

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)

The screenshot shows the Open OnDemand website interface. On the left is a dark navigation sidebar with the following menu items: 'OPEN OnDemand', '+ Run Open OnDemand', '+ Administer Open OnDemand', '+ Get Involved', '▼ About Us' (with sub-items: 'Active Deployments', 'Our History & Vision', 'Our Team', 'Press'), '+ Support', '+ Resources', 'Subscribe' (with sub-items: 'Our Partners', 'Newsletter', 'Events'), a search bar, and social media icons for LinkedIn, Facebook, Twitter, and YouTube.

The main content area features a header with the text 'Join us at the Inaugural Global Open OnDemand (OOD) Conference March 18-20 at Harvard University.' and a 'Learn more' link. Below this is the 'About Us' section, which includes the text: 'In a world where limitless information is only a few clicks away, we believe computing should be just as accessible.' and an illustration of a person's head with a thought bubble containing icons for a person, a globe, and a person with a laptop.

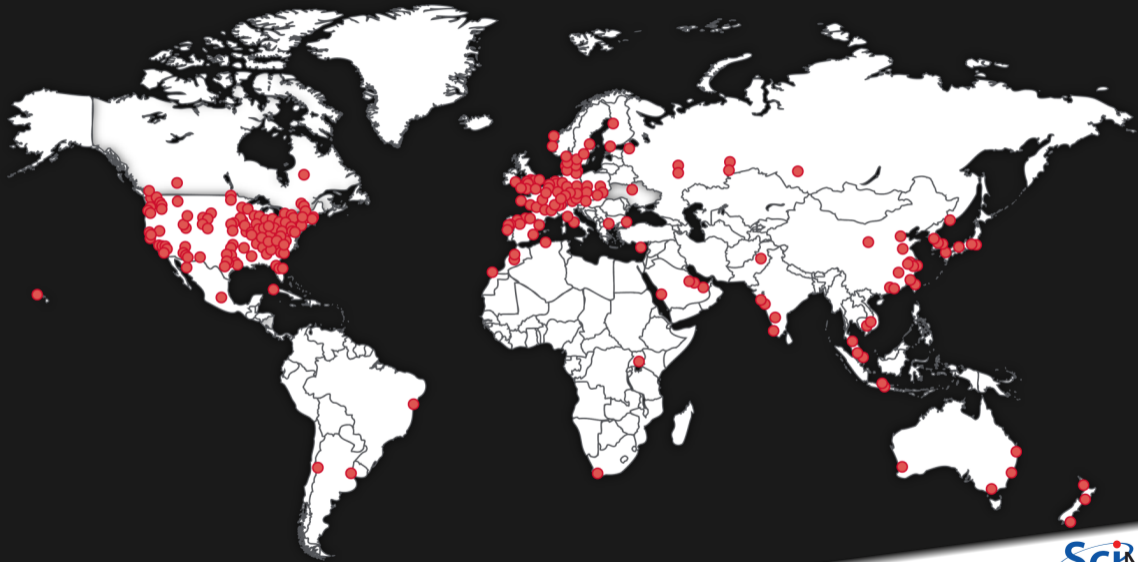
The 'Active Deployments' section states: 'Since Open OnDemand's launch in 2017, nearly 400 HPC centers around the world have deployed the portal.' It features three statistics: '2100+ locations' (in a pink box), '100 countries' (in a teal box), and '6 continents' (in a blue box).

The 'Our History & Vision' section begins with the text: 'In the early 2000s, when smartphones made information accessible anywhere, we had an epiphany: what if you could access supercomputers from anywhere, too? Many years later, this simple idea still guides the work of Open OnDemand, an open-source web portal run by researchers at the Ohio Supercomputer Center and supported by contributors around the world. Our technology has enabled hundreds of institutions to expand access to their supercomputers, transforming the way countless students, researchers, and industry professionals work and learn.'

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

SciNet OnDemand Apps Files Jobs Clusters Interactive Apps My Interactive Sessions

Develop Help Logged in as willis2 Log Out

SciNet

ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

Welcome to the SciNet Open OnDemand platform! If you have any questions, need assistance or would like a new application installed, please contact the SciNet support team at support@scinet.utoronto.ca.

Our regular maintenance window occurs every **Monday from 7:00 AM to 8:00 AM EST**. During this time, some services may be temporarily unavailable as we perform routine updates and improvements to ensure system reliability and performance.

Useful links

- [Quickstart Guide](#)
- [System Status](#)
- [my.SciNet](#)

January 12th: Updates

- The **Open Composer** app now provides templates for MPI, OpenMP, hybrid MPI/OpenMP, Python and R Slurm jobs. In addition to submitting batch jobs directly to Trillium it also supports job submission to Trillium-GPU. This can be useful when you need more resources than the interactive jobs provide, i.e. exclusive access to 792 cores and 755GB of memory on a Trillium compute node. To get started please see our [Open Composer Guide](#).

