ARCATS Training Workshop

High Performance Computing and AI/ML

30 April 2025

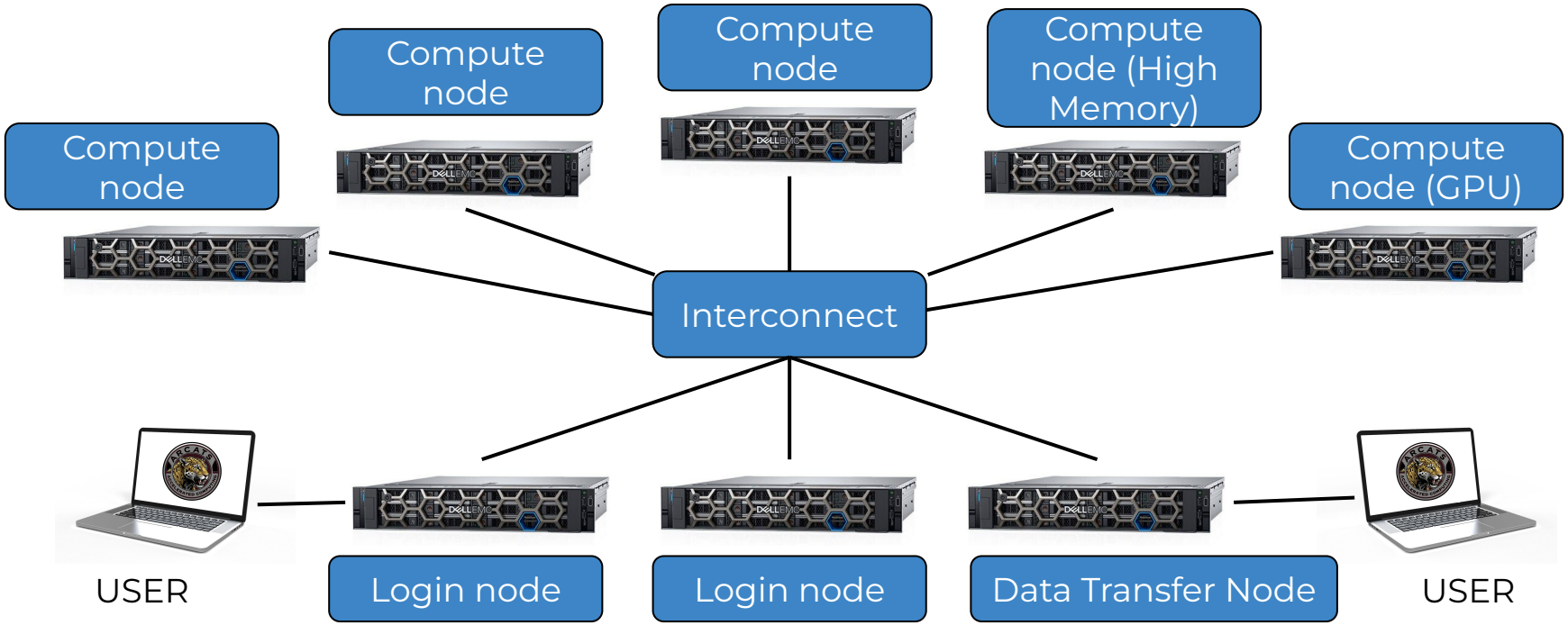
TEXAS A&M UNIVERSITY
SAN ANTONIO

High Performance
Research Computing
DIVISION OF RESEARCH

Workshop Outline

- High Performance Computing (HPC) Architecture
- Getting on Launch
- Overview of Launch Open OnDemand Portal
- Working in JupyterLab
- Coding in Python
- Introduction to AI/ML
- Deploying Deep Learning Models on HPC

