Open OnDemand on Trillium

James Willis (SciNet)

April 15, 2026

Outline

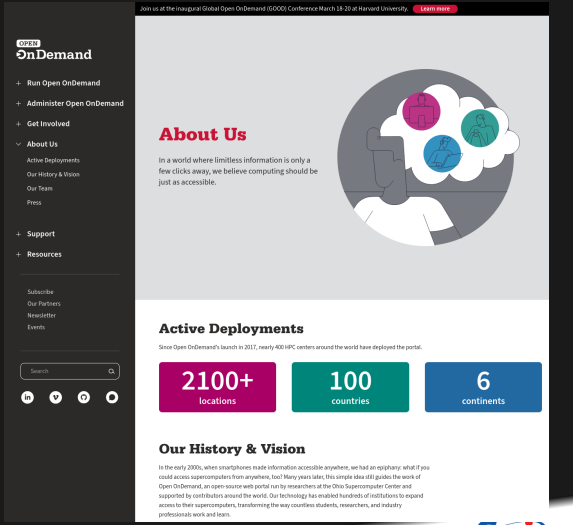
- Motivation
- What is Open OnDemand?
- Key Features
- SciNet's Open OnDemand Portal
- Open OnDemand Demo & Hands-on
- Jupyter Lab & Trillium Desktop Demo & Hands-on
- Job Submission and Monitoring with Open Composer
- Open Composer Demo & Hands-on

Motivation

- Terminal based interfaces can be very daunting for new users with little to no experience
- We need a way to make HPC more accessible to improve the learning curve
- Web-based interfaces are a good solution
- There are many available:
  - JupyterHub
  - RStudio
  - Galaxy
  - **Open OnDemand**

What is Open OnDemand?

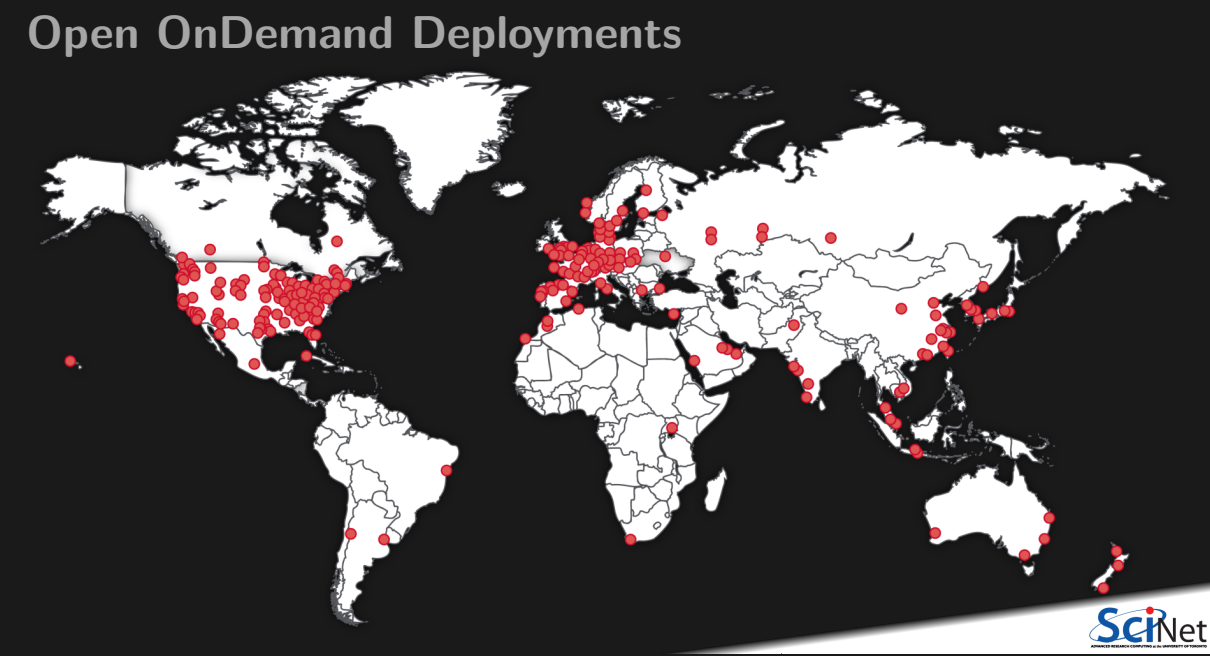
- Open OnDemand (OOD) is a web-based interface that provides access to HPC resources
- Open-source project developed by the Ohio Supercomputer Center (OSC) in 2017
- Funded by the National Science Foundation (NSF)
- Used worldwide across ~400 HPC centres, including:
  - Nibi - SHARCNET
  - Vulcan - University of Alberta, Amii
  - Grex - University of Manitoba
  - Bridges2 - University of Pittsburgh
  - Anvil - Purdue University
  - Expanse - San Diego
  - LUMI - IT Center for Science (Finland)



