

Instance Mode

Choose the right instance type

Selecting the appropriate instance type from Axlflops is a critical step in planning your deployment. The choice of VRAM, RAM, vCPU, and storage, both Temporary and Persistent, can significantly impact the performance and efficiency of your project.

This page gives guidance on how to choose your instance type. However, these are general guidelines. Keep your specific requirements in mind and plan accordingly.

Overview

It's essential to understand the specific needs of your model. You can normally find detailed information in the model card's description on platforms like Hugging Face or in the `config.json` file of your model.

There are tools that can help you assess and calculate your model's specific requirements, such as:

- [Hugging Face's Model Memory Usage Calculator](#)
- [Vokturz' Can it run LLM calculator](#)
- [Alexander Smirnov's VRAM Estimator](#)

You should focus on the following main factors for instance selection:

- **GPU**
- **VRAM**
- **Disk Size**

Each of these components plays a crucial role in the performance and efficiency of your deployment. By carefully considering these elements along with the specific requirements of your project as shown in your initial research, you will be well-equipped to determine the most suitable instance type for your needs.

GPU

The type and power of the GPU directly affect your project's processing capabilities, especially for tasks involving graphics processing and machine learning.

Importance

The GPU plays a vital role in processing complex algorithms, particularly in areas like data science, video processing, and machine learning. A more powerful GPU can significantly speed up computations and enable more complex tasks.

Selection criteria

- **Task Requirements:** Assess the intensity and nature of the GPU tasks in your project.
- **Compatibility:** Ensure the GPU is compatible with your software and frameworks.
- **Energy Efficiency:** Consider the power consumption of the GPU, especially for long-term deployments.

VRAM

VRAM (Video RAM) is crucial for tasks that require heavy graphical processing and rendering. It is the dedicated memory used by your GPU to store image data that is displayed on your screen.

Importance

VRAM is essential for intensive tasks. It serves as the memory for the GPU, allowing it to store and access data quickly. More VRAM can handle larger textures and more complex graphics, which is crucial for high-resolution displays and advanced 3D rendering.

Selection criteria

- **Graphics Intensity:** More VRAM is needed for graphically intensive tasks such as 3D rendering, gaming, or AI model training that involves large datasets.
- **Parallel Processing Needs:** Tasks that require simultaneous processing of multiple data streams benefit from more VRAM.
- **Future-Proofing:** Opting for more VRAM can make your setup more adaptable to future project requirements.

Storage

Adequate storage, both temporary and persistent, ensures smooth operation and data management.

Importance

Disk size, including both temporary and persistent storage, is critical for data storage, caching, and ensuring that your project has the necessary space for its operations.

Selection criteria

- **Data Volume:** Estimate the amount of data your project will generate and process.
- **Speed Requirements:** Faster disk speeds can improve overall system performance.
- **Data Retention Needs:** Determine the balance between temporary (volatile) and persistent (non-volatile) storage based on your data retention policies.

Create and manage your instances

Create Instances

Go to "Workload" -> "Deploy" to create your instance.

Workload Type

General

Suited for General Workloads

Training

Computing Power for LAM/LLM/AI Models Training

Inference

Computing Power for Low Latency Inference

Recommend Products

Custom configuration

NVIDIA T4 * 1 @ USA West-02

USA West-02

CPU: 4 Cores

GPU: NVIDIA T4 * 1

Memory: 16GB

NVIDIA T4 * 1 @ USA West-02

USA West-02

CPU: 4 Cores

GPU: NVIDIA T4 * 1

Workload ^

Node ^

Serverless ^

^

Name

NVIDIA T4 * 1 @ USA West-02

OS/Image

Linux Ubuntu tensorflow2.17.0-cu121-jupyter2.14.2-ubuntu-22.04-lts

Storage (GB)

100

Server Quantity

1

Duration Required

1

Hourly

Daily

Weekly

Estimated Cost

1.3240 AIGT / Hour

Hour

1

Quantity

1

Total Cost

1.3240 AIGT

Deploy

Figure: Create your instance.

Choose the instance type that is right for you by selecting OS, Storage, etc. You can also click on "Custom Configuration" to see more options, including "GPU" and "Location".

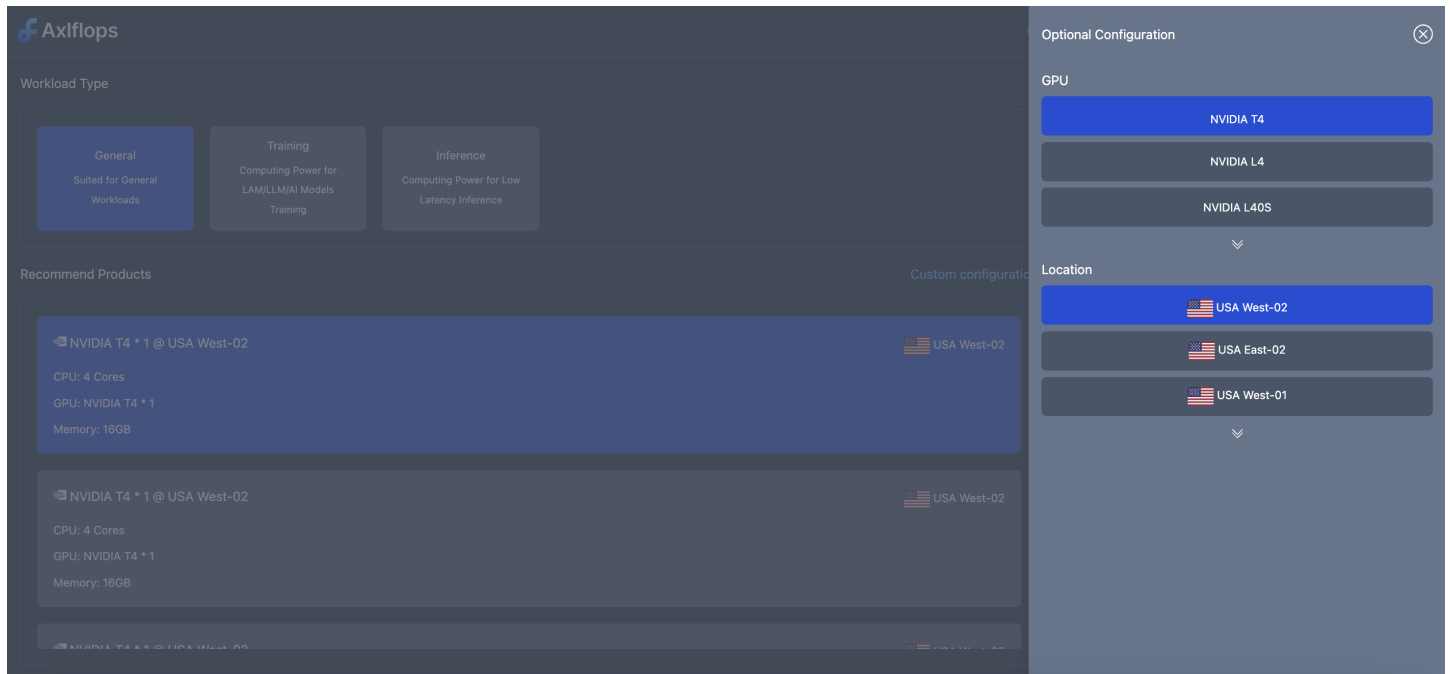


Figure: GPU and Location

And then click on the "Deploy" button to pay for your instance.