HPC Architecture

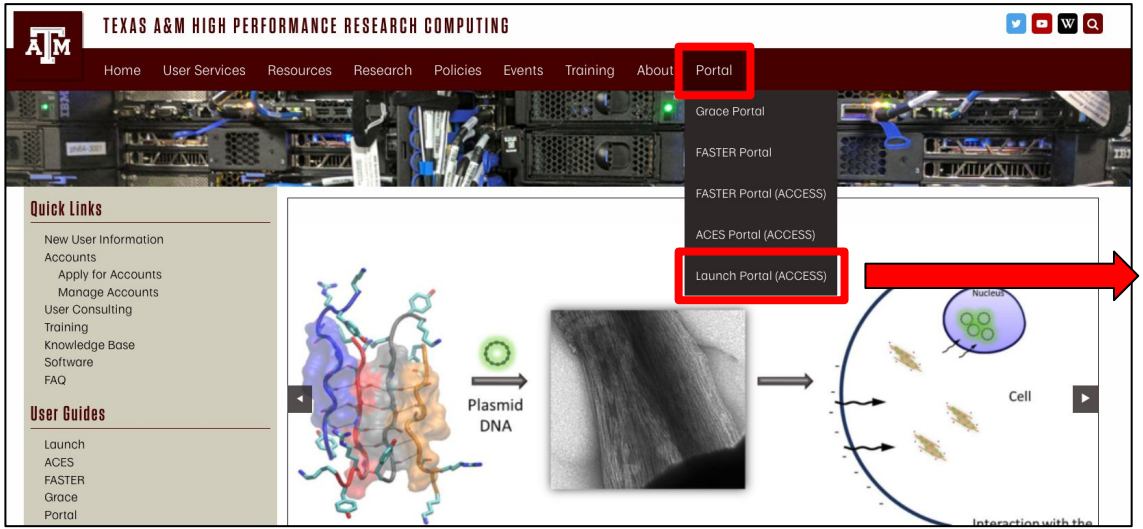


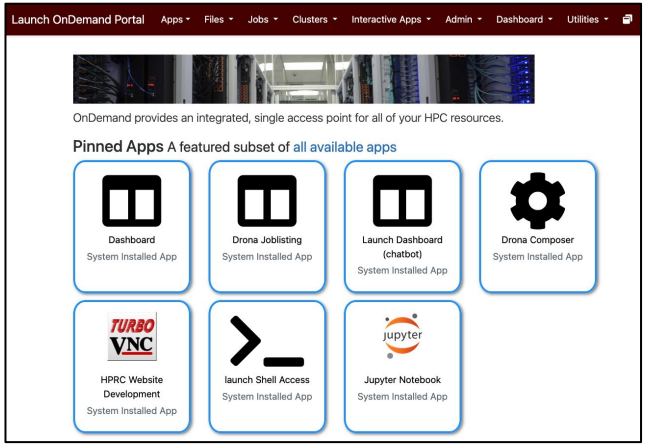
Coordinating multiple users

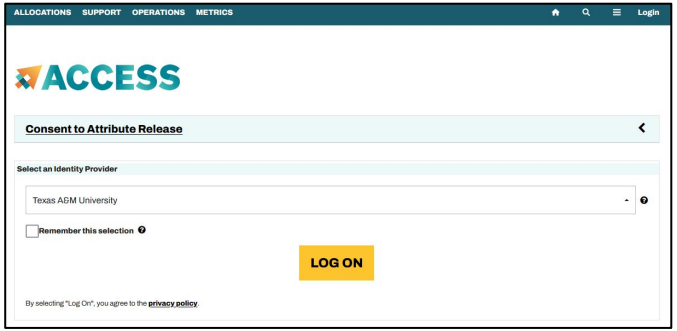
- HPC systems can have thousands of users, many running analyses or scripts simultaneously.
- HPC usage is coordinated by job orchestration software (Slurm).
- Large jobs must be submitted to or run interactively on compute nodes.
- Login nodes are for small jobs (job script creation, data management).
- Interactive GUI applications are automatically launched on compute nodes.

Getting on Launch

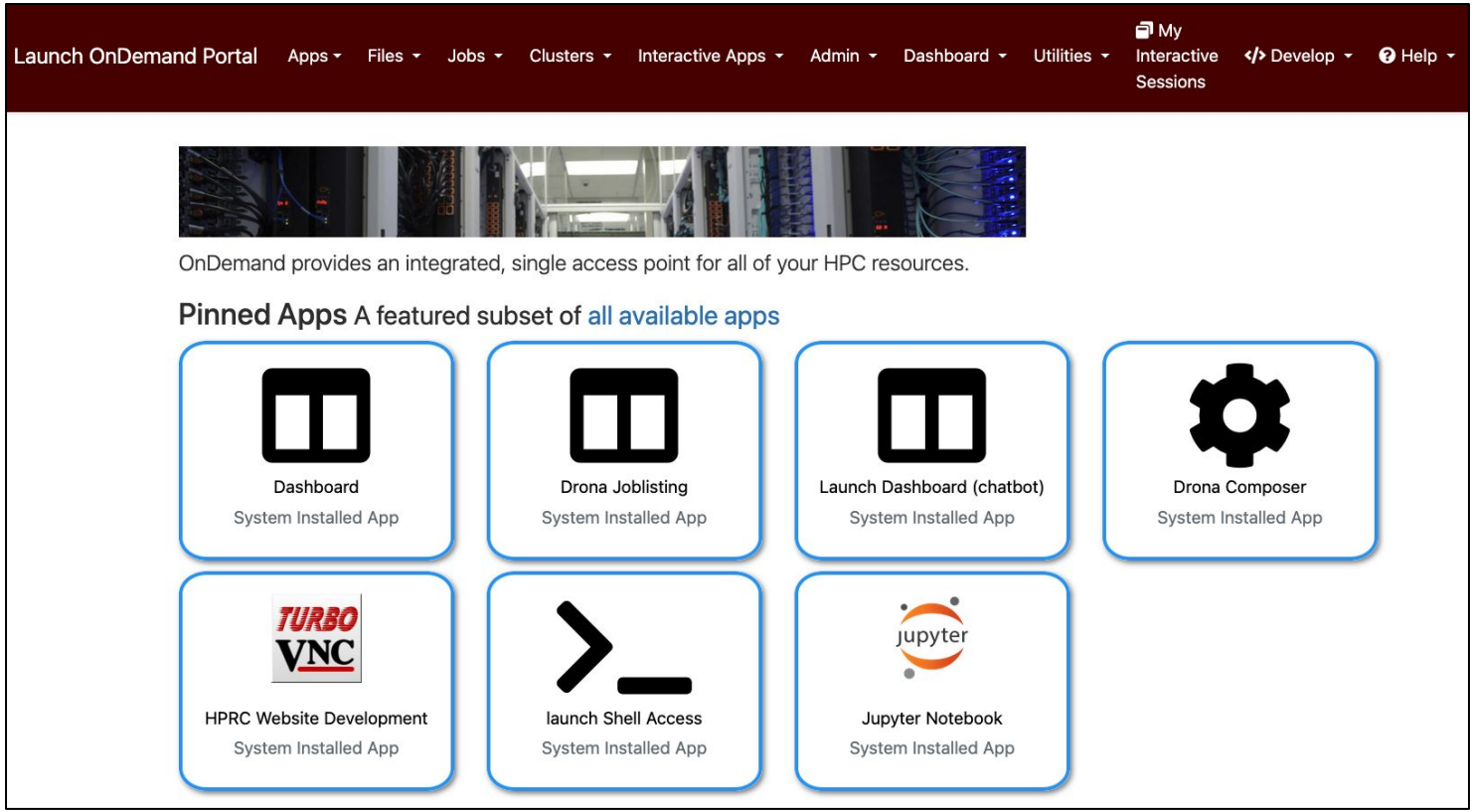
- Go to https://hprc.tamu.edu
- Hover over the “Portal” drop down menu
- Select “Launch Portal (ACCESS)”
- Log in via ACCESS
- Open On Demand Portal



