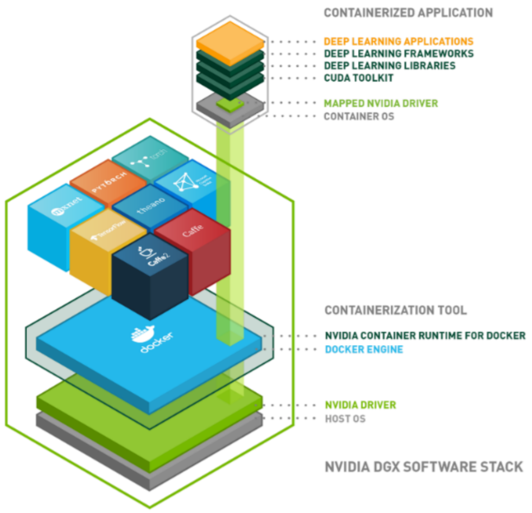
VISTA Lab

Évora University GPU-Cluster

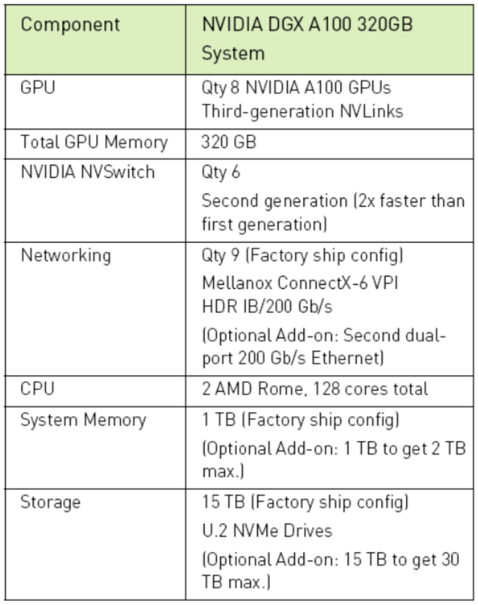
Instructions to launch Jobs

Author: Javier Lamar León
High-performance computer lab for Big Data and Machine Learning
jlamarleon@uevora.pt

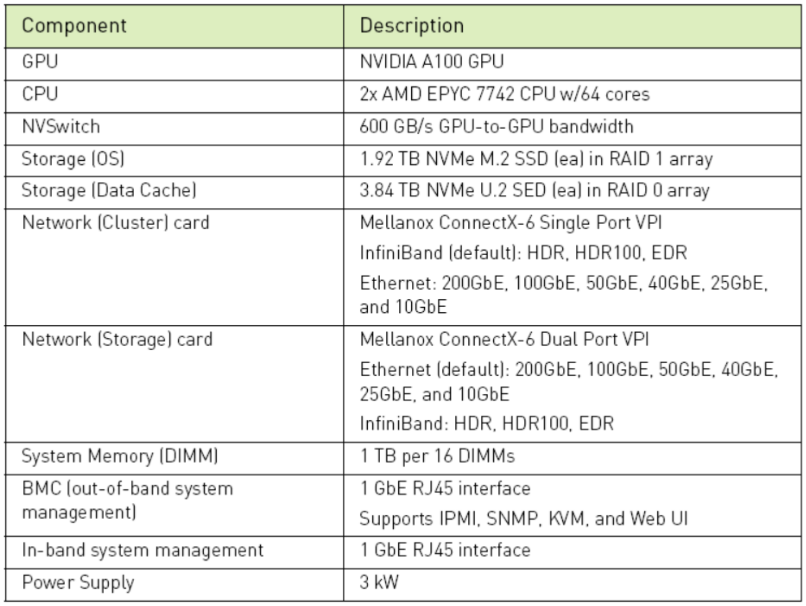
Two NVIDIA DGX™ A100 Stations are deployed in the HPC laboratory "VISTA Lab" in the Évora University. Universal system purpose-built for all AI infrastructure and workloads, from analytics to training to inference. Each station is built on eight NVIDIA A100 Tensor Core GPUs. Docker Containers platform is used to deploy the toolset for high-performance computing (HPC) in the Vista Lab.



DGX-A100 Description



Component	NVIDIA DGX A100 320GB System
GPU	Qty 8 NVIDIA A100 GPUs Third-generation NVLinks
Total GPU Memory	320 GB
NVIDIA NVSwitch	Qty 6 Second generation (2x faster than first generation)
Networking	Qty 9 (Factory ship config) Mellanox ConnectX-6 VPI HDR IB/200 Gb/s (Optional Add-on: Second dual-port 200 Gb/s Ethernet)
CPU	2 AMD Rome, 128 cores total
System Memory	1 TB (Factory ship config) (Optional Add-on: 1 TB to get 2 TB max.)
Storage	15 TB (Factory ship config) U.2 NVMe Drives (Optional Add-on: 15 TB to get 30 TB max.)



