# CONNECTING TO THE CURIOSITY CLUSTER

User instructions [last update October 12th 2022]

# CONTENT

- Introduction
- Specification & Node architecture
- Step by step on how to connect to the raplab-hackathon Cluster
  - STEP 1: Activate your Axis Account
  - STEP 2: Connecting to the Cluster
- Modules & environment
- Storage policies
- Common scheduler commands
  - Interactive mode
  - Batch mode (submission script)
- Containers
  - Singularity
- Transfer files to/from the Curiosity cluster

# INTRODUCTION

About the Cluster:

The hackathon cluster consists of a single CPU-only login node and 10 DGX-A100 compute nodes.

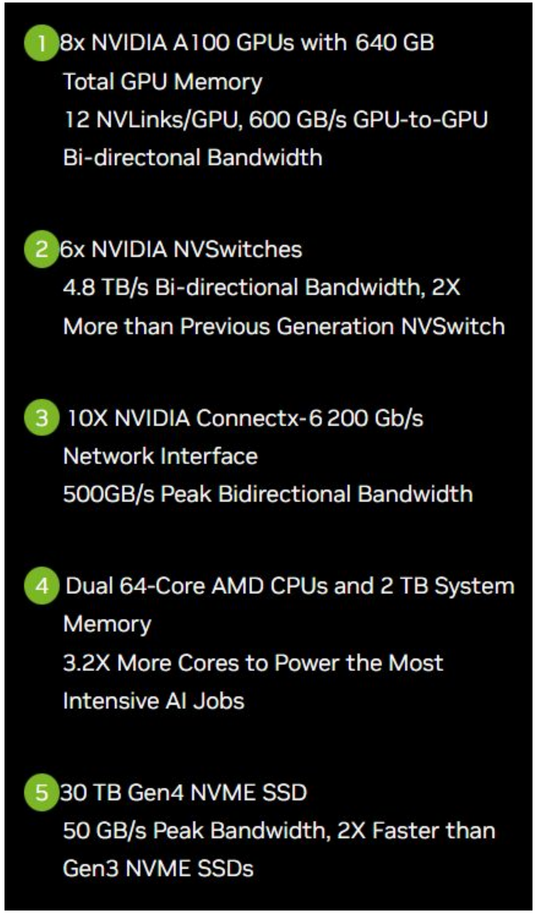
NVIDIA and external users access the cluster via SSH to the login node. Cluster accounts are controlled via a web-based service called Axis.

Join the OpenACC User Group SLACK workspace:

http://www.openacc.org/community#slack

# CURIOSITY CLUSTER

## Specs per node (total of 10 nodes)



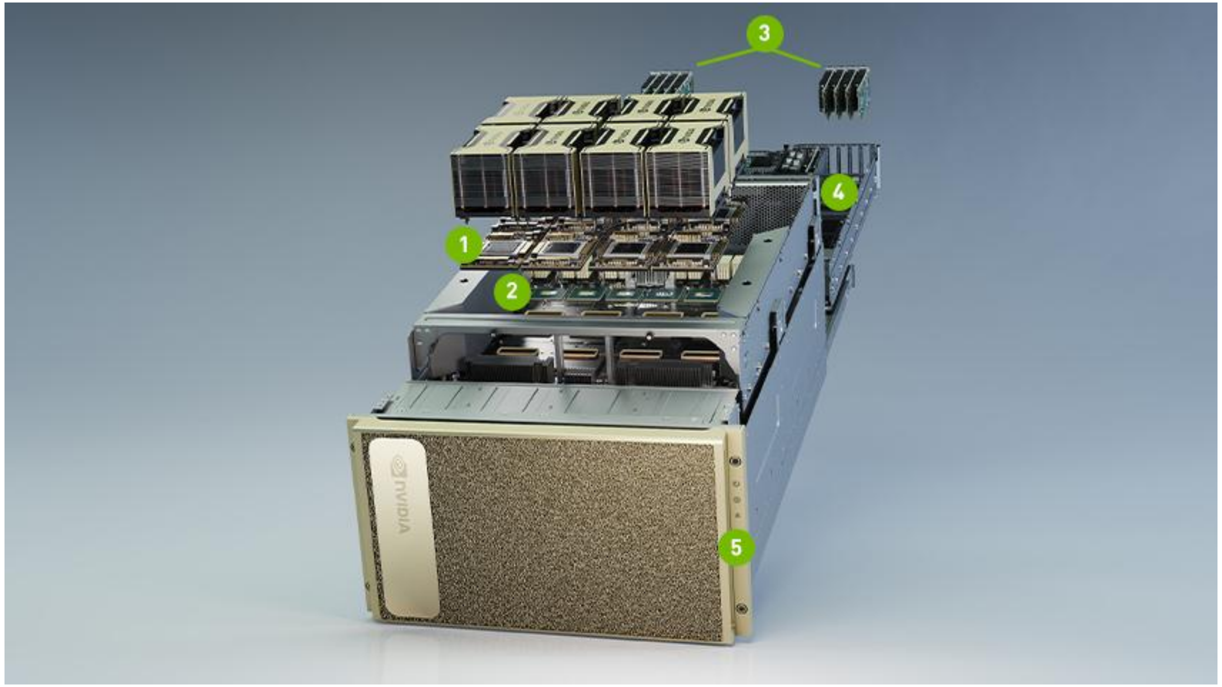
1 8x NVIDIA A100 GPUs with 640 GB Total GPU Memory
12 NVLinks/GPU, 600 GB/s GPU-to-GPU Bi-directonal Bandwidth

2 6x NVIDIA NVSwitches
4.8 TB/s Bi-directional Bandwidth, 2X More than Previous Generation NVSwitch

3 10X NVIDIA Connectx-6 200 Gb/s Network Interface
500GB/s Peak Bidirectional Bandwidth

4 Dual 64-Core AMD CPUs and 2 TB System Memory
3.2X More Cores to Power the Most Intensive AI Jobs

5 30 TB Gen4 NVME SSD
50 GB/s Peak Bandwidth, 2X Faster than Gen3 NVME SSDs



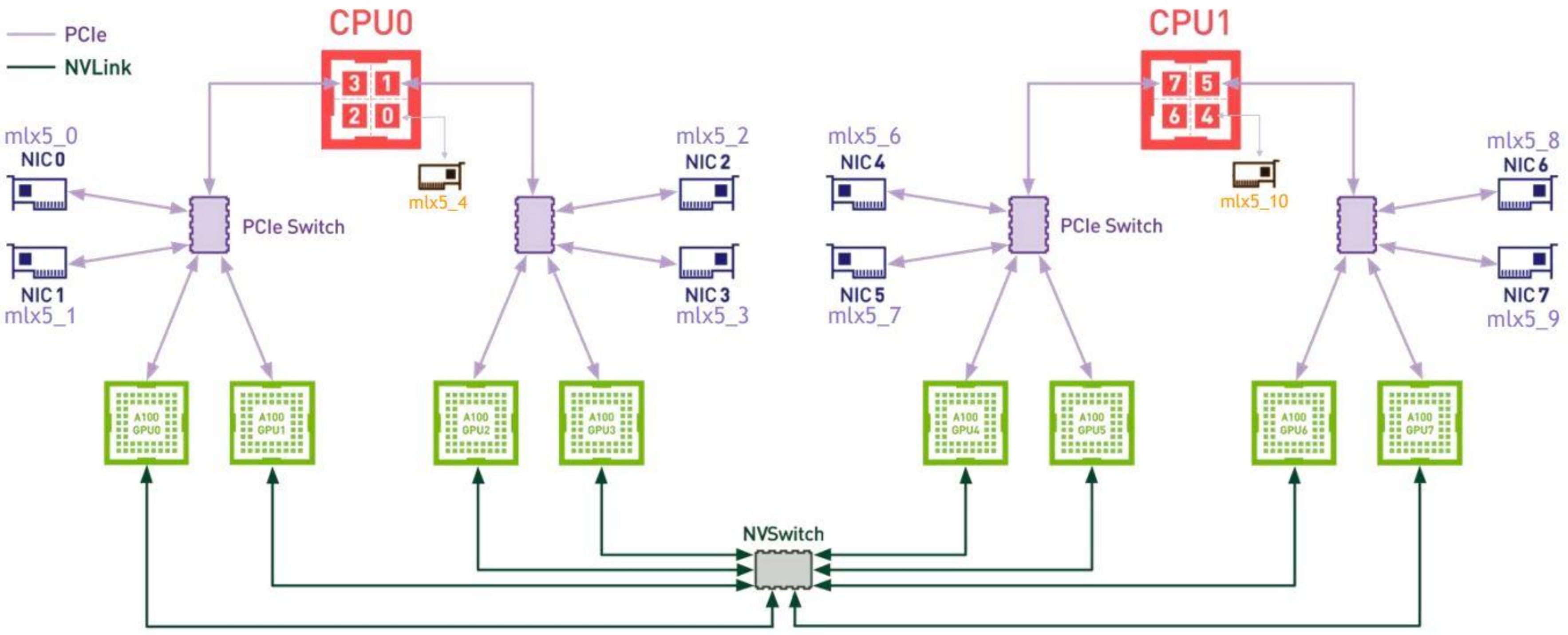
# Node architecture

10 DGX-100A

## DGX A100

Data plane (can be used as eth or IB)
Compute plane (IB)