What is Open OnDemand?

- Designed to make HPC more accessible to users with no prior experience
- Built with Ruby and JavaScript
- Provides a graphical interface to computing resources as opposed to the command line
- Great tool for teaching and learning HPC skills incrementally
- Lowers the barrier to entry for new users who may be intimidated by the command line interface
- You can even access it from your mobile phone!

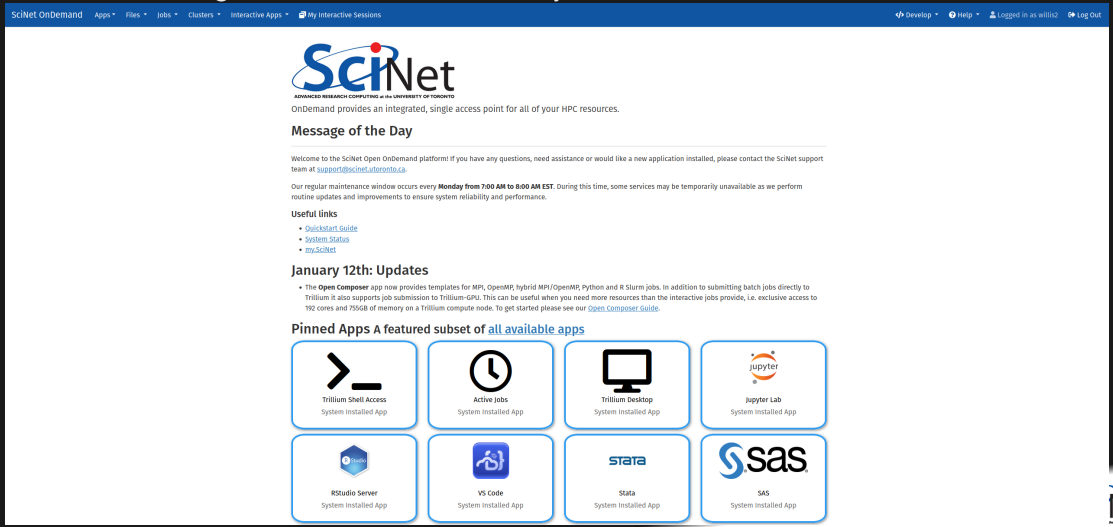