Component	Description
GPU	NVIDIA A100 GPU
CPU	2x AMD EPYC 7742 CPU w/64 cores
NVSwitch	600 GB/s GPU-to-GPU bandwidth
Storage (OS)	1.92 TB NVMe M.2 SSD (ea) in RAID 1 array
Storage (Data Cache)	3.84 TB NVMe U.2 SED (ea) in RAID 0 array
Network (Cluster) card	Mellanox ConnectX-6 Single Port VPI InfiniBand (default): HDR, HDR100, EDR Ethernet: 200GbE, 100GbE, 50GbE, 40GbE, 25GbE, and 10GbE
Network (Storage) card	Mellanox ConnectX-6 Dual Port VPI Ethernet (default): 200GbE, 100GbE, 50GbE, 40GbE, 25GbE, and 10GbE InfiniBand: HDR, HDR100, EDR
System Memory (DIMM)	1 TB per 16 DIMMs
BMC (out-of-band system management)	1 GbE RJ45 interface Supports IPMI, SNMP, KVM, and Web UI
In-band system management	1 GbE RJ45 interface
Power Supply	3 kW

Three Docker-Container open-source applications (`Coldfront`, `Ondemand` and `Xdmod`) work in concert to provide a toolset for high performance computing (HPC).

ColdFront is an allocations management portal that provides users an easy way to request access to allocations for a Center's resources.

OnDemand, is a web-based portal for accessing HPC services. Through OnDemand, users can upload and download files, create, edit, submit and monitor jobs, create and share apps, run GUI applications and connect to a terminal.

XDMoD is a management and monitoring systems, which include: number of jobs, CPUs consumed, wait time, and wall time, with minimum, maximum and the average of these metrics. Performance and quality of service metrics of the HPC infrastructure are also provided, along with application specific performance metrics (flop/s, IO rates, network metrics, etc) for all user applications running on a given resource.

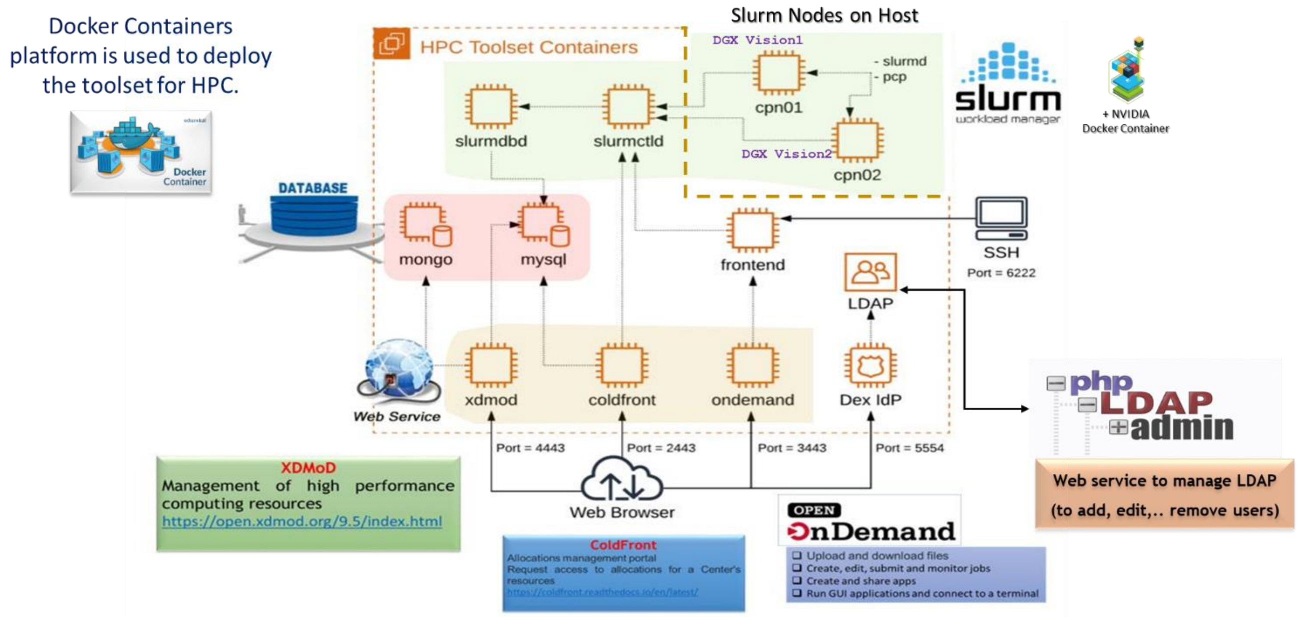
Link Access:

Coldront: `http://vision1.xdi.uevora.pt:2443`

Ondeman: `http://vision1.xdi.uevora.pt`

XDmod: `http://vision1.xdi.uevora.pt:4443`

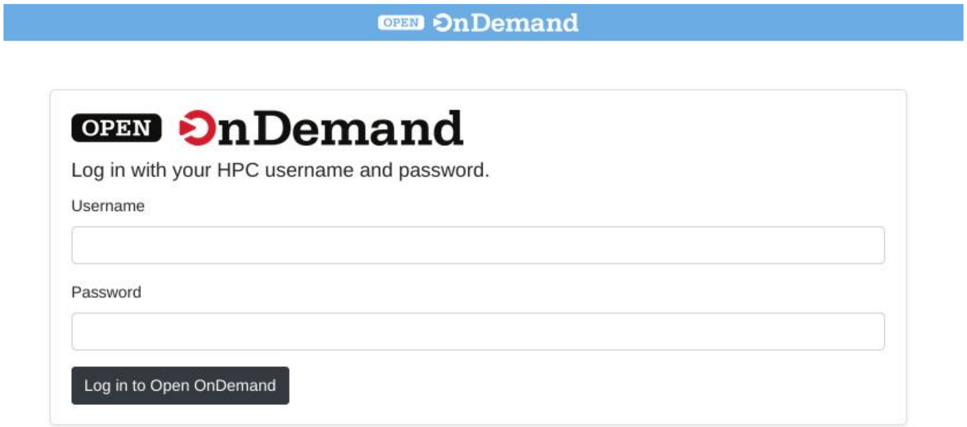
An overview of the docker-containers in the Évora University Cluster:


Docker Containers platform is used to deploy the toolset for HPC.
Docker Container
DATABASE
HPC Toolset Containers
Slurm Nodes on Host
DGX Vision1
DGX Vision2
- slurmd
- pcp
cpn01
cpn02
slurmdbd
slurmctld
mongo
mysql
frontend
SSH
Port = 6222
LDAP
slurm
workload manager
+ NVIDIA
Docker Container
Web Service
xdmod
coldfront
ondemand
Dex IdP
Port = 4443
Port = 2443
Port = 3443
Port = 5554
Web Browser
phpLDAPadmin
Web service to manage LDAP
(to add, edit,.. remove users)
XDMoD
Management of high performance computing resources
https://open.xdmod.org/9.5/index.html
ColdFront
Allocations management portal
Request access to allocations for a Center's resources
https://coldfront.readthedocs.io/en/latest/
OPEN OnDemand
❑ Upload and download files
❑ Create, edit, submit and monitor jobs
❑ Create and share apps
❑ Run GUI applications and connect to a terminal

1. Creating jobs from Ondemand

You should access to Portal **Ondemand** app using any browser (Firefox,Chrome, etc) through:

https://vision1.xdi.uevora.pt



OPEN OnDemand

OPEN OnDemand

Log in with your HPC username and password.

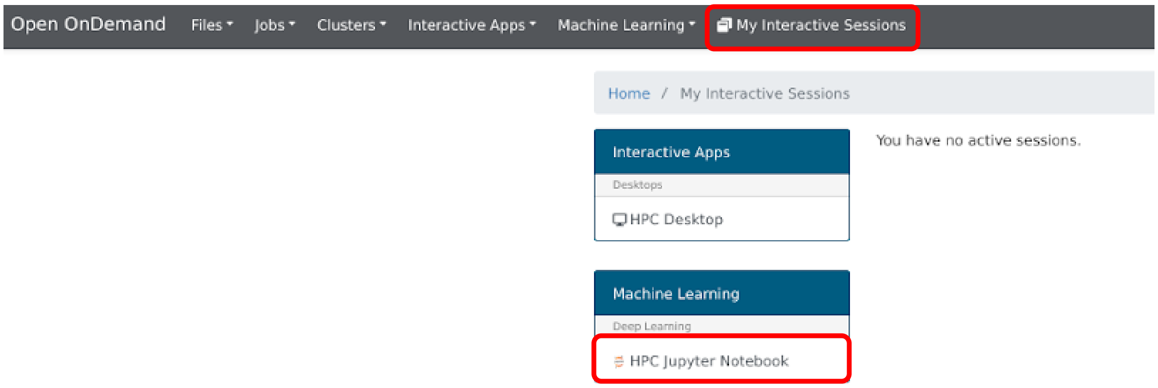
Username

Password

1.1. Creating jobs from Interactive Apps

Interactive applications support graphics without the overhead of installing an X server on your workstation, you can get a full graphical interactive session over your web browser.

Selecting '`My Interactive Sessions`' you can access to setup page for that application.



You have no active sessions.

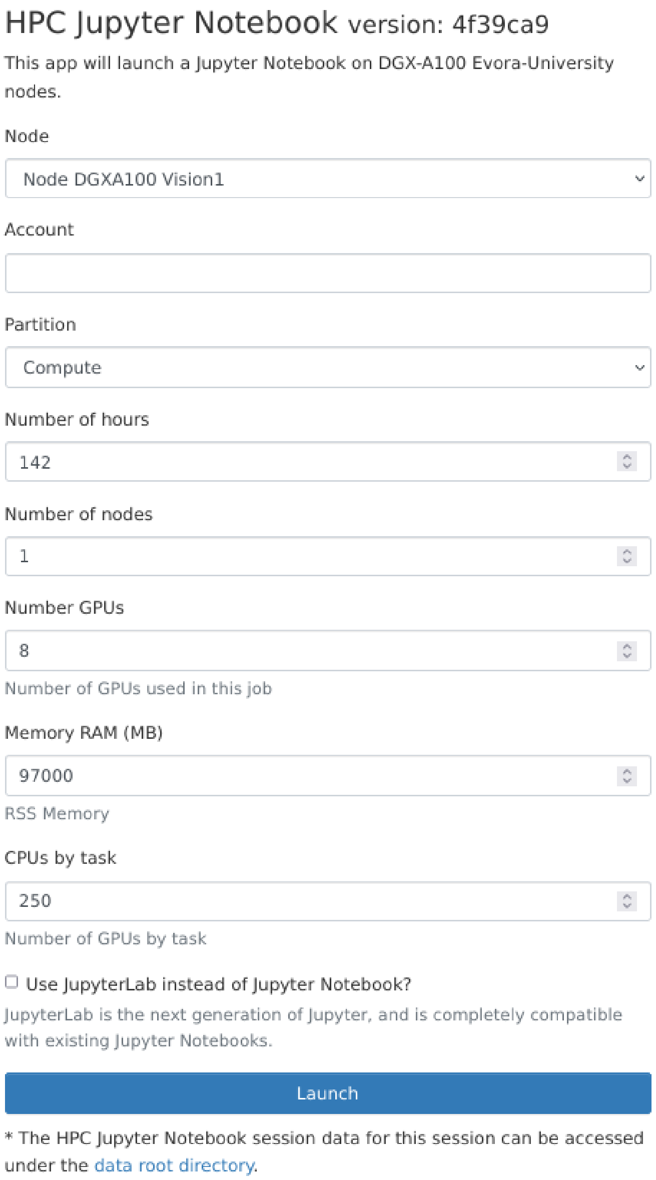
Currently, the following interactive applications are supported:

- HPC jupyter Notebook (or laboratory)
- HPC desktop

1.1.1. Interactive Jupyter Session

This interactive application will allow you to start a jupyter notebook server on one of the compute nodes.

The follow figure shows the different parameters to launch a job.



HPC Jupyter Notebook version: 4f39ca9

This app will launch a Jupyter Notebook on DGX-A100 Evora-University nodes.

Node: Node DGXA100 Vision1

Account:

Partition: Compute

Number of hours: 142

Number of nodes: 1

Number GPUs: 8

Number of GPUs used in this job

Memory RAM (MB): 97000

RSS Memory

CPUs by task: 250

Number of GPUs by task

□ Use JupyterLab instead of Jupyter Notebook?

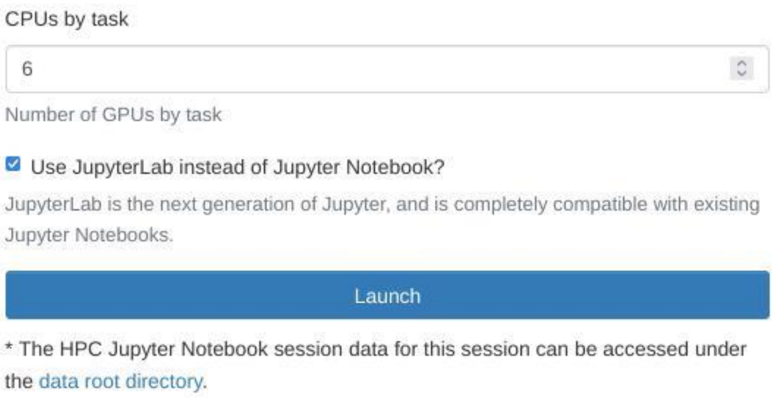
JupyterLab is the next generation of Jupyter, and is completely compatible with existing Jupyter Notebooks.

* The HPC Jupyter Notebook session data for this session can be accessed under the data root directory.

Account: This specifies which allocation account you wish the job to charge. This field is only of concern if you belong to multiple projects, or if you belong to a project with both standard and high priority allocation accounts, and even then only if you wish to use an allocation account other than your default. If you only belong to a single allocation account or wish to use your default allocation account, just leave this blank. Otherwise, enter the name of the allocation account to have the interactive job charge.

Partition: This is for selecting which partition your job is submitted. It is a drop down list allowing you to choose between Compute and Debug. Debug partition is limited to 1 GPU, 1 CPU and ~120GB RAM during 15 min maximum. It is useful for quick testing or checking something. On the other hand, the Compute partition will send your job to either the high-priority or the standard node depending on your allocation account.

You can use Jupyterlab instead of Jupyter Notebook selecting the option according to the following figure.



CPUs by task

6

Number of GPUs by task

☑ Use JupyterLab instead of Jupyter Notebook?

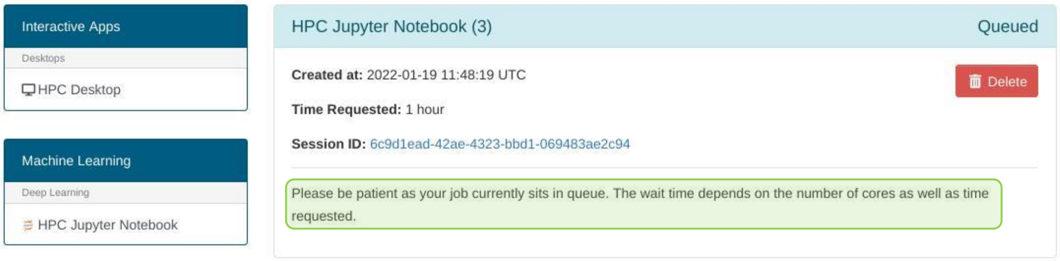
JupyterLab is the next generation of Jupyter, and is completely compatible with existing Jupyter Notebooks.

Launch

* The HPC Jupyter Notebook session data for this session can be accessed under the data root directory.

Now, you can `Launch` the job using the blue bottom.

Firstly, the job is `queued` and according to allocations and cluster load, the job will be launched.



