jetstream2 still has hundreds of CPU-based compute nodes for general purpose virtual machines/computing. Jetstream2 also features a small number of large memory nodes with up to 1 terabyte of RAM. Jetstream2 makes available 90 GPU-enabled nodes with four **NVIDIA A100 GPUs** each. These are subdivided using **NVIDIA virtual GPU (vGPU)** to allow Jetstream2 allocations to utilize from 1/5th of a GPU to an entire GPU in their instances, facilitating everything from educational use requiring a minimal amount of GPU processing power to a full GPU for research workloads.

## Interfaces

Jetstream2 features an interface called [Exosphere](#) as the primary, recommended GUI interface. Jetstream2 also provides API services utilizing both the OpenStack Horizon GUI and a robust command-line interface (CLI). Atmosphere has evolved into a new tool called [Containerized Atmosphere for Continuous Analysis Orchestration \(CACAO or simply Cacao\)](#), which is built on the principles of Atmosphere (abstracting complicated functions such as firewalls and virtual networking). Because Jetstream2 will no longer have separate operating domains for Cacao and API operations, those utilizing Jetstream2 can switch between interfaces easily, seeing all virtual machines and other assets created in any interface. This single namespace also allows for third-party interfaces that can manage any OpenStack created resource to be used with Jetstream2.

## Containers and Orchestration

Jetstream2 is bringing support for containers to the forefront of services. It will support managing and scaling container-based workloads via the cloud-native functionality of OpenStack Magnum or using standalone Kubernetes deployments. Users will be able to deploy Docker Swarm, Apache Mesos, or Kubernetes container orchestration engines to manage and run their container-based research workloads. In addition, the features of Cacao will provide similar functionality to individuals who have no desire to access the OpenStack API directly. Both approaches allow researchers and educators to scale their workloads dynamically according to their needs.

## Additional Services

Services such as OpenStack Heat will be available for researchers and developers, as well. OpenStack Heat is a service that allows individuals to instantiate complex resources with dependencies via a declarative YAML-based language. Further, tools such as HashiCorp's [Terraform](#) programmable infrastructure, with the ability to deploy on Jetstream2, private clouds, and commercial clouds easily and consistently, will allow developers and researchers to have their environments where they need, when they need it. These capabilities build on one of the fundamental aspects of cloud computing: the ability of users to create, manage, and orchestrate use of tools autonomously, based on need, without involving sysadmins to install or enable new software.

## Virtual Clusters

In addition to the ability for individuals to control their infrastructure programmatically, Jetstream2 provides the capability to spin up [elastic HPC virtual clusters \(VCs\)](#) at the push of a button. These have been tested extensively on Jetstream1, with about thirty VCs running in production at different times. These Slurm-powered virtual clusters allow individuals to transition easily between cloud and HPC resources, acting as both a test-bed environment for custom software, and a highly-available production resource for projects with modest computational needs. The deployment process for these resources in Jetstream2 will be streamlined, allowing individuals to deploy an instance, acting as a head node, that is ready to accept jobs. Once jobs are submitted, worker instances will be automatically created and destroyed as needed. The Singularity/Apptainer container runtime environment will be built into these VCs, allowing individuals to use containerized scientific software without lengthy installation processes.

### Note

**Push-button virtual cluster deployment is available in [Exosphere](#) now.**

### Historical Reference

Jetstream1 utilized two primary clouds, geographically dispersed and powered by [OpenStack](#), to provide thousands of concurrent virtual machines to researchers, educators, and students from hundreds of institutions across the United States. Jetstream1 featured Atmosphere, a user-friendly graphical user environment, as the means of access for the majority of users. Jetstream1 also allowed API access for infrastructure-driven projects like science gateways.

There are quite a few similarities between Jetstream1 and Jetstream2. Jetstream2 will be powered by a more recent version of OpenStack and will still have a primary mission of providing virtual machine services to research and education faculty, staff, and students. Jetstream2, however, will build on what Jetstream1 delivered and provide a number of improvements and new services.

## 3.3 Configuration and specifications

---

### Primary Cloud Specifications

#### Compute Nodes (384 nodes)

System Configuration	Aggregate information	Per Node (Compute Node)
Machine type	Dell	Dell PowerEdge C6525
Operating system	Ubuntu	Ubuntu
Processor cores	49,152	128
CPUs	768 (AMD Milan 7713)	2
RAM	192 TiB	512 GiB
Network	100 Gbps x 4 to Internet2	100 Gpbs to switch
Storage	14 PB Total Ceph Storage	240gb SSD

#### Large Memory Nodes (32 nodes)

System Configuration	Aggregate information	Per Node (Large Memory Node)
Machine type	Dell	Dell PowerEdge R7525
Operating system	Ubuntu	Ubuntu
Processor cores	4,096	128
CPUs	64 (AMD Milan 7713)	2
RAM	32 TiB	1024 GiB
Network	100 Gbps x 4 to Internet2	100 Gpbs to switch
Storage	14 PB Total Ceph Storage	480gb SSD

#### GPU Nodes (90 nodes)

System Configuration	Aggregate information	Per Node (GPU Node)
Machine type	Dell	Dell PowerEdge XE8545
Operating system	Ubuntu	Ubuntu
Processor cores	11,520	128
CPUs	180 (AMD Milan 7713)	2
RAM	45 TiB	512 GiB
GPUs	360 (NVIDIA A100 SXM4 40GB)	4
Network	100 Gbps x 4 to Internet2	100 Gpbs to switch
Storage	14 PB Total Ceph Storage	960gb SSD

## 3.4 Jetstream2 Network Configuration and Policies

---

### Summary of Network Configuration and Policies

#### Hardware Configuration:

The Jetstream2 primary cloud configuration features:

- 100 Gbps network connectivity from the compute hosts to the cloud's internal network infrastructure
- 2x100 Gbps uplinks from the cloud infrastructure to the data center infrastructure
- 100 Gbps connectivity from the site infrastructure to the Internet2 backbone
- 100 Gbps connectivity to the ACCESS research network via virtualized link
- Individual instances have full access to this infrastructure with no added speed limits.

*It is important to note that connections coming from commercial/commodity internet will likely not be as fast as those coming from Internet2 sites.*

#### Persistent IPs:

A key difference between Jetstream1 and Jetstream2 is that no special or additional access is required to get a persistent IP address. Some of the GUI interfaces like [Exosphere](#) and [Cacao](#) release IPs by default when a VM is deleted. [Horizon](#) and the Jetstream2 [CLI](#) require you to explicitly release the IP address.

We do ask that you release any unused IPs back to the public pool. There are a finite number of IPs available and allocations hoarding them may inadvertently cause issues for other Jetstream2 researchers.

The Jetstream administration team reserves the right to release any IP addresses not associated with a VM as needed.

#### Network Security Policies:

In general, Jetstream2 does not restrict inbound or outbound access to virtual machines. There are a handful of blocks at the institutional level that are outside of the control of the Jetstream2 staff. In general, though, the most common Unix service ports (eg. 22/ssh, 80/http, 443/https, etc) are not restricted in any way. Whether they are open by default will be dependent on which user interface you're launching your VM with.

Please refer to the [Security FAQ](#) for additional information.

---

---

---

### In-depth exploration of Jetstream2 Networking

This section describes the network architecture of Jetstream2's primary cloud at Indiana University. Regional sites may be configured differently.

There are three kinds of networks used on Jetstream2.

- The **Cluster Network**, which carries all of the tenant/workload/data traffic. It *underlays* the Neutron overlay networks for cloud workloads. It also carries all storage traffic and OpenStack control plane traffic.
- The **Neutron overlay networks**, an entirely software-defined (virtual) networking stack which runs on top of the cluster network.
- The **Management Network**, which is fully physically separate from the cluster network. It is used for managing hosts and their [iDRACs](#).

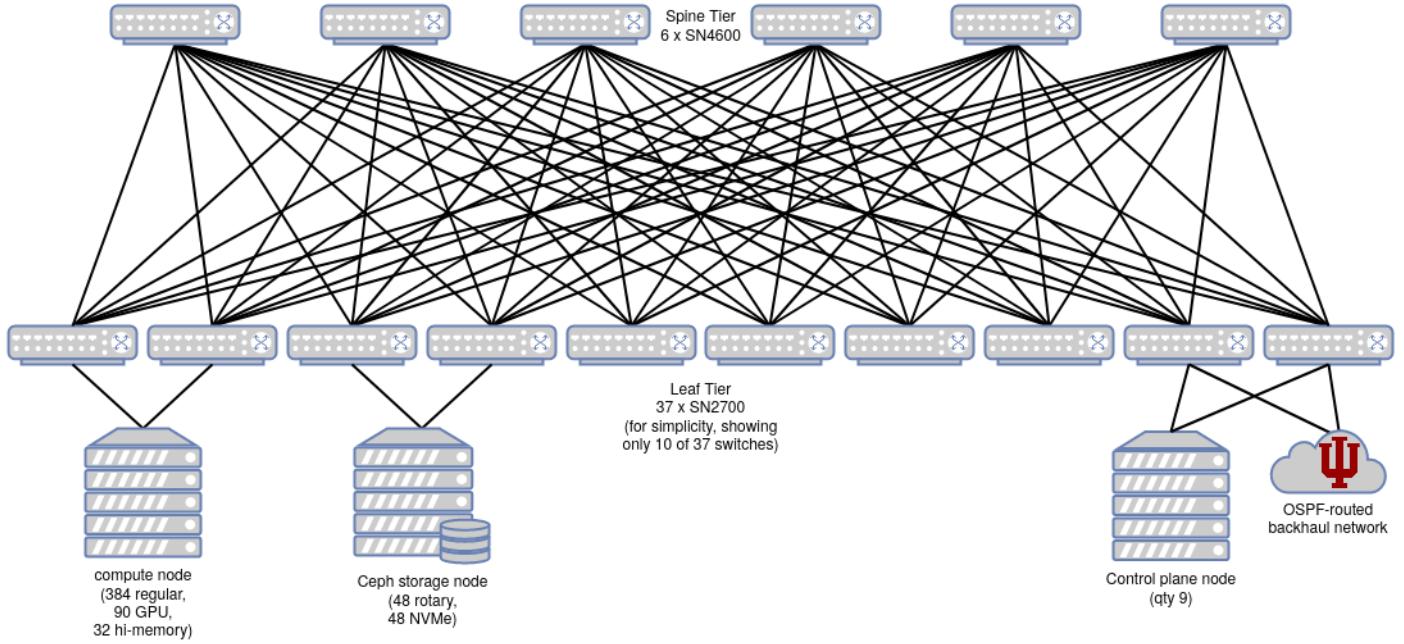
This document primarily looks at the **Cluster Network** and may delve into the **Neutron overlay networks**. The Management Network is not in the scope of user-facing documentation.

#### 3.4.1 Cluster Network

---

[BGP in the Data Center](#) is recommended reading. It provides background and orientation to Clos networks and BGP the way that it is used here. It is a short, accessible read for someone with general IT networking knowledge. For a quicker version, read chapters 1, 2, 4, and 6. Skim chapters 3 and 5.

The Jetstream2 network is a two-tier spine-and-leaf network in a [fat tree](#) or [Clos](#) topology.



There are no inter-switch links (ISLs). Traffic that transits the network will generally traverse a leaf switch, a spine switch, and another leaf switch.

Two of the leaf switches connect the Cluster network to the campus WAN and internet. These happen to be the two leaf switches that the control plane nodes are also connected to.

## Physical Switches

Switches use the [Cumulus Linux](#) operating system.

Switch types in the cluster network:

Model	Topology Layer	Switch Quantity	Port speed	Port Qty Per Switch
Mellanox SN4600	Spine	6	100 GbE	64
Mellanox SN2700	Leaf	37	100 GbE	32

## Dual-attached Servers

Each host (server) is *dual-attached* to the Cluster network via two separate leaf switches, using two 100 GbE interfaces. Dual-attaching serves both purposes of redundancy and load-balancing (via equal-cost multipathing).

Redundancy is particularly important here, because many workloads that run on Jetstream2 will *not* be ‘cloud-native’ or tolerant of partial-cluster network connectivity failure. Many users will hand-configure their analyses and services on a single instance (virtual machine)

that runs on a single host in a single cabinet. Dual-attaching every physical server means that a single network switch failure will not cause a complete or prolonged outage for these non-cloud-native workloads. The physical compute node will quickly notice that one of its network links is offline, and re-route all network traffic to use the remaining functional link.

### **Server cross-wiring**

Throughout the cluster, a given host may be variously connected to:

- 2 switches in the same cabinet, or
- 2 switches in an adjacent cabinet, or
- 1 switch in the same cabinet and 1 in an adjacent cabinet.

The benefit of this cross-wiring between cabinets is increased utilization of switch ports, and reduction of the number of switches needed. This is especially true in Jetstream2 because different types of hosts (e.g. compute, GPU, storage) are different physical sizes, i.e. different densities in a cabinet. The cost of cross-wiring is limited cabinet mobility, because hosts in one cabinet may have many connections to a switch in the next cabinet over, but cabinets are not expected to be moved over the lifetime of Jetstream2.

### **Border Gateway Protocol**

Jetstream2 uses [Border Gateway Protocol](#) (BGP) to route and deliver packets across the entire cluster network. BGP was originally created to handle traffic between large network service providers (a.k.a. carriers), but is also adapted for use within data centers. In this application, BGP replaces use of [MLAG](#) to route traffic over redundant physical paths between switches. Jetstream1 used MLAG, but Jetstream2 does not, for the following reasons:

- MLAG is fragile when making config changes
- “Rigorous and conservative interface state management needed. Temporary loops or duplicates not acceptable” ([source](#))
- MLAG does not maximize use of aggregate throughput supported by redundant physical links.
- MLAG requires inter-switch links (ISLs), which would require many (about 96) additional links between switches, which would require more cabling and purchasing switches with more physical ports.

Use of BGP solves all of these drawbacks with MLAG. (See also [MLAG on Linux: Lessons Learned](#). For a comparison of BGP and other internal routing protocols like Open Shortest Path First (OSPF), see pages 22 and 24 of [BGP in the Data Center](#) (labeled as pages 14 and 16)).

Some more configuration details:

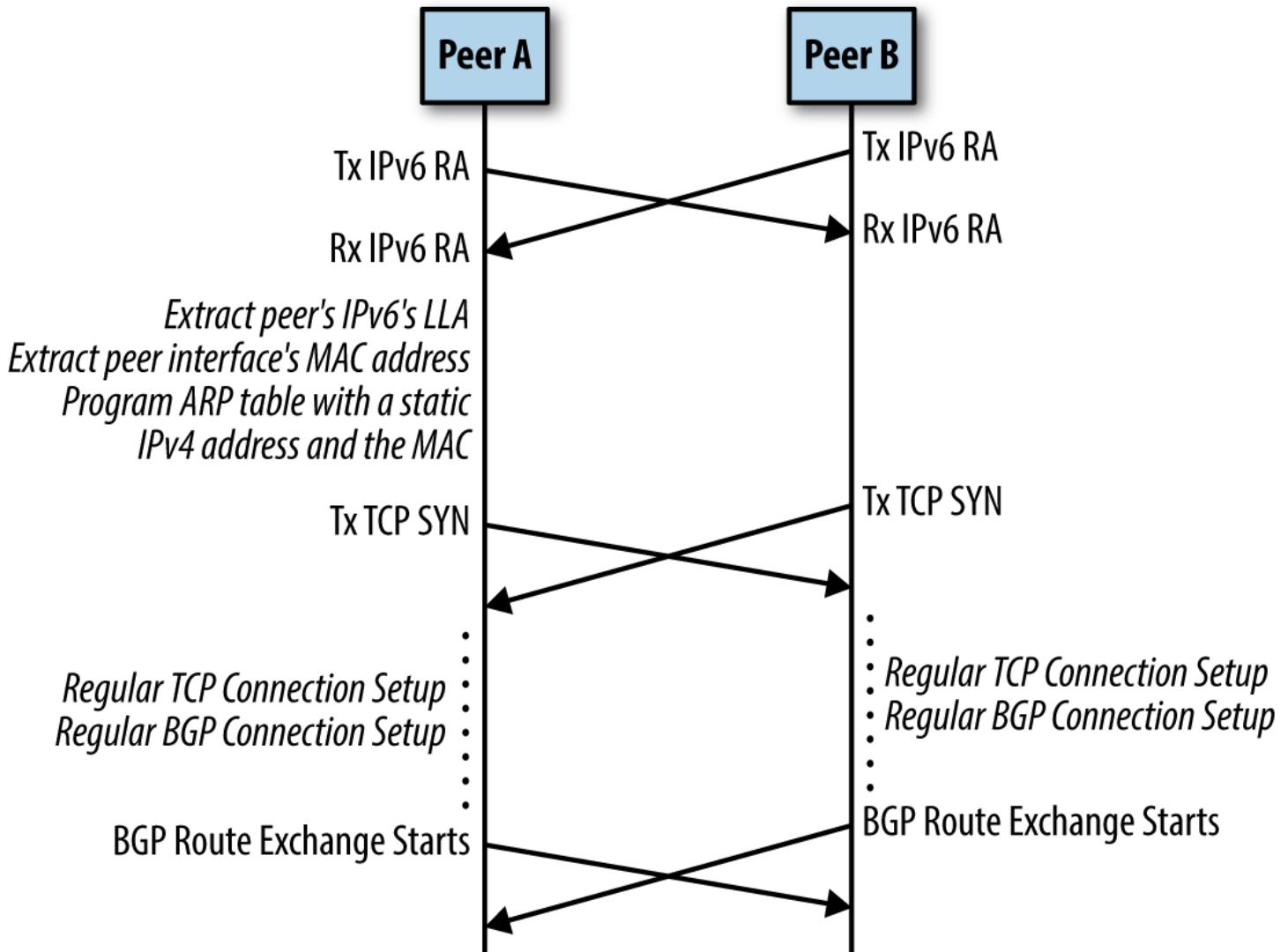
- On the Cluster network, peering between all nodes uses eBGP (external BGP), because each host and each leaf (but not spine) switch has its own individually-assigned private autonomous system number (ASN). iBGP (internal BGP) is not used.
- The spine switches all share a common ASN, 64512. This avoids path hunting problems as described in pages 26-29 of [BGP in the Data Center](#) (labeled as pages 18-21).
- [Equal-cost multipathing](#) allows packet forwarding to a single destination to be load-balanced across multiple network paths.