Open OnDemand Deployments



Key Features

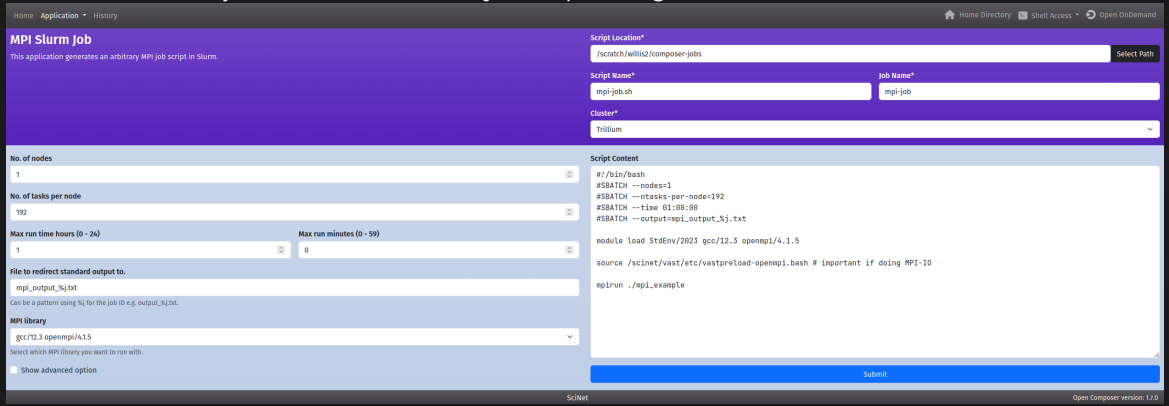
Web-Based Access

- Users can access HPC resources through a web browser without needing to install specialised software or configure SSH connections manually

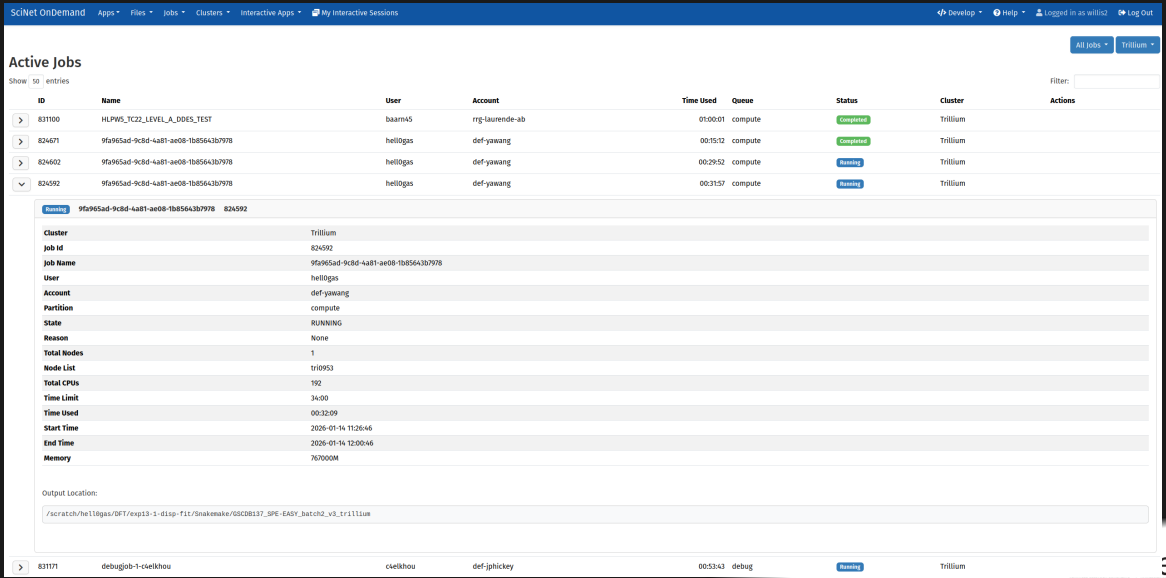


Job Management

- It provides an intuitive way to submit, monitor, and manage batch jobs using the Slurm scheduler. Users can easily create and customise job scripts using web forms