Pinned Apps A featured subset of [all available apps](#)

 Trillium Shell Access System Installed App	 Active Jobs System Installed App	 Trillium Desktop System Installed App	 Jupyter Lab System Installed App
 RStudio Server System Installed App	 VS Code System Installed App	 Stata System Installed App	 SAS System Installed App

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

Home Application History Home Directory Shell Access Open OnDemand

MPI Slurm job

This application generates an arbitrary MPI job script in Slurm.

Script Location*
/scratch/willis2/composer-jobs Select Path

Script Name* mpi-job.sh **Job Name*** mpi-job

Cluster*
Trillium

No. of nodes
1

No. of tasks per node
192

Max run time hours (0 - 24) 1 **Max run minutes (0 - 59)** 0

file to redirect standard output to.
mpi_output_%j.txt
Can be a pattern using %j for the job ID e.g. output_%j.txt.

MPI library
gcc/12.3 openmpi/4.1.5
Select which MPI library you want to run with.

Show advanced option

Script Content

```
#!/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 ./mpi_example
```

Submit

SciNet Open Composer version: 1.7.0

Job Monitoring

Active Jobs

All Jobs Trillium

Show 50 entries

Filter:

ID	Name	User	Account	Time Used	Queue	Status	Cluster	Actions
>	831100	HLPWS_TC22_LEVEL_A_DDES_TEST	baarn45	rrg-laurende-ab	01:00:01	compute	Completed	Trillium
>	824671	9fa965ad-9c8d-4a81-ae08-1b85643b7978	hellogas	def-yawang	00:15:12	compute	Completed	Trillium
>	824602	9fa965ad-9c8d-4a81-ae08-1b85643b7978	hellogas	def-yawang	00:29:52	compute	Running	Trillium
▼	824592	9fa965ad-9c8d-4a81-ae08-1b85643b7978	hellogas	def-yawang	00:31:57	compute	Running	Trillium

Running 9fa965ad-9c8d-4a81-ae08-1b85643b7978 824592

Cluster	Trillium
Job Id	824592
Job Name	9fa965ad-9c8d-4a81-ae08-1b85643b7978
User	hellogas
Account	def-yawang
Partition	compute
State	RUNNING
Reason	None
Total Nodes	1
Node List	tri0953
Total CPUs	192
Time Limit	34:00
Time Used	00:32:09
Start Time	2026-01-14 11:26:46
End Time	2026-01-14 12:00:46
Memory	767000M

Output Location:

/scratch/hellogas/DFT/exp13-1-disp-f1t/Snakeake/GSCDB137_SPE-EASY_batch2_v3_trillium

>	831171	debugjob-1-c4elkhou	c4elkhou	def-jphickey	00:53:43	debug	Running	Trillium
---	--------	---------------------	----------	--------------	----------	-------	---------	----------

File Management

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

The screenshot displays the SciNet OnDemand file management interface. The top navigation bar includes 'SciNet OnDemand', 'Apps', 'Files', 'Jobs', 'Clusters', 'Interactive Apps', and 'My Interactive Sessions'. On the right, it shows 'Develop', 'Help', 'Logged in as willis2', and 'Log Out'. Below the navigation bar, there are several action buttons: 'Open in Terminal', 'Refresh', 'New File', 'New Directory', 'Upload', 'Download', 'Globe', 'Copy/Move', and 'Delete'. The main content area shows a file browser view for the directory '/ home / willis2 /'. A sidebar on the left lists the 'Home Directory' and several project directories: 'Scratch Directory', 'Project: def-willis2', 'Project: def-willis2-ab', and 'Project: scinet'. The main area contains a table of files and folders with columns for 'Type', 'Name', 'Size', and 'Modified at'. The table lists various directories such as 'benchmarks', 'bin', 'ccenv', 'composer', 'ddt-examples', 'Desktop', 'Documents', 'Downloads', 'easybuild-easyconfigs', 'gnu_parallel', 'go', 'gpu-workshop', 'Install', 'jetstream', 'lib-scripts', 'libs', and 'links'. Each row includes a checkbox, a type icon, a name, a size, and a modified date. At the bottom, there is a footer with 'James Willis (SciNet)', 'Open OnDemand on Trillium', and a date 'April 15, 2025'.