### **BGP Unnumbered**

In the cluster network, there is no network *bridging* or explicitly-defined shared subnets between connected switches (or between switches and hosts). There is only *routing* of packets between hosts. BGP requires underlying TCP/IP connectivity in order to create a connection, so how can that happen if we have no explicitly-defined subnets or bridges?

In very short, each physical 100 GbE interface on each node (host or switch) has only a [link-local IPv6 address](#) assigned. IPv6 Router Advertisement (RA) is a link-level protocol that periodically announces an interface's IPv6 addresses, including the link-local address. This is how each node identifies the IP of its peer. RFC 5549 allows a system to advertise IPv4 routes, and route IPv4 packets, over a pure IPv6 network. It also adds an “extended nexthop” capability to BGP, which negotiates use of RFC 5549 over a BGP peering session.

This strategy is described in greater detail in chapter 4 of [BGP in the Data Center](#). The result is that there is no need to define a shared subnet between any nodes. There is also no need to explicitly define neighbor-specific information in a node's configuration. Two peers connected with a network cable will discover each others' link-local addresses, exchange IPv6 route advertisements, and initiate a BGP route exchange.



### Routing on the Host

In traditional enterprise networking, packet routing (at layer 3 in the OSI model) is generally performed *off the host*, i.e., by router hardware that is physically separate from servers running the workloads. Sometimes there is purpose-dedicated router hardware, and sometimes routing is a role performed by the ethernet switches. In either case, servers do not make most packet routing decisions. Servers are *bridged* to routers at layer 2, on a common subnetwork. Servers determine where to send packets using ARP (Address Resolution Protocol). **This is not how the Jetstream2 Cluster network operates.**

Instead, the BGP routing fabric is extended all the way to each server. Each host has its own BGP autonomous system number (ASN), and runs the same FRRouting software as the switches. The host's primary IPv4 address (in private, RFC 1918 space) will appear on the loopback interface, and it is configured as a `/32` in CIDR notation – as its very own single-IP subnetwork, not bridged anywhere else!

A server advertises BGP routes for its own internal IPv4 address. These routes are advertised to both switches that the server is attached to, via IPv6 link-local connectivity. This overall strategy is called “Routing on the Host”, and it is described in chapter 6 of [BGP in the Data Center](#).

Again, in the cluster network there is no bridging between physical nodes, only routing.

Some more background reading (if you don’t like the book): - [Linux Routing On The Host With FRR \(archived version\)](#) - [Independence from L2 Data Centers](#) - [Border Gateway Protocol \(archived version\)](#)

## Conclusion

All of this BGP unnumbered architecture supports the ‘Abstraction layer’ of one or more IPv4 addresses on each host, and the ability to route packets from any host to any other host by its IPv4 address. At this level of abstraction and above, there is no concern with (or visibility of) the BGP routing layer or link-local IPv6 addresses. The Neutron overlay networks do not see it, or know about its complexity.

---

Areas of future discussion:

### **3.4.2 Neutron Overlay Networks**

TODO

#### **Layer 2 (Data Link)**

VXLAN / Open vSwitch / Linux Bridge?

#### **Layer 3 (Network)**

Neutron Distributed Virtual Routing (DVR)

Typically with DVR, egress traffic leaves a compute node directly to the WAN, while ingress traffic passes through a network node.

See also: - <https://docs.openstack.org/neutron/latest/admin/config-service-subnets.html>

## 4. Frequently Asked Questions

### 4.1 Jetstream2 FAQ Home

For specific FAQs, see the following pages:

- [Allocations FAQ](#)
- [Troubleshooting](#)
- [Security FAQ](#)
- [Software FAQ](#)
- [Gateways FAQ](#)

#### 4.1.1 General FAQs

##### I'm getting an error trying to log into Exosphere or Horizon

If you get an error that looks like this trying to log in to Exosphere or Horizon:

To use Jetstream2, your account must belong to an ACCESS allocation that is active on a Jetstream2 resource. Please visit <https://docs.jetstream-cloud.org/faq/general-faq>

you need to confirm that you either are the PI of an allocation or are on a valid allocation. If you need to be added to an allocation, you'll need to contact the PI of that allocation to do this. We are unable to add users to allocations.

If you are indeed on an allocation or are the PI of an allocation, please note that it can take up to four hours for allocations/users to become active. This is outside of our control. If you are still unable to log into [Exosphere](#) or [Horizon](#) after four hours from being added, please open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#).

**Use ACCESS CI credentials if you're having trouble authenticating**

We know ACCESS-CI credentials work correctly and recommend using those unless you are 100% sure you have linked credentials PREVIOUSLY in CILogon.

Please note that other CILogon credentials (i.e. for your home institution or Google) must be set up outside of this process. Follow these [Linking instructions](#) to link your home institution or other organization (e.g. Google, ORCID, etc) credentials.

**I was added to an allocation but can't access Jetstream2?**

It can take up to four hours for ACCESS to sync up everything to Jetstream2. If you were added recently, we recommend trying later. If you still cannot access your Jetstream2 account and your PI or resource manager has confirmed that you are on a valid allocation, [contact Jetstream2 help](#).

Use ACCESS CI credentials if you're having trouble authenticating - see just above!

**How do I cite Jetstream2 ?**

The Jetstream2 team requests that all researchers using Jetstream2 cite us in any publications that utilize results that were computed or created on Jetstream2. We keep the most up to date citation information here - [Jetstream2 Citation Information](#)

**I need a root disk larger than the maximum size for Jetstream2 instances. Can you create a custom flavor for me?**

In OpenStack, flavors define the compute, memory, and storage capacity of instances. We may also refer to it in support tickets or documentation as the VM's size.

We won't create custom flavors, but there are ways to get larger root disks. You can [review the flavors](#) and see if moving up from one of the smaller VMs to a slightly larger one would yield a larger root disk. The other option is to use a custom sized root disk using what Openstack calls "boot from volume", or a "volume-backed" instance. What this means is that instead of an ephemeral boot disk for the instance, a volume is used to be the root disk.

There are several upsides to this:

- You can have a root disk large enough to fit your needs
- The disk becomes reusable as a volume if necessary
- Shelving the instance is extremely fast compared to shelving a typical instance

The downside is that using boot from volume will count against your Jetstream2-Storage allocation whereas root disks do not.

Instructions for using boot from volume are here:

- [Exosphere: Choose a Root Disk Size](#)
  - Cacao (link coming)
  - [Horizon](#)
  - [CLI](#)
- 

### **How do I get access to one of the Jetstream2 regional clouds?**

The regional clouds of Jetstream2 (Arizona State University, Cornell University, University of Hawai'i, and Texas Advanced Computing Center) are available via invitation only. They have limited resources and the local PIs have discretion as to which projects may run there.

All allocations gain access to the primary Jetstream2 cloud at Indiana University. If you have an allocation and wish to have it added to a regional cloud, you can [request access using this form](#)

*As the regional clouds have autonomy over their access and because resources are limited, it's important to note that not all requests may be accommodated.*

---

### **Will there be Microsoft Windows support on Jetstream2 ?**

Microsoft Windows is not officially supported on Jetstream2. We will be making a limited number of images available for experimental use.

It is not known at this time whether GPUs will work on Microsoft-based instances. We will test this as time permits.

More information may be found on the [Microsoft Windows on Jetstream2](#) page.

---

### **How do I share a volume between virtual machines?**

You can't easily share volumes in OpenStack without deploying a Shared File System service. However, the native Openstack Manila filesystems-as-a-service option is available.

Instructions for using manila on Jetstream2 are here - [Manila - Filesystems-as-a-service - on Jetstream2](#)

Please note, there are different quotas for block storage (volumes) and shares. There will be a self-service tool for managing those quotas soon, but for now, if you need to have your Jetstream2 Storage quota adjusted between block and share storage, please [contact us via the Jetstream2 contact form](#) with the amount you wish to move between the storage types.

---

### **Can I set the password for a user on my virtual machine?**

We generally don't recommend using password authentication on Jetstream2, recommending that you use SSH keys for access. That said, if you need to set a password for console access or for some other reason, you can do it like this:

```
sudo passwd *username*
```

---

### **How can I get my public IP number from my VM ?**

If your VM has a public IP address and you need to find that IP (and don't have ready access to the Jetstream2 interfaces), use wget or curl from the command line to get your public IP:

```
wget http://169.254.169.254/latest/meta-data/public-ipv4 -q0 - wget http://ipinfo.io/ip -q0 - curl http://169.254.169.254/latest/meta-data/public-ipv4 cur
```

\*Note: http://169.254.169.254/latest/meta-data/public-ipv4 works even in conditions in which external DNS servers are not accessible.

---

### **How do I get a jetstream-cloud.org DNS name for my instance?**

Right now, every allocation's `auto_allocated_network` is set up with OpenStack Designate to automatically create DNS hostnames for instances. DNS hostnames are formatted as:

```
instance-name.allocation-name.projects.jetstream-cloud.org
```

Or a more real-life example:

```
genome-browser.bio123456.projects.jetstream-cloud.org
```

The DNS hostname points to your instance's public (a.k.a. floating) IP address, so this won't currently work if you remove an instance's public IP (or create an instance without a public IP).

If DNS hostnames are not working for your allocation or you'd like to set up Designate for another network, please [create a support ticket](#).

### Note

For information about hostnames outside the [projects.jetstream-cloud.org](https://projects.jetstream-cloud.org) space, see "[How do I set up a custom domain name for my gateway, site, or web service?](#)"

---

### Are there backups of Jetstream2 storage resources ?

Jetstream2 provides multiple forms of storage, including block (volumes), shares (Manila), and an object store. All of these storage formats are on erasure coded partitions that should provide protection against data loss in the case of hardware issues or outages.

Jetstream2, however, does not provide any backup service for data. Since the storage is not a shared filesystem as on traditional HPC, there are issues with providing consistent backup for all of the user-defined instance root disks, volumes, shares, and object store buckets. As such, we recommend that you keep a copy of any crucial data offsite. This is a good practice no matter what computing system you may be on.

Additionally, there are NSF-funded resources like the Open Storage Network (OSN) available for research storage for work in [active use](#). This is also an ACCESS allocated resource. More information may be found on the [Open Storage Network website](#)

---

### What are the IP ranges / CIDR blocks for Jetstream2 ?

The current CIDR blocks for Jetstream2 are:

[149.165.152.0/22](https://projects.jetstream-cloud.org) [149.165.159.0/24](https://projects.jetstream-cloud.org) [149.165.168.0/21](https://projects.jetstream-cloud.org)

---

## How do I keep a program running on a VM even if I log out or get disconnected?

If you want to start a workflow and have it run, that's a unix capability.

Tutorials like [How to Run Linux Commands in Background](#)

show a number of methods shown here on how to start a program and leave it running, even if your connection disconnects.

We would suggest looking at the nohup command. As that page says:

*“Another way to keep a process running after the shell exit is to use nohup.”*

*The nohup command executes another program specified as its argument and ignores all SIGHUP (hangup) signals. SIGHUP is a signal that is sent to a process when its controlling terminal is closed.”*

To run a command in the background using the nohup command, type:

```
nohup command &
```

If you want to log any terminal output to a file, you can use one of these commands below.

To redirect standard output and standard error to different files:

```
nohup myprogram > myprogram.out 2> myprogram.err
```

or to the same file:

```
nohup myprogram > myprogram.out 2>&1
```

## 4.2 Troubleshooting

---

Jetstream2 requires that you be on a [valid ACCESS allocation](#). If you are having issues accessing the various Jetstream2 interfaces, please first verify that you're on a valid allocation via the [ACCESS allocations summary page](#)

Troubleshooting Jetstream2 Interfaces:

- [Exosphere](#)
  - [Cacao](#)
  - [Horizon](#)
  - [CLI](#)
- 

### 4.2.1 General Troubleshooting:

**My instance/image that was migrated to a bootable volume won't mount on Exosphere**

Most instances that were migrated from Jetstream1 were partitioned whereas volumes on Jetstream2 are generally the entire device. What this means is that in almost all cases, migrated instances will have their information on `/dev/sdb1` assuming it's the first volume you're attaching to an instance.

Exosphere, in the quest to keep things simple, only tries to mount the entire devices, e.g. `/dev/sdb`, which means it won't find the data on `/dev/sdb1`.

The best way to work around this is to use Horizon to attach the volume and then do

```
sudo mount /dev/sdb1 /mnt
```

and you'll be able to see the contents of the volume in `/mnt`

**Ubuntu 22 Snaps: “not a snap cgroup”**

When attempting to launch applications that rely on [Snaps](#) you may receive something similar to the following error message:

```
/user.slice/user-1001.slice/session-5.scope is not a snap cgroup
```

This is a known issue with the Ubuntu 22 distribution. You can read about the bug report and its status in [this Ubuntu launchpad](#). Software delivered via snap may not be reliable at this time, especially web browsers and other graphical applications.

For the time being, the only known workaround is to install software via other means if possible; for example, with `apt`, a `.deb` file, or built from source.

#### 4.2.2 Resolved Issues:

The following are solutions to historical known issues (now resolved), preserved for the sake of archival:

##### In Ubuntu 20 or 22 web desktop, I can't load items from the JS2 Software Collection.

This issue should only affect instances created on or before June 10th, 2022, and has been resolved in current featured images.

The [Jetstream2 Software Collection](#) requires `Lmod` modules to work. By default, the Gnome terminal in the the Ubuntu web desktop did not act like a login shell – meaning it didn’t source the normal Bash login/environment files setting up your path and other environment variables.

You may be able to fix this in the terminal preferences.

With Terminal as the active application:

- Go to Edit -> Profile Preferences.
- Select the “Unnamed” Profile
- Select the **Title and Command** tab.
- Check the “Run command as login shell” checkbox

You’ll need to start a new terminal window, but that new session should allow you to do commands like `module avail`

##### Firefox doesn't work in Ubuntu 22

This issue should only affect instances created on or before October 10th, 2022, and has been resolved in current featured images.

This was a known issue with the Ubuntu 22 distribution, related to the above: [Ubuntu 22 Snaps: “not a snap cgroup”](#)

If you are affected, the only easy workaround is to use another web browser. You can install Chrome from a terminal command line as one possibility:

```
 wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb sudo apt install ./google-chrome-stable_current_amd64.deb
```

Chrome can also be launched from the command-line by typing `google-chrome`. You can also launch it from the activity bar.

#### I can't ping or reach a public/floating IP from an internal, non-routed host

This is a change in the network between Jetstream 1 and 2. You cannot presently ping a floating IP address from internal, non-routable IPs. You can only talk from non-routed nodes to bastion hosts and floating IPs only work from other routable hosts.

To communicate from an internal only Jetstream2 instance to an instance on Jetstream2 with a floating IP, you'll need to use the internal networking address or name.

## 4.3 Jetstream2 Allocations FAQ

### I was added to an allocation but I get an error like this trying to log in

To use Jetstream2, your account must belong to an ACCESS allocation that is active on a Jetstream2 resource. Please visit <https://docs.jetstream-cloud.org/faq/general-faq>

The allocation updates and user additions are created by ACCESS and synced to Jetstream2 (and other ACCESS resources). It can take up to four hours for users added to an allocation to become active. This is outside of our control. If you are still unable to log into [Exosphere](#) or [Horizon](#) after four hours from being added, please open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#).

### Use ACCESS CI credentials if you're having trouble authenticating

We know ACCESS-CI credentials work correctly and recommend using those unless you are 100% sure you have linked credentials PREVIOUSLY in CILogon.

Please note that other CILogon credentials (i.e. for your home institution or Google) must be set up outside of this process. Follow these [Linking instructions](#) to link your home institution or other organization (e.g. Google, ORCID, etc) credentials.

### Is there an overview of the types of allocations available as well as any restrictions those allocations have?

The [Getting Started guide](#) describes the overall allocations process and using ACCESS resources. To review the types of allocations ACCESS and the process to get an allocation, here are some links you might find useful:

- [Allocations overview](#)
- [Allocation Details](#)
- [Allocation Requirements & Request Instructions](#)
- [Allocation policies](#)
- [Allocations Management Portal](#)
- [Allocation How-To Guides](#)

**Is there a example or demonstration of how to get an allocation that I could follow?**

The requirements for allocations have changed considerably with ACCESS. An overview of the details and differences between allocation tiers can be found on the [Allocation Details](#) page.

Instructions on how to apply to each tier as well as their requirements can be found on the [Allocation Requirements & Request Instructions](#)

- Explore ACCESS - up to 400,000 credits, basically requires an abstract and other standard items listed on the Explore ACCESS page.
- Discover ACCESS - up to 1,500,000 credits, requires an abstract, one page description, and other standard items listed on the Discover ACCESS page.
- Accelerate ACCESS - up to 3,000,000 credits, requires an abstract, three page description, and other standard items listed on the Discover ACCESS page.
- Maximize ACCESS - most closely resembles XSEDE research allocations. Submission periods are twice a year and have requirements of up to 10 page main document, additional code and scaling performance documents, and the other standard materials like CVs required by all requests.

Presently there are no sample documents though we expect those will materialize in time. That said, with the requirements for documentation for all but the largest allocations being up to three pages, following the ACCESS pages linked above for submissions should yield all of the information you need.

---

**How do I let other ACCESS accounts use my allocation?**

You can add users to (or remove them from) your ACCESS allocation via the ACCESS Allocations Portal. Users must have already created their ACCESS accounts before they can be added to an allocation.