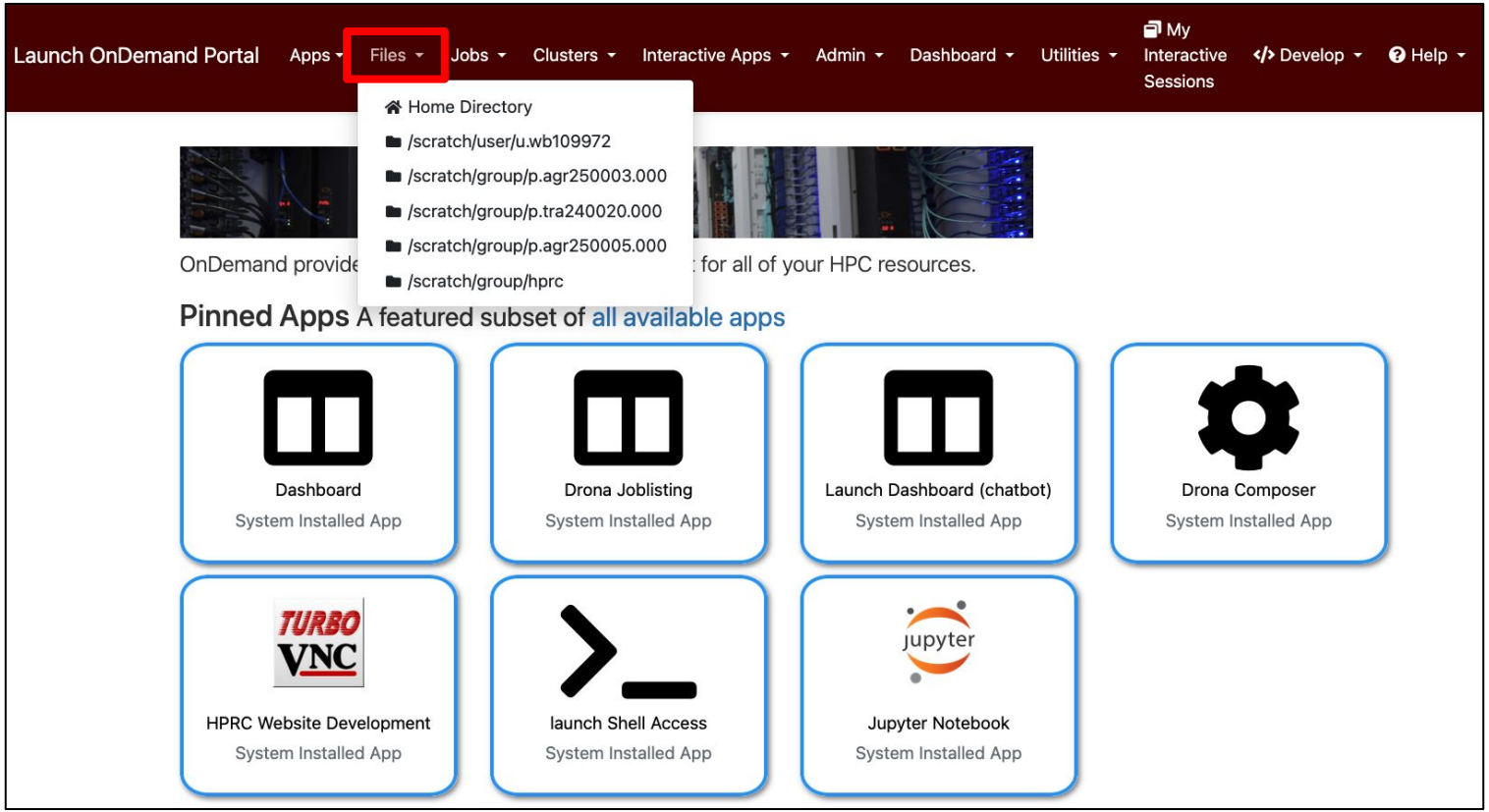




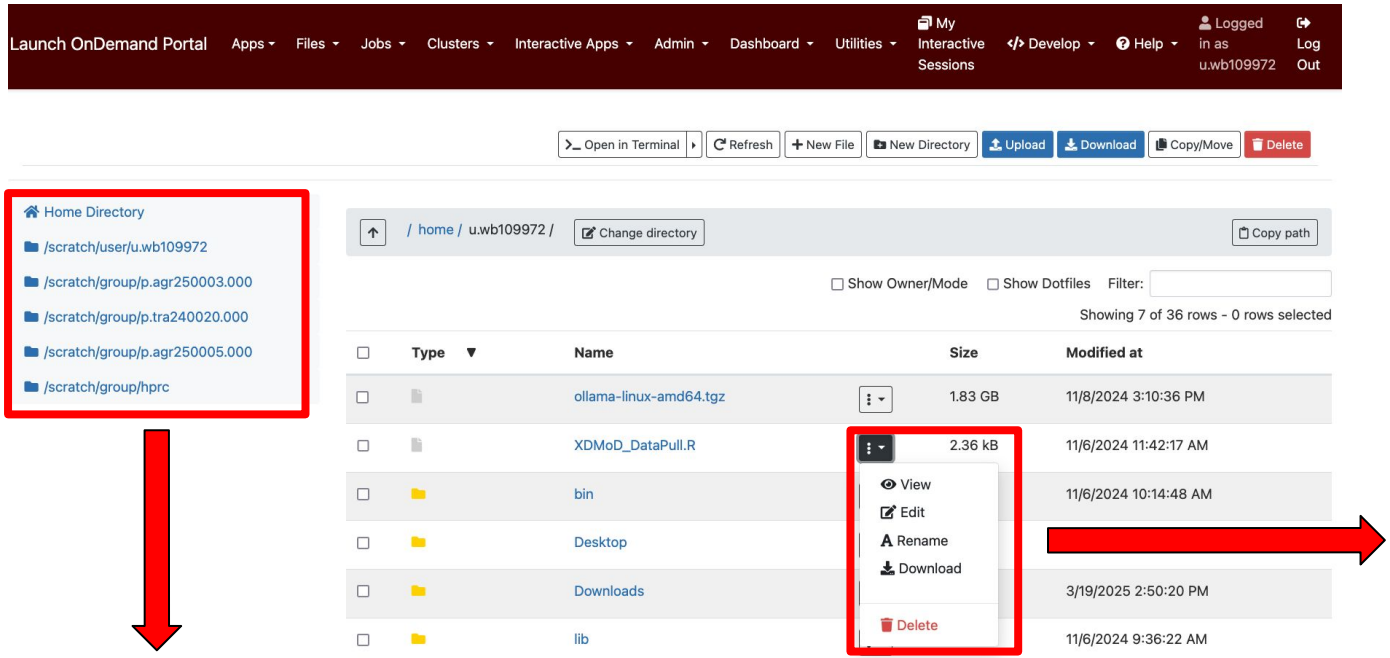
Launch Open OnDemand Portal



Launch Open OnDemand Portal: File Browser