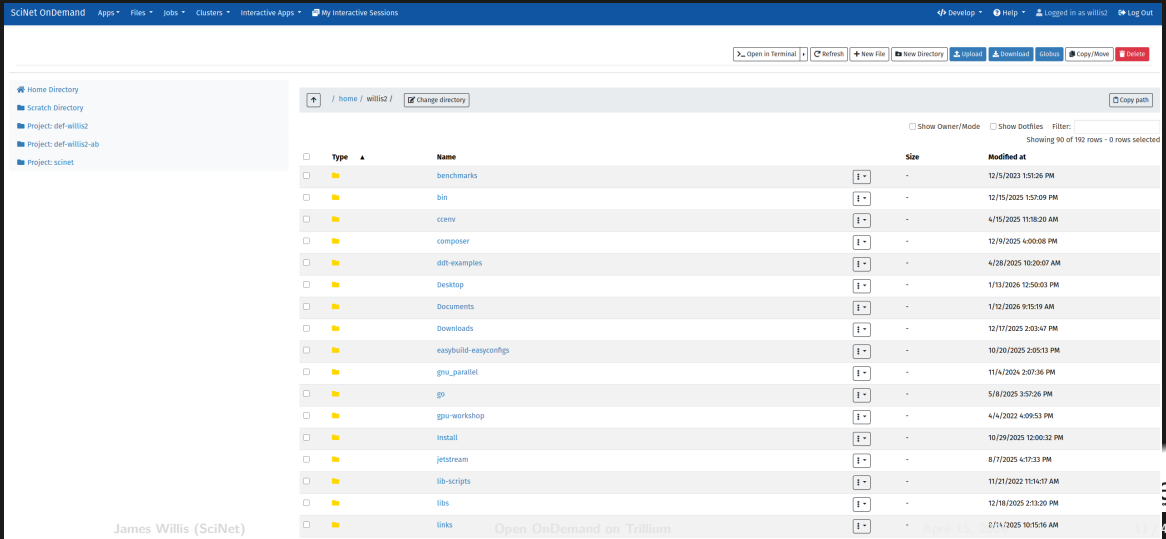


Job Monitoring



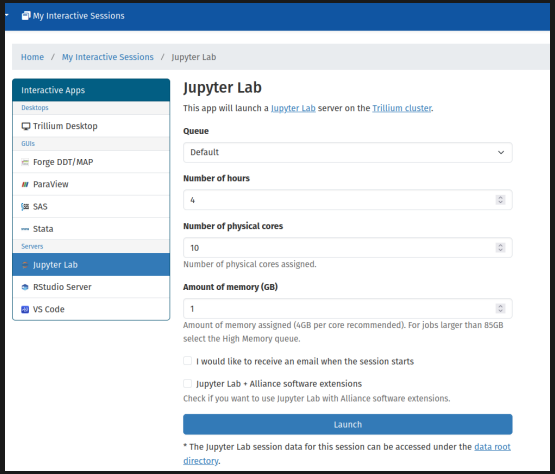
File Management

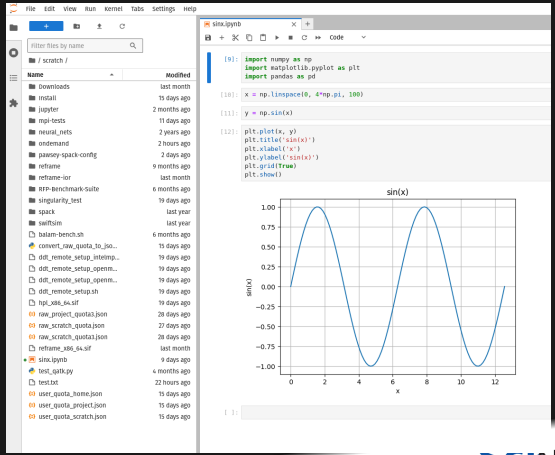
- Users can upload, download, edit, and manage files stored on the cluster through a built-in file browser



Interactive Applications

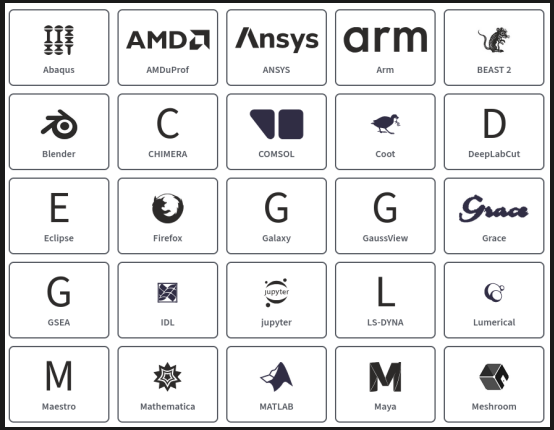
- It supports running interactive applications like Jupyter Notebooks, RStudio, VS Code or remote desktop sessions, making it useful for data analysis, visualisation, and interactive computational tasks