To add users to, or remove them from, an active Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support ([ACCESS](#)) allocation, the principal investigator, co-principal investigator, or can use the ACCESS tools here (*requires ACCESS login*):

- [Allocations Management Portal](#)
- See the [Allocation How-To Guides](#) for more information

**Note**

***Please do note that only the PI is automatically added to resources like Jetstream2 resources. This is a function of ACCESS' allocations system and is outside of our control. The PI, co-PI, or resource managers will explicitly need to add users to resources via the User Management link above***

**It can take up to four hours for users added to an allocation to become active.**

---

**Can I keep renewing my allocation(s)?**

Unlike XSEDE, ACCESS does not differentiate between most allocation types. You can renew all allocation types now, though **Maximize ACCESS** allocations are only renewable during the enrollment periods noted in the **Maximize Access** section of the [Preparing your ACCESS Request](#) page.

---

**I'm running out of Service Units (SUs) or storage. How do I request more?**

If you already have an ACCESS allocation and need to request additional service units (SUs), the PI, co-PI, or delegate may submit a request via the ACCESS Portal. For instructions on how to submit the request, see [Requesting additional SUs, other Jetstream resources, or storage for Jetstream – Supplemental Allocations](#).

**Note**

Supplements only add SUs/storage. If you need more time on your allocation, you need to pursue a renewal or extension.

**I am at or exceeding the quota limits for my allocation.**

How do I request additional resources such as CPUs and memory?

You may contact [help@jetstream-cloud.org](mailto:help@jetstream-cloud.org) or open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#) with those requests.

It's important to note that Jetstream Trial Allocation quotas are fixed and will **NOT** be increased under any circumstances.

For other allocation types, justification will be required and will be granted at the discretion of the Jetstream2 staff based on the justification and available resources. Please note that large memory and GPU resources are limited so requests for those will require strong justification for success or partial success. We strive to make resources available to all researchers that require them, so striking a balance between the needs of one versus many is often necessary.

**Can you extend my allocation for me or give me access to my allocation for just a few days/weeks/months more?**

If your allocation is expired or out of SUs, you may request an extension, renewal, or supplement. Please see one of the following links:

- [Requesting additional SUs, other Jetstream resources, or storage for Jetstream – Supplemental Allocations](#)
- [Jetstream2 Allocation Extensions and Renewals](#)

Jetstream2 staff are unable to take these actions on your behalf.

**Note**

Supplements only add SUs. If you need more time, you need an extension or renewal (preferred).

Extensions only add time. If you need more or different resources, you need to pursue a supplement.

For allocations that are about to expire or have expired, ***renewals are ALWAYS preferred over extensions***

---

**My PI got a Jetstream2 supplement added to our existing allocation but we cannot access Exosphere or Horizon**

When you get a supplement for a resource that you have not previously had, users on your allocation are not automatically enabled for it. To enable users for a new resource like Jetstream2, your PI, Co-PI, or resource managers may need to go to their [Allocations Management page](#) and select the appropriate allocation to manage.

**I have an Explore ACCESS allocation but need to move to a larger Discover or Accelerate allocation size. How do I do that and keep my allocation number (and VMs/volumes in my current allocation)?**

When you are ready to move up to the next ACCESS Opportunity, please keep a few things in mind.

1. First make sure you use up all your ACCESS Credits. Please submit an Exchange Request if you have any remaining ACCESS Credits.
2. Next, you will want to focus on using any SUs you have left on the resources.
3. When you get close to using up all your SUs, please submit a ticket to the [ACCESS Allocations Team](#) when you have about 90% of the resources used. The Allocations Team will then help you move up to the next ACCESS Opportunity while keeping the same allocation number.

## 4.4 GPU Related FAQ

### There is a known issue with suspending GPU instances

We will update this [Status IO Incident](#) with details/

There is an issue/bug with suspending GPU instances with the version of libvirt Jetstream2 is using for virtualization.

DO NOT SUSPEND GPU instances.

We will have to upgrade the compute nodes to resolve it. This is on the near-term timeline but we do not have a precise date at this time.

In the meantime, please only use stop or shelve with GPU instances.

### I don't have GPU instances available to me when I go to launch an instance

Jetstream2 consists of [four distinct resources](#). You must explicitly have access to the Jetstream2-GPU resource to access GPUs. Having access to Jetstream2 (CPU) does not give you access to GPUs. We also highly encourage you familiarize yourself with the [VM instance sizes/flavors](#) and note the difference in burn rate (SU cost per hour).

### How do I use multiple GPUs on an instance for my research?

The short answer is that you cannot use multiple GPUs on a single instance at this time.

The longer answer is that this is a limitation of the NVIDIA GRID vGPU driver for our hypervisors. Basically, even with NVLINK present, the driver cannot gang multiple GPUs together into a single VM. Recent updates indicate that we may be able to use multiple *fractional* vGPUs on an instance. Engineers are currently looking into this and we will update this FAQ and the documentation overall accordingly if there is a means to do this.

### CentOS 7 does not work with my GPU

Due to issues with the NVIDIA GRID driver, we have discontinued support for GPUs using CentOS 7. We will be removing CentOS 7 from the featured images once we have a stable Rocky 9 build available.

## My GPU is not usable after a kernel update

The NVIDIA drivers are built as kernel modules and **should** rebuild on a kernel update. If they do not, you can do this on Ubuntu 20.04 instances:

```
ls /var/lib/initramfs-tools | sudo xargs -n1 /usr/lib/dkms/dkms_autoinstaller start
```

For Ubuntu 22.04 instances, you can try:

```
ls /usr/lib/modules | sudo xargs -n1 /usr/lib/dkms/dkms_autoinstaller start
```

This doesn't work on redhat-based instances like Rocky or Alma. We're working on a simple solution for that.

## The CUDA debugger (cuda-gdb) doesn't work on GPU instances

If you use the nvhpc module with nvcc compiler and try to use the cuda-gdb debugger, you will get an error like this:

```
fatal: One or more CUDA devices cannot be used for debugging
```

There is an issue with vGPU and our configuration that cannot be readily resolved. We are looking into options to work around this problem. There is no estimate for when a workaround will be in place. We apologize for any inconvenience.

## Unified memory doesn't work on GPU instances

We can confirm that unified memory is not working under the NVIDIA drivers we're using. We have reached out to NVIDIA for a timeline on when we might expect that functionality.

Regardless of the time, it is not expected that unified memory will work on fractional (slices of) GPUs, only on full GPU flavors.

We will update this FAQ entry when we have additional information.

## Is nvcc/CUDA available on the images or in the software store ?

The [NVIDIA HPC SDK](#) is available from the [Jetstream Software Store](#).

You can do

```
module avail
```

on featured imgs to see available software packages. You should see several with names like ***nvhpc*** that will have the HPC SDK software.

For other GPU software, where possible, we highly recommend using containers from NVIDIA if they are available. The [NVIDIA Docker Container Catalog](#) is the repository.

### What CUDA version do I need for Jetstream2 GPUs ?

We recommend using the same major revision as reported by `nvidia-smi`; however, NVIDIA maintains that CUDA versions are backward compatible, up to one major revision ago. For example, if `nvidia-smi` reports:

```
+-----+ | NVIDIA-SMI 525.60.13 Driver Version: 525.60.13 CUDA Version: 12.0 | +--
```

then it is “safe” to use CUDA 11.x, though CUDA 12.0 is recommended. In this example, CUDA 10.x and older will not work.

## 4.5 Security FAQ

---

### Do I need to patch/update my VMs?

The featured images provided by the Jetstream team have unattended security updates enabled. Instances will not reboot, but they will apply any update marked as a security update. It's still a good idea to update your VM periodically.

CentOS 7: `sudo yum update`

Rocky 8: `sudo dnf update`

Ubuntu: `sudo apt-get update ; sudo apt-get upgrade`

If the kernel or glibc/libc packages are being updated, rebooting is necessary to implement those updates

Always run updates before taking snapshots so your new instances will have the latest patches in place.

---

### Do I need to secure services running on my VMs?

If you have a public IP (also called a floating IP in OpenStack terminology), your instance may be exposed to the internet. While you can use security groups or host-based firewalls like `iptables` or `UFW` to limit access, it's best to take a posture of defense in depth and always secure any listening services.

### We strongly recommend

- limiting access to any service ports via security group and/or firewall
- making sure any non-public services like databases have secure passwords for all accounts.
- If your service only needs to be accessible to the VM it's running on, check the documentation to see if it can be bound only to the localhost port. (Not all services have this ability.)

## Do I need to run a firewall on my VM?

Jetstream2 staff encourages a defense-in-depth approach to security. This potentially involves several methods of restricting access and securing instances.

Firewalls are **not enabled by default** on Jetstream2 instances. Depending on the user interface you launched your instance from, you may have different security groups established for your instance. (See [What is the default security profile for Jetstream2 VMs?](#) on this page for more information on that.)

We encourage keeping your instances patched, rebooting as needed for any kernel or glibc patches, limiting access to all services as much as possible, utilizing security groups if your interface allows it, and running your own host-based firewall if you're comfortable administering it.

Please refer to [Jetstream2 Virtual Machine Firewalls](#) for more information.

---

## What is the default security profile for Jetstream2 VMs?

That depends on the interface.

The CLI and Horizon by default allow egress only. You have to apply the appropriate security groups for ingress. Please refer to the [CLI](#) or [Horizon](#) for more information on managing security groups in those interfaces.

For Exosphere, the default security group allows all egress and inbound access.

For CACAO, the default security group will be announced when it is available for general usage.

---

## What is the *default* security group I can see in Horizon or from the CLI?

The *default* security group that you can see from Horizon or the CLI is provided by OpenStack for every allocation. It contains default egress rules for all VMs on that allocation and is automatically applied to each VM.

We highly recommend not changing the *default* security group at all. Security groups are additive, so you can create additional groups for the access you need and add them to your host at launch or at any time after launch.

Please refer to the [CLI](#) or [Horizon](#) for more information on managing security groups in those interfaces.

**Is there any security benefit to creating a new network or subnet for VMs in my project?**

Unlike physical networking, there is no real advantage to “isolating” hosts on their own subnet or network.

All traffic is filtered on the hypervisor and must be explicitly allowed by security groups even if the two hosts are on the same neutron subnet.

Only network traffic destined for ports on the specific hypervisor a VM is on will even make it to that hypervisor. Since the networking happens at a layer above the virtual machines, promiscuous mode sniffing will only reveal traffic that was already destined for the VM(s) in question.

We recommend using the `auto_allocated_network` and `auto_allocated_subnet_v4` that are automatically created for each allocation. If you do not specify a network or subnet when creating a VM, it should automatically use those via an OpenStack configuration/feature.

---

**In Exosphere, there is a way to get a passphrase for any instance. Can I prevent other users on my allocation from accessing my instance(s)?**

The way Jetstream2 is currently architected, all users on an allocation have access to all resources on the allocation. By default, Exosphere hides some resources created by other users, but this is only a convenience and it cannot assure separation of access.

It is possible to make it less straightforward for another user on the same allocation to access your running instance. You can do this by changing the password for the default exouser account. Changing the password does not prevent any access, but makes it more difficult.

Note that currently, changing the exouser account password will break Web Shell, Web Desktop, and some other Exosphere-powered instance interactions. This may change in the future.)

We note how to change a user password here:

<https://docs.jetstream-cloud.org/faq/general-faq/#can-i-set-the-password-for-a-user-on-my-virtual-machine>

you can do:

```
sudo passwd exouser
```

and that solves the issue of them being able to access your instance using the credential listed on the Exosphere page. While you can use the console option still, we HIGHLY suggest utilizing ssh keys for your instances to ensure you have access. That's covered here:

[https://docs.jetstream-cloud.org/ui/exo/create\\_instance/](https://docs.jetstream-cloud.org/ui/exo/create_instance/)

under the advanced options. You can also manually add your key to an already running instance.

## 4.6 Software FAQ

---

Jetstream2 provides some software packages for all researchers. The sheer number of available packages across all of the scientific disciplines makes it impossible to meet the desires of everyone in installing and maintaining software for research.

---

### **How can I see what software is maintained by the Jetstream2 staff?**

The Jetstream2 software collection is documented here : [Software Collection](#)

---

### **How do I access the software in this collection?**

This software store is mounted automatically when an instance is launched from a [Jetstream2 featured image](#). Please see the following links about using the Jetstream Software Collection:

- [Using the Jetstream2 Software Collection from the command line](#)
  - [Using the Jetstream2 Software Collection from the web desktop](#)
- 

### **Can I request to have other software packages added to the software collection?**

Since everyone on Jetstream2 allocations potentially has root access on their VMs, you may install any software you need on a per VM basis. You can then snapshot and save your customized images.

If you feel a package would be broadly useful to the Jetstream User community, you may request software using the [contact form](#) and staff will review the package.

Please do keep in mind that we cannot add and maintain all requested software packages. We will review any requests and track those requests.

## 4.7 Science Gateways and related services FAQ

### How can I federate logins on my gateway or JupyterHub service?

The recommended and supported means for doing so on Jetstream2 is to use the Custos software provided by the [Cyberinfrastructure Integration Research Center](#)

Information on federating your gateway or Jupyter service is outlined in on [Federating a gateway or JupyterHub on Jetstream2](#)

### How do I set up a custom domain name for my gateway, site, or web service?

If you want to access your site/service with a hostname (rather than a raw public IP address), you will need to set up appropriate DNS (Domain Name Service) records.

It is possible to configure automatic DNS hostnames for instances in the [projects.jetstream-cloud.org](#) subdomain; for more information, see [“How do I get a jetstream-cloud.org DNS name for my instance?”](#)

Second level “designer” domains like “myresearchproject.org” or “ourcoolproject.info” would need to be purchased from an [ICANN-accredited registrar](#). Many popular registrars will include DNS configuration in the price of a domain. Before purchasing a domain name, consider investigating whether your sponsoring university or organization offers such services; for example, projects affiliated with Indiana University may be able to obtain an [IU subdomain](#).

Once you have a domain registered, you will need to set up DNS records to point that domain to the public IP address of your Jetstream2 instance(s). If your service/site is served on the usual ports 80 and 443 for HTTP and HTTPS respectively, this can be accomplished with a simple [A record](#); otherwise, you may need to investigate [using SRV records to redirect traffic to a specific port](#).

## 5. General Usage Information

### 5.1 Jetstream2 Resources

As noted on the [Key Differences](#) page, Jetstream2 consists of four distinct ACCESS-allocated resources. These resources are:

- [Jetstream2 \(CPU only\)](#)
- [Jetstream2 Large Memory](#)
- [Jetstream2 GPU](#)
- [Jetstream2 Storage](#)

The specifications for each are linked above.

With the exception of Jetstream2 Storage, these resources may be allocated individually. Jetstream2 Storage requires the PI apply for or already have an allocation on any one or more of the three Jetstream compute resources.

#### Default Storage

By default, all allocations receive 1TB (1000 GB) of Jetstream2 Storage for volumes. If you will not need more than this, you do not need to request Jetstream2 Storage with your allocation request.

***Any additional storage must be justified in your allocation request.***

#### Limits and important notes:

- Storage defaults to 1 TB.
- There are no restrictions on runtime for the Jetstream2 CPU resource. As long as you have an active allocation and SUs remaining, you may run your Jetstream2 CPU VM(s) continuously or on demand.
- Jetstream2 GPU and Jetstream2 Large Memory may have runtime restriction placed at a future date. Present policy notes that Jetstream2 Staff may limit runtime on these resources to two weeks at a time. As long as resources are not in contention, we *may* opt to allow continuous running of VMs/services. Any change in resource limits will be noted via [ACCESS Infrastructure News](#) and other communications channels.

## 5.2 ACCESS Credits and Jetstream2

ACCESS allocations are measured in ACCESS credits. Those credits then may be converted to resource-defined service units.

Jetstream2 allocations Jetstream core hours or service units (SUs).

### ACCESS to Jetstream2 SUs

For simplicity, we've aligned our SU value such that **1 ACCESS Credit = 1 Jetstream2 SU**.

SUs are consumed at a rate of:

- Jetstream2 (CPU) - 1 SU per vCPU\_core-hour (use of one virtual core of a CPU per hour).
- Jetstream2-LM (Large Memory) - 2 SUs per vCPU\_core-hour
- Jetstream2-GPU - 4 SUs per vCPU\_core-hour

Please refer to [VM Sizes and configurations](#) to see available VM flavors and per hour cost on Jetstream2.

- **SUSPENDED** instances will be charged .75 of their normal SU value. (75%)
- **STOPPED** instances will be charge 0.50 of their normal SU value. (50%)
- **SHELVED** instances will not be charged SUs. (0%)

For Large Memory and GPU allocations, the vCPU core hour cost is 2x and 4x respectively as noted above.

The reason for continuing to charge for VMs that are not in a usable state is that they still consume resources if they are suspended or stopped. In those states, they still occupy allocable/usable space on the hypervisor, preventing other users from using those resources.

***If your VM is active, even if you are not logged in and using it, it is still being charged for use. If you do not wish to be charged, shelve your instance***

## Usage reporting

Usage reported to ACCESS is not in real time. There may be a delay of 12-24 hours for usage reporting.

For instructions on managing instances, please see:

- [Cacao instance management actions](#)
- [Exosphere instance management actions](#)
- [Horizon instance management actions](#)
- [Command line instance management actions](#)

## 5.3 Instance Flavors

---

Jetstream2 provides a range of instance flavors (also known as sizes), each under three different types with different capabilities. These instances consume an allocation's service units (SUs) according to how much physical hardware they occupy. Larger instance flavors consume SUs at a greater rate.

The unit of allocation for Jetstream is based on a virtual CPU (vCPU) hour: *1 service unit (SU) is equivalent to 1 vCPU for 1 hour of wall clock time*. The tables below outline the instance types and flavors for Jetstream2.