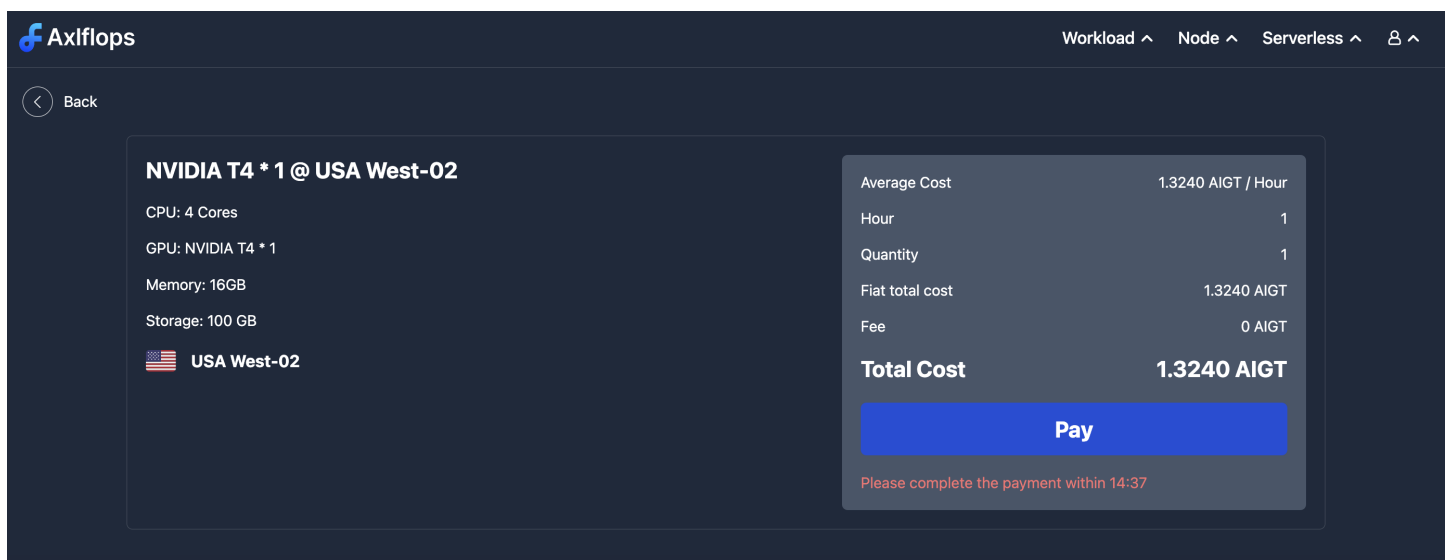


Figure: Make payment for your instance

You will be prompted to input your 2FA-Code. Input your 2FA-Code which is generated by the Authenticator App you've bond at the My Account page, and click on the "Submit" button.

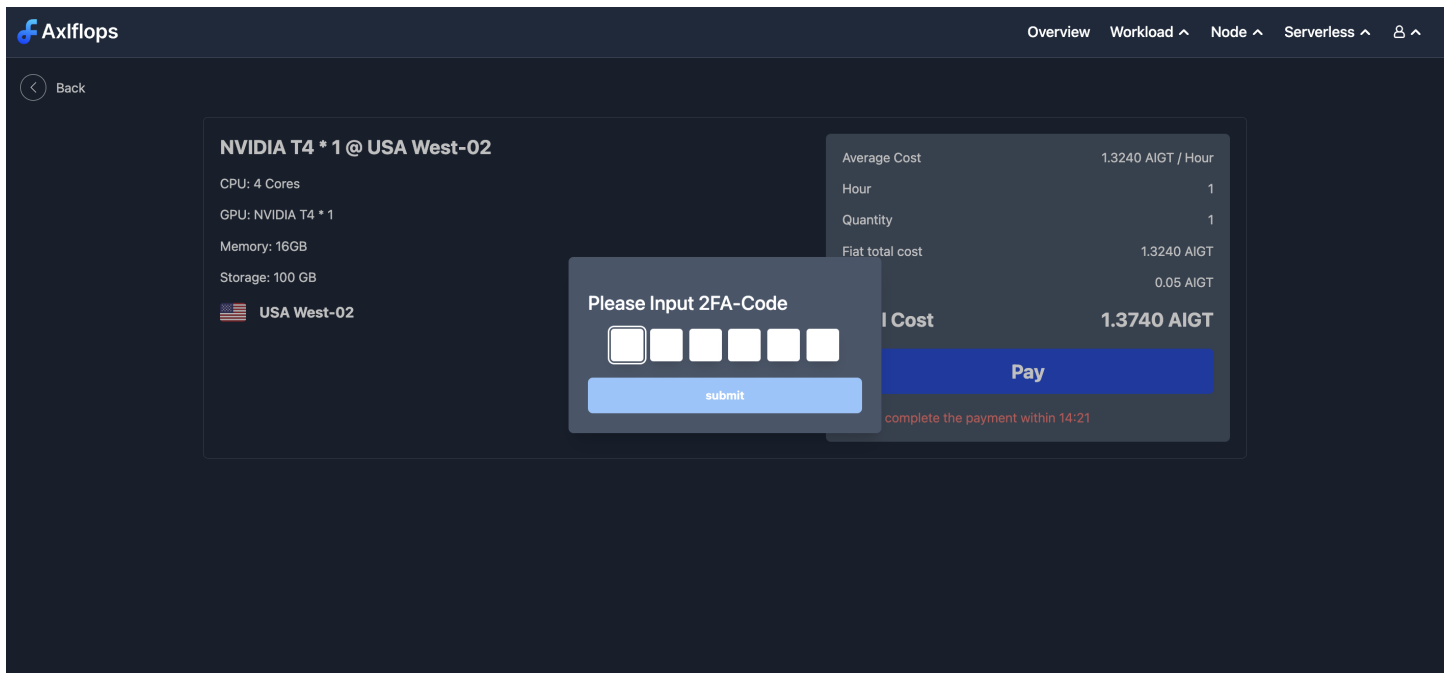


Figure: 2FA-Code

Click on the "Connect" button to connect your crypto wallet for payment. (Or you can choose to click on "Pay with QR" for payment.)