Interactive Apps

Desktops

HPC Desktop

Machine Learning

Deep Learning

HPC Jupyter Notebook

HPC Jupyter Notebook (3) Queued

Created at: 2022-01-19 11:48:19 UTC

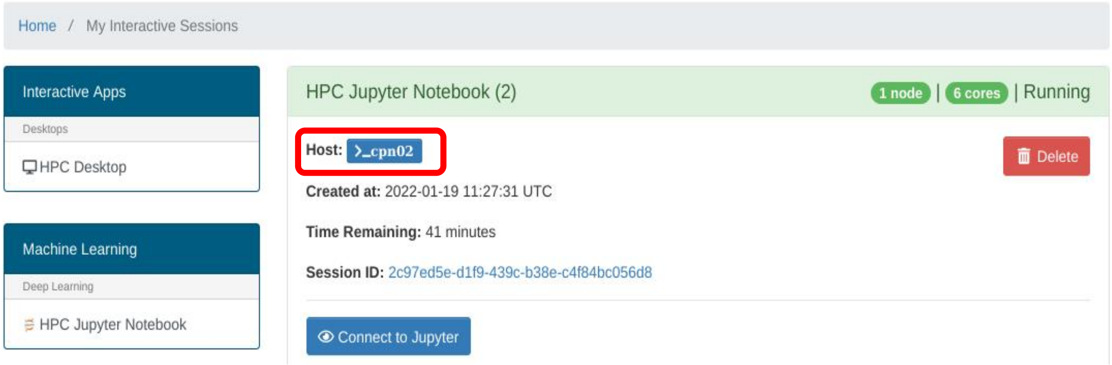
Time Requested: 1 hour

Session ID: 6c9d1ead-42ae-4323-bbd1-069483ae2c94

Delete

Please be patient as your job currently sits in queue. The wait time depends on the number of cores as well as time requested.

After the job is launched you can access the node via shell how is indicated in the following figure.



Home / My Interactive Sessions

Interactive Apps

Desktops

HPC Desktop

Machine Learning

Deep Learning

HPC Jupyter Notebook

HPC Jupyter Notebook (2) 1 node | 6 cores | Running

Host: >_cpn02

Delete

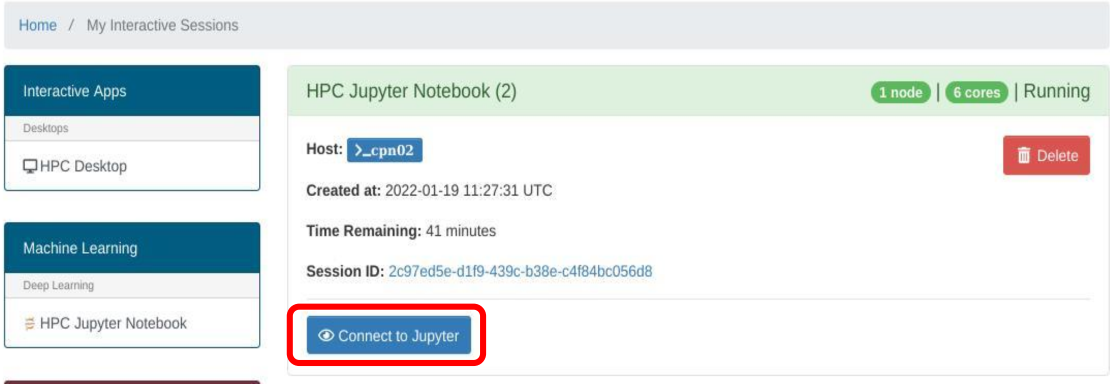
Created at: 2022-01-19 11:27:31 UTC

Time Remaining: 41 minutes

Session ID: 2c97ed5e-d1f9-439c-b38e-c4f84bc056d8

Connect to Jupyter

`Connect to Jupyter` tab will show on the Files-Panel your home directory from where you can navigate to existing notebooks (`file extension: .ipynb`).



HPC Jupyter Notebook (2) | Running

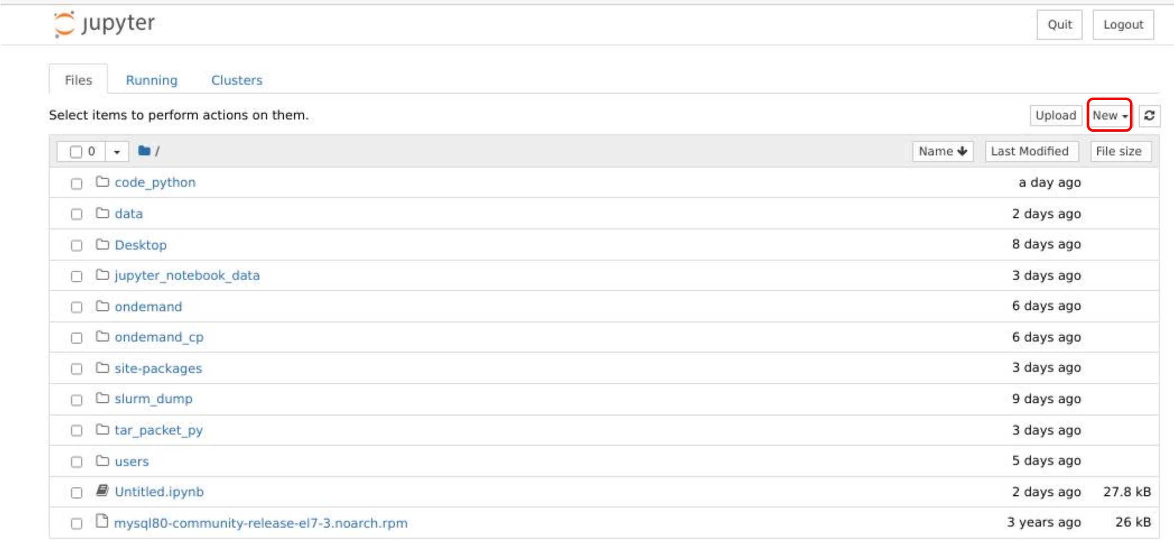
Host: >_cpn02

Created at: 2022-01-19 11:27:31 UTC

Time Remaining: 41 minutes

Session ID: 2c97ed5e-d1f9-439c-b38e-c4f84bc056d8

You can start a new notebook-workspace by selecting the "`New + python3`" option.