Launch Open OnDemand Portal: File Browser



This drop down menu provides access to edit, rename, or download files.

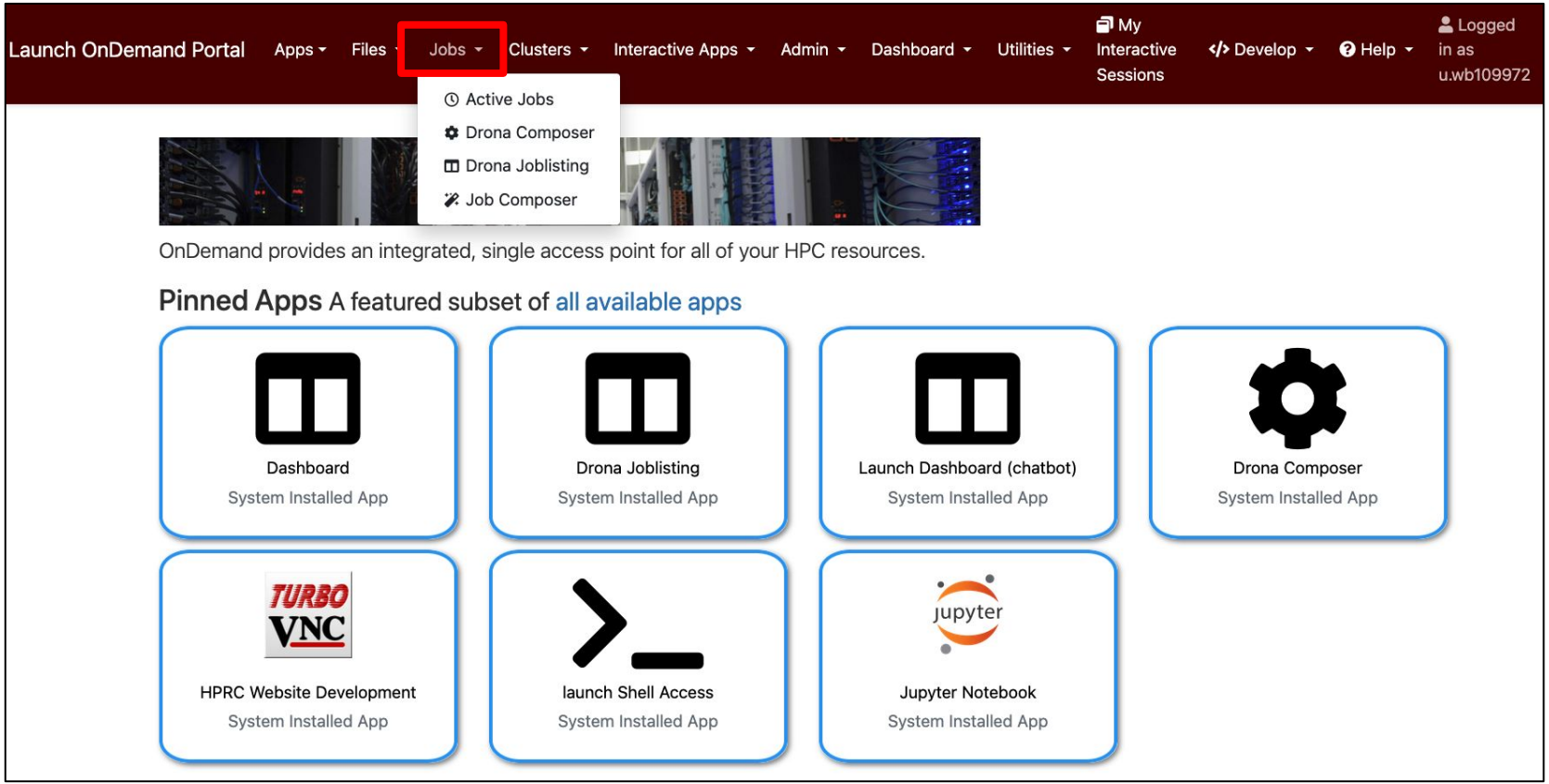
Navigate to home or scratch space here.