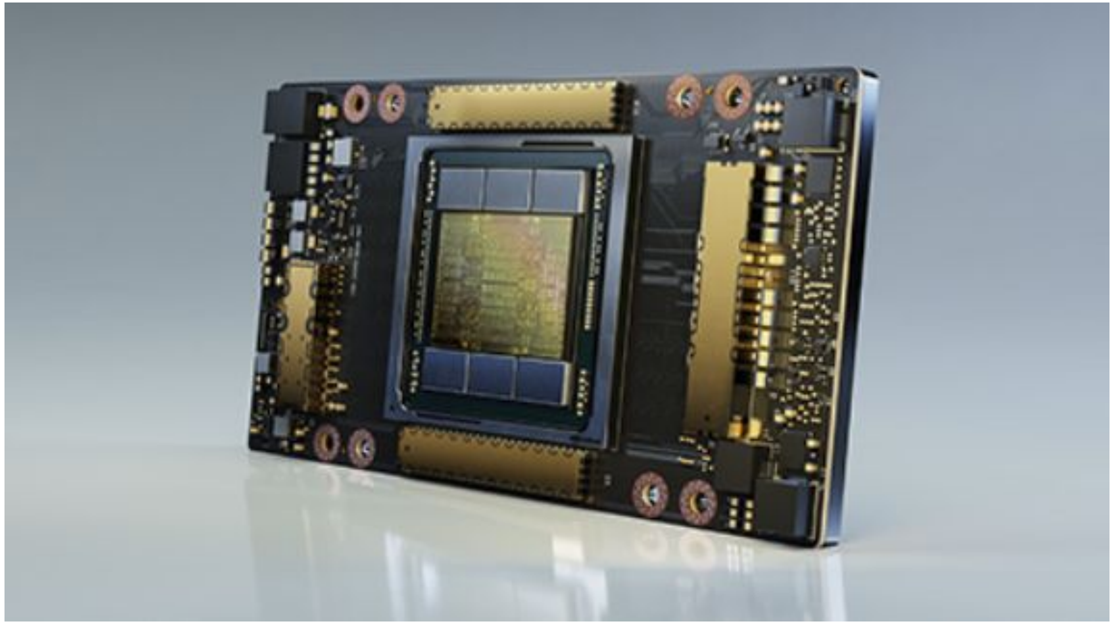
High-level Topology Overview (with options)



# Node architecture

NVIDIA AMPERE A100

- 4 Tensor cores;
- 64 Single-precision (FP32) cores;
- 64 integer (INT32) cores;
- 32 double precision (FP64) cores;
- 256 KB of register file (RF);
- Up to 164 KB shared memory (configurable).



Tesla A100 peak computational throughput is:

- 312/624 Peak Tensor TOPS/TFLOPS
- 9.7 Peak FP64 TFLOPS
- 19.5 Peak FP32 TFLOPS

# Activate your Axis Account : STEP 1

# AXIS REGISTRATION

1: You should have received a welcome email from AXIS Security to activate account.

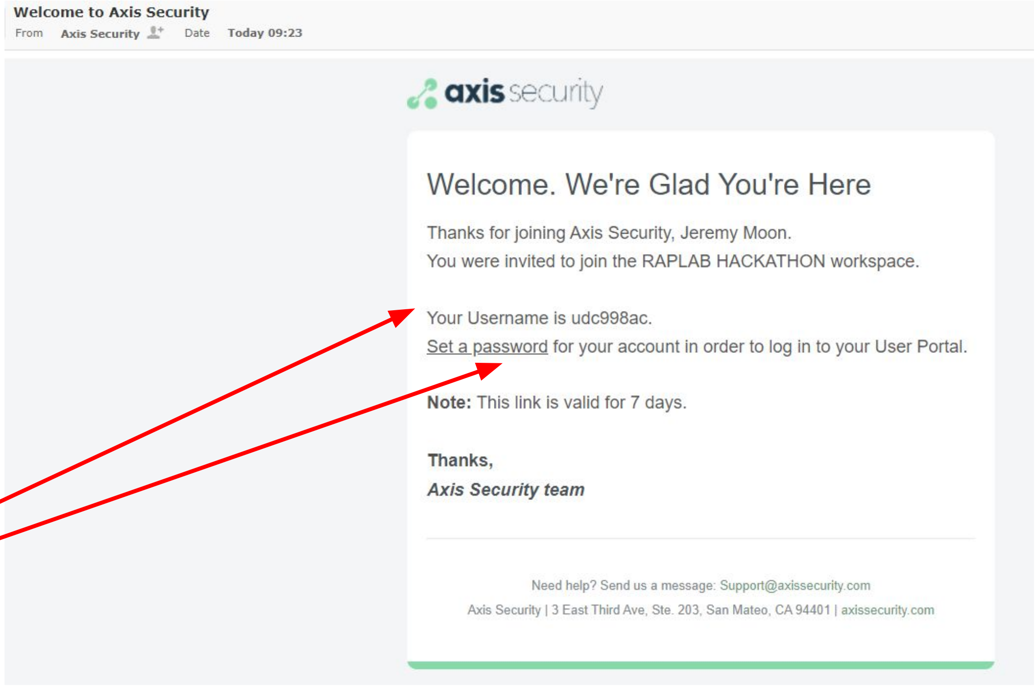
2: The welcome email will list your username.

3: Click on the Set a password link.

4: This will bring up a windows for you to set your password for AXIS.

5: Upon successfully setting your password you will be sent to the login page. Your AXIS account has now been activated.

1



**Welcome to Axis Security**
From Axis Security Date Today 09:23

axis security

Welcome. We're Glad You're Here

Thanks for joining Axis Security, Jeremy Moon.
You were invited to join the RAPLAB HACKATHON workspace.

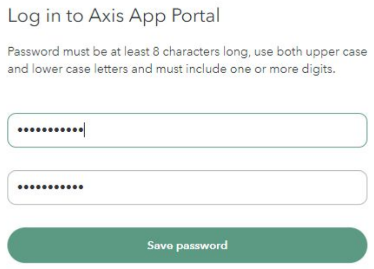
Your Username is udc998ac.
Set a password for your account in order to log in to your User Portal.

**Note:** This link is valid for 7 days.

**Thanks,**
***Axis Security team***

Need help? Send us a message: Support@axissecurity.com
Axis Security | 3 East Third Ave, Ste. 203, San Mateo, CA 94401 | axissecurity.com

2



Log in to Axis App Portal

Password must be at least 8 characters long, use both upper case and lower case letters and must include one or more digits.

•••••••••••

•••••••••••

# Connecting to the Cluster : STEP 2

# Connecting to the Cluster

➔ **Login to Axis with your credentials**

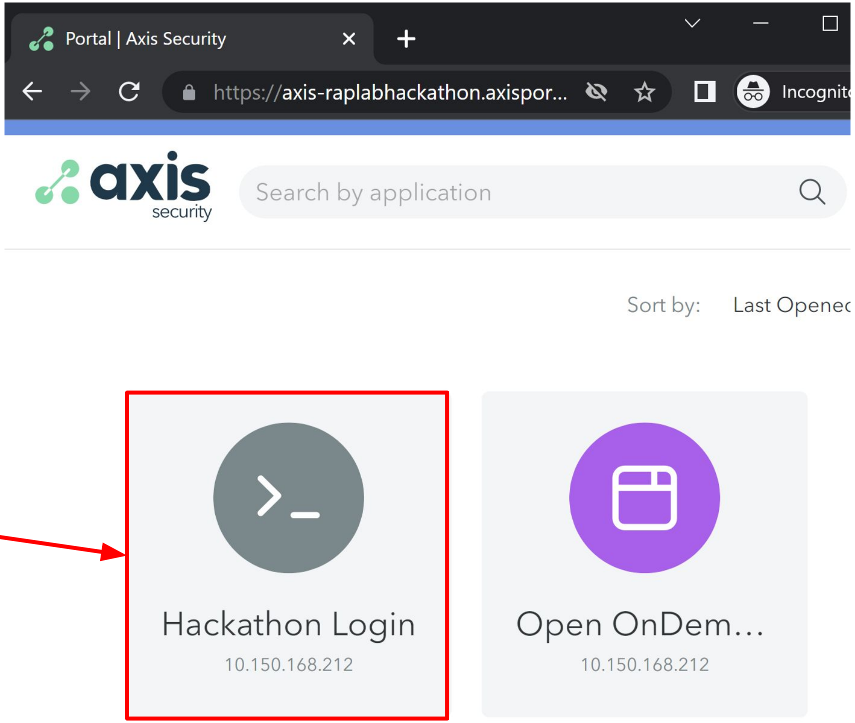
- Link : https://axis-raplabhackathon.axisportal.io/apps
- Use Chrome browser or make sure your browser does not block pop ups

# Connecting to the Cluster

1. Login to Axis with your credentials: [https://axis-raplabhackathon.axisportal.io/apps](https://axis-raplabhackathon.axisportal.io/apps)

*Use Chrome or make sure your browser does not block pop ups!