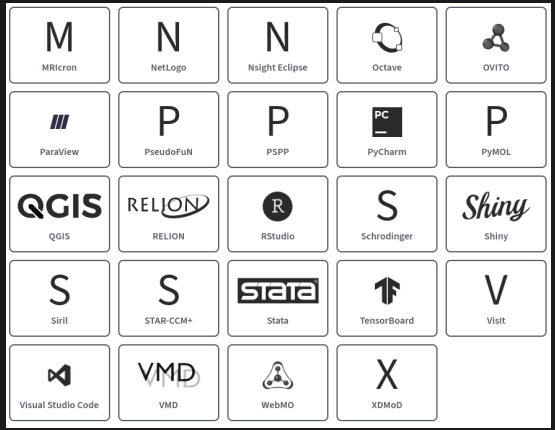




Interactive Applications

- Full list of supported apps:

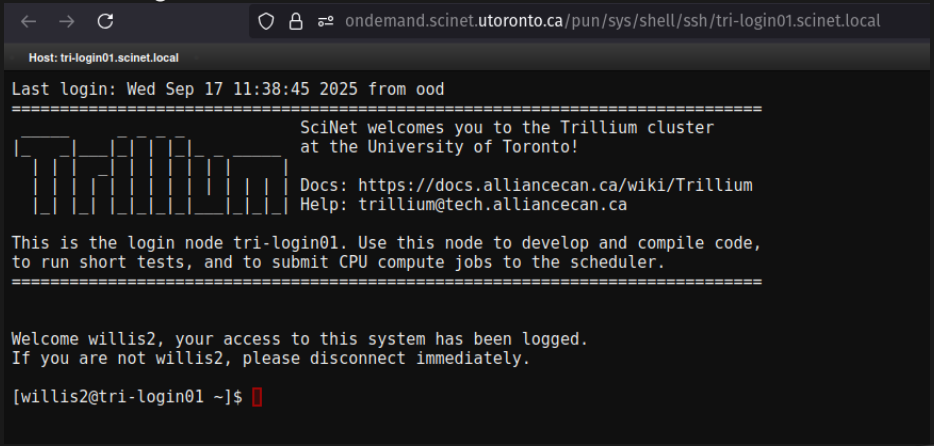




- Unsupported applications can be developed and added to the growing community of Open OnDemand apps

Terminal Access

- It provides a web-based shell to run command-line tasks directly from the browser, which can ease new users into using a terminal



SciNet's Open OnDemand Portal

- Live since Dec 2024
- As of April 2026 we have ~180 weekly active users
- Provides access to the Trillium HPC cluster
- Users can access the SciNet OOD portal at: **https://ondemand.scinet.utoronto.ca**
- You will need to have a Trillium account to log in
- Login credentials are the same as your Alliance account
- MFA is enabled, so you will be prompted to authenticate with Duo
- There is a quickstart guide on our wiki:
  https://docs.alliancecan.ca/wiki/Trillium_Open_OnDemand_Quickstart
- **Recently added a new OOD portal for the Teach cluster:**
  https://ondemand-teach.scinet.utoronto.ca

SciNet Deployment

- We have the following setup available on our portal:
  - File Browser
  - Job Submission
  - Job Monitoring
  - Module Browser
  - Storage Quotas
  - Terminal
  - Jupyter Lab
  - RStudio
  - VS Code
  - Remote Desktop
  - ParaView
  - MATLAB
  - Forge DDT/MAP
  - Stata (requires a license)
  - SAS (requires a license)
- All applications are provided by the software stack available on Trillium
- **Note: we are continuously adding new features/applications, if you require an application that is not supported please let us know and we can add it.**