Select items to perform actions on them.

Name	Last Modified	File size
code_python	a day ago	
data	2 days ago	
Desktop	8 days ago	
jupyter_notebook_data	3 days ago	
ondemand	6 days ago	
ondemand_cp	6 days ago	
site-packages	3 days ago	
slurm_dump	9 days ago	
tar_packet_py	3 days ago	
users	5 days ago	
Untitled.ipynb	2 days ago	27.8 kB
mysql80-community-release-el7-3.noarch.rpm	3 years ago	26 kB

Adding a new python packages

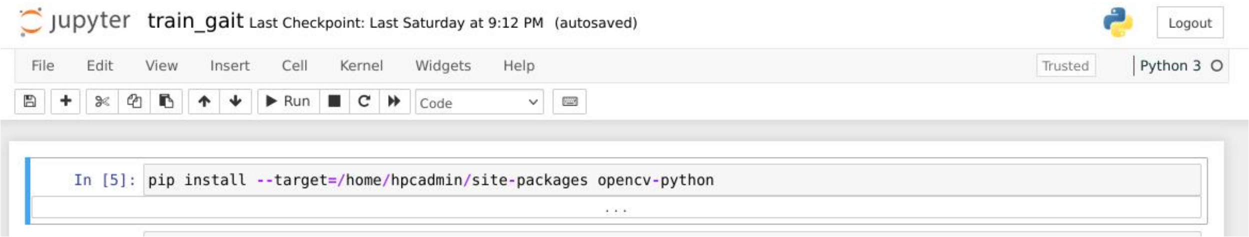
```
pip install --target=/home/$USER/site-packages new-packages
```

Where:
`$USER`: name user
`site-packages`: directory in your `/home`. This directory is already created.
`new-packages`: package name according to PyP site https://pypi.org/.

Example: Install `Opencv` package from https://pypi.org/project/opencv-python/:

```
pip install --target=/home/$USER/site-packages  opencv-python
```

Note: you must change `$USER` by your `user-name`.



```
In [5]: pip install --target=/home/hpcadmin/site-packages opencv-python
```

On the other hand, the path `/home/$USER/site-packages` have been added to your `~/.bashrc` file. Then, if you run `$cat ~/.bashrc` via shell:

Host: >_cpn02

Some like this you can see in the file content:

```
export PYTHONPATH="${PYTHONPATH}:/home/$USER/site-packages
```

Using Multi-GPU in Tensorflow + Keras to train deep leaning

In the following link, you can get an easy sample to run a deep learning network over multi-GPUs.

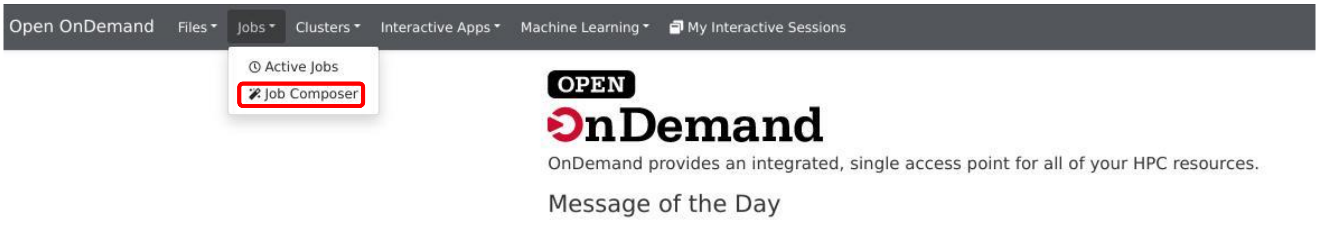
https://www.tensorflow.org/guide/distributed_training#use_tfdistributestrategy_with_keras_modelfit

1.2 Creating jobs from Batch

In many case production work on HPC clusters is done via batch jobs. Typically, one sets up a calculation that will run for many hours, or even days, and submits it to the cluster to run when resources become available (which in itself might take hours), and then return to look at the results after the job is done.

1.2.1. Creating jobs from template

The `Job Composer` allows one to create, edit, submit, and manage jobs based on templates or previously submitted jobs.