2. Click on the “Hackathon Login” app



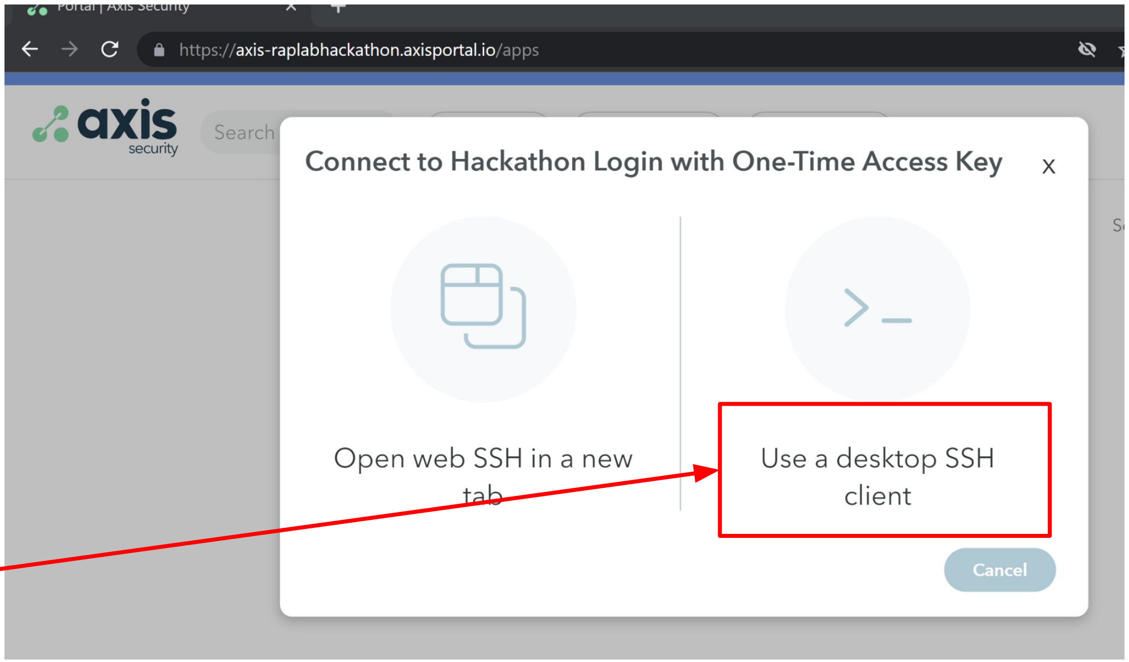
# Connecting to the Cluster

There are 2 methods to connect to the cluster.

1) Using a desktop client or terminal.
2) Web based SSH.

For Example for using a desktop client:

- Click on the “Use a desktop SSH client”

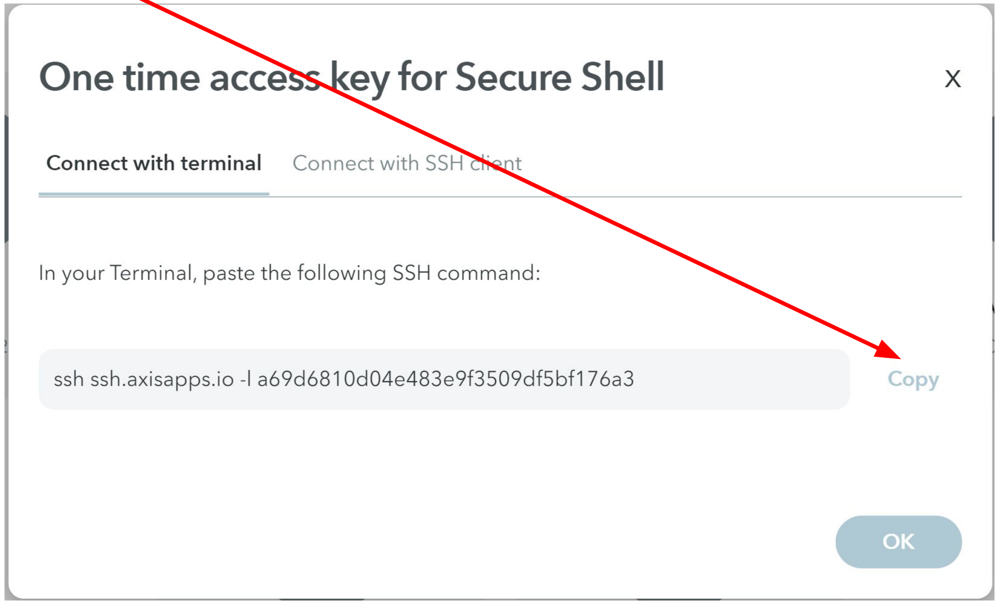


Portal | Axis Security
https://axis-raplabhackathon.axisportal.io/apps
axis
security
Search
Open web SSH in a new tab
Use a desktop SSH client
Cancel

Connect to Hackathon Login with One-Time Access Key

# Connecting to the Cluster

- Click “Copy” and open your preferred terminal (Powershell etc)



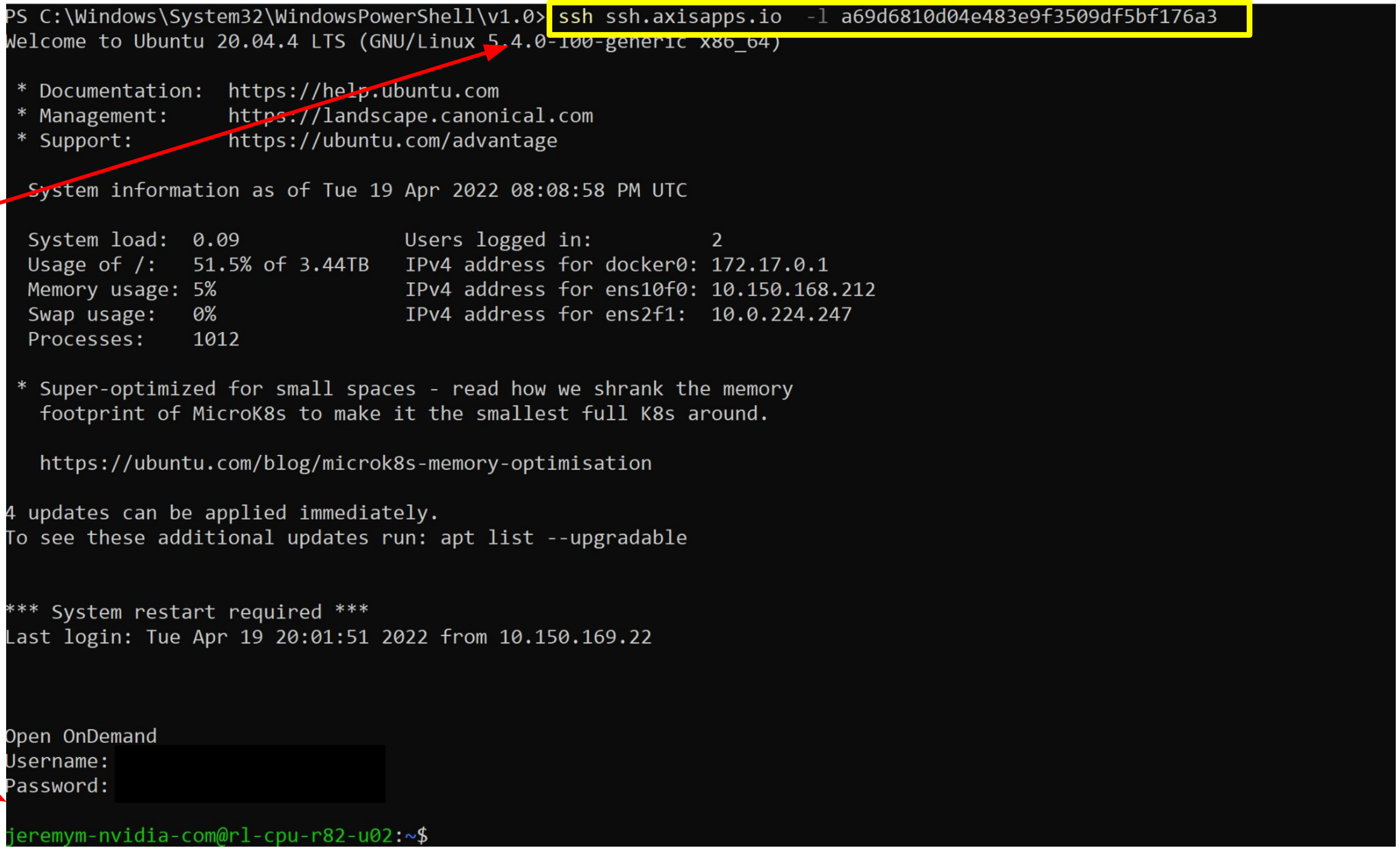
In your Terminal, paste the following SSH command:

```
ssh ssh.axisapps.io -l a69d6810d04e483e9f3509df5bf176a3
```

# Connecting to the Cluster

- Enter the command you copied in the last step and paste it into your terminal and click your enter key.