Slurm Scheduling and Resources

- We have set up a new scheduler for interactive OOD jobs
- Specs:
  - 65 nodes (`ood0[01-64]` and `ood073`)
  - Each standard node has 40 cores and 180GB memory
  - `ood001`, `ood002` and `ood073` are large memory nodes with 40 cores and 1TB memory
  - Scheduled by-core
  - Can request how much memory you need in GB
  - 3 day job time limit
  - 20 cores max per job
  - Default queue allows up to 85GB memory per job
  - High memory queue allows up to 500GB memory per job and 1 job per user

Slurm Scheduling and Resources

- We have recently added a GPU node to the OOD cluster
- Specs:
  - 1 node (good001)
  - 4 x NVIDIA H100 80GB GPUs
  - 96 CPU cores
  - 810GB memory
  - MIG (Multi-Instance GPU) has been enabled with the **h100_1.10** profile
  - Provides access to 28 x GPU instances with 10GB memory each and 1/8 compute resources of a full NVDIA H100 GPU
  - Each job always gives 27GB of CPU memory and 3 cores per GPU instance

Open OnDemand Demo

Demos

- Dashboard Overview
  - Apps
  - Files
  - Jobs
  - Clusters
  - Interactive Apps
  - Interactive Sessions
  - Help
- Front page
  - Updates/Announcements
  - Recently used apps
  - Active sessions
  - Documentation links
  - App shortcuts
- File Browser
  - Upload/download files
  - Edit files
  - Manage files/folders
  - Globus integration
- Terminal
  - Open terminal session
  - Basic commands
  - Cluster status
- Module browser
- Storgage quotas

Dashboard Hands-on

- Access the SciNet OOD portal at: **https://ondemand.scinet.utoronto.ca** using your Alliance credentials
- Explore the **Dashboard** and familiarise yourself with the interface
- Use the **File Browser** to upload a file, create a directory and edit a text file
- Open a **Terminal** session and run some basic commands (e.g. `ls`, `cd`, `pwd`, `htop`)
- Look at the **Module Browser** to see what software is available
- Check your **Storage Quotas**

Jupyter Lab & Desktop Demos

Jupyter Hands-on

- Launch the **Jupyter Lab** app and create a new notebook
- Plot a simple graph using Matplotlib (e.g. a sine wave):

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 4*np.pi, 100)
y = np.sin(x)
plt.title('Sine Wave')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.plot(x, y)
```

Trillium Desktop Hands-on

- Use the **Trillium Desktop** app to run Octave and perform a simple calculation:
  - Launch the Trillium Desktop app
  - Open a terminal within the desktop session
  - Load the Octave module (Hint: Use the module browser to find the correct command)
  - Run the Octave GUI by typing `octave --gui` in the terminal
  - Plot a simple graph using Octave (e.g. a cosine wave):