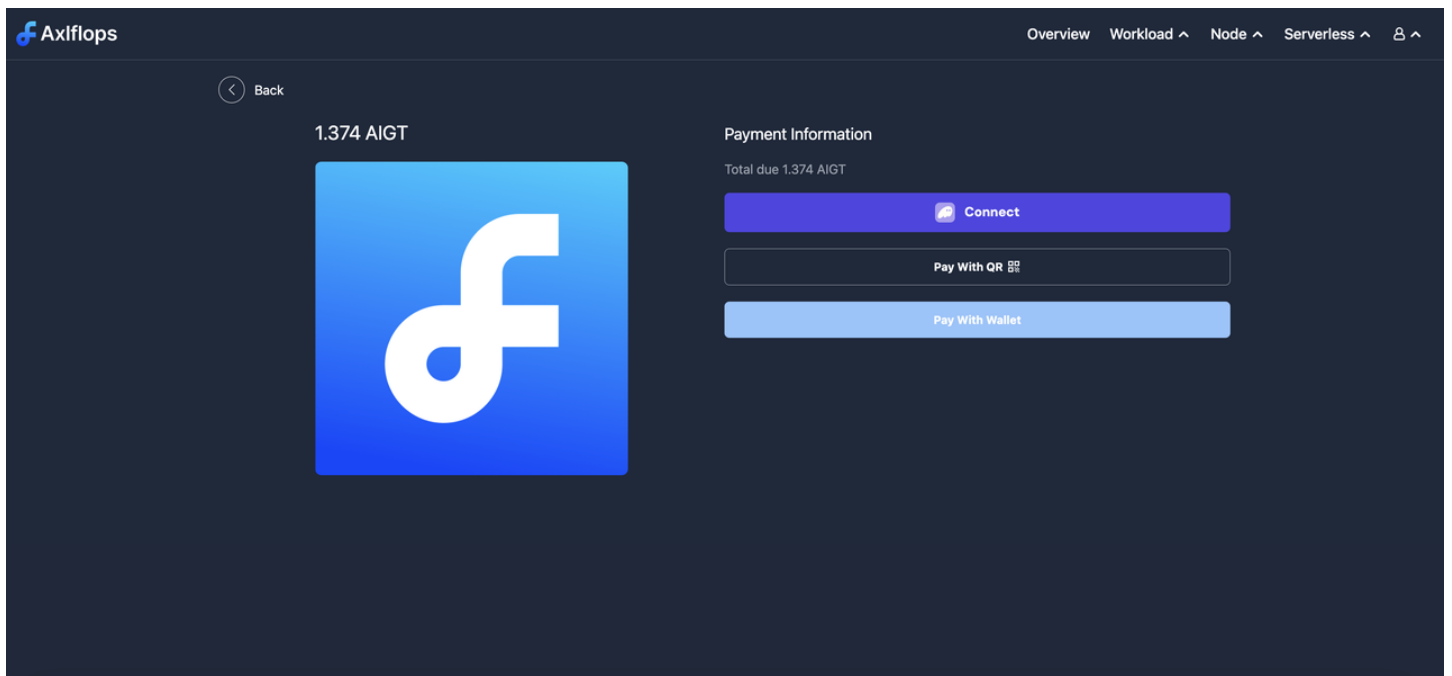


Figure: Connect your crypto wallet

Choose from "Phantom" or "Solfare" crypto wallet (If you don't have any of them, configure one of the crypto wallets first).

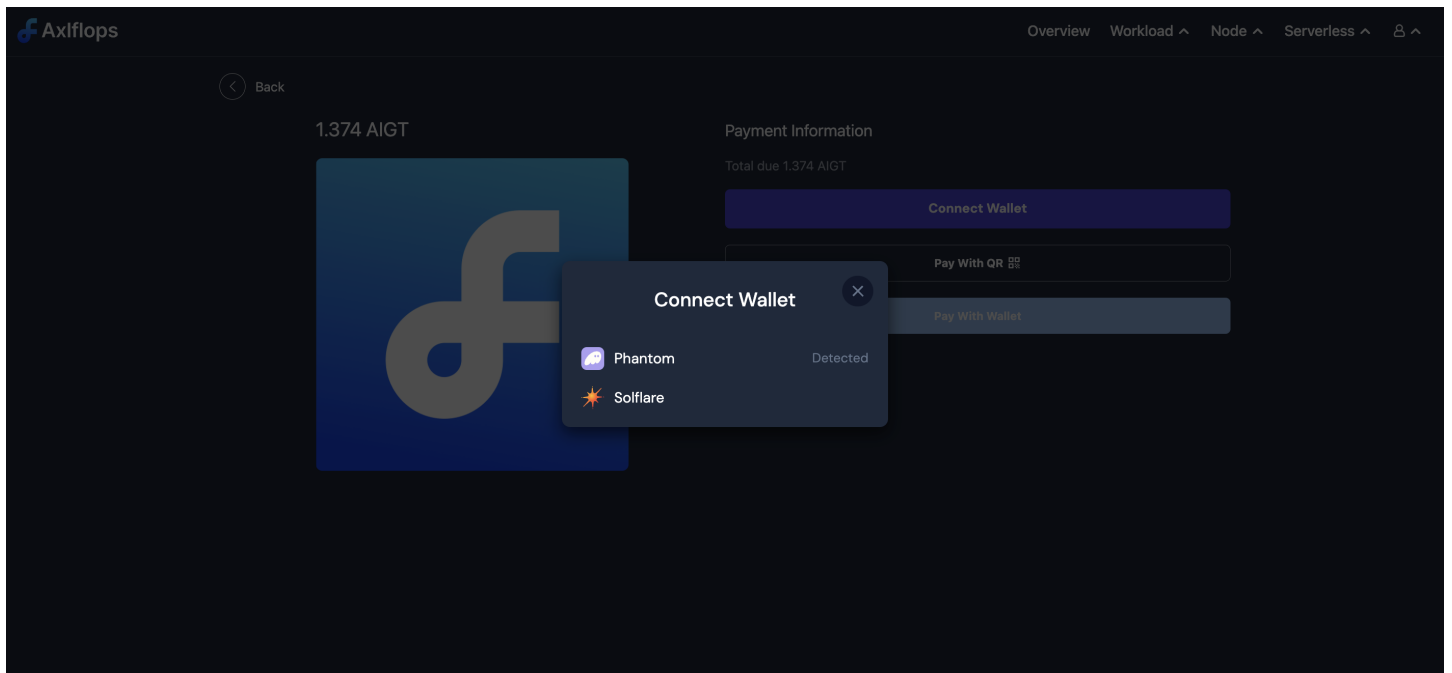


Figure: Phantom or Solflare crypto wallet

After your crypto wallet is chosen, your crypto wallet address will be shown.

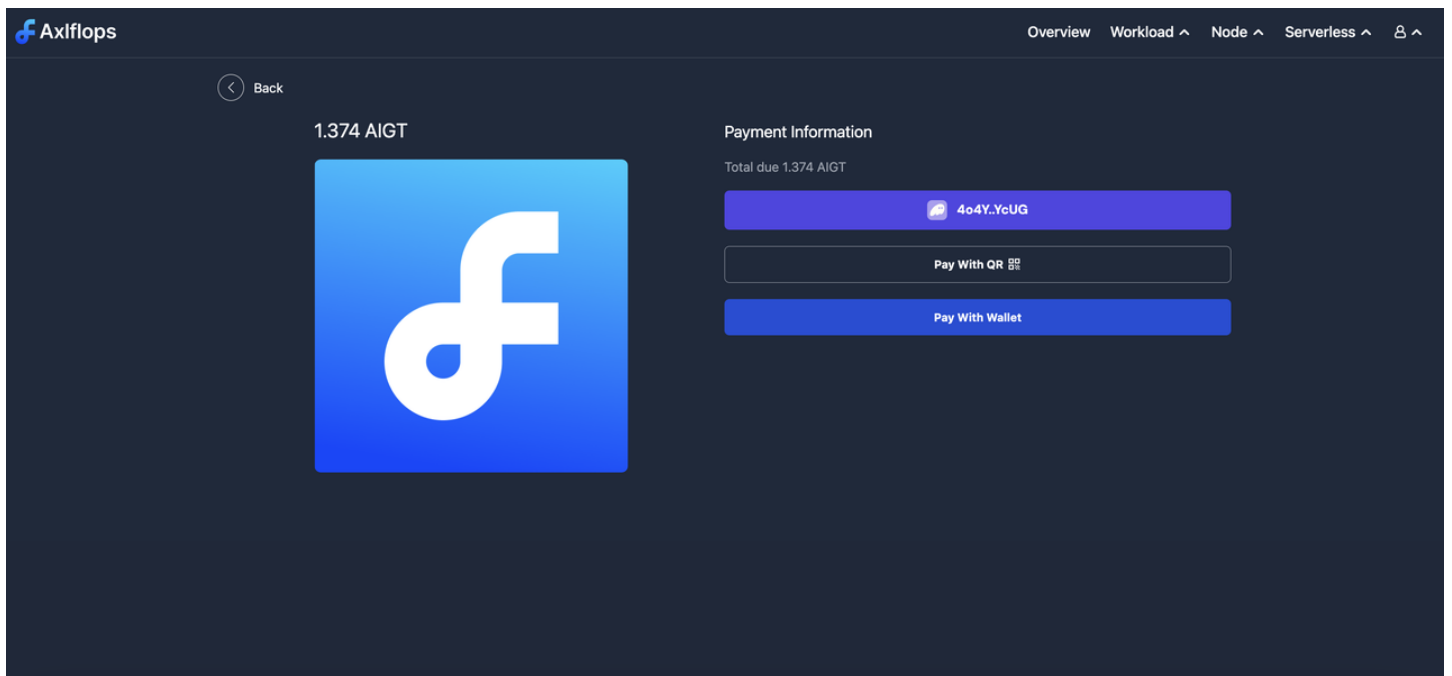


Figure: Crypto wallet address

Click on the "Confirm" button to confirm your payment.