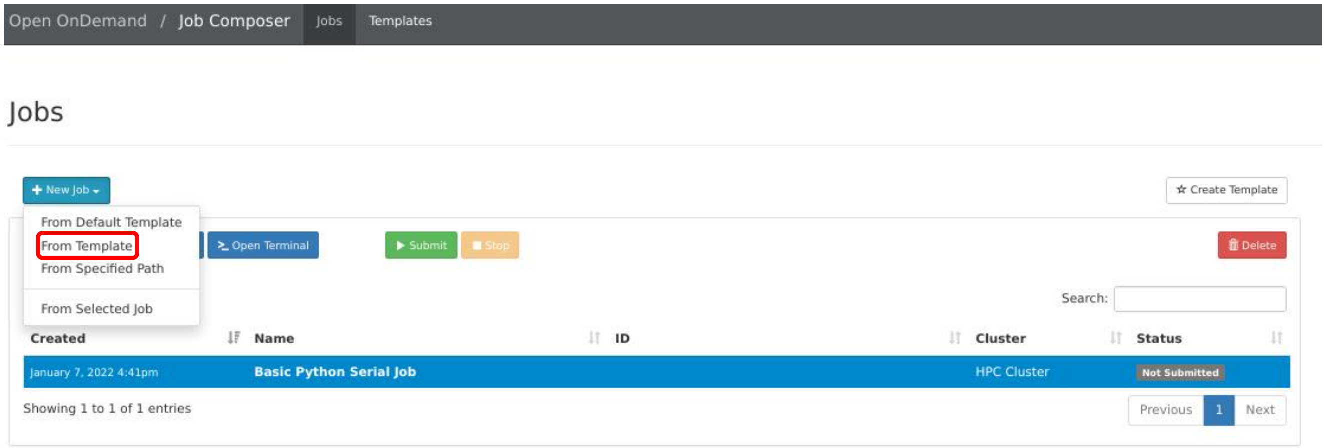
OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

Options for creating a new job from template:

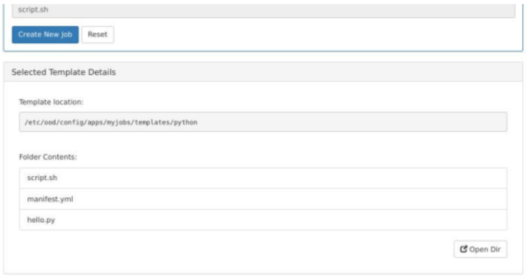
1. From Default Template: this will create a job from the default template. The default template is typcially a fairly basic and generic sequential job, so this is probably not a great choice in production. However, it is useful for learning to use the composer.
2. From Template: This will switch you to the "Templates" section of the Job Composer and gives you a selection of templates to choose.
3. From Specified Path: this will allow you to create a Job Composer job from a job created outside of the Job Composer. You will need to specify the path to the directory containing the job, and some other parameters. The contents of the directory you specify will be copied into the newly created Job Composer job directory.
4. From Selected Job: this will copy an existing Job Composer job to a new Job Composer job. Before you can use this, you must have jobs in your job list (the table marked L in the screenshot) and have selected one of those jobs.

To create a new job-template, select "`From Template`"



Created	Name	ID	Cluster	Status
January 7, 2022 4:41pm	Basic Python Serial Job		HPC Cluster	Not Submitted

Showing 1 to 1 of 1 entries



Template location:

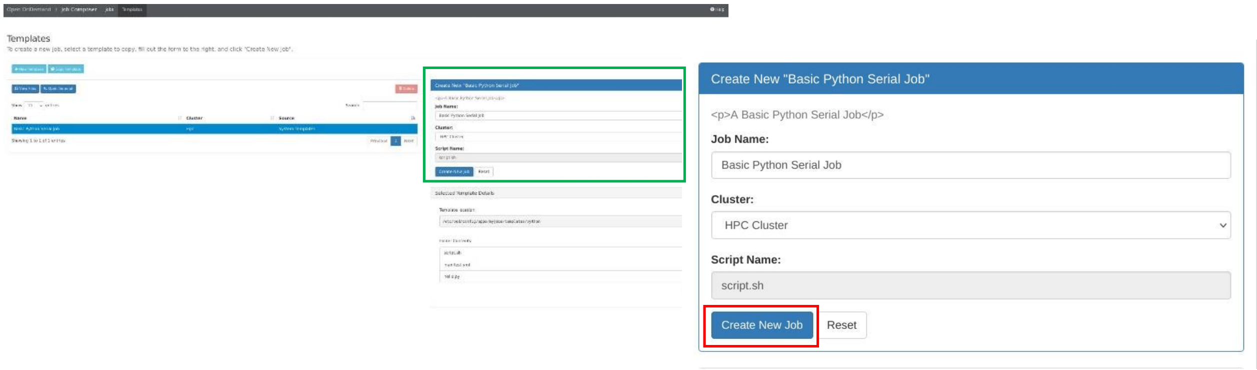
/etc/ood/config/apps/myjobs/templates/python

Folder Contents:

script.sh
manifest.yml
hello.py

Selecting "`Create New Job`", is created a new directory under the ondemand folder in your home directory and relevant files to that directory are copied. You then can edit, add, and/or delete files in that directory.