```
x = -10:0.1:10;
plot(x, cos(x));
title('Cosine Wave');
xlabel('x');
ylabel('cos(x)');
```

- Load the Octave module: `module load StdEnv/2023 gcc/12.3 openmpi/4.1.5 octave/7.2.0`

Job Submission and Monitoring

Slurm Job Submission with Open Composer

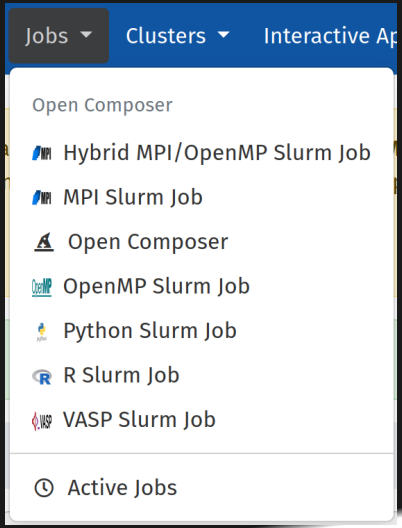
- For some tasks you may require more compute resources than what is available through the interactive apps
- Running a job on a Trillium compute node gives you exlusive access to 192 cores and 755GB of memory
- To submit a job to the compute nodes, you need to use the Slurm scheduler
- Slurm can be intimidating for new users, especially when it comes to writing job scripts
- Open OnDemand provides an easy to understand interface to Slurm called **Open Composer**

Open Composer Overview

- Open Composer provides a graphical interface to create and submit Slurm job scripts
- It allows you to specify job parameters, resource requests, and other options through a web form
- It also provides templates for common job types to help you get started quickly such as Python, R, MPI and OpenMP
- You can also track the status of your submitted jobs and view job history via the History page

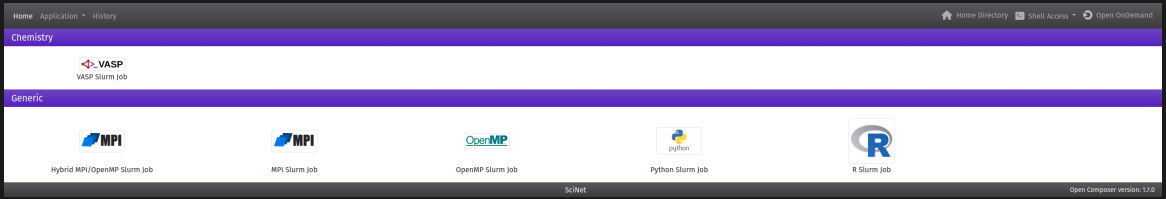
Accessing Open Composer

- You can access Open Composer from the dashboard by clicking on the **Jobs** tab in the navigation bar and selecting either **Open Composer** or one of the job templates



Dashboard

- The dashboard provides access to:
  - Available job templates
  - Job history page
  - Terminal access to Trillium and Trillium-GPU login nodes
  - Link back to the main Open OnDemand dashboard



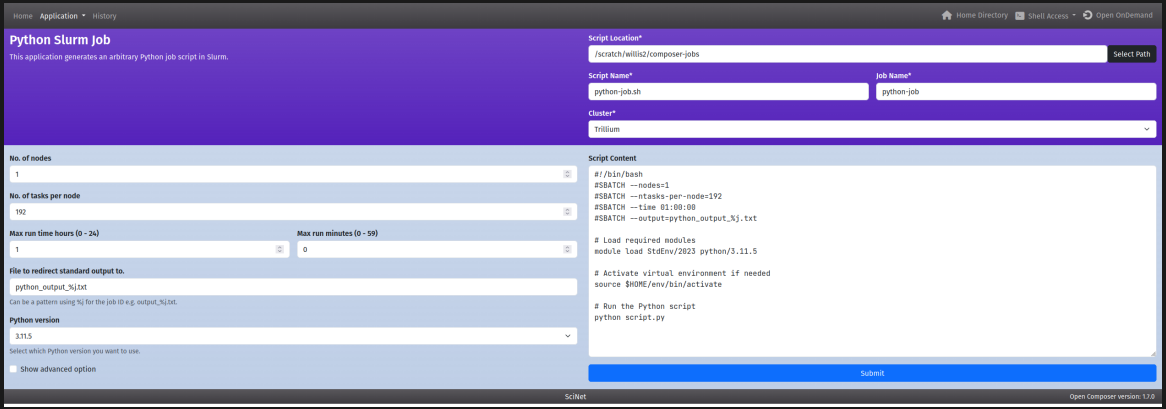
Job Submission

- After clicking on a job template, you will be taken to the job submission page where you can specify the details of your job
- This page is split between the job parameters on the left and the job script itself on the right
- The parameters form allows you to specify:
  - Number of nodes
  - Number of tasks per node
  - Job time limit
  - Job standard output file
  - Email notifications
  - Extra parameters specific to the job type (e.g. Python/R version, MPI library, etc.)

Job Submission

- There are also extra fields at the top of the page that affect how your job is submitted and where:
  - **Script Location**: specifies the directory where the job script will be saved and where your job will be run from
  - **Script Name**: specifies the name of the job script file
  - **Job Name**: specifies the name of the job that will appear in the job queue
  - **Cluster**: allows you to change which cluster to submit your job to, e.g. Trillium (default) or Trillium-GPU
- The job script is automatically updated based on the parameters you specify
- The script can also be edited manually if needed