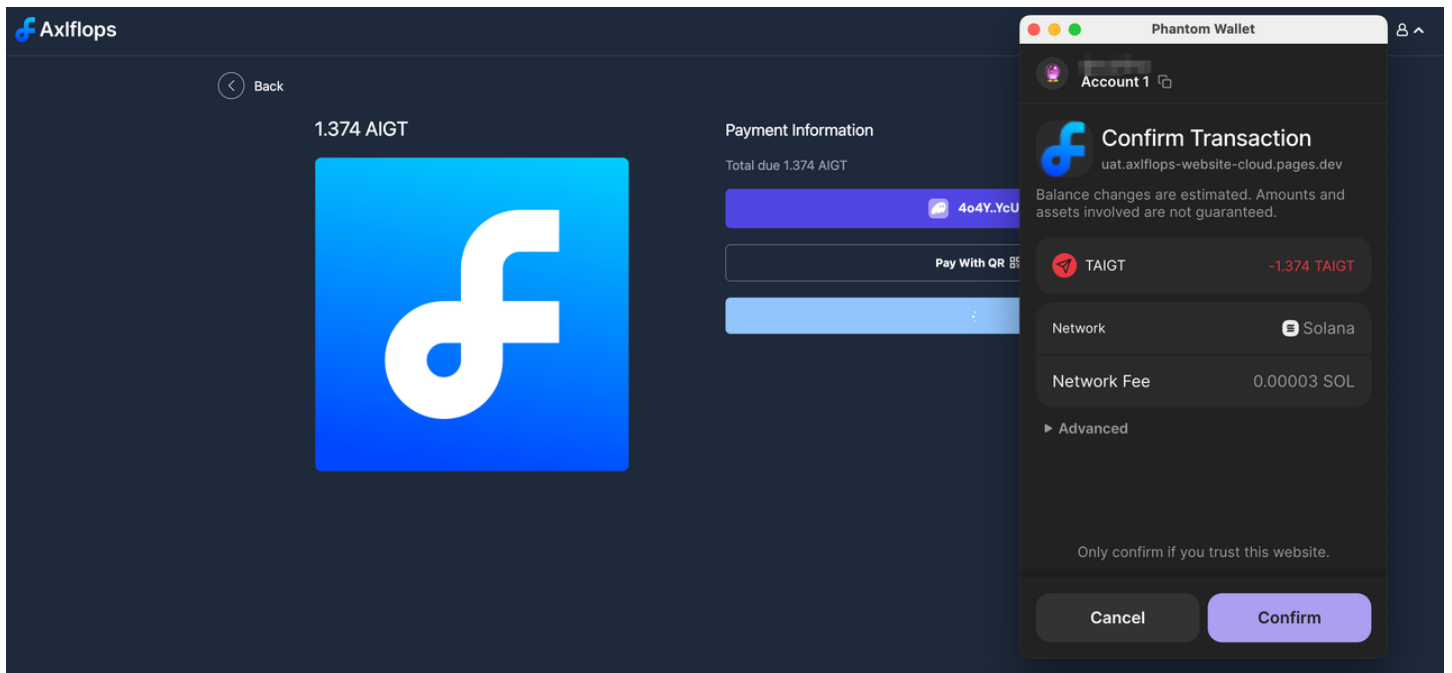


Figure: Confirm payment

List existing instances

Go to "Workload" and then "Search" to view your existing instances.

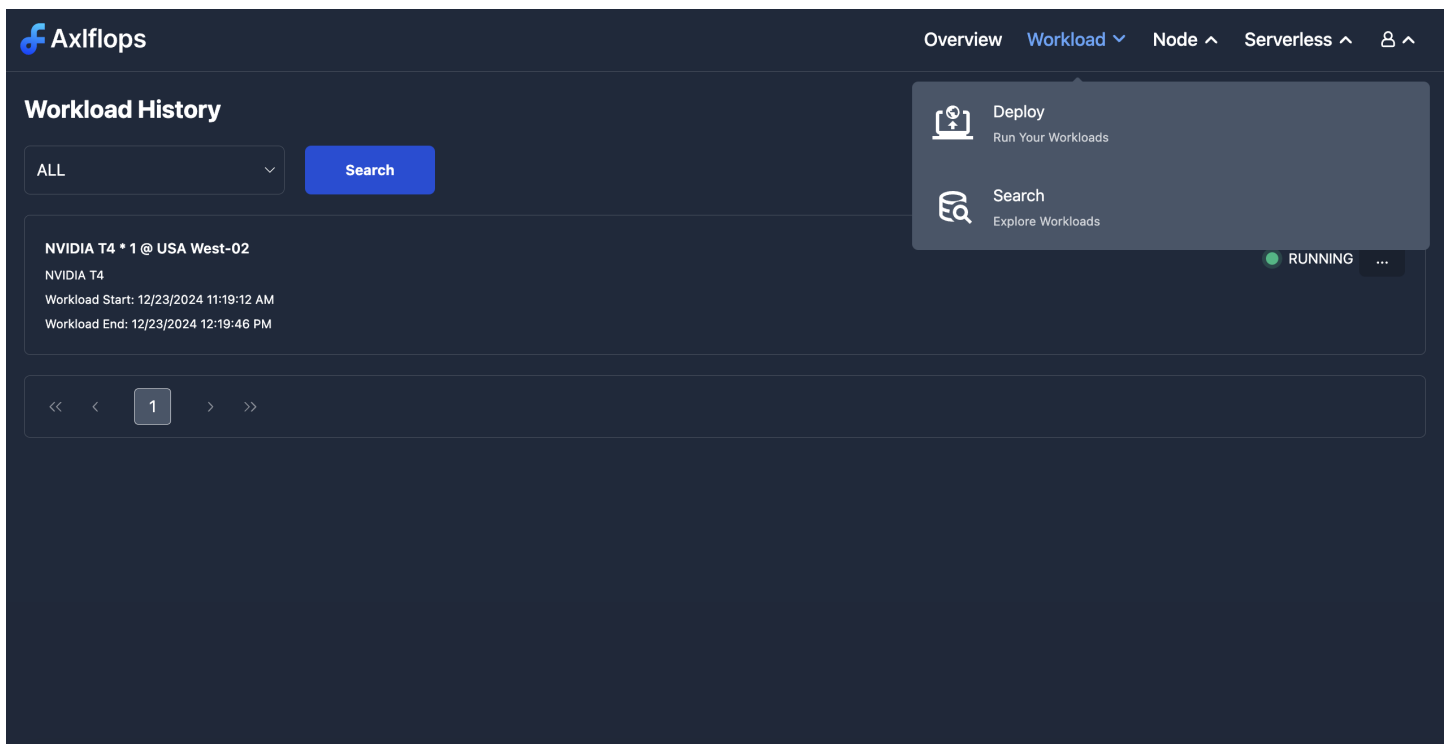


Figure: List existing instances

You can also search for instances based on their status, such as "Creating", "Running" and "Stopped".