- You are now logged into the Cluster.

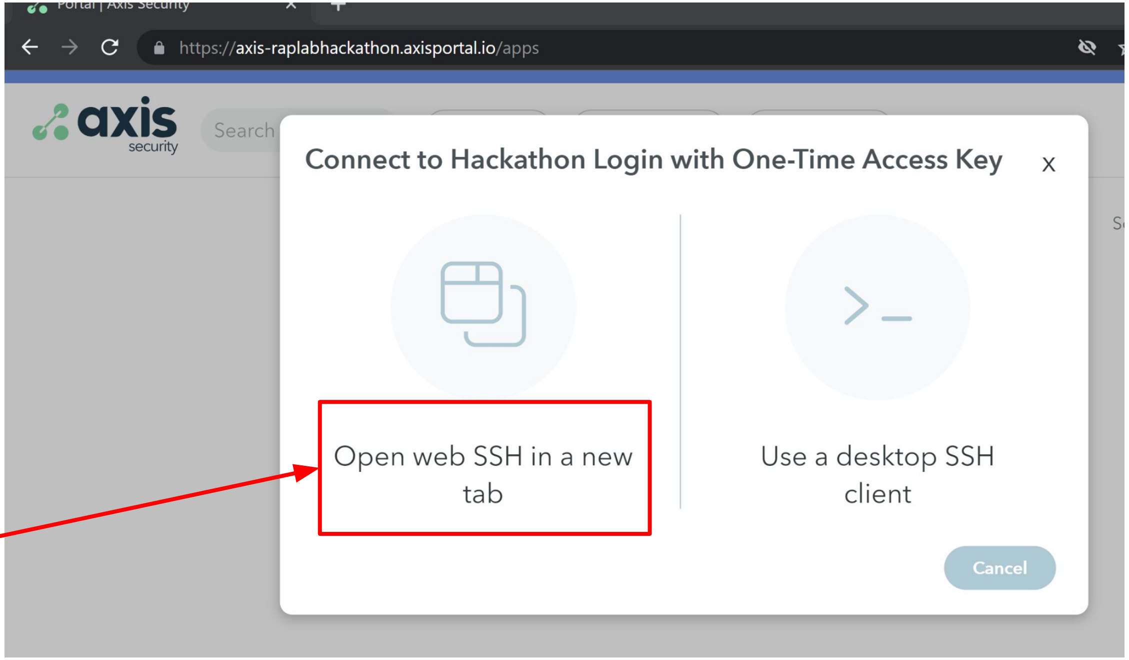


```
PS C:\Windows\System32\WindowsPowerShell\v1.0> ssh ssh.axisapps.io  -l a69d6810d04e483e9f3509df5bf176a3
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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

  System information as of Tue 19 Apr 2022 08:08:58 PM UTC

  System load:  0.09                Users logged in:          2
  Usage of /:   51.5% of 3.44TB     IPv4 address for docker0: 172.17.0.1
  Memory usage: 5%                  IPv4 address for ens10f0: 10.150.168.212
  Swap usage:   0%                  IPv4 address for ens2f1:  10.0.224.247
  Processes:    1012

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

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


*** System restart required ***
Last login: Tue Apr 19 20:01:51 2022 from 10.150.169.22


Open OnDemand
Username: [redacted]
Password:

jeremym-nvidia-com@rl-cpu-r82-u02:~$
```

# Connecting to the Cluster

There are 2 methods to connect to the cluster.

1) Using a desktop client or terminal.
2) Web based SSH.



https://axis-raplabhackathon.axisportal.io/apps

Connect to Hackathon Login with One-Time Access Key

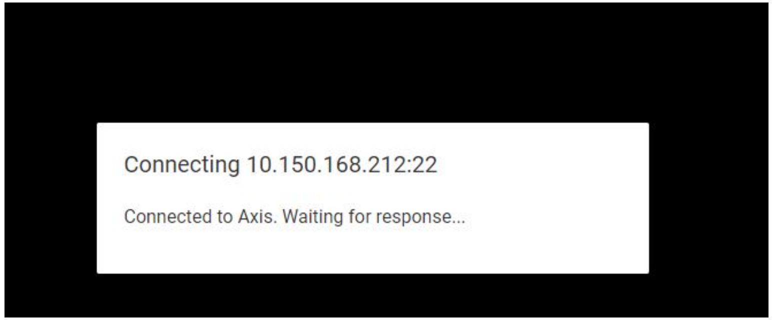
2)Using the web based SSH:

- Click on the "Open web SSH in a new tab"

The main advantages of web-based SSH can be summarized as follows: Accessibility: Web-based SSH as described in this article requires no local installation of client software. It is thus possible to access SSH servers through a web browser from anywhere