Launch Open OnDemand Portal: File Browser

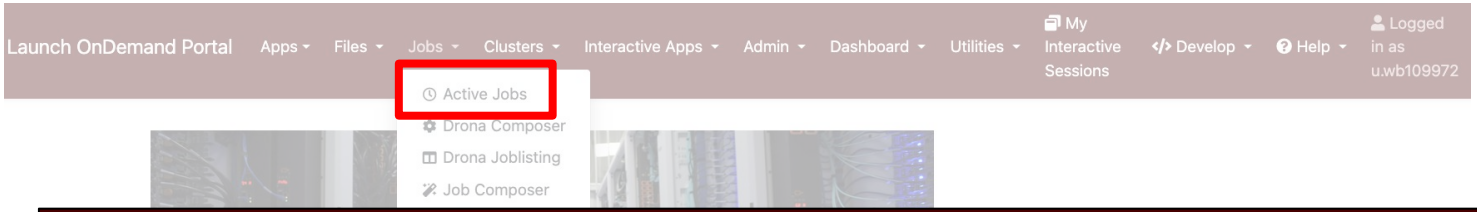
Save /home/u.wb109972/XDMoD_DataPull.R | Key Bindings: Default | Font Size: 12px | Mode: Text | Theme: Solarized Light

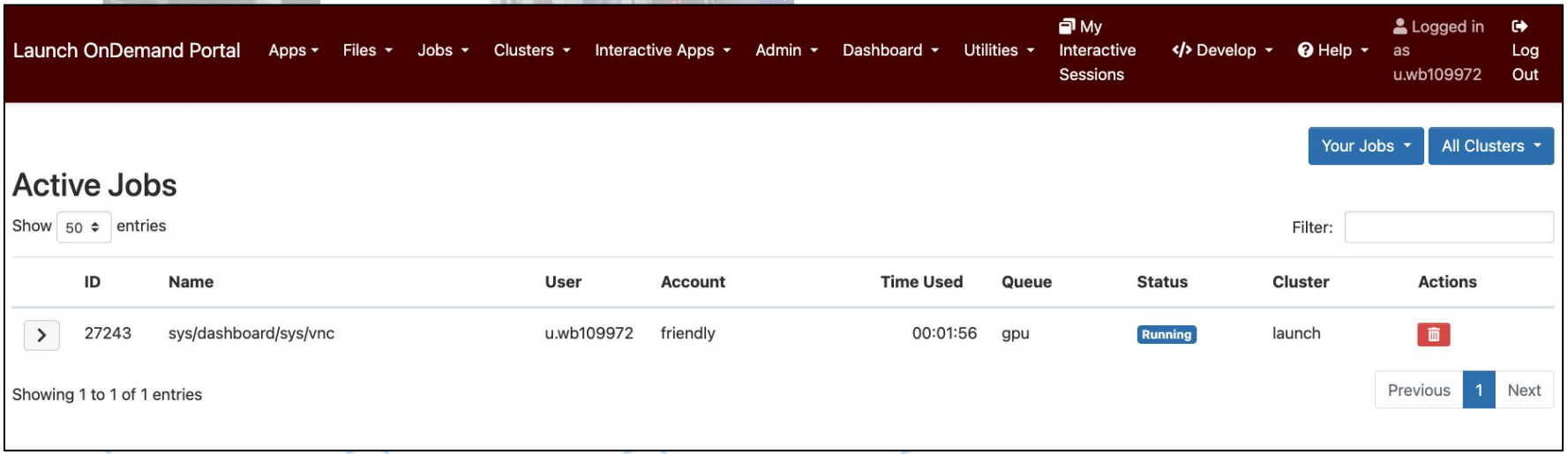
```
######## Before running:
# Get api key from XDMoD
# add api key to .Renviron
# XDMOD_API_TOKEN=<my secret xdmod api token>

# Load the required packages and set up the environment
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(plotly))
suppressPackageStartupMessages(library(reticulate))
use_condaenv("xdmod-notebooks")
suppressPackageStartupMessages(library(rxdmod))
suppressPackageStartupMessages(library(ggthemes))

# Get XDMoD connection
dw <- xdmod_get_datawarehouse('https://xdmod.access-ci.org')

# Get data
xdmod_df = xdmod_get_raw_data(dw,
            duration = c('2024-10-01', '2024-10-31'),
            realm = 'Jobs',
            filter=list(Resource=c("Texas A&M U FASTER"))
            )


# Pie Chart of Number of Users by Parent Science
unique_user = xdmod_df[!duplicated(xdmod_df$User),]

to_pie = as.data.frame(table(unique_user$`Parent Science`))
colnames(to_pie)[1] = paste("Parent Science")

ggplot(to_pie, aes(x="", y=Freq, fill=`Parent Science`)) +
  geom_bar(stat="identity", width=1, color = "white") +
  coord_polar("y", start=0) +
  scale_fill_manual(values=c("#8B1E3F", "#3C153B", "#89BD9E",
                             "#F0C987", "#DB4C40", "#2081C3",
                             "#60E1E0", "#B88E8D", "#34435E")) +
  theme_void()