### Jetstream2 Resources

***The three instance types are all separate ACCESS resources. When you exchange ACCESS credits, you must select Jetstream2 (CPU), Jetstream2 GPU, or Jetstream2 Large Memory in order to use each of them. Having access to one does NOT include access to all.***

While the root disk sizes are fixed for each instance flavor, there is an option called “volume-backed” that allows you to specify a larger root disk, using quota from your storage allocation. Instructions for this are in the user interface-specific documentation for creating an instance ([Exosphere](#), [Horizon](#), [CLI](#)).

#### Jetstream2 CPU

Flavor	vCPUs	RAM (GB)	Local Storage (GB)	Cost per hour (SU)
m3.tiny	1	3	20	1
m3.small	2	6	20	2
m3.quad	4	15	20	4
m3.medium	8	30	60	8
m3.large	16	60	60	16
m3.xl	32	125	60	32
m3.2xl	64	250	60	64
m3.3xl*	128	500	60	128

\* m3.3xl are not available by default. This flavor is available by request and with proper justification.

## Jetstream2 Large Memory

Jetstream2 Large Memory instances have double the memory (RAM) of equivalently-resourced CPU instances. They cost 2 SUs per vCPU hour, or 2 SUs per core per hour.

Flavor	vCPUs	RAM (GB)	Local Storage (GB)	Cost per hour (SU)
r3.large	64	500	60	128
r3.xl	128	1000	60	256

## Jetstream2 GPU

Jetstream2 GPU instances include a partial or full NVIDIA A100 GPU, with up to 40 GB of GPU RAM. Jetstream2 GPU instances cost 4 SUs per vCPU hour, or 4 SUs per core per hour.

Flavor	vCPUs	RAM(GB)	Local Storage (GB)	GPU Compute	GPU RAM (GB)	Cost per hour (SU)
g3.small	4	15	60	20% of GPU	5	16
g3.medium	8	30	60	25% of GPU	10	32
g3.large	16	60	60	50% of GPU	20	64
g3.xl	32	125	60	100% of GPU	40	128

Note: If you are using a partial-GPU flavor, and the remainder of the underlying physical GPU is idle, your instance may provide higher compute performance than the flavor strictly allots. In other words, the GPU compute for each flavor is a minimum value, while the GPU RAM is a maximum value.

**Jetstream2 instance types, flavors, and associated policies are subject to change in the future.**

**Example of SU estimation:**

Note: You can now estimate your SU needs using the usage estimation calculator here: [Usage Estimation Calculator](#)

- First determine the compute resource appropriate to your needs (CPU only, large memory, GPU):
  - If your work requires 24 GB of RAM and 60 GB of local storage:
    - you would request 8 SUs per hour to cover a single *m3.medium* instance.
  - If your work requires 10 GB of local storage in 1 core using 3 GB of RAM:
    - you would request 2 SUs per hour for an *m3.small* instance.
  - If your work requires 1TB of RAM:
    - you would request 256 SUs per hour for an *r3.xl* instance on Jetstream Large Memory
  - If you work requires 20 GB of GPU RAM:
    - you would request 64 SUs per hour for a *g3.large* instance on Jetstream GPU
- You then would calculate for the appropriate resource (refer to the tables above):
  - For Jetstream2 CPU, you would then multiply by the number of hours you will use that flavor instance in the next year and multiply by the number of instances you will need.
  - For Jetstream2 Large Memory and GPU, either refer to the SU cost per hour in the last column, or multiply hours times 2 for LM or 4 for GPU
- To calculate the number of SUs you will need in the next year, first estimate the number of hours you expect to work on a particular project. For example, if you typically work 40 hours per week and expect to spend 25% of your time on this project that would be 10 hours per week.
- Next, calculate the total number of hours per year for this project:
  - Total hours = 10 hours per week \* 52 weeks per year
  - Total hours = 520
- Finally, calculate the total SUs for the year for a single instance:
  - Total SUs = 520 hours per year \* vCPUs
    - e.g. For a Medium instance: Total SUs = 520 hours per year \* 8vCPUs
    - Total SUs = 4160
- If your project requires more than 1 instance, multiply the total SUs by the number of instances that you will need:
  - Total SUs needed for 3 medium flavor instances = 3 \* 4160
  - Total SUs = 12480

The calculations above assume that your instance is shelved when not in use. For instructions see:

- [Cacao instance management actions](#)
- [Exosphere instance management actions](#)
- [Horizon instance management actions](#)
- [Command line instance management actions](#)

#### SU Estimation for Infrastructure or “Always On” allocations

For jobs that may need to run for extended periods or as “always on” infrastructure, you can take this approach:

instance cost (SUs) x 24 hours/day x 365 days = single instance cost per year

or as an example for each resource, an m3.large, r3.large, and g3.large each running for a year:

`m3.large (16 cores) x 24 hours/day x 365 days = 140,160 SUs r3.large (64 cores x 2 SUs/hour) x 24 hours/day x 365 days = 1,121,280 SUs g3.large (16 cores x`

## 5.4 Quotas

Jetstream2 tries to apportion the memory, storage, and network available to researchers based on both allocation type and Jetstream2 resource node type.

### 5.4.1 Limits

#### Jetstream2 quotas

\*\*\*Please note that these quotas are currently under review as we determine how to make them best work with the new allocation types. We will update this page once we do so. \*\*\*

Quotas are a function of: [allocation](#) and [resource](#)

**allocation type:**

Startup/Research	Education	Jetstream Trial Allocations (JTA)
default allocations for collaborative users in a common research project.	special allocations for teaching and workshops. Quotas are set slightly higher because of typical larger user counts	Sample allocations with minimal capacity quotas designed for evaluation purposes

**resource type:**

CPU (compute)	GPU	LM (large memory compute)
normal bulk compute resources with higher available core counts	GPU accelerated resource with additional ceph storage	2X-memory compute resource with additiona ceph storage
75% of JS2	19% of JS2	6% of JS2
3.9 GB per RAM per core	7.8 GB RAM per core	3.9 GB RAM per core
<b>STARTUP/RESEARCH</b>	<b>defaults *</b>	
<input type="button" value="CPU"/>	<input type="button" value="GPU"/>	<input type="button" value="LM"/>
150 cores	64 cores	128 cores
585 GB RAM	250 GB RAM	1000 GB RAM
10 floating IPs	4 floating IPs	2 floating IPs
<b>EDUCATION</b>	<b>defaults *</b>	
<input type="button" value="CPU"/>	<input type="button" value="GPU"/>	<input type="button" value="LM"/>
200 cores	320 cores	128 cores
780 GB RAM	1250 GB RAM	1000 GB RAM
25 floating IPs	25 floating IPs	2 floating IPs
<b>JTA</b>	<b>defaults</b>	
<input type="button" value="CPU"/>	<input type="button" value="GPU"/>	<input type="button" value="LM"/>
2 cores	none	none
6 GB RAM		
2 floating IPs		

\* modifications available with proper documentation

## 5.4.2 Other Limits

[Jetstream2 policies](#) may also impact quotas, particular regarding networking.

## 5.5 Instance Management Actions

---

Please orient yourself to the following instance management actions. These will help you use Jetstream2 effectively and conserve your allocation.

Each user interface for Jetstream2 has instance actions in a slightly different place – see guidance for [Exosphere](#), [Cacao](#), [Horizon](#), and the [OpenStack CLI](#).

### 5.5.1 Basic Actions

---

#### Shelve and Unshelve

**When your instance is not performing work or otherwise in active use, please *shelve* it.** Shelving an instance shuts it down and frees up resources on the cloud for other users. It also conserves the SUs (service units) on your allocation.

Shelving an instance shuts down its operating system. The instance's disk contents are preserved, but any running programs will exit, so please save any unfinished work before shelving.

Shelving and unshelving each take a few minutes, so shelving doesn't make sense for very short periods of inactivity. In other words, shelve your instance when you're done for the day or the week, not merely for your lunch break.

A shelved instance will not accept shell, SSH, or any other connections. So, if your instance runs a server that you want to provide others the ability to connect to at any time, you must leave it active. If this describes your instance, consider re-sizing it to the smallest flavor that will work for your server needs. This will conserve SUs on your allocation.

#### Lock and Unlock

Locking an instance helps prevent anyone from accidentally deleting or performing other actions on it, from Exosphere and all other Jetstream2 interfaces. If your instance is running an important job or used in 'production' capacity, consider keeping it locked. You must unlock your instance again before performing other actions (such as shelving it). Locking and unlocking are non-disruptive actions – they do not affect a running instance.

Be aware that locking an instance does not prevent:

- another user on your allocation from unlocking it.
- modifications to the instance's filesystem(s) or running software. For example, someone with access to the instance could still log in and delete files. Locking only prevents instance actions at the cloud (OpenStack) level.
- the instance from shutting off when your allocation expires or is exhausted.

## Reboot

Rebooting an instance is just like restarting a computer. The cloud will first attempt a graceful or “soft” reboot, where all of your programs are allowed to exit. If that fails then OpenStack will perform a “hard” reboot, which will lose any work that is not yet written to disk. If you cannot connect to your instance, rebooting is a good troubleshooting step before creating a support ticket.

---

### 5.5.2 Advanced Actions

The following actions are for more sophisticated use cases. If you're a new cloud user, it's okay to skip reading about these for now.

## Resize

Resizing allows you to choose a different flavor for your instance. When you resize, your instance will shut down and then restart with the new flavor (so please save any work in progress first). You can now resize using Exosphere, Horizon, or the CLI.

Consider resizing if you find yourself in one of these situations:

- Your instance exhausts its compute resources, e.g. you run out of working memory (RAM) or you want it to process work faster.
- Your instance's CPU is sitting idle most of the time, in which case a smaller flavor would burn your allocation more slowly.
  - Exosphere's instance resource usage graphs are a useful guide here.
- You launched a GPU flavor, then later find that you no longer need the GPU, but want to keep using the instance.
- You launched a non-GPU flavor, then later find that you want to use the same instance with a GPU.

If your software stack sometimes needs a large flavor to run a compute-intensive job, but you can develop and tune it on a smaller flavor, consider resizing *down* to a small flavor for development work, and back *up* when you’re ready to run it at a larger scale. This gets you best performance when you need it, while conserving your allocation when you don’t.

Moving to a larger flavor is generally **not** appropriate in these situations:

- The speed of your workload is limited by a process that is single-threaded (not parallelized). If this process cannot be parallelized then resizing is unlikely to speed it up.
- The speed of your workload is limited by disk or network transfer speed. Larger instances do not have faster storage or network connectivity.
- Your instance is running out of storage. Instead, create a volume, attach it, and move your data to the volume. If you’re installing a lot of software that is not easily moved to a volume, resizing *may* be appropriate – open a ticket and ask for advice.

When resizing, you must select your desired new flavor. After the resize is complete, the instance will be in status “Resize verify”. At that time, access the instance (e.g. using Web Shell or SSH) and confirm that it is working, then choose the “Confirm resize” action. If the resize process broke something and you need to return to the previous flavor, choose the “Revert resize” action.

## Image

When you create an image, you capture the entire state of your instance’s root disk. New instances can be launched from that image, which means that you can use an image to ‘snapshot’ and ‘clone’ an instance.

After you specify the image name, it will take several minutes for the image to finish creating and become active.

Consider creating an image in the following situations:

- You want to create a new instance that is a *clone* of your existing instance. In this case, create an image of the existing instance and launch your clone(s) from that image.
- You are about to perform a possibly destructive action on your instance (like installing, upgrading, or removing software), and you need the ability to go back and get to the prior disk state of the instance if something goes wrong.
- You are building a software stack that other people will consume via their own instance (e.g. you are teaching a class). Providing an image can be an easy way for other people to get a new instance just like yours.

Be aware that system images quickly fall behind on operating system updates. As more time passes since an image was created, the more software will need to be updated when a new instance is created for it. This can lead to excessively long instance launch times and other problems. For this reason, custom images are not the right tool for sharing software or workflows more than a few months into the future. If this describes your situation, please open a support ticket and ask for advice.

### CAUTION:: Image Sharing

You can set the **visibility** property to `community` (all can see and boot), `shared` (only those you specify can see and boot), `private` (only your allocation can see and boot). Only in VERY special cases will Jetstream2 allow `public` visibility, such as staff-featured images. Limiting the number of fully `public` images in the catalog improves Jetstream2 reliability and performance.

For more information on Glance **visibility** properties, see <https://wiki.openstack.org/wiki/Glance-v2-community-image-visibility-design>

Currently, **visibility** can only be modified in the **Horizon** and **CLI** interfaces.

### Suspend and Resume

Suspending an instance is like placing a computer on standby (a.k.a. sleep). When you resume the instance, all running programs will be in the state they were in prior to entering standby. (Still, it is wise to save any work in progress before suspending.)

Please consider suspending instead of shelving *only* if you are running software that was complex or labor-intensive to *start* (not install), and you only need to leave it suspended for a relatively short time (e.g. a few days). Suspended instances still occupy resources on the cloud, and they continue to consume your allocation at a reduced rate.

## 5.6 Featured Images

Jetstream2 will have a limited set of featured images. These are images that the Jetstream2 team curates and maintains.

A key difference between Jetstream1 and Jetstream2 is that there will not be application specific featured images. We will maintain a [software collection](#) that will be made available on every virtual machine at boot. Software will be loaded via [Lmod Modules](#).

At this time, the featured images will be:

- Ubuntu 22.04
- Ubuntu 20.04
- Rocky 8 / Alma 8
- CentOS 7 (*will retire soon – will NOT support GPU operations*)

All featured images are named ***Featured-yyyyyyyy*** (e.g. Featured-Ubuntu22) on Jetstream2.

These featured images will evolve over time. As distributions leave support (e.g Ubuntu 18 will end support in April 2023), we will replace them with newer, supported versions. NVIDIA drivers will be present on all featured images so any of the featured images will work with Jetstream2 GPUs.

### CentOS 7 and GPUs

Due to issues with the NVIDIA GRID driver, we have discontinued support for GPUs using CentOS 7. We will be removing CentOS 7 from the featured images once we have a stable Rocky 9 build available.

Our goal is to maintain a minimum of featured images but keep them updated via automated pipeline on a weekly basis.

## 5.7 Microsoft Windows on Jetstream2

Microsoft Windows is not officially supported on Jetstream2.

We will be making a limited number of images available for experimental use.

While these images will be created by one of the Jetstream2 parters in conjunction with Jetstream2 staff, it is NOT supported nor guaranteed to work. It is unlikely that the Jetstream2 staff will be able to do more than general VM troubleshooting as we do not have Windows expertise on staff. No license will be installed so you will need to bring your own license.

It is not known at this time whether GPUs will work on Microsoft-based instances. We will test this as time permits.

### 5.7.1 Creating an Instance

As noted above, Jetstream2 staff makes a limited number of Microsoft Windows images publically available. Instances can be spawned from these images with some considerations:

- In order to be able to access your instance later, you will need to provide a **brand-new** SSH keypair in PEM format. One can be generated with `ssh-keygen -m pem` in an OpenSSH-enabled terminal. Be sure to set a strong passphrase for now.
- Be sure that your root disk is large enough to deploy the image; Windows tends to be larger than most Unix/Linux images. The default 20 GB size of some flavors is too small. To get around this, either select a different flavor or use a **volume-backed instance**.
- Creating a Windows instance through Exosphere will cause the resulting instance to appear to be stuck in the “Building” status indefinitely, and many of the platform’s creature comforts like `exouser` and resource monitoring will not function properly, if at all.

### Image information

Presently the image supplied by Jetstream2 is

**Windows-Server-2022-JS2-Beta | bbad1676-40f1-485d-8575-ca9eac7e211e**

### 5.7.2 Retrieving the Admin Password

In order to retrieve the password for a Windows Server instance, an SSH keypair is **required** at setup. If you did not provide your instance with a keypair, you will be locked out, since recovering the Windows password without one is impossible.

You can find the password for your Windows instance via Horizon. Navigate to your [instance's management actions dropdown](#) and select “Retrieve Password.” To decrypt the password, you will need to provide your SSH keypair’s private key file or contents. In order for Horizon to recognize the private key, it must have no passphrase and be in PEM format (it should start with `-----BEGIN RSA PRIVATE KEY-----`). It is, however, bad practice to leave your SSH keys unprotected, so we recommend **temporarily** removing the passphrase, retrieving your instance’s password, then re-setting a passphrase on the key:

### SSH Private Keys are Secret.

Horizon processes your private key locally only, and does not transmit it across the internet. Just like a password or passphrase, your private key should **never** be shared, even with good intentions. Jetstream2 support personnel will **never** ask you for your private key.

1. Remove the passphrase on your key and ensure it is in PEM format (when prompted for a passphrase, simply hit `enter / return`):

```
ssh-keygen -p -f <PATH_TO_PRIVATE_KEY> -m pem
```

2. Select your key file in Horizon or paste its contents directly into the text box, then click “Decrypt Password.”
3. Take note of the resulting password.
4. Reset the passphrase on your private key to something strong:

```
ssh-keygen -p -f <PATH_TO_PRIVATE_KEY>
```

### 5.7.3 Accessing a Windows Instance from Horizon

On the left side bar, navigate to [Project](#) → [Compute](#) → [Instances](#), then click on your instance’s name. Navigate to the [Console](#) tab near the top.

The screenshot shows the Jetstream2 web interface. In the top navigation bar, 'Project' is selected under 'Compute'. Below it, 'Instances' is highlighted in blue. The main title is 'zegraber-windows-test'. Underneath, there are tabs: Overview, Interfaces, Log, **Console**, and Action Log. The 'Console' tab is circled in green. Below the tabs, the title 'Instance Console' is displayed. A note says: 'If console is not responding to keyboard input: click the grey status bar below. [Click here to show only console](#). To exit the fullscreen mode, click the browser's back button.' Below the note is a screenshot of the RDP console window, which is dark and shows the SPICE logo. A green arrow points from the word 'Console' to this window.

## 5.7.4 Accessing a Windows Instance with Remote Desktop (RDP)

**Note:** like SSH, RDP requires your instance to have a public (Floating) IP associated with it in order to access from outside the Jetstream2 network.