SciNet OnDemand Apps Files Jobs Clusters Interactive Apps My Interactive Sessions Develop Help Logged in as willis2 Log Out

Open in Terminal Refresh New File New Directory Upload Download Globe Copy/Move Delete

Home Directory
Scratch Directory
Project: def-willis2
Project: def-willis2-ab
Project: scinet

/ home / willis2 / Change directory Copy path

Show Owner/Mode Show Dotfiles Filter: Showing 90 of 192 rows - 0 rows selected

Type	Name	Size	Modified at
Folder	benchmarks	-	12/5/2023 1:51:26 PM
Folder	bin	-	12/15/2025 1:57:09 PM
Folder	ccenv	-	4/15/2025 11:18:20 AM
Folder	composer	-	12/9/2025 4:00:08 PM
Folder	ddt-examples	-	4/28/2025 10:20:07 AM
Folder	Desktop	-	1/13/2026 12:50:03 PM
Folder	Documents	-	1/12/2026 9:15:19 AM
Folder	Downloads	-	12/17/2025 2:03:47 PM
Folder	easybuild-easyconfigs	-	10/20/2025 2:05:13 PM
Folder	gnu_parallel	-	11/4/2024 2:07:36 PM
Folder	go	-	5/8/2025 3:57:26 PM
Folder	gpu-workshop	-	4/4/2022 4:09:53 PM
Folder	Install	-	10/29/2025 12:00:32 PM
Folder	jetstream	-	8/7/2025 4:17:33 PM
Folder	lib-scripts	-	11/21/2022 11:14:17 AM
Folder	libs	-	12/18/2025 2:13:20 PM
Folder	links	-	April 15, 2025 10:15:16 AM

James Willis (SciNet) Open OnDemand on Trillium April 15, 2025 10:15:16 AM

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

My Interactive Sessions

Home / My Interactive Sessions / Jupyter Lab

Interactive Apps

- Desktops
 - Trillium Desktop
- GUIs
 - Forge DDT/MAP
 - ParaView
 - SAS
 - Stata
- Servers
 - Jupyter Lab**
 - RStudio Server
 - VS Code

Jupyter Lab

This app will launch a [Jupyter Lab](#) server on the [Trillium cluster](#).

Queue

Default

Number of hours

4

Number of physical cores

10

Number of physical cores assigned.

Amount of memory (GB)

1

Amount of memory assigned (4GB per core recommended). For jobs larger than 85GB select the High Memory queue.

I would like to receive an email when the session starts

Jupyter Lab + Alliance software extensions

Check if you want to use Jupyter Lab with Alliance software extensions.

Launch

* The Jupyter Lab session data for this session can be accessed under the [data root directory](#).

File Edit View Run Kernel Tabs Settings Help

filter files by name

/ scratch /

Name	Modified
Downloads	last month
install	15 days ago
jupyter	2 months ago
mpi-tests	11 days ago
neural_nets	2 years ago
ondemand	2 hours ago
paersey-spac-config	2 days ago
reframe	9 months ago
reframe-ior	last month
RRP-benchmark-suite	6 months ago
singularity_test	19 days ago
spack	last year
swiftsim	last year
balam-bench.sh	6 months ago
convert_raw_quota_to_iso...	15 days ago
did_remote_setup_intelimp...	19 days ago
did_remote_setup_openim...	19 days ago
did_remote_setup_sh	19 days ago
hpi_x86_64.sif	19 days ago
raw_project_quota3.json	28 days ago
raw_scratch_quota3.json	27 days ago
raw_scratch_quota3.json	28 days ago
reframe_x86_64.sif	last month
sinx.ipynb	9 days ago
test_gatk.py	4 months ago
test.txt	22 hours ago
user_quota_home.json	15 days ago
user_quota_project.json	15 days ago
user_quota_scratch.json	15 days ago

```
[9]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

[10]: x = np.linspace(0, 4*np.pi, 100)

[11]: y = np.sin(x)

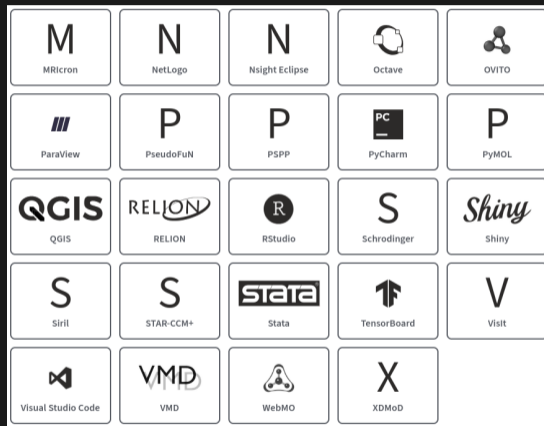
[12]: plt.plot(x, y)
plt.title('sin(x)')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.grid(True)
plt.show()
```

sin(x)

[]

Interactive Applications

- Full list of supported apps:



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

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 × 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 × GPU instances with 10GB memory each and 1/8 compute resources of a full NVIDIA 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
- Storage 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