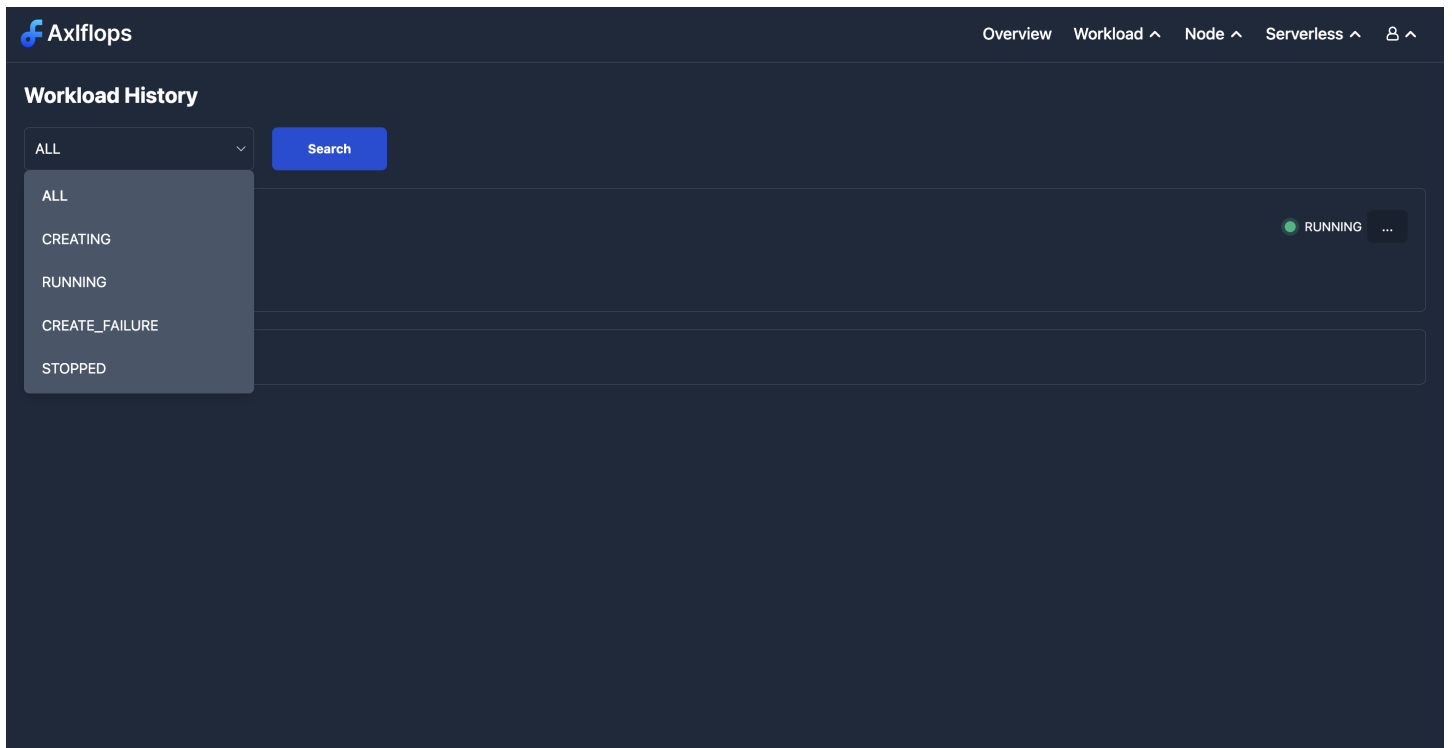


Figure: Search for instances based on status

Check instance details

Click on the 3 dots near instance status and then "Details" to check instance details.

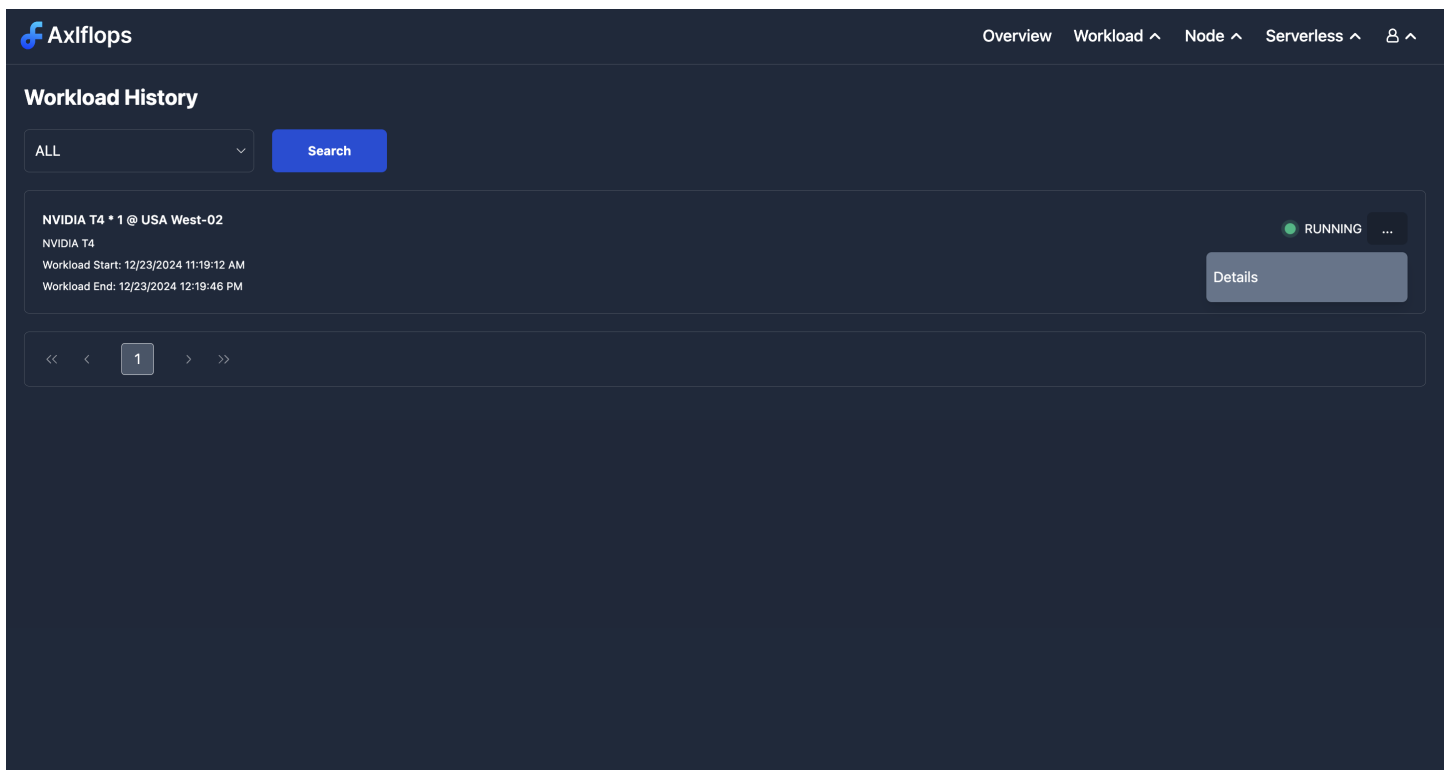


Figure: Instance Details button

On the details page, you can check the intance name, instance specification, instance location, connectivity, etc. You can also start using the instance by clicking on the "Web Console" or "Jupyter" button.