Job Submission



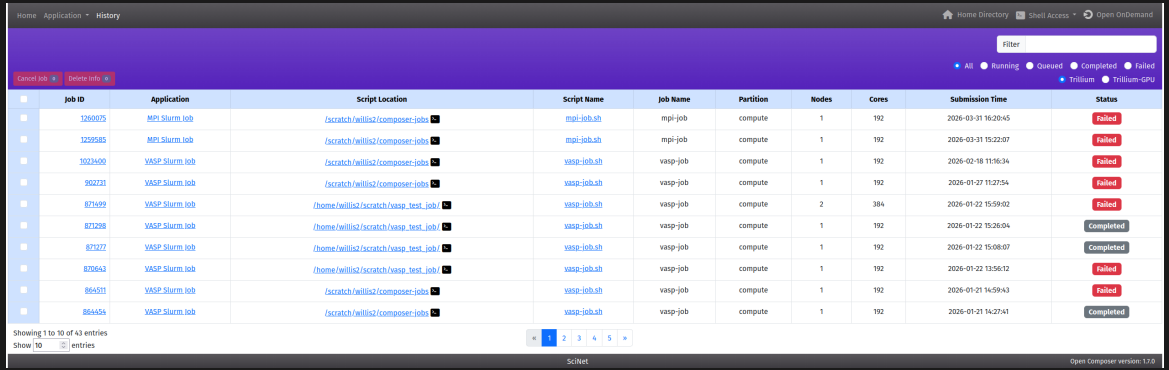
Job Monitoring

- After submitting a job, you can monitor its status and view details about it through the **History** page
- The History page provides a list of all your submitted jobs along with their status, job ID and other related job information
- You can filter the jobs by using the **Filter** text box at the top right or by using the checkboxes below

Job Monitoring

- Clicking on different column fields give further functionality:
  - **Job ID**: opens the job in **my.SciNet** which displays performance statistics (Note: my.SciNet may show 'Not found or not permitted' if the job hasn't started yet or was cancelled)
  - **Application**: opens the job script editor of the template you used
  - **Script Location**: opens a file browser window at the location of the job script. Clicking on the small terminal icon will open a terminal in the job script location
  - **Script Name**: displays the job script that was submitted to the scheduler
- To resubmit or modify a previously run job click on the job script under the **Script Name** column and click **Load Parameters**

Job Monitoring



Open Composer Demo

Open Composer Hands-on

- Use **Open Composer** to create and submit a simple Slurm job script:
  - Launch Open Composer from the dashboard
  - Choose a job template (e.g. MPI)
  - Fill out the job parameters (e.g. 1 node, 10 minute time limit)
  - Fill out the job script (e.g. a job that runs `mpirun hostname` to print the node name)
  - Submit the job
  - Monitor the job status through the History page and view the output once it completes

Open Composer Hands-on

- Example job script:

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=192
#SBATCH --time 01:00:00
#SBATCH --output=mpi_output_%j.txt

module load StdEnv/2023 gcc/12.3 openmpi/4.1.5

source /scinet/vast/etc/vastpreload-openmpi.bash # important if doing MPI-IO

mpirun hostname
```

Feedback

- Please give Open OnDemand a try and let us know what you think!
- We welcome feedback and are open to suggestions for improvement
- Let us know if you'd like to see any specific applications or features added

Summary

- We introduced Open OnDemand as an alternative to terminal-based interfaces for accessing HPC resources:
  - Graphical web-based interface
  - No need to install software locally
  - Great for new users and teaching HPC skills
  - No setting up SSH connections
- Discussed its key features and how it can make HPC more accessible to new users
- Showed the SciNet Open OnDemand portal and the applications available
- Demonstrated how to use Open OnDemand for file management, interactive applications, and job submission/monitoring
- Please email any questions to: **support@scinet.utoronto.ca**