```

Launch Open OnDemand Portal: Jobs

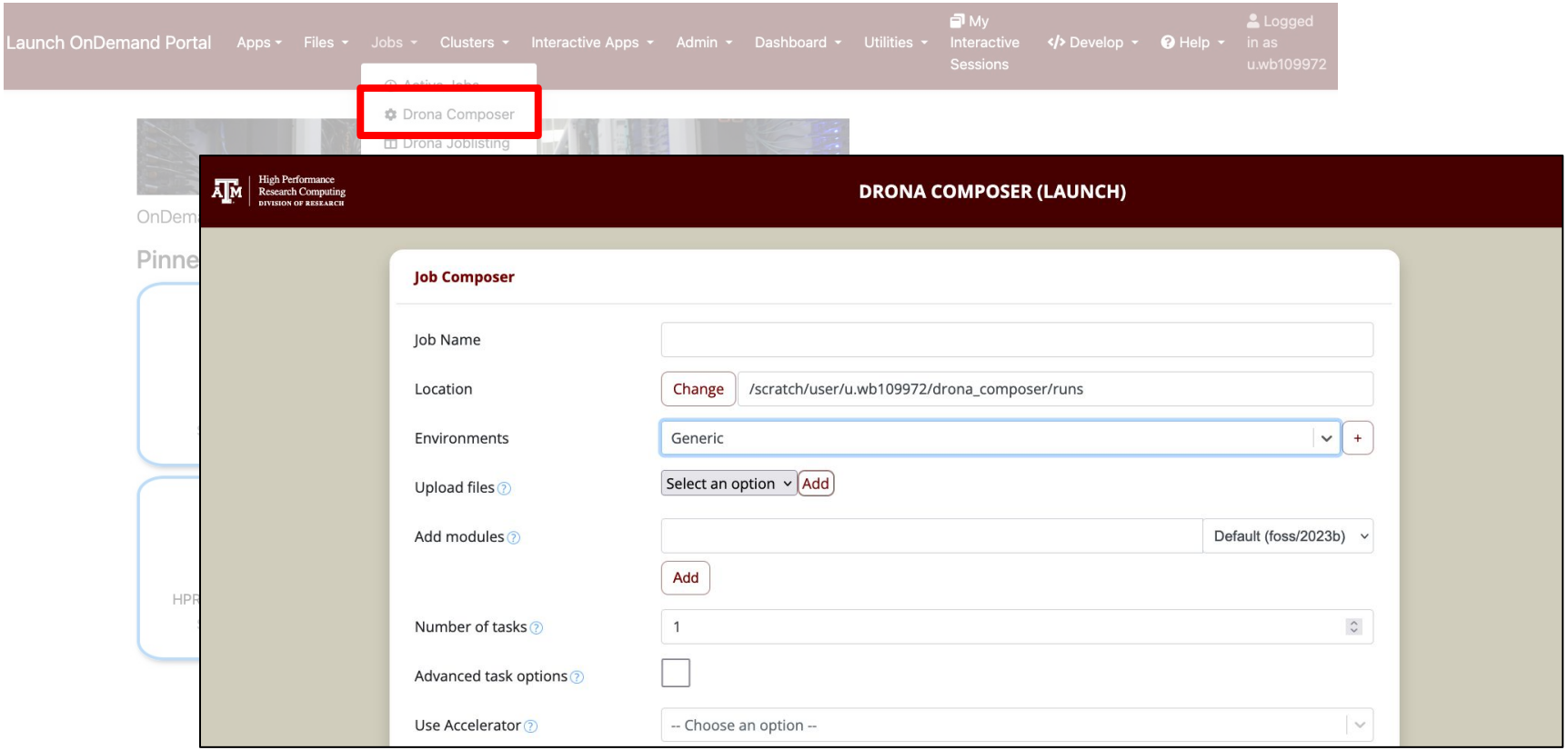


Launch Open OnDemand Portal: Jobs

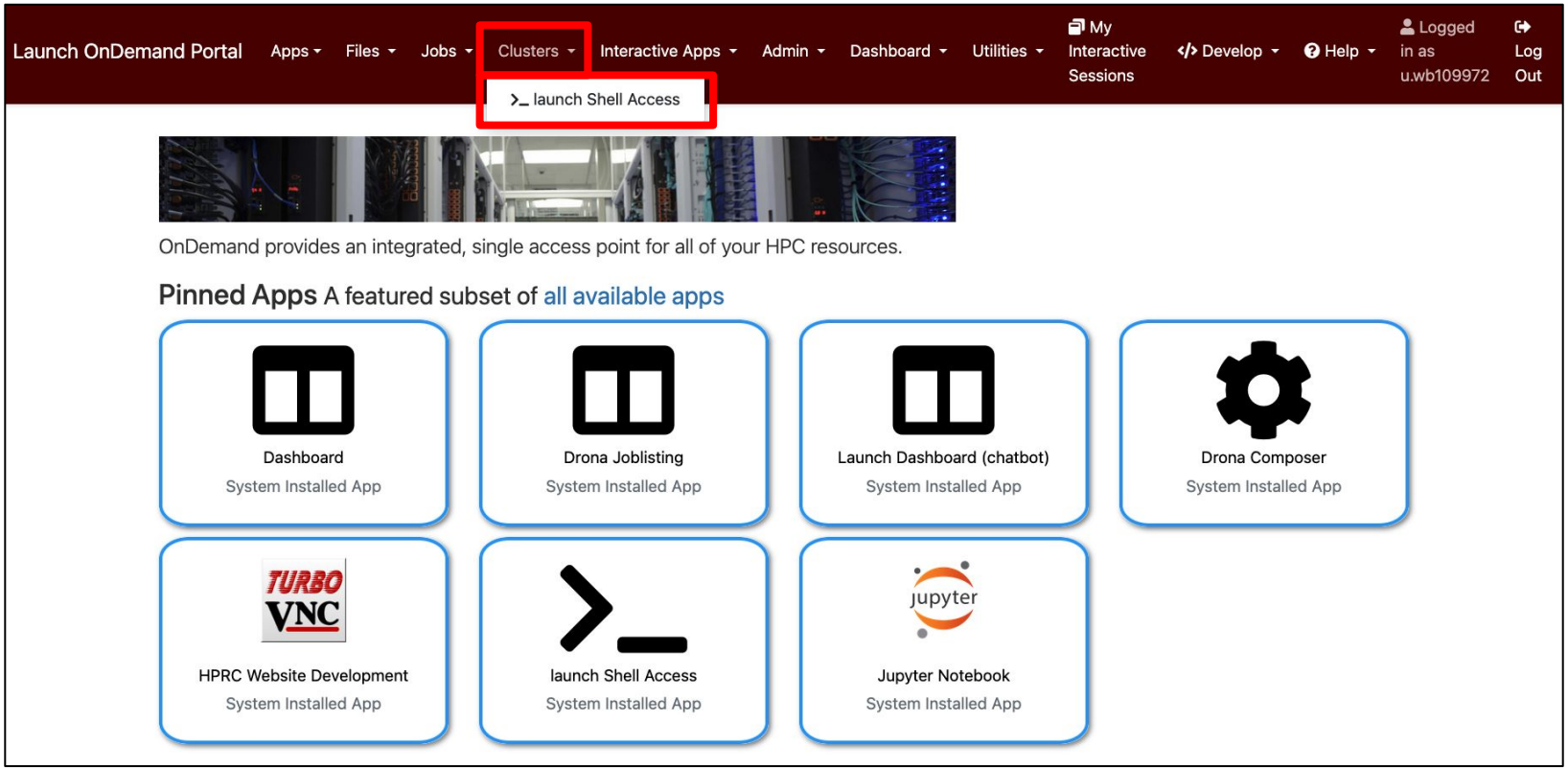