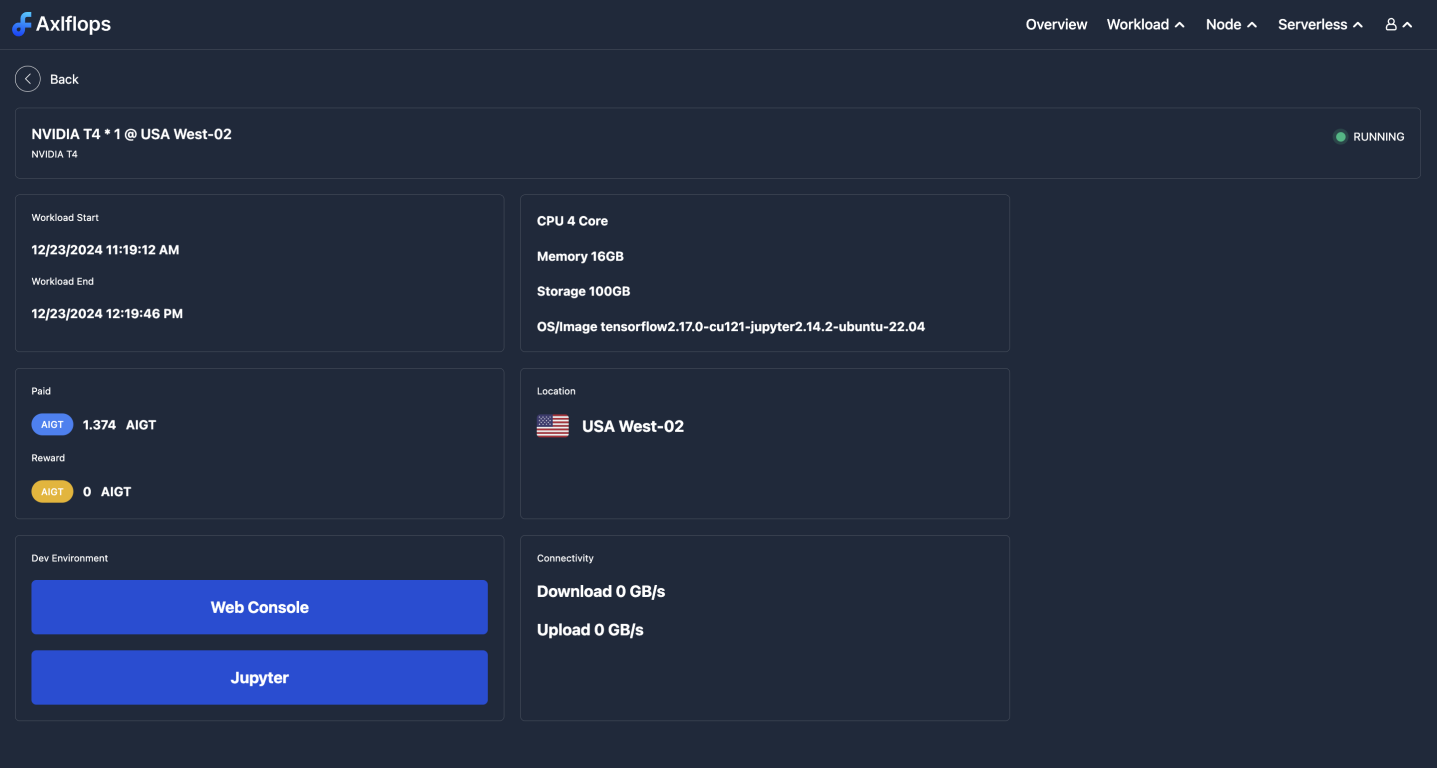


Figure: Instance Details page

Connect to an instance via Web Console

After creating an instance, you can connect to it via Web Console. Go to the instance details page and click on the "Web Console" button.

Axlflops

Overview Workload ^ Node ^ Serverless ^

< Back

NVIDIA T4 * 1 @ USA West-02
NVIDIA T4

WORKLOAD

Workload Start
12/23/2024 11:19:12 AM

Workload End
12/23/2024 12:19:46 PM

CPU 4 Core

Memory 16GB

Storage 100GB

OS/Image tensorflow2.17.0-cu121-jupyter2.14.2-ubuntu-22.04

PAID

AIGT 1.374 AIGT

REWARD

AIGT 0 AIGT

Location

USA West-02

Dev Environment

Web Console

Jupyter

Connectivity

Download 0 GB/s

Upload 0 GB/s

RUNNING

Figure: Instance Details page

Then you are logged in to the instance via your browser without having to use any terminals.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-38-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

*** System restart required ***
Last login: Wed Apr 19 18:55:55 2023 from 10.143.90.13
```

Figure: Web Console

You have now logged into your instance and have access to the power needed for your own usage. At any time, you can quit your connection with the command `exit`.

Execute AI tasks in Jupyter

After creating an instance, you can execute AI tasks in your instance using Jupyter. Go to the instance details page and click on the "Jupyter" button.

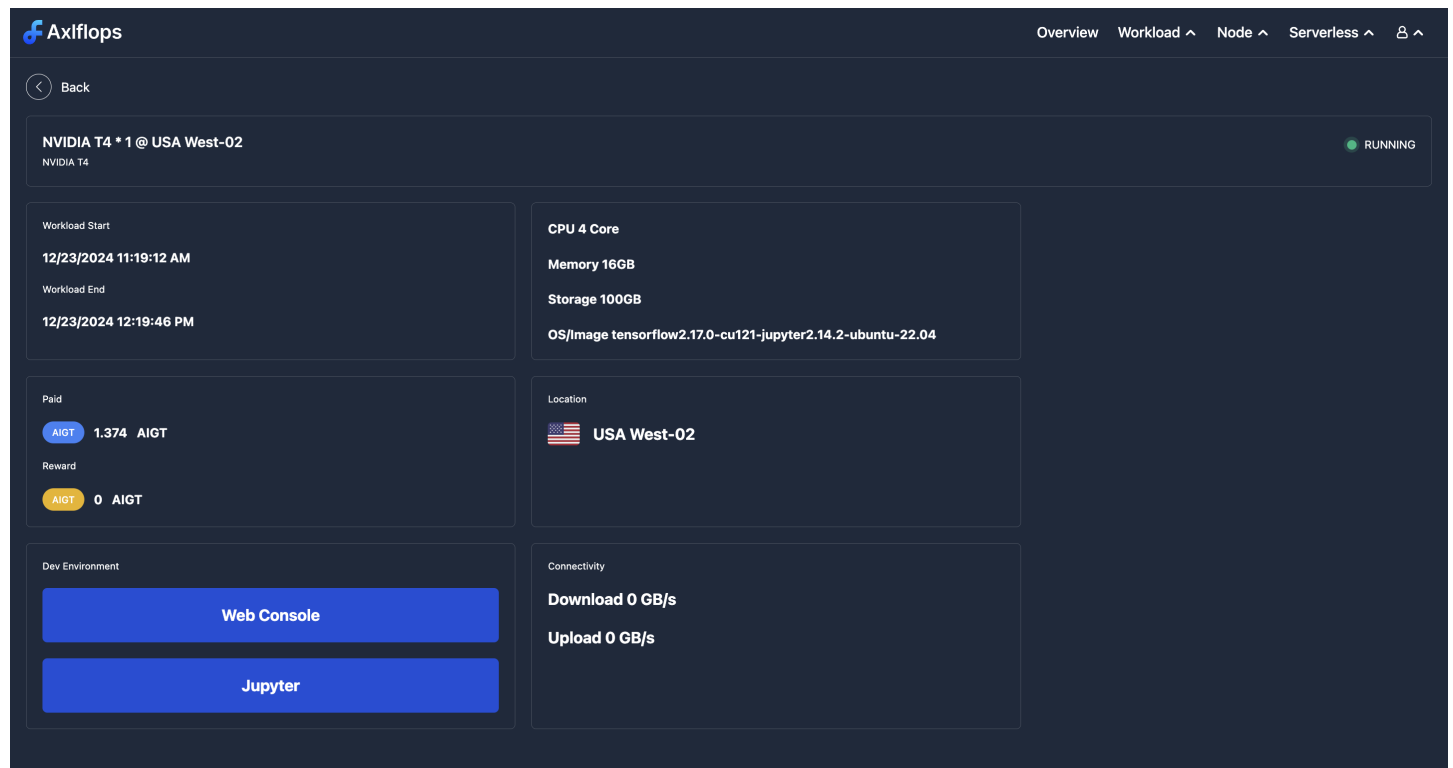


Figure: Instance Details page

Type the following command in your instance to run Jupyter Notebook:

```
$ jupyter notebook --ip=0.0.0.0
```

Now, in your local computer, open your browser of choice and enter URL like:

<https://23.12.123.12:8888>

You will notice that a Token is required to access Jupyter Notebook



Password or token:

Log in

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

Currently running servers:

```
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See [the documentation on how to enable a password](#) in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

Figure: Launch Jupyter Notebook

In your Axlflops instance, you will find the token in the provided URLs (in this example, the third URL has `token=XXXXXXX` which is the token we want to paste into the Jupyter Notebook in our local browser.

Do not paste the URL that is instructed as it is the wrong IP address, as you should be using the URL from the previous step.

```
kernel (enter to skip confirmation):
[W 23:24:12.850 NotebookApp] No web browser found: could not locate runnable bro
wser.
[C 23:24:12.851 NotebookApp]

To access the notebook, open this file in a browser:
    file:///home/user/.local/share/jupyter/runtime/nbserver-1469-open.html
Or copy and paste one of these URLs:
    http://0248a0a8-1c02-439c-a86d-1e03f21a8ed6:8888/?token=a3718607fd78a1cc
30f001cefeeca207dd94d51b3cd9cef7
    or http://127.0.0.1:8888/?token=a3718607fd78a1cc30f001cefeeca207dd94d51b3cd
9cef7
[I 23:28:01.327 NotebookApp] 302 GET / (128.148.204.159) 0.680000ms
[I 23:28:01.439 NotebookApp] 302 GET /tree? (128.148.204.159) 0.840000ms
```

Figure: Token for Jupyter Notebook

Enter the token into Jupyter Notebook. Success! You have logged into Jupyter Notebook on your Axlflops instance.



Figure: Jupyter Notebook

Connect to an instance via SSH

After creating an instance, you can connect to it via SSH.

Go to your email box and follow the the instruction in the email containing SSH log-in info.