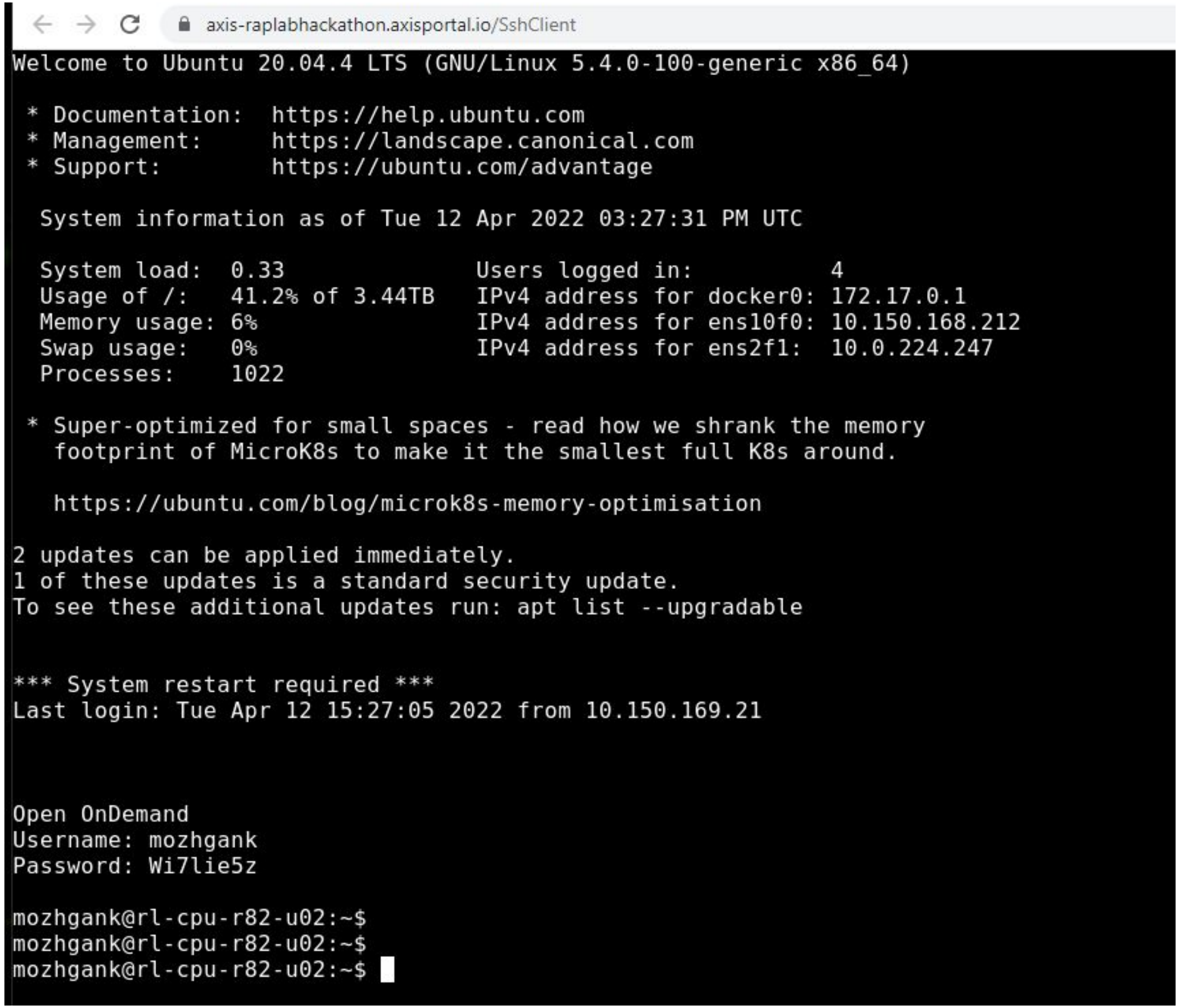
# Connecting to the Cluster

## Be patient ..



Connecting 10.150.168.212:22

Connected to Axis. Waiting for response...



```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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

  System information as of Tue 12 Apr 2022 03:27:31 PM UTC

  System load:  0.33               Users logged in:          4
  Usage of /:   41.2% of 3.44TB    IPv4 address for docker0: 172.17.0.1
  Memory usage: 6%                 IPv4 address for ens10f0: 10.150.168.212
  Swap usage:   0%                 IPv4 address for ens2f1:  10.0.224.247
  Processes:    1022

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

2 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Tue Apr 12 15:27:05 2022 from 10.150.169.21



Open OnDemand
Username: mozhgank
Password: Wi7lie5z

mozhgank@rl-cpu-r82-u02:~$
mozhgank@rl-cpu-r82-u02:~$
mozhgank@rl-cpu-r82-u02:~$
```

You are now connected to the cluster

# Modules & Environment

# Common module commands

List currently loaded modules.

```
module list
```

List available packages.

```
module avail
```

Description of specified module.

```
module help [modulefile]
```

Displays information about specified module, including environment changes, dependencies, software version and path.

```
module show [modulefile]
```

Loads module or specifies which dependencies have not been loaded.

```
module load [modulefile]
```

Unloads specified module from environment.

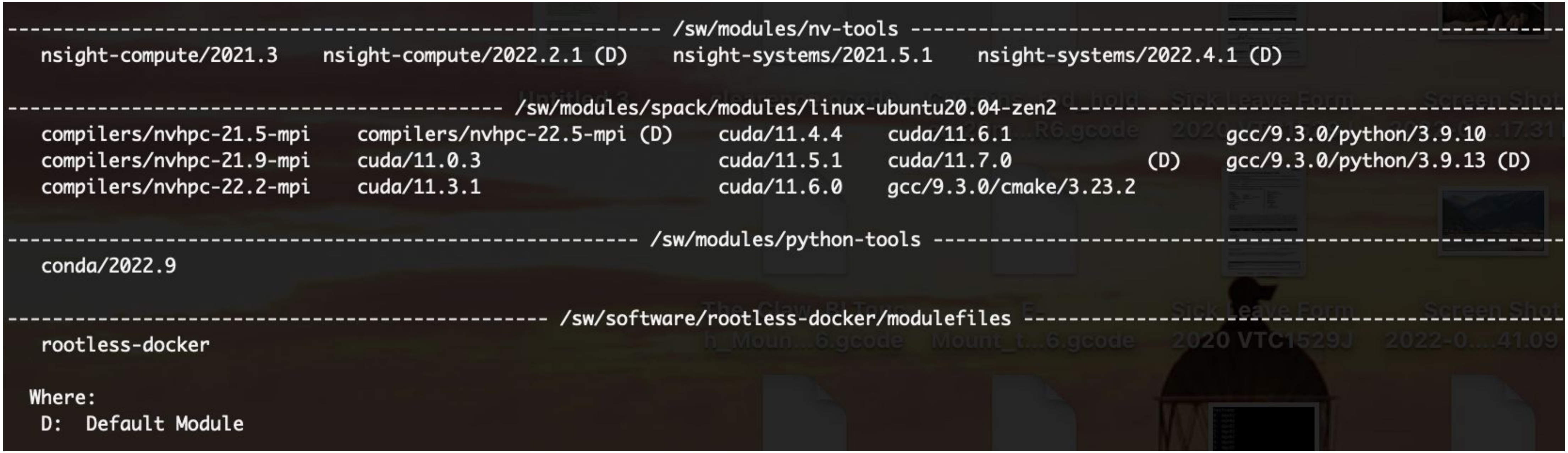
```
module unload [modulefile]
```

Unloads all loaded modules

```
module purge
```

# Overview environment

## Current pre-installed modules



```
---------------------------------------------------------- /sw/modules/nv-tools ----------------------------------------------------------
   nsight-compute/2021.3    nsight-compute/2022.2.1 (D)    nsight-systems/2021.5.1    nsight-systems/2022.4.1 (D)

-------------------------------------------- /sw/modules/spack/modules/linux-ubuntu20.04-zen2 --------------------------------------------
   compilers/nvhpc-21.5-mpi    compilers/nvhpc-22.5-mpi (D)    cuda/11.4.4    cuda/11.6.1                   gcc/9.3.0/python/3.9.10
   compilers/nvhpc-21.9-mpi    cuda/11.0.3                     cuda/11.5.1    cuda/11.7.0            (D)    gcc/9.3.0/python/3.9.13 (D)
   compilers/nvhpc-22.2-mpi    cuda/11.3.1                     cuda/11.6.0    gcc/9.3.0/cmake/3.23.2

-------------------------------------------------------- /sw/modules/python-tools --------------------------------------------------------
   conda/2022.9

------------------------------------------------ /sw/software/rootless-docker/modulefiles ------------------------------------------------
   rootless-docker

  Where:
   D:  Default Module
```

# Storage Policies

# Storage policy

- Central storage is via NFS
  - If you compile on the compute node then please use `/tmp` or local scratch
  - If you have heavy I/O then please please use `/tmp` or local scratch
- There is no quota on `$HOME`
  - If you create lot of temporary files then please use `/tmp` on the compute node
  - If you create big profile traces then delete them as soon as you do not need them
- The `/tmp` location on compute node is not persistent!
  - When you job ends or die, `/tmp` gets cleaned up by the scheduler/`OS`
- When the hackathon ends and the account deleted, all `$HOME` are purged. The is no backup
  - Save your files.

# Common Scheduler commands

# Slurm Common Commands

For full list of common Slurm command, follow this link.

| Command | Description | Detailed |
|---|---|---|
| sinfo | Reports the state of the partitions and nodes managed by Slurm | PARTITION: the name of the partition<br>AVAIL: whether the partition is up or down<br>TIMELIMIT: the maximum length a job will will run in the format Days-Hours:Minutes:Seconds<br>NODES: the number of nodes of that configuration<br>STATE: down* if jobs cannot be ran, idle if it is are available for jobs, alloc if all the CPUs in the partition are allocated to jobs, or mix if some CPUs on the nodes are allocated and others are idle.<br>NODELIST: specific nodes associated with that partition. |
| sacct | Lists the jobs that are running or have been run. | |
| squeue | Lists the state of all jobs being run or scheduled to run.<br>Use squeue -u username to view only the jobs from a specific user | JOBID: number id associated with the job<br>PARTITION: name of partition running the job<br>NAME: name of the job ran with sbatch or sinteractive<br>USER: who ordered the job to be ran<br>ST: State of the job, PD for pending, R for running<br>TIME: how long the job has been running in the format Days-Hours: Minutes:Seconds<br>NODES: number of nodes allocated to the job<br>NODELIST(REASON): either the name of the node running the job of the reason the job is not running such as JobHeldAdmin (job is prevented from running by the administrator). |
| scancel | Signals or cancels a job. One or more jobs separated by spaces may be specified. | |

# Job Submission

## Resource Requests

To run your job, you will need to specify what resources you need.  These can be memory, cores, nodes, GPUs, etc. There is a lot of flexibility in the scheduler to get specifically the resources you need.

| Options | Description |
|---|---|
| --nodes | The number of nodes for the job (computers) |
| --mem | The amount of memory per node that your job need |
| -n | The total number of tasks you job requires |
| --gres gpu:# | The number of GPUs per node you need in you job |
| --gres=gpu:type:# | you can also specify the type of gpu.  We have mostly p100s, but also 2 v100s for testing |
| --qos | the QOS you want to run in, currently normal or debug |
| --mem-per-cpu= | The amount of memory per cpu your job requires |
| -N | The minimum (and maximum) number of nodes required |
| --ntasks-per-node=# | tasks per node. |
| --exclusive | this will get you exclusive use of the node |
| --constraint= | constrain to particular nodes.  use skylake or broadwell for particular processor types |

# Scheduler (SLURM) policies

Each user can (by default)...