Windows Remote Desktop Protocol (RDP) will most likely provide a better, more feature-rich experience than the Horizon (SPICE) console.

### Adding a security group for RDP

Under normal circumstances, RDP listens for connections on TCP port 3389 and may be accelerated by exposing UDP port 3389. If TCP port 3389 is not exposed on your instance, you will be unable to connect with RDP. Luckily, Horizon provides a prefabbed security rule for RDP access.

You can add this rule to an existing [security group](#) or create a new one, then apply the security group to your instance by navigating to “Edit Security Groups” in the [management actions dropdown](#).

## Minimize security risks.

We **strongly** advise limiting remote access to RDP port(s) to the smallest possible CIDR or, even better, single IP(s), rather than the default `0.0.0.0/0` (which allows access from any public IP); an improperly configured RDP service could allow remote access to a host in an unintended manner in addition to exposing SSL/TLS certificate information.

### Add Rule



Rule \*

#### Description:

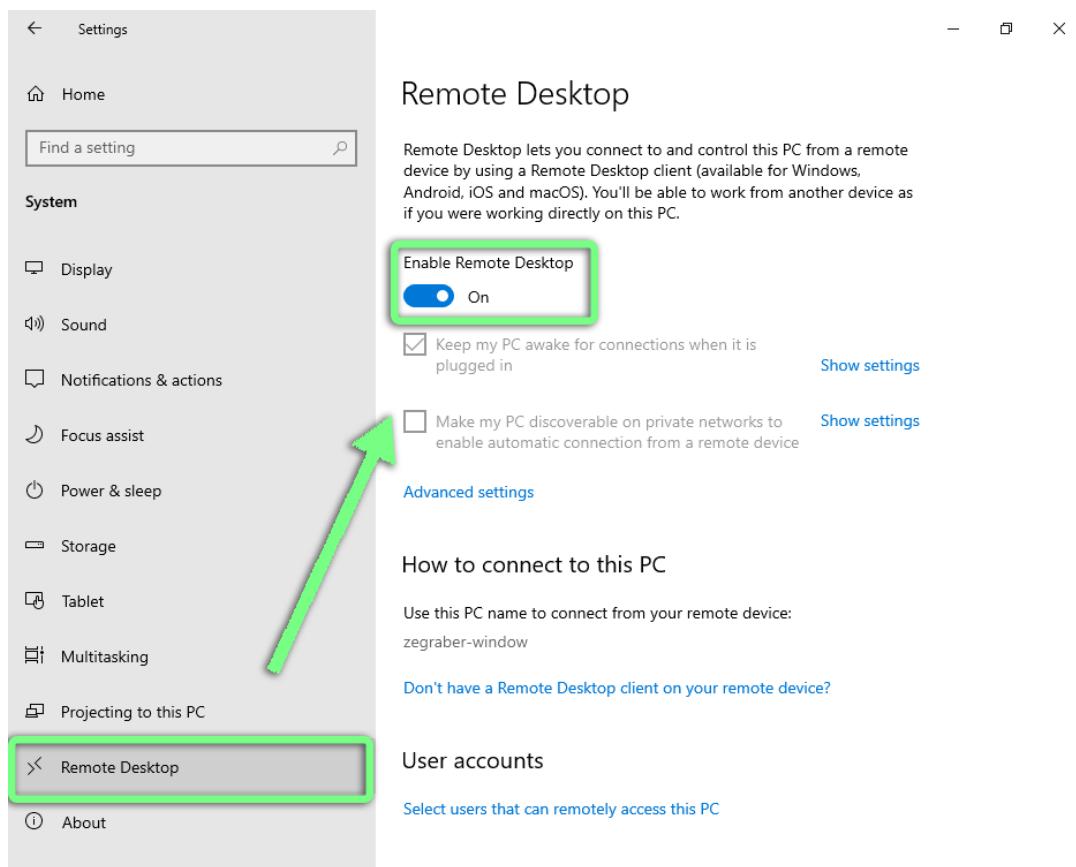
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

**Rule:** You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Description ⓘ

### Enabling Remote Desktop

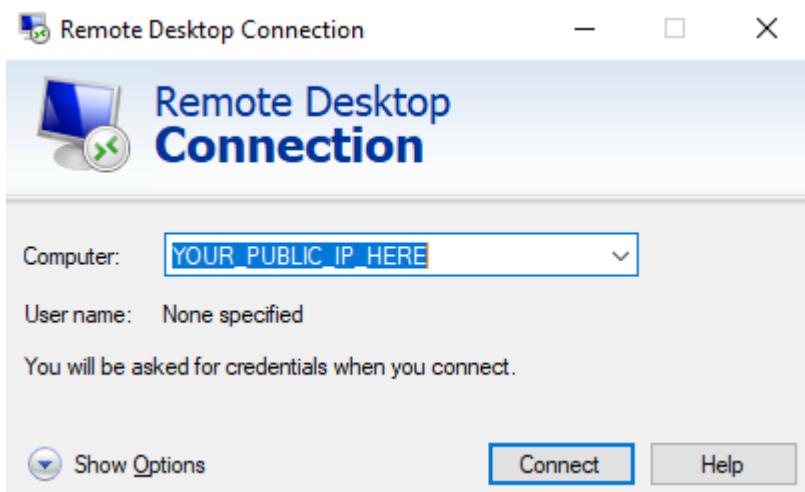
Remote Desktop may need to be enabled on your instance before attempting to connect. First, access your instance through Horizon, as [described above](#). Open the machine's Settings app, then navigate to the "Remote Desktop" section on the left. Ensure that "Enable Remote Desktop" is set to "On."



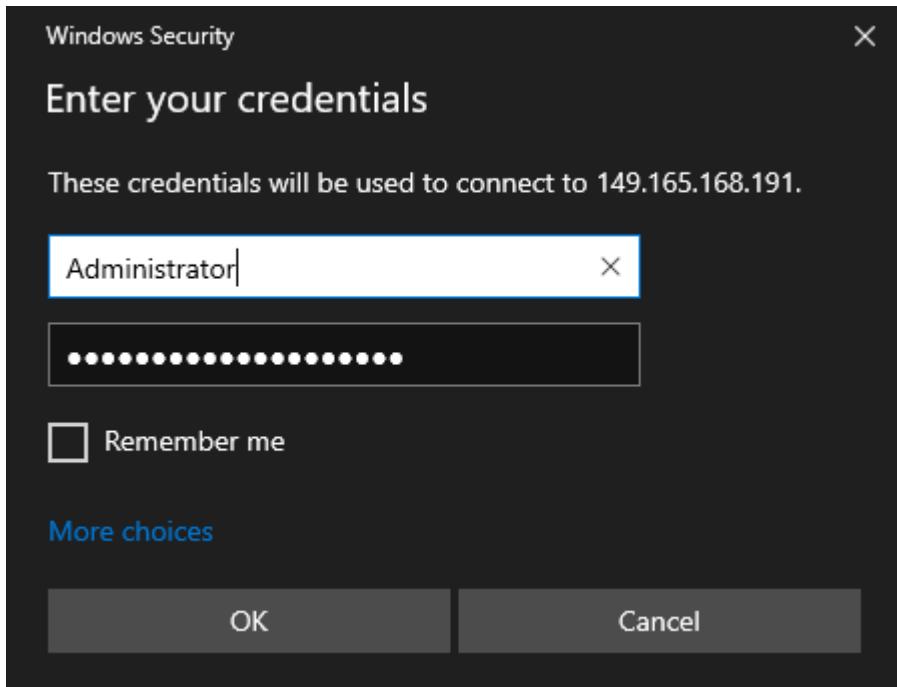
## Connecting

If you are on a Windows computer, you can follow these steps to connect via RDP:

1. Search for “Remote Desktop Connection”
2. In the **Computer** field, enter your instance’s public IP address.



3. You will be prompted for credentials. Depending on the version of Windows Server installed, the **User name** will be either “Administrator” or “Admin”. Your password can be [found in Horizon](#). Click “OK”.



4. You will likely be prompted with a security certificate. View the certificate, determine if it is safe to continue, then (if so), click "Yes".



5. You should now be connected to the remote machine.

## 6. Allocations

### 6.1 Jetstream2 Allocations Overview

#### XSEDE to ACCESS transition

Please note that as XSEDE concludes operation and transitions to ACCESS, all of the allocations documentation will need to be updated. We will get it updated to reflect ACCESS as soon as possible. Thank you for your patience.

*Access to Jetstream2 is available solely through Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) allocations. You must be on a valid allocation or the PI of a valid allocation to have access to Jetstream2.*

Jetstream allocations are awarded exclusively through ACCESS. ACCESS provides user accounts free of charge. ACCESS allocations require that the Principal Investigator (PI) be a US-based researcher.

#### Get an account

You may apply for an account via the [ACCESS User Registration page](#).

A guide with more detail on the allocations processes is available from the [ACCESS Allocations Overview](#) page.

#### Allocation types:

Allocations have fundamentally changed with ACCESS. They are no longer groups by type, but rather by size.

#### ACCESS Credits and Thresholds

Researchers have opportunities to request ACCESS allocations at four levels.

Allocation	Credit Threshold
Explore ACCESS	400,000
Discover ACCESS	1,500,000
Accelerate ACCESS	3,000,000
Maximize ACCESS	Not awarded in credits

Information for each tier is below. Complete information including how to apply for allocations may be found on the [ACCESS Get Your First Project Page](#) page.

#### **Explore ACCESS**

Explore ACCESS allocations are intended for purposes that require small resource amounts. Researchers can try out resources or run benchmarks, instructors can provide access for small-scale classroom activities, research software engineers can develop or port codes, and so on. Graduate students can conduct thesis or dissertation work.

These allocations can be requested at any time. They are available for the duration of your supporting grant or for 12 months without a grant (with possible extensions up to five years). Multiple requests are allowed if you have more than one supporting grant or need to keep your classroom or other non-research activity separate from your research efforts.

#### **Discover ACCESS**

Discover ACCESS projects are intended to fill the needs of many small-scale research activities or other resource needs. The goal of this opportunity is to allow many researchers, Campus Champions, and Gateways to request allocations with a minimum amount of effort so they can complete their work.

These allocations can be requested at any time. They are available for the duration of your supporting grant or for 12 months without a grant (with possible extensions up to five years). Multiple requests are allowed if you have more than one supporting grant or need to keep your classroom or other non-research activity separate from your research efforts.

To submit a request, you will need to provide a one-page description of the project that addresses how you plan to use ACCESS resources. You can also ask for an advisory review from the community to guide you to appropriate resources.

#### **Accelerate ACCESS**

Accelerate ACCESS allocations support activities that require more substantial, mid-scale resource amounts to pursue their research objectives. These include activities such as consolidating multi-grant programs, collaborative projects, preparing for Maximize ACCESS requests, and supporting gateways with growing communities.

These allocations can be requested at any time. They are available for the duration of your supporting grant or for 12 months without a grant (with possible extensions up to five years). Multiple requests are allowed if you have more than one supporting grant or need to keep your classroom or other non-research activity separate from your research efforts.

Researchers are expected to have reasonably well defined plans for their resource use and to submit a three-page project description for merit review. Reviewers will look more closely at how your resource usage plan addresses the review criteria.

#### **Maximize ACCESS**

For projects with resource needs beyond those provided by an Accelerate ACCESS project, a Maximize ACCESS request is required. ACCESS does not place an upper limit on the size of allocations that can be requested or awarded at this level, but resource providers may have limits on allocation amounts for specific resources. The review process is similar to the XSEDE process for large allocations.

---

You'll need to have an ACCESS account to request access to the Jetstream2 Trial Allocation, be added to a PI's allocation, or apply for an allocation.

#### **To create an ACCESS account if you do not have one:**

Please see [ACCESS User Registration](#) for up to date information on creating an ACCESS user account.

#### **Start the allocations process**

The [General allocations page](#) has the steps you would take for all but the largest (Maximize) allocation size.

The Maximize opportunity size has some special requirements. We recommend reading here about [Maximize ACCESS allocations](#) if you are contemplating an allocation larger than 3 million SUs.

## 6.2 Jetstream2 Trial Allocations (JTA)

---

Jetstream2 trial allocations provide expedited, limited access to Jetstream2 resources. With a trial allocation, approved users can “test drive” Jetstream2 prior to requesting their own ACCESS allocations.

Trial allocations feature:

- 1000 service units (SUs) on the main Jetstream2 cloud at Indiana University (access to regional clouds is not provided)
- 1 m3.tiny (single core) or 1 m3.small (2-core) virtual machine instance
- 1 virtual machine backup snapshot per instance
- 10 GB disk external storage volume

This is meant to provide enough capacity for new users to experience “cloud native” virtual computing. It is not intended as a permanent or long-term solution.

Please note that these limits are intrinsic to the allocation and cannot be adjusted. While [virtual GPUs](#) are available on Jetstream2, they are not currently part of the trial allocation program.

### 6.2.1 Signing up for a trial allocation

---

To begin the process of using a Jetstream2 trial allocation:

- [Register for your ACCESS ID](#)
- Visit the [Jetstream2 Trial Allocation Portal](#) and select “enroll” (it may take up to an hour for enrollment to go into effect)
- If using [Exosphere](#), the allocation is named with your ACCESS username plus “JTA User”
- If using [Horizon](#), it is labeled with only your ACCESS username

## 6.3 Jetstream2 Allocations

---

Under ACCESS, there are no longer allocations dedicated solely to research (including research startups) or education. Instead, there are categories of allocations based on the request size. We discuss those in the [Allocations Overview](#) and ACCESS covers that on their [Prepare Requests: Overview](#) page.

Startup allocations generally represented researchers getting started with the national cyberinfrastructure or research groups that needed modest amounts of resources generally equal to startup values. This allowed researchers that needed more modest awards to renew yearly without submitting to the allocations committee.

ACCESS has embraced this methodology and all allocations under 3,000,000 SUs may be awarded outside of the allocations committee. The allocation types and sizes are covered in the [Allocations Overview](#).

Researchers have opportunities to request ACCESS allocations at four levels.

Allocation	Credit Threshold
Explore ACCESS	400,000
Discover ACCESS	1,500,000
Accelerate ACCESS	3,000,000
Maximize ACCESS	Not awarded in credits

If you have a clear picture of what resources your project requires, you can estimate your SU needs using the usage estimation calculator here: [Usage Estimation Calculator](#)

Otherwise, we would suggest reviewing:

- [Budgeting for Common Usage Scenarios](#)
- [Jetstream2 VM Sizes](#)
- [ACCESS Credits and Jetstream2](#)
- [Allocations Overview](#)

The process is that you as the PI will:

1. [Create your user account](#)
2. [Request an allocation/ACCESS credits](#)
3. Then you will “spend” (exchange) those credits on Jetstream2 resources
4. Have any colleagues or student researchers get their ACCESS accounts
5. [Add them to your allocation](#)

Resource managers, students, or other necessary users for your allocation may be added via [ACCESS User Management](#)

#### Note

**It can take up to four hours for allocations or users to become active AFTER your receive the email notification.**

[ACCESS Allocations home page](#) has complete information on these topics:

- [Create your user account](#)
- [Select an opportunity \(choose the size of your allocation\)](#)
- [Request an allocation/ACCESS credits](#)
- [Receive credits](#)
- [Exchange/spend those credits](#)

You'll need a copy of your CV as the PI and the CVs of any co-PIs. We recommend providing a detailed abstract regardless of allocation size. Intermediate allocation sizes, Discover and Accelerate, will require 1 page and 3 page project descriptions respectively. This is covered on the [Prepare Requests: Overview](#) page.

**Allow 1-2 business days for your application to go through the approval process.**

If you have questions about the the allocations process please open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#)

## 6.4 Jetstream2 Education Allocations

---

Under ACCESS, there are no longer allocations dedicated solely to research or education. Instead, there are categories of allocations based on the request size. We discuss those in the [Allocations Overview](#) and ACCESS covers that on their [Prepare Requests: Overview](#) page.

Since educational usage is a prime use case for Jetstream2, we want to help instructors/educators figure out how to best size their request and make that request to ACCESS.

Researchers have opportunities to request ACCESS allocations at four levels.

Allocation	Credit Threshold
Explore ACCESS	400,000
Discover ACCESS	1,500,000
Accelerate ACCESS	3,000,000
Maximize ACCESS	Not awarded in credits

If you know how many instances, what sizes/flavors, and how long you need, you can estimate your SU needs using the usage estimation calculator here: [Usage Estimation Calculator](#).

Otherwise, we would suggest reviewing:

- [Budgeting for Common Usage Scenarios](#)
- [Jetstream2 VM Sizes](#)
- [ACCESS Credits and Jetstream2](#)
- [Allocations Overview](#)

---

The process is that you as the PI will:

1. [Create your user account](#)
2. [Request an allocation/ACCESS credits](#)
3. [Then you will “spend” those credits on Jetstream2 resources](#)
4. [Have any associate instructors or teaching assistants get ACCESS accounts](#)
5. [Add those associates to your allocation as resource managers](#)
6. [Your students will need to get ACCESS accounts](#)
7. [You add them to your allocation](#)

Resource managers, students, or other necessary users for your allocation may be added via [ACCESS User Management](#)

### Note

**It can take up to four hours for allocations or users to become active AFTER you receive the email notification.**

[ACCESS Allocations home page](#) has complete information on these topics:

- [Create your user account](#)
- [Select an opportunity \(choose the size of your allocation\)](#)
- [Request an allocation/ACCESS credits](#)
- [Receive credits](#)
- [Exchange/spend those credits](#)

You'll need a copy of your CV as the PI and the CVs of any co-PIs. We would recommend adding the course/workshop/tutorial syllabus in PDF format.