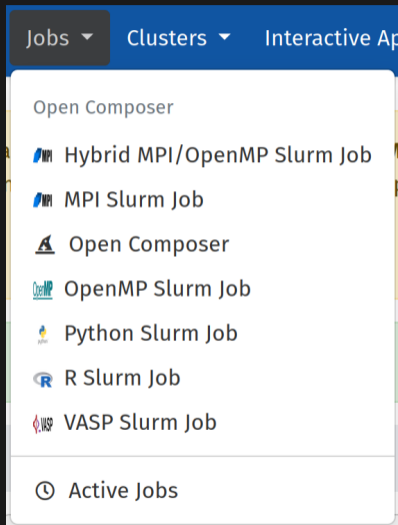
- For some tasks you may require more compute resources than what is available through the interactive apps
- Running a job on a Trillium compute node gives you exclusive access to 192 cores and 755GB of memory
- To submit a job to the compute nodes, you need to use the Slurm scheduler
- Slurm can be 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

The screenshot displays the Open OnDemand dashboard interface. At the top, there is a navigation bar with links for 'Home', 'Application', and 'History'. On the right side of the navigation bar, there are links for 'Home Directory', 'Shell Access', and 'Open OnDemand'. Below the navigation bar, the dashboard is divided into two main sections: 'Chemistry' and 'Generic'. The 'Chemistry' section features a single job template for 'VASP Slurm Job', represented by a red and white VASP logo. The 'Generic' section contains five job templates: 'Hybrid MPI/OpenMP Slurm Job' (MPI logo), 'MPI Slurm Job' (MPI logo), 'OpenMP Slurm Job' (OpenMP logo), 'Python Slurm Job' (Python logo), and 'R Slurm Job' (R logo). At the bottom of the dashboard, the text 'SciNet' is centered, and 'Open Composer version: 1.7.0' is displayed on the right.

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

Home Application History Home Directory Shell Access Open OnDemand

Python Slurm Job

This application generates an arbitrary Python job script in Slurm.

Script Location*
/scratch/willisz/composer-jobs Select Path

Script Name* python-job.sh **Job Name*** python-job

Cluster*
Trillium

No. of nodes
1

No. of tasks per node
192

Max run time hours (0 - 24) 1 **Max run minutes (0 - 59)** 0

File to redirect standard output to.
python_output_%j.txt
Can be a pattern using %j for the job ID e.g. output_%j.txt.

Python version
3.11.5
Select which Python version you want to use.

Show advanced option

Script Content

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

# Load required modules
module load StdEnv/2023 python/3.11.5

# Activate virtual environment if needed
source $HOME/env/bin/activate

# Run the Python script
python script.py
```

Submit

SciNet Open Composer version: 1.7.0

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

Home Application History Home Directory Shell Access Open OnDemand

Filter

All Running Queued Completed Failed
 Trillium Trillium-GPU

[Cancel Job](#) [Delete Info](#)

Job ID	Application	Script Location	Script Name	Job Name	Partition	Nodes	Cores	Submission Time	Status
1260075	MPI Slurm job	/scratch/willis2/composer-jobs	mpi-job.sh	mpi-job	compute	1	192	2026-03-31 16:20:45	Failed
1259585	MPI Slurm job	/scratch/willis2/composer-jobs	mpi-job.sh	mpi-job	compute	1	192	2026-03-31 15:22:07	Failed
1023400	YASP Slurm job	/scratch/willis2/composer-jobs	vasp-job.sh	vasp-job	compute	1	192	2026-02-18 11:16:34	Failed
902731	YASP Slurm job	/scratch/willis2/composer-jobs	vasp-job.sh	vasp-job	compute	1	192	2026-01-27 11:27:54	Failed
871499	YASP Slurm job	/home/willis2/scratch/vasp_test_job	vasp-job.sh	vasp-job	compute	2	384	2026-01-22 15:59:02	Failed
871298	YASP Slurm job	/home/willis2/scratch/vasp_test_job	vasp-job.sh	vasp-job	compute	1	192	2026-01-22 15:26:04	Completed
871277	YASP Slurm job	/home/willis2/scratch/vasp_test_job	vasp-job.sh	vasp-job	compute	1	192	2026-01-22 15:08:07	Completed
870643	YASP Slurm job	/home/willis2/scratch/vasp_test_job	vasp-job.sh	vasp-job	compute	1	192	2026-01-22 13:56:12	Failed
864511	YASP Slurm job	/scratch/willis2/composer-jobs	vasp-job.sh	vasp-job	compute	1	192	2026-01-21 14:59:43	Failed
864456	YASP Slurm job	/scratch/willis2/composer-jobs	vasp-job.sh	vasp-job	compute	1	192	2026-01-21 14:27:41	Completed

Showing 1 to 10 of 43 entries
Show entries

« 1 2 3 4 5 »

SciNet Open Composer version: 17.0

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**