- Run one job at a time
- Request max 8 hours walltime
- Request 2 GPU and 32 cores (HyperThreading is enabled and MIG Disabled)
- Each team is limited to 2 GPU

If you are ready for benchmarks, scaling 1~8 GPUs, exploring NVLink effects...

- Request to mentors/sysadmins to see a possibility to enable the user access to a full node (8 GPUs and 128 cores)
- Walltime remains 8 hours but, if you have very long tests, an extension can be granted

# Common scheduler commands

## Interactive mode, 1 GPU

```
srun --ntasks=5 --nodes=1 --cpus-per-task=2 \
     --partition=gpu --time=4:00:00 --gres=gpu:1 \
     --pty /bin/bash
```

*Please note this is a sample run. The number of tasks will depend on the number of GPU core assigned per team.

# Scheduler (SLURM) policy

## Batch mode, 1 GPU (serial job)

```
#!/bin/bash
#SBATCH --partition=gpu
#SBATCH --nodes=1             # Max is 1
#SBATCH --ntasks=16           # Max is 16 (1/8 of 2x 64 AMD EPYC CPUs)
#SBATCH --cpus-per-task=2     # Max is 2 (Clustered Multithreading is on)
#SBATCH --gres=gpu:1          # Max is 1 (1 single A100)
#SBATCH --time=4:00:00        # Max is 4 hours

module purge > /dev/null 2>&1
#module load ...

export MYEXE=...
export OMP_PROC_BIND=true
export OMP_NUM_THREADS=5

./${MYEXE} 2>&1 | tee out.${SLURM_JOBID}
```

# Scheduler (SLURM) policy

## Batch mode, 1 GPU (parallel job)

```
#!/bin/bash

#SBATCH --partition=gpu
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=1
#SBATCH --gres=gpu:1
#SBATCH --time=4:00:00

module purge > /dev/null 2>&1
module load compilers/nvhpc-21.9-mpi
export MYEXE=hello_world

mpicc -o hello_world hello_world.c
mpirun -np 2 --report-bindings ${MYEXE} 2>&1 | tee out.${SLURM_JOBID}
```

# Visual profiling tools

Nsight Compute (**ncu-ui**)

```
module load nvhpc/22.1 nsight-compute/2021.3
```

Nsight Systems (**nsys-ui**)

```
module load nvhpc/22.1 nsight-systems/2021.5.1
```

NVIDIA Visual Profiler (**nvvp**)

```
module load nvhpc/22.1 gcc/9.3.0/python/3.9.10
```

# Containers

# Containers

The Curiosity cluster supports singularity containers.

If you need to create your own containers without sudo. Singularity provides with a `--fakeroot` option, so when creating custom singularity containers.

# NGC Containers

If you are looking for pre-configured containers, please look at GPU-optimized AI, Machine Learning, & HPC Software | NVIDIA NGC, you can also ask your Mentor for assistance.

# Singularity

# Running singularity containers

- To transfer the images to the cluster, please visit the “Transfer files to/from hackathon cluster” section.
- You must first request and interactive session using `srun` command, then get an interactive shell in to the image, using the following command: `singularity shell path/to/imagefile.simg`
- To manually mount the existing directories in the image to your own path use `–B` flag: `singularity shell –B path/to/file:/file path/to/imagefile.simg`

# Create your own Singularity image

```
Bootstrap: docker
From: nvcr.io/hpc/pgi-compilers:ce

%runscript

    "$@"

%post

    apt-get -y update
    apt-get -y install --no-install-recommends python3-pip python3-setuptools zi
p
    rm -rf /var/lib/apt/lists/*
    pip3 install --no-cache-dir jupyter

%files

labs /labs

%environment
XDG_RUNTIME_DIR=
```

To build the container from the example def file (above) on CURIOSITY, use below command:

```
singularity build --fakeroot myimage.simg Singularity
```

To run interactively and using NVIDIA GPUs in the image: `singularity shell --nv myimage.simg`

To convert Docker containers to singularity, checkout: https://github.com/NVIDIA/hpc-container-maker

Checkout the singularity examples : https://sylabs.io/docs/

# Singularity Commands

## Build

**Build** takes a target as input and outputs a Container. Build is a multipurpose tool:

```
$ singularity [global options...] build [local options...] <IMAGE PATH> <BUILD SPEC>
```

**Option1**: Pull and Build a singularity container from DockerHub

```
$ singularity build docker://tensorflow/tensorflow:latest-gpu
```

## **Option 2**: Pull and Build a singularity container from Definition Script (Below)

```
Bootstrap: docker
From: nvcr.io/nvidia/tensorflow:20.11-tf2-py3

%runscript

    "$@"

%post

    apt-get -y update
    apt-get -y install --no-install-recommends python3-pip python3-setuptools zip build-essential
    rm -rf /var/lib/apt/lists/*
    pip3 install --no-cache-dir jupyter

%files

    English/ /labs

%environment
XDG_RUNTIME_DIR=

%labels
```

```
$ sudo singularity build tensorflow.simg Singularity
```

**OR**

```
$ singularity --fakeroot build tensorflow.simg Singularity
```

# Singularity Commands

## Run + Exec + Shell

- **Run**: runs the Singularity runfile

```
$ singularity run <singularity_container.simg>
```

- **Exec**: execute container commands from outside the container (like cat /etc/os-release)

```
$ singularity exec <singularity_container.simg> cat /etc/os-relase
```

- **Shell**: opens a shell inside the container

```
$ singularity shell <singularity_container.simg>
```

# Singularity Commands

## Nvidia GPUs

**--nv** flag enables the container to setup the environment to use NVIDIA GPUs and the CUDA libraries.  Run, exec and shell commands can take the flag.

```
$ singularity build tensorflow.simg docker://tensorflow/tensorflow:latest-gpu
```

```
$ singularity run --nv tensorflow.simg
```

# Singularity Commands

## Loading local folders

You can use local files from within your container

```
$ echo “hello world” > file1.txt

$ singularity exec tensorflow.simg cat file1.txt

>hello world
```

Singularity binds /home/$USER, /tmp, and $PWD at runtime, to bind other directories use:

```
$ singularity exec --bind /<host dir>:/test  tensorflow.simg ls /test
```

This example binds the /test directory to a <host dir> inside the container so you can access it, allowing you to do an `ls` on it.

# Singularity Commands

## Making changes to containers

Build as **--sandbox** for writable:

```
$ sudo singularity build --sandbox test_tensorflow tensorflow.simg
```

**--fakeroot** option: fakeroot lets a user have nearly all the powers of root but only inside the container and does not extend to filesystem.  Sudo is not required.

```
$ singularity build --fakeroot test_tensorflow tensorflow.simg
```

# Transfer files to/from the Curiosity Cluster

# Transfer files to/from the Curiosity Cluster

➔ Login to Axis with your credentials

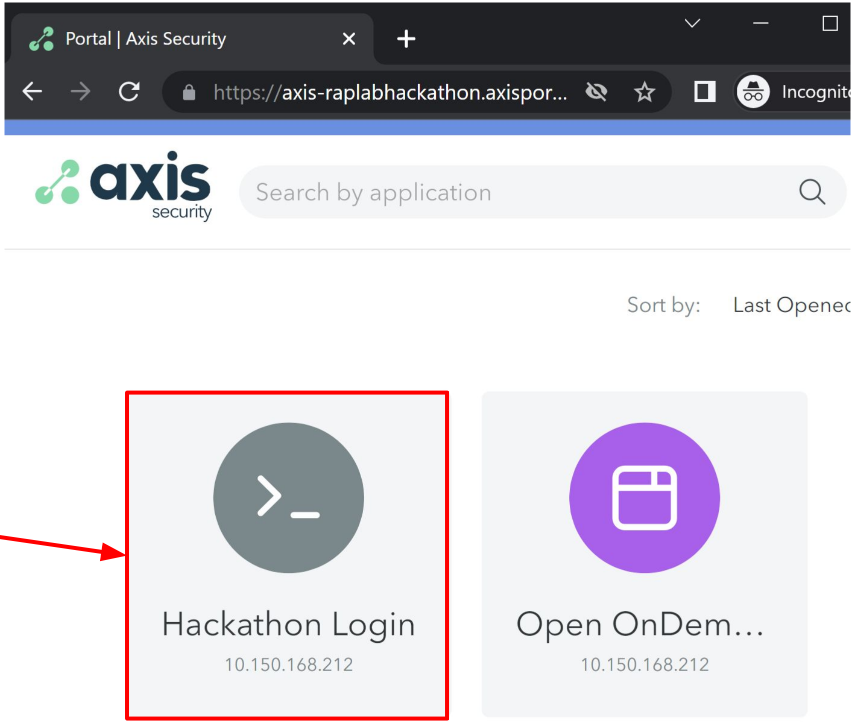
- Link : https://axis-raplabhackathon.axisportal.io/apps
- Use Chrome browser or make sure your browser does not block pop ups

# Transfer files to/from the Curiosity Cluster

1. Login to Axis with your credentials: [https://axis-raplabhackathon.axisportal.io/apps](https://axis-raplabhackathon.axisportal.io/apps)

*Use Chrome or make sure your browser does not block pop ups!