**Allow 1-2 business days for your application to go through the approval process.**

If you have questions about the the allocations process please open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#)

## 6.5 Jetstream2 Research Allocations (Maximize ACCESS)

---

Under XSEDE, Research allocations were generally the larger allocations, greater than startup values, up to millions of SUs intended for scientific research. With the advent of ACCESS and the new allocation convention, the closest category to this would be **Maximize ACCESS** where the allocation sizes exceed 3 million SUs.

All allocation types now may be used for research or education and the requirements vary based on the number of SUs requested. We discuss this in the [Allocations Overview](#) and ACCESS covers that on their [Prepare Requests: Overview](#) page.

Most researchers will fall in the 3 categories of allocations that require a minimum of preparation and that can be submitted at any time. For those that will need 3 million SUs or more, you may start with a smaller allocation size (anything up to 3 million SUs) and then submit during the limited windows (twice a year) for a Maximize ACCESS allocation. It's highly suggested that you read [Preparing Your Maximize ACCESS Request](#) to see the requirements for submissions.

We highly encourage all potential applicants to view:

- [Writing and Submitting a Maximize ACCESS Proposal Webinar](#) and also the webinar for code and scaling documentation.
- *Example Research Requests - coming soon*
- [Required Components](#) section of the ACCESS documentation. *If your research allocation request does not have all of the required components, it may be rejected without review.*

[Science Gateway research requests](#) may have some different criteria. We recommend reviewing this section of the document if you will be submitting a science gateway oriented research request.

*Please do not forget to review the [formatting guidelines](#) - failure to adhere to the guidelines may also result in a submission being rejected without review*

---

If you have a clear picture of what resources your project requires, you can estimate your SU needs using the usage estimation calculator here: [Usage Estimation Calculator](#)

Otherwise, we would suggest reviewing:

- [Budgeting for Common Usage Scenarios](#)
  - [Jetstream2 VM Sizes](#)
  - [ACCESS Credits and Jetstream2](#)
  - [Allocations Overview](#)
- 

As this process is more involved and are reviewed by a panel twice a year, the turn around time to receive the award will be weeks or months after submission. Please plan accordingly.

---

[ACCESS Allocations home page](#) has complete information on these topics:

- [Create your user account](#)
  - [Select an opportunity \(choose the size of your allocation\)](#)
  - [Request an allocation/ACCESS credits](#)
  - [Receive credits](#)
  - [Exchange/spend those credits](#)
- 

If you have questions about the the allocations process please open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#)

## 6.6 Jetstream2 Allocation Supplements

Supplemental allocations allow you to get more SUs for a resource you currently use or access to a different resource you haven't previously had access to. With XSEDE allocations, you could add SUs beyond your initial request. Under ACCESS, you can spend additional credits that you have requested for your allocation type.

The following text is from the [RAMPS policies for ACCESS](#)

The ACCESS Credits for the Explore, Discover, and Accelerate opportunities are awarded in two phases. The first half of the Credits are available to you upon approval of your initial request. You can request the second half when needed by submitting a supplement request accompanied by a brief Progress Report.

If you require additional credits beyond the posted Credit limit for your type of project, you will need to prepare a request for an allocation opportunity with a higher limit. To be fair to all researchers, we cannot exceed the Credit limits of a given opportunity.

For Maximize ACCESS projects, supplements are available and have no upper limit (in theory). However, supplement requests for Maximize ACCESS projects are subject to resource availability and Resource Provider review and approval.

Additional information about supplements is available from the [Managing allocations: overview](#) page.

### Supplements and User Access

It's important to note that when you add a resource via a supplement that you have not previously had, users on your allocation are not automatically enabled for it. To enable users for a new resource like Jetstream2, you may need to go to your [Manage Users page](#) and select the appropriate project.

It can take up to 4 hours for any users added to an allocation to become active.

### Supplements vs Extensions

Supplements only add SUs/storage. If you need more time on your allocation, you need to pursue a renewal (*preferred*) or extension.

## Instructions for requesting a supplements

- Go to the [Manage Allocations page](#)
  - This will require you to authenticate to ACCESS if you have not already
- Find your allocation. Under the Action menu for your allocation, select “Supplement”
  - *Note: If your allocation is going to expire in 30 days or less, you will not see the Supplement option. You should see the Extension option, though. Choose a 3 month extension and once it's approved, the supplement button should appear.*
- Select the appropriate Jetstream2 resources(s) from the list.
- Once you have selected the resources you need, you'll fill in the SUs needed and any comments.
  - For storage requests, you'll fill in the value in gigabytes - e.g. 1000gb = 1TB.
  - For the number of SUs you'll need. The [Virtual Machine Sizes and Configurations page](#) can help you estimate VM sizes needed and SUs that will consume.
  - For more information about the resources, please see the [Jetstream2 Resources page](#)
- You will have to include a PDF “Progress Report” – basically a paragraph or two on the need for the storage or SU request.
  - We do like to see any publications that have been published or submitted during your allocation period as part of this report.
  - Any other milestones or achievements would also be welcomed in the progress report along with user or job counts if it's a science gateway resource.
- After you have added that, you may submit the request.

**Requests are typically reviewed within 1 to 2 business days. You'll receive notification from ACCESS Allocations on the status of your request once it has been reviewed by all service providers on the request.**

## 6.7 Jetstream2 Allocation Extensions and Renewals

If your allocation is expired or close to expired and has adequate SUs remaining for the resource(s), you may request an extension if you need 3-6 months to complete your work or to prepare for a renewal allocation. If you are planning to continue your research for another year (or more), you should pursue a renewal of your allocation.

***The option to extend allocations will be available when 90 days or less remain on the active allocation. A button for “Extension” will be available under the “Actions” tab under My Allocations.***

Further information about the extension request process is available from the [RAMPS / Policies FAQs](#) page.

### About Extensions

Extensions only add time. If you need more or different resources, you need to pursue a supplement or use additional ACCESS credits.

Extensions are not a substitution for the renewal process. **A renewal is ALWAYS preferred over an extension.**

You may continue to renew your Explore, Discover, or Accelerate-sized allocation at any time without going through the allocation committee process.

- Allocations may be extended **once** with *adequate justification*.
- A renewal is **always** preferred over an extension. Since the renewal process for most allocations is considerably simpler under ACCESS, there's little reason to extend versus renew
- Maximize ACCESS allocations must be reviewed yearly by the allocations committee. More information may be found on the [Preparing Your Maximize ACCESS Request](#) page

### Instructions for requesting an extension

- Go to the [Manage Allocations](#) page
  - This will require you to authenticate to ACCESS if you have not already
- Find your allocation. Under the Action menu for your allocation, select “Extension”
  - *Note: You will not see the extension option until your allocation is within 90 days of expiring.*
- Choose “Start Extension”
- Select the duration from the list.

- Enter all comments and justifications for the extension.
- After you have added that, you may submit the request.

## Instructions for requesting a renewal

- Please review [Prepare requests: overview](#) prior to submission to determine the opportunity size/category you want to request and the materials you'll need for submission.
- - Go to the [Submit a Request page](#)
  - This will require you to authenticate to ACCESS if you have not already
- Your allocation will show as a renewal opportunity in **EACH** allocation size/category. Choose the appropriate opportunity based on the categories you previously reviewed
  - *Note: You will not see the renewal option until your allocation is within 30 days of expiring*
- Choose “Renewal [Your allocation number] as [Opportunity Size]“
- Fill in the fields for the Request Information, the optional Opportunity Questions, and following sections.
- At the very bottom, you’ll select the ACCESS credits needed and then submit.
- Once the initial renewal has been approved, you’ll need to spend your ACCESS credits - please refer to the [Use credits overview page](#) for more information.
- To exchange credits from your renewal request to a specific resource, you’ll go to [My Allocations](#) page and select “Exchange” from the dropdown in the first column for your allocation.

## Request approval process

There are two requests with ACCESS – renewals and credit exchanges. Requests are typically reviewed within 1 to 2 business days. You’ll receive notification from ACCESS Allocations on the status of your request once it has been reviewed by all service providers on the transfer. It may take up to four hours for your allocation to become active after the EXCHANGE has been approved.

## 6.8 Jetstream2 Allocations FAQ

### I was added to an allocation but I get an error like this trying to log in

To use Jetstream2, your account must belong to an ACCESS allocation that is active on a Jetstream2 resource. Please visit <https://docs.jetstream-cloud.org/faq/general-faq>

The allocation updates and user additions are created by ACCESS and synced to Jetstream2 (and other ACCESS resources). It can take up to four hours for users added to an allocation to become active. This is outside of our control. If you are still unable to log into [Exosphere](#) or [Horizon](#) after four hours from being added, please open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#).

### Use ACCESS CI credentials if you're having trouble authenticating

We know ACCESS-CI credentials work correctly and recommend using those unless you are 100% sure you have linked credentials PREVIOUSLY in CILogon.

Please note that other CILogon credentials (i.e. for your home institution or Google) must be set up outside of this process. Follow these [Linking instructions](#) to link your home institution or other organization (e.g. Google, ORCID, etc) credentials.

### Is there an overview of the types of allocations available as well as any restrictions those allocations have?

The [Getting Started guide](#) describes the overall allocations process and using ACCESS resources. To review the types of allocations ACCESS and the process to get an allocation, here are some links you might find useful:

- [Allocations overview](#)
- [Allocation Details](#)
- [Allocation Requirements & Request Instructions](#)
- [Allocation policies](#)
- [Allocations Management Portal](#)
- [Allocation How-To Guides](#)

## **Is there a example or demonstration of how to get an allocation that I could follow?**

The requirements for allocations have changed considerably with ACCESS. An overview of the details and differences between allocation tiers can be found on the [Allocation Details](#) page.

Instructions on how to apply to each tier as well as their requirements can be found on the [Allocation Requirements & Request Instructions](#)

- Explore ACCESS - up to 400,000 credits, basically requires an abstract and other standard items listed on the Explore ACCESS page.
- Discover ACCESS - up to 1,500,000 credits, requires an abstract, one page description, and other standard items listed on the Discover ACCESS page.
- Accelerate ACCESS - up to 3,000,000 credits, requires an abstract, three page description, and other standard items listed on the Discover ACCESS page.
- Maximize ACCESS - most closely resembles XSEDE research allocations. Submission periods are twice a year and have requirements of up to 10 page main document, additional code and scaling performance documents, and the other standard materials like CVs required by all requests.

Presently there are no sample documents though we expect those will materialize in time. That said, with the requirements for documentation for all but the largest allocations being up to three pages, following the ACCESS pages linked above for submissions should yield all of the information you need.

---

## **How do I let other ACCESS accounts use my allocation?**

You can add users to (or remove them from) your ACCESS allocation via the ACCESS Allocations Portal. Users must have already created their ACCESS accounts before they can be added to an allocation.

To add users to, or remove them from, an active Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support ([ACCESS](#)) allocation, the principal investigator, co-principal investigator, or can use the ACCESS tools here (*requires ACCESS login*):

- [Allocations Management Portal](#)
- See the [Allocation How-To Guides](#) for more information

**Note**

***Please do note that only the PI is automatically added to resources like Jetstream2 resources. This is a function of ACCESS' allocations system and is outside of our control. The PI, co-PI, or resource managers will explicitly need to add users to resources via the User Management link above***

**It can take up to four hours for users added to an allocation to become active.**

---

**Can I keep renewing my allocation(s)?**

Unlike XSEDE, ACCESS does not differentiate between most allocation types. You can renew all allocation types now, though **Maximize ACCESS** allocations are only renewable during the enrollment periods noted in the **Maximize Access** section of the [Preparing your ACCESS Request](#) page.

---

**I'm running out of Service Units (SUs) or storage. How do I request more?**

If you already have an ACCESS allocation and need to request additional service units (SUs), the PI, co-PI, or delegate may submit a request via the ACCESS Portal. For instructions on how to submit the request, see [Requesting additional SUs, other Jetstream resources, or storage for Jetstream – Supplemental Allocations](#).

**Note**

Supplements only add SUs/storage. If you need more time on your allocation, you need to pursue a renewal or extension.

**I am at or exceeding the quota limits for my allocation.**

How do I request additional resources such as CPUs and memory?

You may contact [help@jetstream-cloud.org](mailto:help@jetstream-cloud.org) or open a ticket via either the [ACCESS Help Form](#) or the [Jetstream Help Form](#) with those requests.

It's important to note that Jetstream Trial Allocation quotas are fixed and will **NOT** be increased under any circumstances.

For other allocation types, justification will be required and will be granted at the discretion of the Jetstream2 staff based on the justification and available resources. Please note that large memory and GPU resources are limited so requests for those will require strong justification for success or partial success. We strive to make resources available to all researchers that require them, so striking a balance between the needs of one versus many is often necessary.

---

**Can you extend my allocation for me or give me access to my allocation for just a few days/weeks/months more?**

If your allocation is expired or out of SUs, you may request an extension, renewal, or supplement. Please see one of the following links:

- [Requesting additional SUs, other Jetstream resources, or storage for Jetstream – Supplemental Allocations](#)
- [Jetstream2 Allocation Extensions and Renewals](#)

Jetstream2 staff are unable to take these actions on your behalf.

**Note**

Supplements only add SUs. If you need more time, you need an extension or renewal (preferred).

Extensions only add time. If you need more or different resources, you need to pursue a supplement.

For allocations that are about to expire or have expired, ***renewals are ALWAYS preferred over extensions***

---

**My PI got a Jetstream2 supplement added to our existing allocation but we cannot access Exosphere or Horizon**

When you get a supplement for a resource that you have not previously had, users on your allocation are not automatically enabled for it. To enable users for a new resource like Jetstream2, your PI, Co-PI, or resource managers may need to go to their [Allocations Management page](#) and select the appropriate allocation to manage.

**I have an Explore ACCESS allocation but need to move to a larger Discover or Accelerate allocation size. How do I do that and keep my allocation number (and VMs/volumes in my current allocation)?**

When you are ready to move up to the next ACCESS Opportunity, please keep a few things in mind.

1. First make sure you use up all your ACCESS Credits. Please submit an Exchange Request if you have any remaining ACCESS Credits.
2. Next, you will want to focus on using any SUs you have left on the resources.
3. When you get close to using up all your SUs, please submit a ticket to the [ACCESS Allocations Team](#) when you have about 90% of the resources used. The Allocations Team will then help you move up to the next ACCESS Opportunity while keeping the same allocation number.

## 6.9 Budgeting for Common Usage Scenarios

### Note

If you already have a clear picture of what resources your project requires, you can now estimate your SU needs using the usage estimation calculator here: [Usage Estimation Calculator](#)

Because Jetstream2 is available solely through [SUs converted from ACCESS credits](#), it is often helpful to have a plan for what compute resources your project intends to use and to budget SU expenditure before submitting an exchange request. The goal of this document is to provide examples of common usage scenarios for Jetstream2 and how one might budget for them.

### 6.9.1 General Considerations

Although needs will vary largely depending on the type of project at hand, there are still some best practices and general considerations to keep in mind:

- Please consider padding your budget to add room for any extra time spent in development and other incidentals. We usually recommend budgeting at least 10% of your time to development.
- If you are entirely unsure about what kind of system resources your application needs, it may be wise to start with an [Explore allocation](#) and run your own performance and scaling measurements to determine what sort of VM structure you'll need.
- Try to budget high, but start low. Users often overestimate how much compute power their application really needs, **especially when it comes to GPUs**. Since [resizing an instance](#) to a larger flavor is always easy, we recommend starting small then scaling vertically as necessary.

### Storage

As noted in the [Jetstream2 Resources article](#), if you do not explicitly request storage, the default cap for allocations of 1 TB (1,000 GB) is included. For information on quota limits and how to request more storage, see [Storage Overview](#).

**Note**

The recommendations that follow are intended to give a best guess on how to forecast, but nothing is perfect. **Under-budgeting and running out of SUs is not the end of the world!** In most cases, it is relatively uncomplicated to obtain additional credits and/or move up to the next ACCESS tier (at least until you reach the [Maximize ACCESS](#) threshold). If you would like to consult about your specific use case and allocation types or budgeting, please reach out to [help@jetstream-cloud.org](mailto:help@jetstream-cloud.org).

### 6.9.2 Virtual Workstations and Traditional “Burst” Workloads

Because Jetstream2 instances give the user total control, including `sudo` access, a common use case is setting up an instance as a “virtual workstation” of sorts, allowing a researcher to offload computation from their local workstation or laptop, often to one that is also more powerful.

Although direct comparisons are fraught with peril due to differences in architectures, optimization technologies, cooling solutions, and more, the following might serve as a decent frame of reference for different flavors’ performance:

- **m3.quad:** roughly a typical mid-range laptop equipped with 4-6 CPU cores and 8-16 GB of RAM.
- **m3.medium:** a higher-end laptop or research workstation might have upwards of 8 CPU cores and 32 GB of RAM

Since they are not providing services or infrastructure that needs to be accessible 24/7, these kinds of instances can be [shelved](#) when you are not actively working on them to conserve SUs. This makes budgeting a bit trickier.

For example, if you intend to work on your instance for a quarter of an 8-hour work day, but not on the weekends, then you might plug the following into the [Usage Estimation Calculator](#) to budget for the year: 2 hours/day, 5 days/week, and 52 total weeks.

### 6.9.3 “Always-On” Infrastructure

Because of its lack of restrictions on runtime for CPU resources and the ability to route external traffic to instances (for example with a public IP address), an extremely common use case for Jetstream2 is hosting persistent or “always-on” infrastructure for projects.

Given a selected [instance flavor \(VM size\)](#), these types of workloads are simple to budget for, given that they run constantly. For example, one could simply use the default 24 hours/day, 7 days/week, and 52 total weeks on the [Usage Estimation Calculator](#) to find approximate usage for a year.

## Static Web Servers

### Examples:

- A simple site showcasing information and images about a research group and their project
- A gateway sharing research data, manuscripts, or media
- An endpoint serving static files/data to another service or application

These kinds of web servers are usually not too resource intensive, but the amount of compute required might depend on several factors, including:

1. The type of static content being served (small web pages vs. bulky data files)
2. The size/amount of content being served
3. The density of expected traffic to the server (1 visitor/hr vs. 1000 visitors/hr)

Servers for static pages, small amounts of data, and light to moderate traffic don't generally require more resources than those offered by an m3.quad instance (or even m3.small to m3.tiny on the lighter side). If you plan to serve large amounts of data or media, or you expect a high amount of traffic, you may see benefits by upsizing.