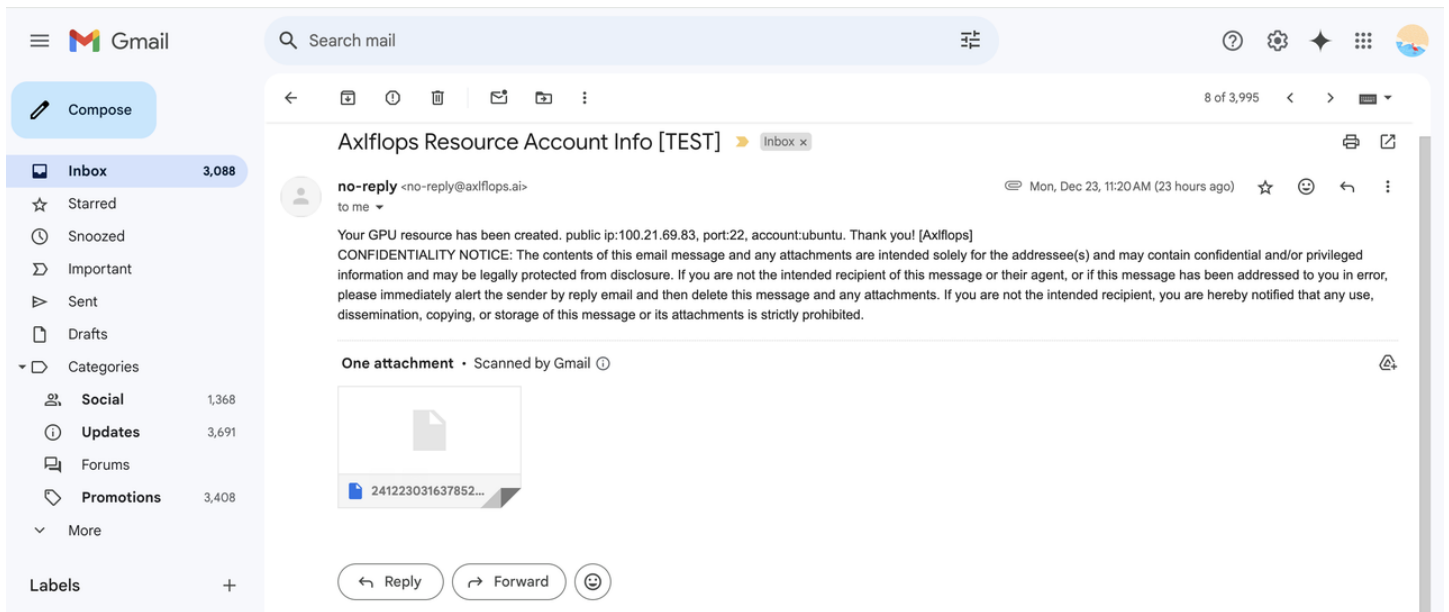


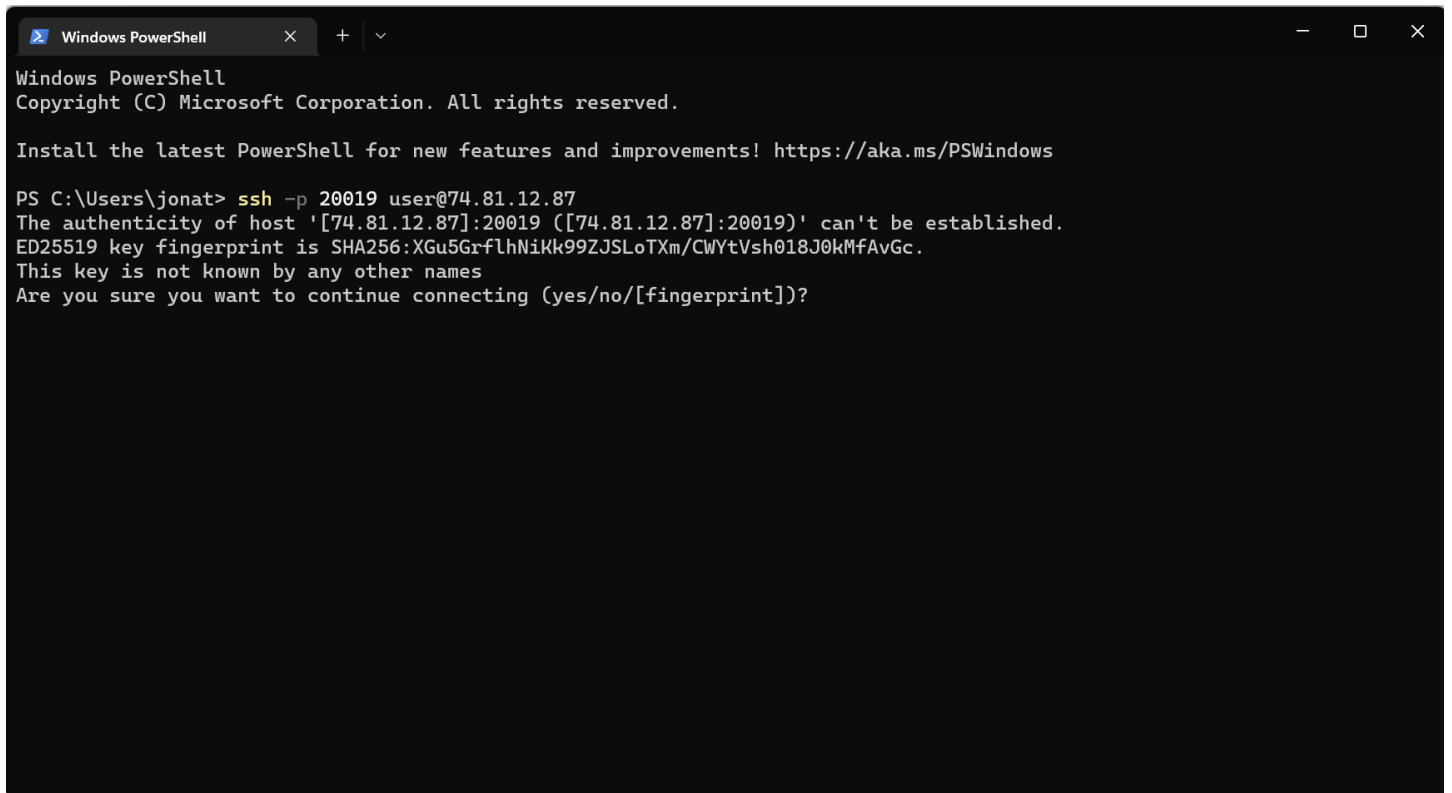
Figure: Email containing SSH log-in info.

Run the command like the following in your operating system's terminal software to access your instance:

```
ssh user@202.98.32.115
```

Your terminal software can be:

- Windows: Powershell
- Linux: Terminal
- MacOS: Terminal

A screenshot of a Windows PowerShell terminal window. The title bar shows 'Windows PowerShell' with standard window controls. The terminal text includes: 'Windows PowerShell', 'Copyright (C) Microsoft Corporation. All rights reserved.', a message to install the latest PowerShell, and an SSH command: 'PS C:\Users\jonat> ssh -p 20019 user@74.81.12.87'. It then displays a warning about the host's authenticity, showing a SHA256 fingerprint, and asks for confirmation to continue connecting.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\jonat> ssh -p 20019 user@74.81.12.87
The authenticity of host '[74.81.12.87]:20019 ([74.81.12.87]:20019)' can't be established.
ED25519 key fingerprint is SHA256:XGu5GrflhNiKk99ZJSLoTXm/CWYtVsh018J0kMfAvGc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Figure: Example for Windows Powershell

You have now SSHed into your server and have access to the power needed for your own usage. At any time, you can quit your connection with the command `exit`.

File transfer to an instance

You can transfer files from your local computer to your instance using SCP.

SCP is a command-line utility that allows you to securely copy files and directories between two locations.

With `scp`, you can copy a file or directory:

- From your local system to a remote system.
- From a remote system to your local system.
- Between two remote systems from your local system.

When transferring data with `scp`, both the files and password are encrypted so that anyone snooping on the traffic doesn't get anything sensitive.

SCP General Syntax

Before going into how to use the `scp` command, let's start by reviewing the basic syntax.

The `scp` command syntax take the following form:

```
scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2
```

- `OPTION` - [scp options](#) such as cipher, ssh configuration, ssh port, limit, recursive copy ...etc.
- `[user@]SRC_HOST:]file1` -Source file.
- `[user@]DEST_HOST:]file2` -Destination file

Local files should be specified using an absolute or relative path, while remote file names should include a user and host specification.

`scp` provides a number of options that control every aspect of its behavior. The most widely used options are:

- `-P` Specifies the remote host ssh port.
- `-p` Preserves files' modification and access times.
- `-q` Use this option if you want to suppress the progress meter and non-error messages.
- `-C` This option forces `scp` to compresses the data as it is sent to the destination machine.
- `-r` This option tells `scp` to copy directories recursively.

Before you begin

The `scp` command relies on `ssh` for data transfer, so it requires an ssh key or password to authenticate on the remote systems.

The colon (`:`) is how `scp` distinguish between local and remote locations.

To be able to copy files, you must have at least read permissions on the source file and write permissions on the target system.

Warning: Be careful when copying files that share the same name and location on both systems, `scp` will overwrite files without warning.

Copy a Local File to an Axlflops instance

To copy a file from a local to an Axlflops instance, run the following command:

```
$ scp -P 12345 file.txt  
remote_username@10.10.0.2:/remote/directoryCopy
```

Where `file.txt` is the name of the file we want to copy, `remote_username` is the user on the remote server (likely `user`), `10.10.0.2` is the server IP address. The `/remote/directory` is the path to the directory you want to copy the file to. If you don't specify a remote directory, the file will be copied to the remote user home directory.

Omitting the filename from the destination location copies the file with the original name. If you want to save the file under a different name, you need to specify the new file name:

Because SSH is listening on a forwarded port (rather than the default 22) we will have to specify the port using the `-P` argument. You can see in Figure 1 that port `20496` forwards to `22`, so we will include `-P 20496` in our scp command, as seen in Figure 2.

You will be prompted to enter the user password, and the transfer process will start.

```
(base) thech@MacBook-Air-196 ~ % scp -P 20496 Downloads/secret-code.txt user@spain-a.tensorockmarketplace.com:secret.txt  
user@spain-a.tensorockmarketplace.com's password:  
secret-code.txt 100% 197 1.5KB/s 00:00  
(base) thech@MacBook-Air-196 ~ %
```

Figure: Here we copy a secret file onto an Axlflops instance using the forwarded port 20496

```
user@b1a2f785-fa5f-4506-b601-599525d4a821:~$ ls  
Documents secret.txt
```

Figure: The Axlflops instance has received the secret file

The command to copy a directory is similar to copying files. The only difference is that you need to use the `-r` flag for recursive.

To copy a directory from a local to remote system, use the `-r` option:

```
$ scp -P 12345 -r /local/projects  
remote_username@10.10.0.2:/remote/projects
```

Copy a file from Axlflops instance to your local system

To copy a file from an Axlflops instance to a local system, use the remote location as the source and local location as the destination.

For example to copy a file named `file.txt` from a remote server with IP `10.10.0.2` runs the following command:

```
$ scp -P 12345 remote_username@10.10.0.2:/remote/file.txt  
/local/directoryCopy
```

Don't forget to include the `-P` option for port forwarding like in the previous section.

You will again be asked to enter your password.

```
(base) thech@MacBook-Air-196 ~ % scp -P 20496 user@spain-a.tensorDockmarketplace.com:secret.txt Downloads/tensordocksecret.txt  
user@spain-a.tensorDockmarketplace.com's password:  
secret.txt
```

Figure: Example of copying a file from the Axlflops instance to your local computer