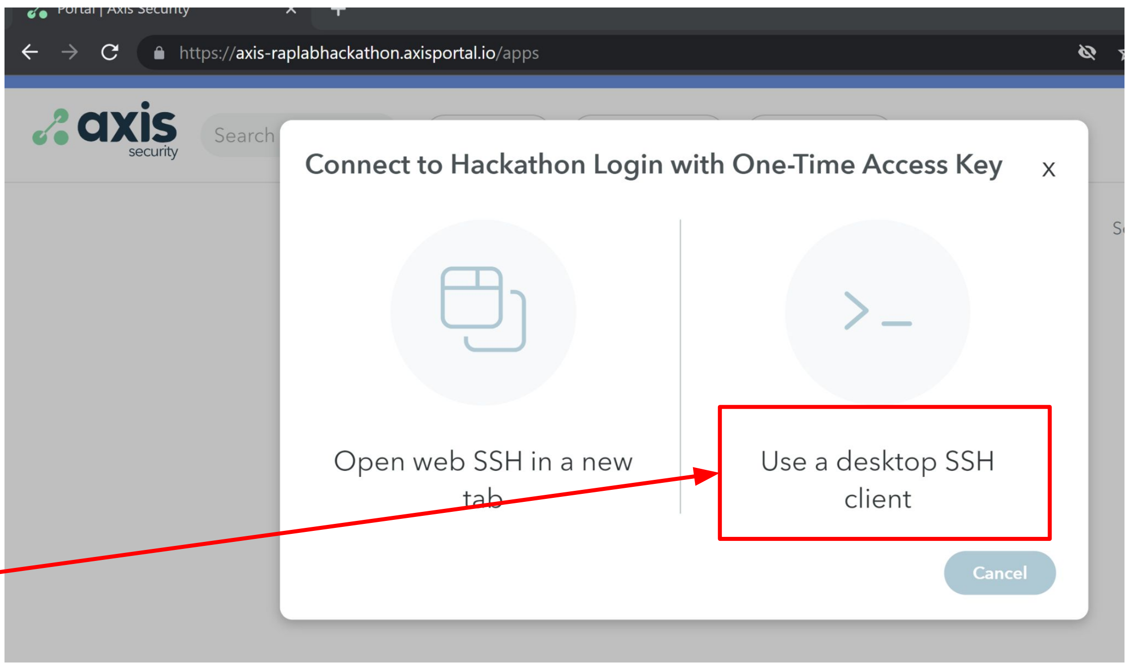
2. Click on the "Hackathon Login" app



# Transfer files to/from the Curiosity Cluster

There are 2 methods to connect to the cluster.

1) Using a desktop client or terminal.
2) Web based SSH.



https://axis-raplabhackathon.axisportal.io/apps

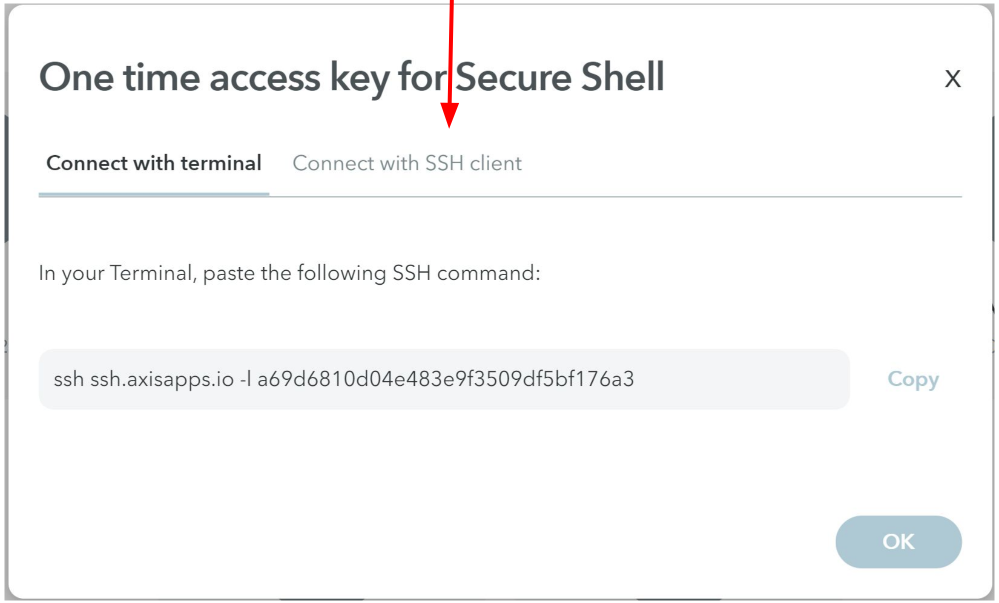
Connect to Hackathon Login with One-Time Access Key

1) Using a desktop client or terminal:

- Click on the “Use a desktop SSH client”

# Transfer files to/from the Curiosity Cluster

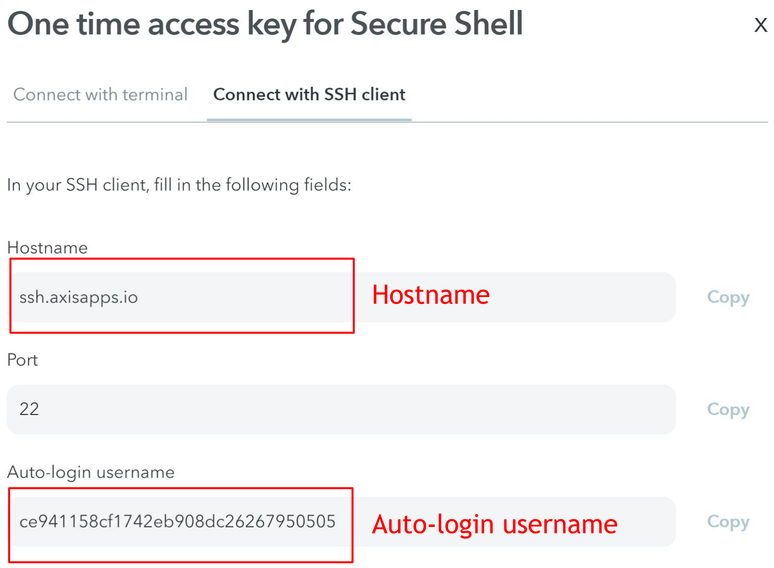
- Click on connect with SSH client, and make note of the Hostname and Auto-login username



One time access key for Secure Shell

In your Terminal, paste the following SSH command:

```
ssh ssh.axisapps.io -l a69d6810d04e483e9f3509df5bf176a3
```



One time access key for Secure Shell

In your SSH client, fill in the following fields:

Hostname: ssh.axisapps.io

Hostname

Port: 22

Auto-login username: ce941158cf1742eb908dc26267950505

Auto-login username

# Transfer files to/from the Curiosity Cluster

- Before sending your files to the cluster it is recommended to zip them
- In your terminal (PowerShell or another) enter the following command replacing the hostname and username with the hostname and auto-login username from the previous step.

```
scp event-container.zip <username>@<hostname>:~
```

- And now unzip **to** a folder

```
unzip event-container –d event-container
```

- To transfer **from** Curiosity cluster

```
scp <username>@<hostname>:~/event-container.zip event-container.zip
```