## Dynamic Web Apps and Other

### Examples:

- A website with a large amount of interactivity or dynamic elements
- An RStudio Server or JupyterHub host
- A REST API that performs significant server-side processing for requests
- An endpoint that dynamically processes and streams large amounts of data to/from an application

This type of application is difficult to make uneducated recommendations for due to the widely variable nature of usage from person to person. Because of this, it is especially important for you to be familiar with the performance requirements of your application, or as mentioned above, to be willing to explore a bit.

## 6.10 Usage Estimation Calculator

---

Below is a web calculator allowing PIs to generate an estimate of their credit needs.

## 7. Storage

### 7.1 Jetstream2 Storage Overview

Jetstream2 storage is an ACCESS-allocated resource. All allocations will be given a default storage amount (as noted on the [Storage](#) page), and any needs beyond this initial quota require a discrete allocation on the “Indiana Jetstream2 Storage” resource.

This storage is usable by all users on that allocation so the PI may want to institute per user quotas or discuss proper usage etiquette with the members of their allocation. Jetstream2 staff will not institute per user storage quotas, with the exception of the Jetstream2 Trial Allocation.

#### Storage Quotas

While storage is available to facilitate research, Jetstream2 is not primarily a storage service; large capacity storage is beyond the scope of Jetstream2.

#### Default Storage

By default, all allocations receive 1TB (1000 GB) of Jetstream2 Storage quota. If you will not need more than this, you do not need to request Jetstream2 Storage with your allocation request.

If your project requires it, additional Jetstream2 storage is an ACCESS-marketplace-allocated resource and can be requested with an exchange request from the [ACCESS Allocations Management Portal](#) by exchanging credits to the [Indiana Jetstream2 Storage](#) resource at a rate of [1 ACCESS credit == 1 GB](#). Please note that all storage requests will require appropriate justification.

Also note that the default 1 TB quota is only a floor, and storage allocations are absolute. In other words, the default 1 TB is **replaced** by a discrete allocation, **not added** on top of it. For example, if a project has a Jetstream2 Storage allocation through ACCESS of 3,000 credits/SUs/GB, their total storage quota will be 3 TB (not [3 TB + 1 TB = 4 TB](#)). In this sense, preparing storage exchange requests is simple—just request the amount of total GB needed.

#### Limits on Jetstream2 Storage\*\*

- Explore allocations are generally limited to 5TB max

- Discover allocations are generally limited to 5-10TB max
- Accelerate allocations are generally limited to 20TB max
- Maximize allocations are generally limited to 40TB max

All are subject to proper justification in the [allocations](#) process. Maximum values may be adjusted with proper justification and if there are adequate resources available. This is entirely at the discretion of the Jetstream2 team.

### 7.1.1 Storage Options

---

Jetstream2 (JS2) supports a number of different methods for data storage, including:

- [Volumes](#): mountable block storage
- [Manila](#): Filesystems-as-a-service
- [Object Store](#): experimental Openstack Swift and S3 storage

Please refer to the following pages for more information on using Jetstream2 storage under the various interfaces:

- [Using Jetstream2 Storage Under Exosphere](#)
- [Using Jetstream2 Storage Under Horizon](#)
- [Using Jetstream2 Storage Under the CLI](#)
- [Using Jetstream2 Storage Under Cacao](#)
- [Using Jetstream2 Storage with Manila](#)
- [Using Jetstream2 Storage with Object Store](#)

## 7.2 Volumes

---

**Volumes:** Small virtual filesystems that may be attached to the Users running/active Instances.

Files/data saved to a Volume are maintained across successive attachment/detachment actions to the User's Instances.

Volume actions (method varies per interface):

- **Creation**
- **Attachment** to an active Instance
- **Detachment**
- **Backup**

### 7.2.1 Using Volumes

Please refer to the following documentation for instructions in creating and managing volumes using our various interfaces:

- [Using Jetstream2 Storage Under Exosphere](#)
- [Using Jetstream2 Storage Under Horizon](#)
- [Using Jetstream2 Storage Under the CLI](#)

### Detaching Volumes

**REMINDER:** Volumes can only be detached if:

1. they are not in active use by a process on the instance Try:

- `fuser -m <volume>` to LIST all processes using a volume
    - You might also try `sudo fuser -m /volume` if nothing comes up – this will check for root processes holding the volume open
  - `fuser -km <volume>` to KILL all processes using a volume
    - If you get results with sudo above, you'll need to do `sudo fuser -km <volume>` to kill the processes
2. the instance to which they are attached is active
- `sudo lsof <volume>` will also show you processes using the volume.

**Volume Count Quota:**

Projects are limited to 10 volumes. Adjustment of **volume count quotas** can be requested via [the Jetstream2 contact form](#).

**Project and Providers:**

As with Instances, Volumes are associated/organized with Projects and with particular regional Providers (e.g. IU, TACC, UH, Cornell, ASU). (i.e. you cannot attach a Volume from one Provider to an instance on a different Provider.)

**Sharing:**

Generally, Volumes may be attached to one active Instance at a time.

You can't easily share volumes in OpenStack without deploying a **Shared File System service**. However, the native OpenStack [Manila - Filesystems-as-a-service](#) option is available.

Volumes may also be shared using standard methods (e.g. NFS) to other active Instances within Jetstream.

**Storage Quotas**

There are different quotas for block storage (volumes) and shares. There will be a self-service tool for managing those quotas soon, but for now, if you need to have your Jetstream2 Storage quota adjusted between block and share storage, please [contact us via the Jetstream2 contact form](#) with the amount you wish to move between the storage types.

**Backup & Exporting:**

Users should regularly backup (via `ssh`, `rsync`, `tar`, or the like: see [File Transfer](#)) any critical data contained on Volumes as no automated backup functions are currently provided by Jetstream.

## 7.3 Manila

---

### 7.3.1 Manila - Filesystems-as-a-service - on Jetstream2

**Use:** With Manila, you can have multiple instances on Jetstream2 share a filesystem.

**Description:** [Manila](#) is the file share service project for OpenStack. Manila provides the management of file shares for example, NFS and CIFS, as a core service to OpenStack. Manila works with a variety of proprietary backend storage arrays and appliances, with open source distributed filesystems, as well as with a base Linux NFS or Samba server.

If you are not using a Jetstream2 Featured image, you'll need to make sure you have these packages installed on your instance: [ceph-commons](#) and [ceph-fuse](#)

You can set up and manage Manila shares via Horizon and the CLI presently:

- [Manila via Horizon](#)
- [Manila via Openstack CLI](#)

Once a Manila share is created, you can mount and use it on your VM managed from any valid Jetstream2 interface.

- [Configuring a VM to use Manila Shares](#)

If there is a need for using Manila shares via FUSE, we do provide documentation for that, as well, though most will use the native kernel drivers above.

- [Configuring a Ceph FUSE client](#)

### 7.3.2 Configuring a VM to use Manila Shares

Please note that any items with  `${something}`  are placeholders for your names that you used when you created the manila shares in Horizon or via the CLI.

#### Pre-requisites

All JS2 featured images will have the necessary Ceph repository and packages installed. You will need to make sure you have *ceph-common* from the Ceph repository. Instructions may be found here:

<https://docs.ceph.com/en/quincy/install/get-packages/>

If you aren't comfortable with adding repositories and installing packages, we highly recommend that you use a featured image.

#### 1. Create a mount point on your instance

This can be anywhere in the filesystem. We recommend using a descriptive name.

```
mkdir /smith-lab-data
```

#### 2. Configuring your instance

Using the name and key of the access rule you created, complete these steps:

- Create the file `/etc/ceph/ceph.client.${accessTo}.keyring` and add the `accessKey`

Example:

```
sudo vi /etc/ceph/ceph.client.smithlabRWaccess.keyring
```

and add

```
[client.smithlabRWaccess] key = AQAHfhZiwTf/NhAAT5ChE4tDXt3Nq1NyjURbMQ==
```

Also, make sure the permissions on the file are rw to the owner only. You can do that with

```
sudo chmod 600 /etc/ceph/ceph.client.smithlabRWaccess.keyring
```

- Edit `/etc/fstab` to include the following line:

Using the path from the share creation and the name and key of the access rule you created, add your entry to the fstab so it mounts at boot time:

```
$path /smith-lab-data ceph name=${accessTo},x-systemd.device-timeout=30,x-systemd.mount-timeout=30,noatime,_netdev,rw 0 2
```

\$path = IP#:ports followed by volume path (/volume/\_no-group/uuid/uuid)

*Please make sure to change the \$variables to whatever you set these values to in Manila or the CLI when you created the share*

Example:

```
149.165.158.38:6789,149.165.158.22:6789,149.165.158.54:6789,149.165.158.70:6789,149.165.158.86:6789:/volumes/_nogroup/fe4f8ad4-2877-4e23-b5d3-46eb8476750b
```

### 3. Mount the share

Mount the manila share created with the following command `mount -a`

If you then run a `df -h|grep vol` you should see something like this:

```
149.165.158.38:6789,149.165.158.22:6789,149.165.158.54:6789,149.165.158.70:6789,149.165.158.86:6789:/volumes/_nogroup
```

### 7.3.3 Configuring a Ceph FUSE client

While most clients will use the native kernel ceph client, there may be times where you want to use the FUSE client for rate limiting or if requested by the Jetstream2 team.

You will need the `ceph-fuse` rpm if not already installed. You can install it with:

```
dnf install ceph-fuse
```

for Redhat-Based OSes like Rocky and Alma. For Ubuntu, you can install it with:

```
apt-get install ceph-fuse
```

To mount the manila share created previously you need to edit the three files listed below :

i. /etc/ceph/ceph.conf

```
[global] admin socket = /var/run/ceph/$cluster-$name-$pid.asok client reconnect stale = true debug client = 0/2 fuse big writes = true mon host = 149.165...
```

where the mon host information is provided to you as part of the manila share.

ii. /etc/fstab

Add the following:

```
none /mnt/ceph fuse.ceph ceph.id=$accessTo,ceph.conf=/etc/ceph/ceph.conf,ceph.client_mountpoint=$volumePath,x-systemd.device-timeout=30,x-systemd.mount-ti...
```

Replace `$accessTo` with the arbitrary name you chose for the access rule in step 3 above and `$volumePath` with ( /volume/\_no-group/...) that is also in the share information that you created.

It should look something like this:

```
none /mnt/ceph fuse.ceph ceph.id=manilashare,ceph.conf=/etc/ceph/ceph.conf,ceph.client_mountpoint=/volumes/_nogroup/fe4f8ad4-2877-4e23-b5d3-46eb8476750b/a...
```

iii. /etc/ceph/ceph.client.`$accessTo`.keyring

Add the following, substituting the name you chose for the access rule and using the access key from that rule:

```
[client.$accessTo] key = $accessKey
```

Your file should look something like this :

```
[client.manilashare] key = AQAHFhZiwTf/NhAAT5ChE4tDXt3Nq1NyjURbMQ==
```

## 7.4 Object Store

### 7.4.1 Using the Object Store on Jetstream2

The object store is only available via Horizon and the CLI presently.

The Jetstream2 object store utilizes Openstack Swift and is S3 compatible. You can utilize it via Horizon or the command line interface (CLI). From the CLI, you can use the python-swiftclient, the AWS CLI's `s3api`, or other compatible tools.

**Trying the object store from the CLI using Swift**

To use the OpenStack CLI natively with the object store, you'll need the Swift client if you have not already installed it. You can install it by doing:

```
pip install python-swiftclient
```

Once you have the Swift client installed, you can test it by doing:

```
swift post my-unique-bucket-name
```

which will create a storage container called “my-unique-bucket-name”. You can then list your buckets by doing:

```
swift list
```

If you want to delete the test bucket, you can do:

```
swift delete my-unique-bucket-name
```

As with all Openstack clients, you can see the full list of commands with

```
swift help
```

**You can also add, remove, and otherwise work with swift containers (buckets in the S3 vernacular) in Horizon on the Project → Object Store → Containers tab.**

## 7.4.2 Configuring Object Storage for AWS S3 Apps

The Jetstream2 object store hosts an S3-compatible API at the endpoint <https://js2.jetstream-cloud.org:8001>. This document will cover how to generate credentials and configure a storage bucket for use with S3-style apps.

There is also a supplemental tutorial from Andrea Zonca on [Using the distributed file format Zarr on Jetstream 2 object storage](#), which leverages [S3FS](#).

### Prerequisites

#### EC2 Credentials

If you want to use S3 compatibility, you'll need to generate EC2 credentials via the OpenStack CLI.

```
openstack ec2 credentials create
```

If you do not have CLI clients installed and an application credential openrc, please see [Installing Openstack Clients](#) and [Authenticating Against the OpenStack CLI \(Logging In\)](#).

Make note of the `access` and `secret` fields—these will be needed for authentication. If forgotten, they can be retrieved again in the future with

```
openstack ec2 credentials list
```

#### AWS CLI

Unfortunately, configuring S3-style settings for buckets is not currently possible through graphical interfaces like [Horizon](#). As such, an API wrapper like the [AWS CLI](#) is needed.

After [installing the AWS CLI](#), create a new file to house your EC2 credentials created above:

```
mkdir -p ~/.aws touch ~/.aws/credentials
```

Edit the file to include the following lines, where the values for `aws_access_key_id` and `aws_secret_access_key` correspond to the aforementioned OpenStack fields `access` and `secret`, respectively:

```
[default] region=RegionOne aws_access_key_id=your_access_here aws_secret_access_key=your_secret_here
```

Test the authentication by querying:

```
aws s3api --endpoint 'https://js2.jetstream-cloud.org:8001' list-buckets
```

Full reference for `aws s3api` interface can be found [here](#).

Note that calls to `aws s3api` require specifying the Jetstream2 endpoint with the `--endpoint` flag, as shown above. One workaround to including this with every command is to set an `alias s3api="aws s3api --endpoint 'https://js2.jetstream-cloud.org:8001'"`.

## Bucket Configuration

Before they can be used for many common S3-compatible applications, buckets will need to be configured properly. On Jetstream2, bucket names need to be **globally unique**, meaning that two buckets with the same name cannot coexist, even on different projects.

### Access Control List (ACL)

Use bucket policies if possible.

Amazon notes that the majority of modern use cases for S3 no longer require the use of ACLs, and instead strongly recommends the use of [policies](#) to control access.

Under Amazon S3, every bucket and object has an [Access Control List \(ACL\)](#) attached that defines both who can access that resource and the level of access they are granted.

There are a number of canned ACLs available for common scenarios. For example, to make a bucket private (only allowing the owner to read or write):

```
aws s3api --endpoint "https://js2.jetstream-cloud.org:8001" put-bucket-acl --acl private --bucket <bucket-name>
```

User-defined ACLs can also be set. See the output of `aws s3api put-bucket-acl help` for more information.

## Bucket Policies

Bucket policies are the preferred method for controlling access to an object storage bucket; Jetstream2 supports a [large subset](#) of the Amazon S3 policy language. To apply a policy with the AWS CLI, first define the policy in a new file (for example, `touch ./policy.json`). Then, run the `put-bucket-policy` command against that file:

```
aws s3api --endpoint "https://js2.jetstream-cloud.org:8001" put-bucket-policy --policy file://policy.json --bucket <bucket-name>
```

Bucket policies are written in JSON. They can be authored by hand or with [Amazon's S3 Bucket Policy generator](#), and generally include the following components:

- **Action:** What actions this policy will govern. For example, `s3>ListBucket` determines whether a user can list the contents of a bucket.
- **Effect:** Whether to explicitly `Allow` or `Deny` the action.
- **Resource:** What AWS resource is governed by this policy, specified by an [ARN](#). In this case, the resource will always be a Jetstream2 bucket or object whose ARN follows the format `arn:aws:s3:::bucket-name` or `arn:aws:s3:::bucket-name/object` respectively.
- **Principal:** To whom this policy applies, usually specified by an ARN. Examples can be found in the [AWS docs](#).
  - Note that on Jetstream2, the Amazon account number is replaced by a project ID; the active project's ID can be found with `openstack project show`, and is **not** the same as an ACCESS grant number (e.g. TRA230001). For example, an ARN to reference all users on the project with ID 12345abcde would be `arn:aws:iam::12345abcde:root`

The following policy example allows anyone to list the objects in `<bucket-name>` and read those objects, and allows authenticated users on project `12345abcde` to upload objects to the bucket:

```
{ "Version": "2012-10-17", "Id": "S3Policy1", "Statement": [ { "Sid": "BucketAllowPublicRead", "Action": [ "s3>ListBucket", "s3GetObject" ], "Effect": "Allow", "Resource": "arn:aws:s3:::<bucket-name>" }, { "Sid": "ProjectAllowUpload", "Action": [ "s3:PutObject" ], "Effect": "Allow", "Resource": "arn:aws:s3:::12345abcde/*", "Principal": "arn:aws:iam::12345abcde:root" } ] }
```

More examples can be found [here](#).

## CORS Configuration

[Cross-Origin Resource Sharing \(CORS\)](#) allows a user's browser to access resources hosted outside the domain of the current page/app. In the context of Jetstream2 object storage, it may be desirable for a web page/app to directly communicate from the client to the S3-compatible API, rather than proxying traffic through the original web server or handling S3 API requests on the backend.