Launch Open OnDemand Portal: Jobs



Launch Open OnDemand Portal: Shell Access



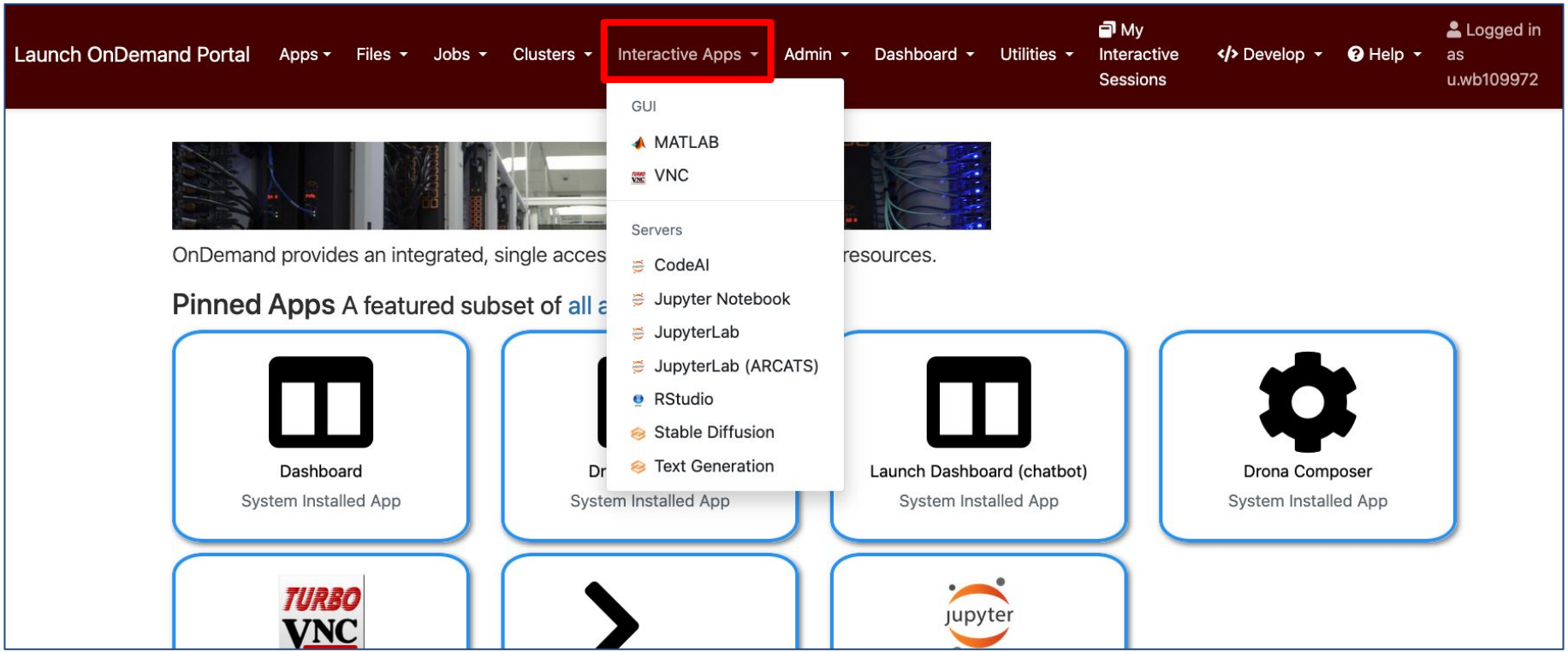
Launch Open OnDemand Portal: Shell Access