# Transfer files to/from the Curiosity Cluster

## Windows only

Install WinSCP

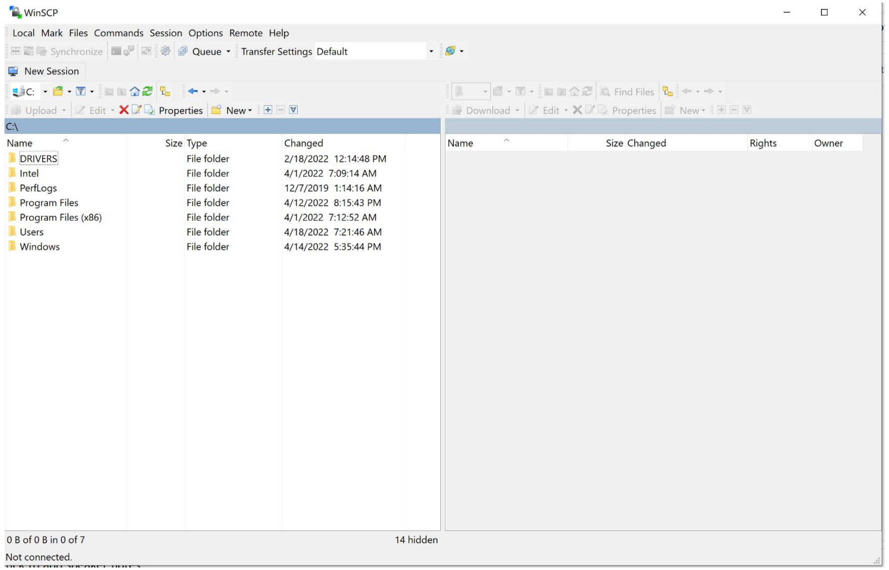
a. Download the WinSCP client: [https://winscp.net/eng/download.php](https://winscp.net/eng/download.php)

b. Run the file after it's downloaded, and accept the License Agreement

c. Click **Next** to select the default installation

d. Click **Install** to begin the installation

e. Click **Finish**



C:\

| Name | Size | Type | Changed |
|---|---|---|---|
| DRIVERS | | File folder | 2/18/2022 12:14:48 PM |
| Intel | | File folder | 4/1/2022 7:09:14 AM |
| PerfLogs | | File folder | 12/7/2019 1:14:16 AM |
| Program Files | | File folder | 4/12/2022 8:15:43 PM |
| Program Files (x86) | | File folder | 4/1/2022 7:12:52 AM |
| Users | | File folder | 4/18/2022 7:21:46 AM |
| Windows | | File folder | 4/14/2022 5:35:44 PM |

0 B of 0 B in 0 of 7 14 hidden

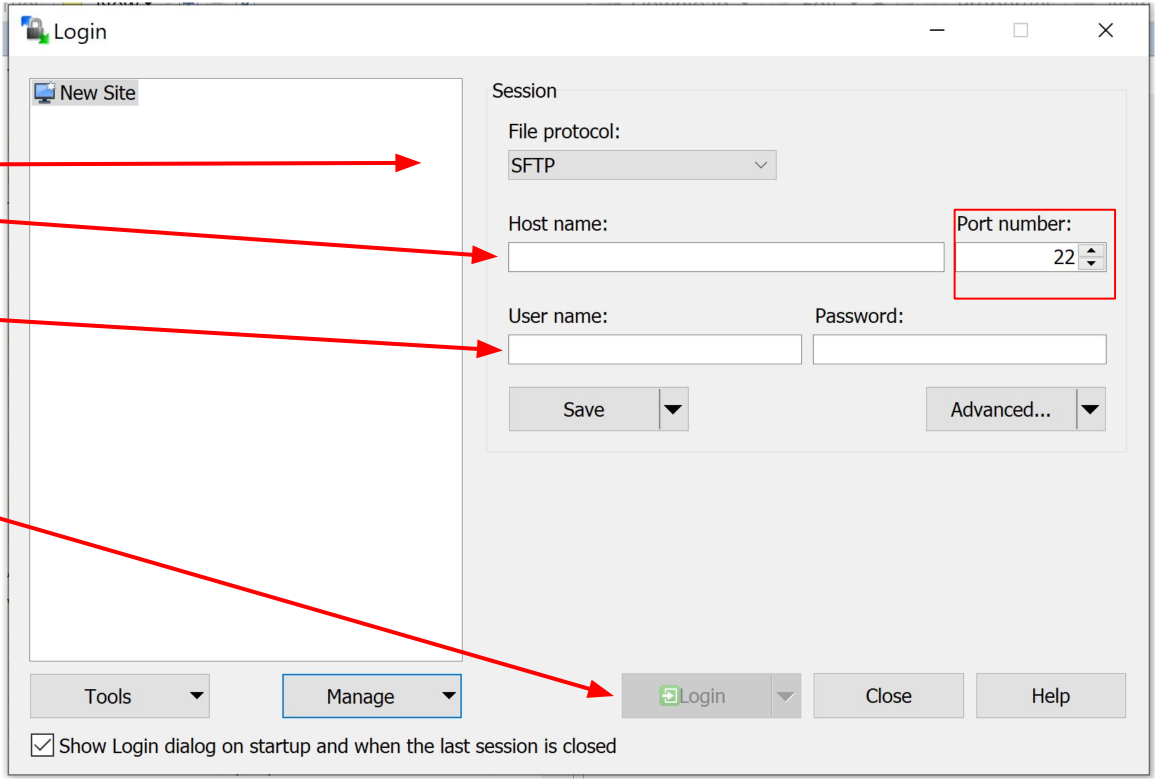
Not connected.

# Transfer files to/from the Curiosity Cluster

## Windows only

- Open WinSCP
- Change File Protocol to SCP
- Enter Hostname from the previous step
- Verify Port number is set for 22
- Enter the “Auto-login username”
- Click “Login”

Note: Please take the “Hostname”, and “autologin username” from slide 41.



Session

File protocol: SFTP

Host name:

Port number: 22

User name:

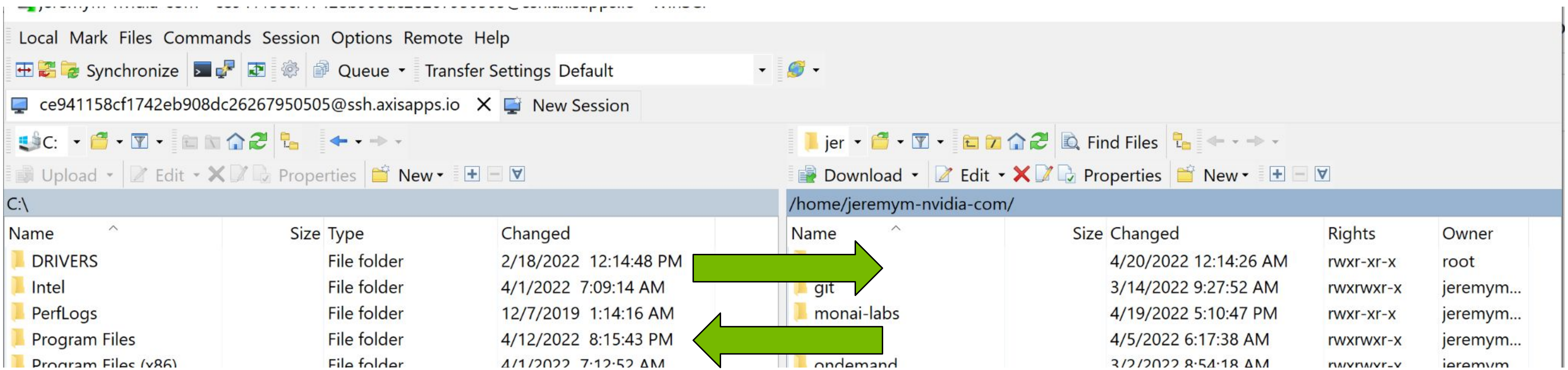
Password:

☑ Show Login dialog on startup and when the last session is closed

# Transfer files to/from the Curiosity Cluster

## Windows only

Now, you can drag and drop files in the panel to transfer files from/to the local machine to the cluster.



Transfer Settings Default

ce941158cf1742eb908dc26267950505@ssh.axisapps.io

C:

C:\

| Name | Size | Type | Changed |
|---|---|---|---|
| DRIVERS | | File folder | 2/18/2022 12:14:48 PM |
| Intel | | File folder | 4/1/2022 7:09:14 AM |
| PerfLogs | | File folder | 12/7/2019 1:14:16 AM |
| Program Files | | File folder | 4/12/2022 8:15:43 PM |
| Program Files (x86) | | File folder | 4/1/2022 7:12:52 AM |

jer

/home/jeremym-nvidia-com/

| Name | Size | Changed | Rights | Owner |
|---|---|---|---|---|
| | | 4/20/2022 12:14:26 AM | rwxr-xr-x | root |
| git | | 3/14/2022 9:27:52 AM | rwxrwxr-x | jeremym... |
| monai-labs | | 4/19/2022 5:10:47 PM | rwxr-xr-x | jeremym... |
| | | 4/5/2022 6:17:38 AM | rwxrwxr-x | jeremym... |
| ondemand | | 3/2/2022 8:54:18 AM | rwxrwxr-x | jeremym |

# Troubleshooting

# Troubleshooting & Workarounds (1/N)

**Q**: NSight Compute does not operate / start properly issuing a warning that a lock ***/tmp/nsight-compute-lock*** is already present or owned by another user

**A**: NSight Compute uses a global lock (see https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html#serialization) since serialization across processes is necessary for the collection of HW performance metrics (some GPU and driver objects can only be acquired by a single process at a time).

A workaround to this can be to use a different TMP directory (not ***/tmp***!). Before run the toop do the following:

***export TMPDIR=/raid/${SLURM_JOB_ID}/tmp***

This change is valid only in the SSH session on the current running interactive job

NVIDIA®