Without CORS	With CORS
--------------	-----------

CORS is configured on a per-bucket basis and specified in JSON through the AWS CLI, similar to bucket policies. Start by creating a new file (e.g. `touch ./cors.json`)—the following example shows a CORS configuration to allow API requests to this bucket to be used by pages on `mycoolsite.com`:

```
{ "CORSRules": [ { "AllowedOrigins": ["https://mycoolsite.com"], "AllowedHeaders": ["*"], "AllowedMethods": ["GET", "PUT", "POST", "HEAD"], "MaxAgeSeconds": 3600 } ] }
```

The configuration can then be applied to the bucket with:

```
aws s3api --endpoint "https://js2.jetstream-cloud.org:8001" put-bucket-cors --policy file://cors.json --bucket <bucket-name>
```

For more information, see the output of `aws s3api put-bucket-cors help`.

Format requests properly.

In order to receive an `Access-Control-Allow-Origin` header from API responses, your app/browser must specify a proper `Origin` header in its request.

## 7.5 Transferring Files

---

There are a number of ways to transfer files to and from your VM instances. Some common ways are listed below.

### Web shell/Web Desktop

The web shell and web desktop allow file transfers to be easily done with a built in facility.

Exosphere : [Transferring files with web shell and web desktop](#)

Cacao : Coming soon!

### SCP/SFTP

Here are some recommended clients for transferring files with SCP or SFTP. These clients, while suggested, are not endorsed by Jetstream in any way. We have limited means to help with support issues related to these clients.

- [Cyberduck](#) is a well-known client for both Windows and Mac OSX to move files to and from individual VMs.
- [WinSCP](#) is another well-known Windows specific option.
- [Filezilla](#) is an option supporting several operating systems, including Windows.

### Globus

[Globus](#) is a fast, reliable, and secure file transfer service for easily moving data to, from, and between digital resources. Information for transferring files with Globus can be found here: <https://www.globus.org/data-transfer>, and a tutorial describing installing and use of Globus Personal Connect can be found here: [globus-connect-personal-linux](#)

## 8. User Interfaces

---

### 8.1 USER INTERFACES

---

Jetstream2 (JS2) supports a number of different interfaces for creating, deploying, and managing [Virtual Machines instances \(VMs\)](#) and [Storage](#).

- JS2 has 3 web-based *Graphical User Interfaces* as well as an *OpenStack Command Line Interface (CLI)*.
  - [Exosphere](#)
  - [Horizon](#)
  - [CACAO](#)
- [OpenStack CLI](#)

CACAO is still in development on JS2 and not in its final production state.

## 8.2 Exosphere

### 8.2.1 Exosphere

Exosphere strives to be the most user-friendly interface for research clouds. If you are new to Jetstream2 and unsure which interface to use, Exosphere is a great place to start. Whether you are exploring new software tools, running compute-intensive jobs, teaching a course/workshop, or building a science gateway, Exosphere can likely help you reach your goals. Use Exosphere in your web browser at <https://jetstream2.exosphere.app/>.

The screenshot shows the Exosphere web interface for an instance named "merely\_resting\_parrot".

**Header:** Instance merely\_resting\_parrot - Mozilla Firefox

**Breadcrumbs:** Home > Project TG-INI210003 > Instances > Instance merely\_resting\_parrot

**User Information:** iu.jetstream-cloud.org - TG-INI210003 (logged in as tg836338) | Remove Project | Create

**Instance Details:**

- Instance name: merely\_resting\_parrot (f06bea73-41d3-4cb3-93e7-de9bc5c27d9)
- Created: 6 days ago
- By user: tg836338
- From image: JS-API-Featured-Ubuntu20-Latest-candidate
- Size: m1.tiny

**Resource Usage (6:40 to 7:10):**

- CPU:** Usage fluctuates between 0% and 25%.
- Memory:** Usage remains constant at approximately 25%.
- Root Disk:** Usage remains constant at approximately 45%.

**Interactions:**

- Terminal (selected)
- Graphical Desktop
- Native SSH: exouser@149.165.170.0
- Console

**Volumes:**

Name	Device	Mount point
My Important Data	/dev/sdb	/media/volume/sdb

**Action History:**

Action	Time
unlock	2 days ago
lock	2 days ago
unlock	6 days ago
lock	6 days ago

Right now, Exosphere supports creating and managing instances. Instances are virtual computers (a.k.a. servers) that run your code, containers, software, and services. When you create an instance, you get helpful features like:

- A one-click web shell (terminal) in your browser
- Optionally, a one-click desktop environment for running graphical software
- A browser-based file upload/download tool
- Resource usage graphs that show how hard your instance is working
- Easy passphrase-based SSH access, if you want it

You can also use volumes to store large data sets, and manage persistent IP addresses for servers and science gateways. More powerful features, like data science workbenches and workflow sharing, are in experimental status now.

With Exosphere, there is no requirement to learn about advanced cloud technologies like virtual networks or SSH keypairs. If your use of Jetstream2 becomes more sophisticated, and you need to reach for more complex tools like the OpenStack CLI or APIs, Exosphere does not get in your way.

#### **How Exosphere compares with the Horizon dashboard, OpenStack CLI, and APIs**

Exosphere supports users who wish to mix their use of Exosphere with other OpenStack interfaces like Horizon dashboard, the OpenStack command-line interface, and the APIs. Generally (and with a few limitations), resources that you create in one interface will show up in other interfaces. They are merely different ways to manage the same infrastructure.

The other OpenStack interfaces support more features of OpenStack that Exosphere doesn't (yet), like Heat for cluster orchestration and Swift for object storage. So, they may better support some advanced cloud use cases than Exosphere, but they are generally less accessible to newer users. The Horizon dashboard, the OpenStack CLI, and the APIs were all built for use by IT engineers, not by researchers and data scientists. For example, in any of these tools you must create a network, subnet, router, security group, and SSH keypair before you can create an instance and connect to it (using an SSH client program and your private SSH key). If your use cases grow sophisticated enough, you may need this lower-level control, but using the Horizon dashboard is sort of like driving a car with a manual transmission. Using the CLI feels somewhat like using Horizon, but now you're shifting gears by typing shell commands instead of clicking buttons. Using the OpenStack APIs directly is like building your own transmission for the car.

Instances created via these other tools do *not* get a one-click shell, desktop, data upload/download tool, or any of the other interactions that Exosphere sets up for you. If you want these with the other OpenStack interfaces, you must set them up yourself with varying degrees of difficulty.

### How Exosphere compares with Cacao (Atmosphere2)

This section will be populated once [Cacao \(Atmosphere2\)](#) is ready enough to explore and compare.

### Getting Help

In addition to the XSEDE ticketing system, there is an #exosphere-user-support channel in the Jetstream Slack workspace. To request access, please open a ticket from the XSEDE user portal. This support option includes *no* promise of immediate, real-time assistance, but the Exosphere core developers monitor it and help when they can. Sometimes it's easier to chat with them than wait for ticket notifications.

### Contributing

If you'd like to help build Exosphere or request a new feature, visit the [project on GitLab](#). The Exosphere maintainers strive to provide a welcoming experience for new contributors. At a broader level than Jetstream, the Exosphere project has a [chat room on Matrix/Element](#) which is used to coordinate development work, but community members are also welcome to join. Further, the Exosphere team discusses project progress and priorities on a weekly video call on Mondays at 16:00 UTC. You can join at <https://meet.jit.si/exosphere> or dial in at +1.512.647.1431, PIN: 3037 7824 88#. (Find the agenda and notes from previous meetings [here](#).)

## 8.2.2 Exosphere: Logging In

Visit Exosphere in your web browser at <https://jetstream2.exosphere.app/>.

### Starting the login/authentication process

To log in to J2 Exosphere, you will need to click on the “Add allocation” card.

A screenshot showing the 'Add Allocation' button on the Exosphere home page

The next screen will ask you to add your ACCESS account or select another login method (“Other Login Methods” button). “Other login methods” allows you to add an OpenStack account, which is not something most users can use with Jetstream2. If you are unsure, select the “Add ACCESS Account” card and follow the instructions below.

A screenshot showing the 'Add ACCESS account' button

### Using ACCESS account (single sign-on)

Once you click on “Add ACCESS account”, you will be redirected to a CILogon page, a platform that manages ACCESS log ins. You will be required to allow authorization via CILogon any time you launch Exosphere unless you are already signed in. To log in using your ACCESS-CI credentials (which should be the same as your XSEDE credentials if you previously used XSEDE), select **ACCESS CI (XSEDE)** from the “Select an Identity Provider” drop down menu.

A screenshot showing the 'Select an Identity Provider' dropdown, with 'ACCESS CI (XSEDE)' selected

Your ACCESS account credentials are the default login method to Exosphere, so it is recommended that you select **ACCESS-CI** from the dropdown box. Please note that this may require multi-factor (Duo) authentication. If you need help setting up or changing your multi-factor authentication method, please open an [ACCESS Help Ticket](#), as Jetstream2 staff cannot fix this for you directly.

### Using other accounts

If you have linked institutional, Google, Orcid, or other credentials **PREVIOUSLY** with CILogon, you’ll be able to use those to authenticate.

**We know ACCESS-CI credentials work correctly and recommend using “ACCESS-CI” as your identity provider unless you are 100% sure you have linked your credentials with CILogon (previously linked credentials in XSEDE using Globus do NOT carry over).**

## Other institutional credentials

Please note that other CILogon credentials (i.e. for your home institution or Google) must be set up outside of this process. Follow these [Linking instructions](#) to link your home institution or other organization (e.g. Google, ORCID, etc) credentials.

## Choosing allocations and regions

After you log in with your ACCESS credentials, Exosphere will prompt you to select from the allocations that you are a member of. Any un-selected allocations will not be added the Exosphere interface, so select all that you may want to use. If you are granted access to a Jetstream2 regional cloud, you will be logged into those allocations in both the main (Indiana University) region and any other regions.

## Logging into non-Jetstream clouds in Exosphere

The Exosphere interface for Jetstream2 also allows you to manage resources on other OpenStack-based research clouds alongside Jetstream2. In order for this to work, these third-party clouds must expose their OpenStack APIs publicly, and you must have OpenStack credentials (or an OpenRC file) to provide. To add other clouds, choose “Add allocation”, select “Other login methods”, and pick the “OpenStack” login method.

If you encounter difficulty adding non-Jetstream2 clouds to Exosphere, Jetstream2 staff will have limited ability to troubleshoot and help, so this capability is not guaranteed to work.

Add Allocation	Other Login Methods	Add OpenStack Account
A screenshot showing the 'Add Allocation' button on the Exosphere home page	A screenshot highlighting the 'Other Login Methods' button	A screenshot highlighting the 'Add OpenStack Account' button

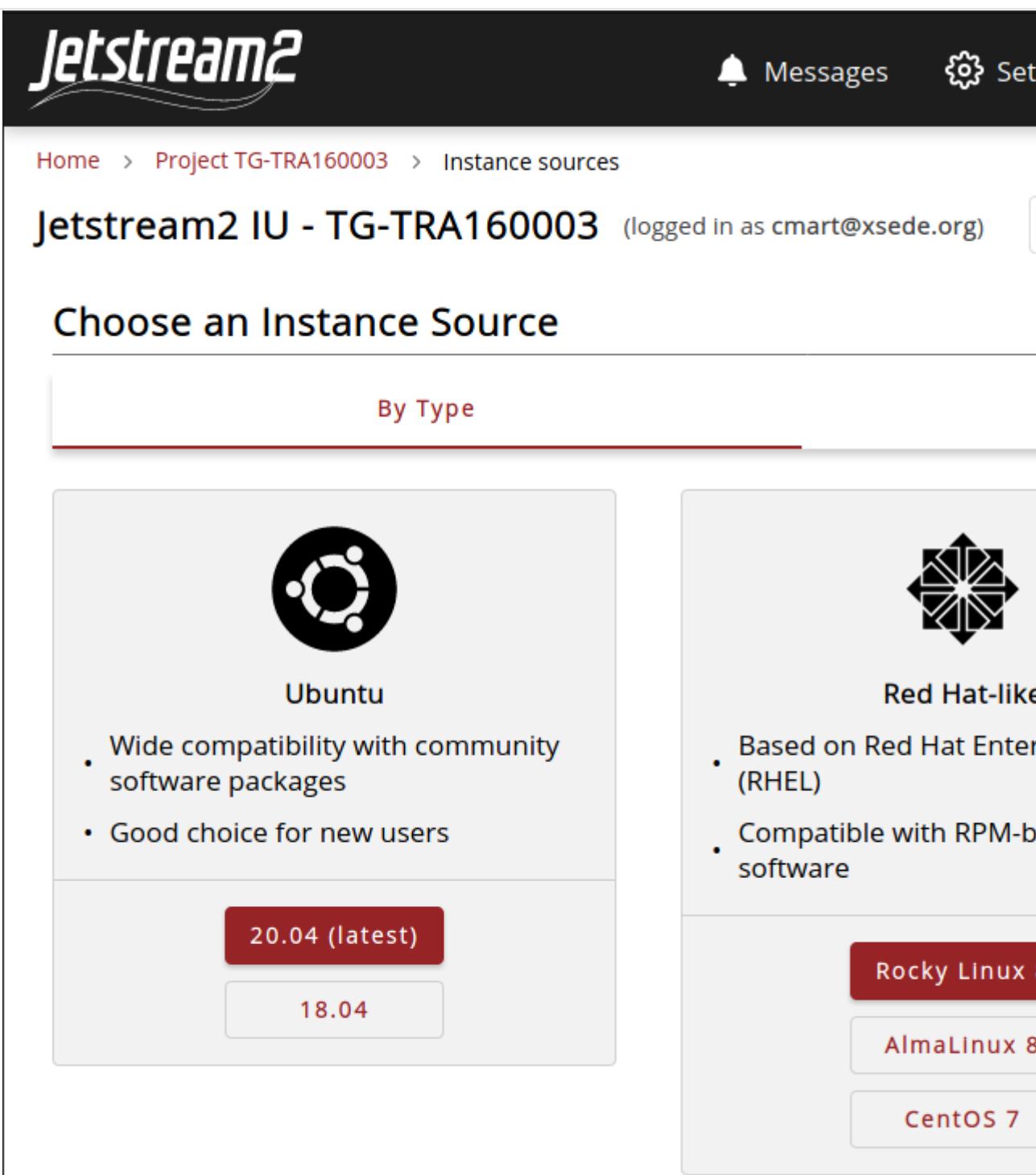
### 8.2.3 Creating an Instance with Exosphere

Once you have logged in and selected an allocation, select “Create” and then “Instance”.

The screenshot shows the Jetstream2 allocation dashboard for Project TG-TRA160003. At the top, there are navigation links: Home, Project TG-TRA160003, Messages, Settings, Get Support, and About. Below the header, the allocation name is displayed as "Jetstream2 IU - TG-TRA160003" and the user is logged in as "cmart@xsede.org". A "Remove Allocation" button is available. On the right, a "Create" button is highlighted in red. The dashboard displays two main sections: "Instances" and "Volumes". The "Instances" section shows "Allocation usage" at 0 of 1,000,000 SUs and "Instances used" at 8 of 100 total. The "Volumes" section shows "Volumes used" at 0 of 10 total. To the right of these sections is a vertical sidebar with three items: "Instance" (selected), "Volume", and "SSH Public Key".

### Choose an Instance Source

Next, choose an instance source. If you are a new user, select your desired operating system “By Type”.

Action	Screenshot
Choose by Type  Select your preferred operating system, or if you don't have a preference, pick the newest Ubuntu version.  Exosphere selects the latest official image for your chosen type automatically.	 <p>The screenshot shows the Jetstream2 Instance Sources page. At the top, there are navigation links: Home &gt; Project TG-TRA160003 &gt; Instance sources. Below that, it says Jetstream2 IU - TG-TRA160003 (logged in as cmart@xsede.org). The main heading is "Choose an Instance Source". A red underline highlights the "By Type" section. There are five options: 1. Ubuntu, represented by its logo, with a list of pros: Wide compatibility with community software packages, Good choice for new users. It has two download buttons: 20.04 (latest) and 18.04. 2. Red Hat-like, represented by its logo, with a list of pros: Based on Red Hat Enterprise (RHEL), Compatible with RPM-based software. It has two download buttons: Rocky Linux 8 and AlmaLinux 8. 3. CentOS 7, represented by its logo.</p>

**Choose by  
Image**

Alternatively, if you want to specify a particular image to create an instance from, select the “By Image” tab. Here, you can browse the entire Jetstream2 image catalog.

The screenshot shows the Jetstream2 Instance Sources page. At the top, there is a navigation bar with the Jetstream2 logo, a messages icon, and a settings icon. Below the navigation bar, the URL is shown as Home > Project TG-TRA160003 > Instance sources. The main title is Jetstream2 IU - TG-TRA160003 (logged in as cmart@xsede.org). A large heading says Choose an Instance Source. Below it, a button labeled By Type is highlighted with a red underline. There is also a button labeled Filter on image name: with a placeholder text "try \"Ubuntu\"". Further down, there are sections for filtering by visibility (Public, Community, Shared, Private) and tags. A checkbox "Show only images owned by this allocation" is present, followed by a "Clear filters (show all)" button. A list of featured images includes: Featured-AlmaLinux8 (featured), Featured-Ubuntu20 (featured), Featured-CentOS7 (featured), Featured-RockyLinux8 (featured), Featured-Ubuntu18 (featured), js2-bench-compute-image-latest (belongs to this allocation), and backup-js2-headnode-11-15-2021 (belongs to this allocation). At the bottom, there are fields for Visibility: Private and Tags: (153/153), and a prominent Create Instance button.

Choose an Instance Source

By Type

Filter on image name:

try "Ubuntu"

Filter on image visibility:  Public  Community  Shared  Private

Filtering on these tags:

Select tags to filter images on:

Show only images owned by this allocation

[Clear filters \(show all\)](#)

> Featured-AlmaLinux8 featured

> Featured-Ubuntu20 featured

> Featured-CentOS7 featured

> Featured-RockyLinux8 featured

> Featured-Ubuntu18 featured

> js2-bench-compute-image-latest belongs to this allocation

▼ backup-js2-headnode-11-15-2021 belongs to this allocation

Visibility: Private

Tags: 153/153

[Create Instance](#)