Access the training materials

Run the following commands in the Launch terminal:

```
# Navigate to your scratch directory
cd $SCRATCH

# Copy the materials
cp -r /scratch/training/arcats .
```

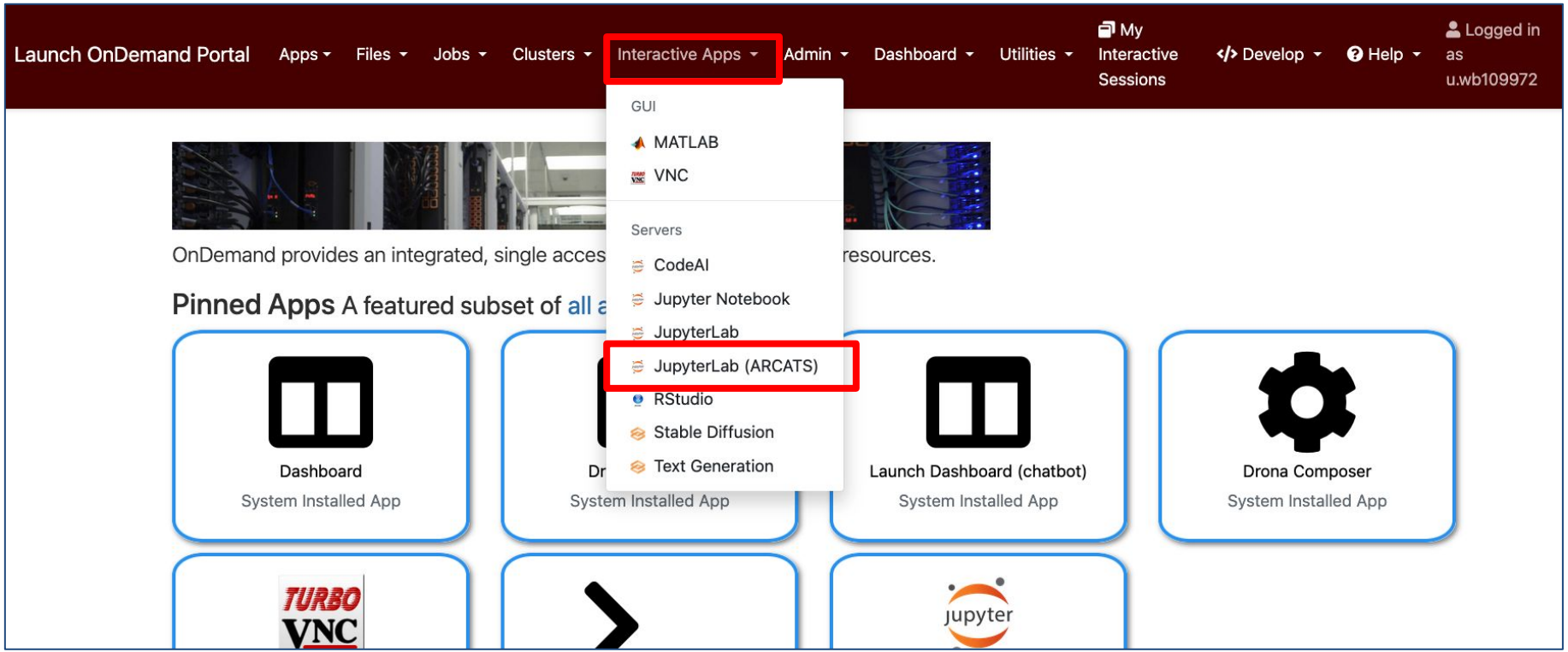
Launch Open OnDemand Portal: Interactive Apps



JupyterLab

- Web-based IDE (Interactive Development Environment)
- Can be used for Python, Julia, and R
- Integrates text and code blocks
- Great for interactive coding and developing workflows

Starting JupyterLab



Starting JupyterLab