`/home/$USER/ondemand/data/sys/myjobs/projects/default/new-directoy_created`



Create New "Basic Python Serial Job"

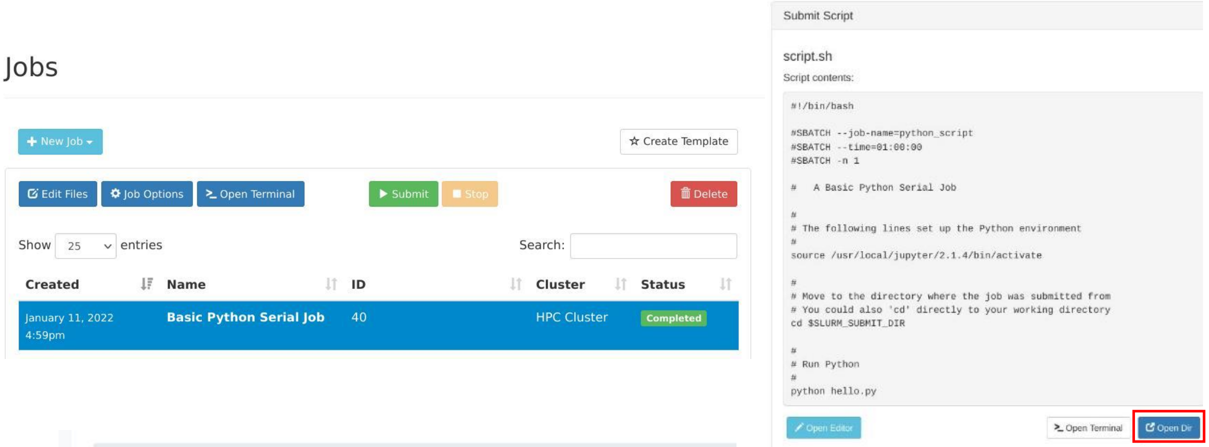
<p>A Basic Python Serial Job</p>

Job Name: Basic Python Serial Job

Cluster: HPC Cluster

Script Name: script.sh

You can access to this directory by selecting "`Open Dir`" or doing click over the file directly.



Show 25 entries Search:

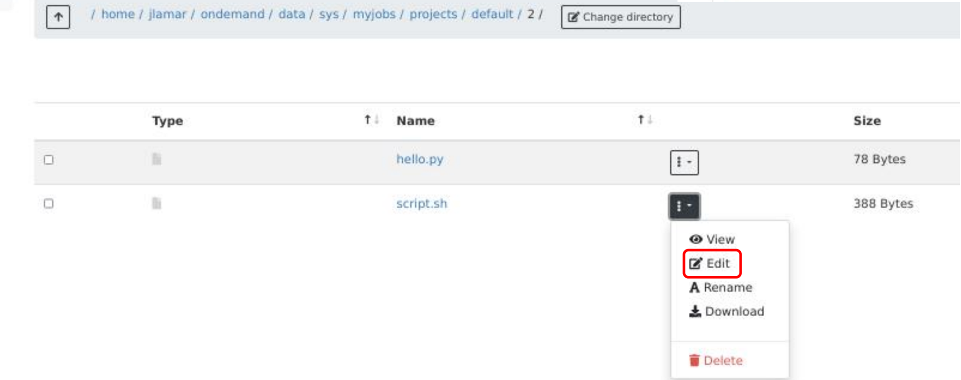
Created	Name	ID	Cluster	Status
January 11, 2022 4:59pm	Basic Python Serial Job	40	HPC Cluster	Completed

Submit Script

script.sh

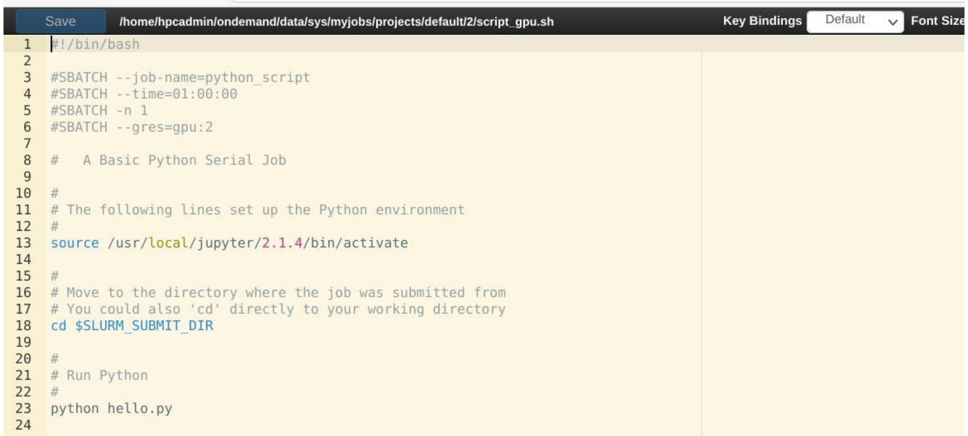
Script contents:

```
#!/bin/bash

#SBATCH --job-name=python_script
#SBATCH --time=01:00:00
#SBATCH -n 1

#   A Basic Python Serial Job

#
# The following lines set up the Python environment
#
source /usr/local/jupyter/2.1.4/bin/activate

#
# Move to the directory where the job was submitted from
# You could also 'cd' directly to your working directory
cd $SLURM_SUBMIT_DIR

#
# Run Python
#
python hello.py
```



/ home / jlamar / ondemand / data / sys / myjobs / projects / default / 2 /

	Type	Name		Size
□		hello.py		78 Bytes
□		script.sh		388 Bytes



/home/hpcadmin/ondemand/data/sys/myjobs/projects/default/2/script_gpu.sh

Key Bindings Default Font Size

```
#!/bin/bash

#SBATCH --job-name=python_script
#SBATCH --time=01:00:00
#SBATCH -n 1
#SBATCH --gres=gpu:2

#   A Basic Python Serial Job

#
# The following lines set up the Python environment
#
source /usr/local/jupyter/2.1.4/bin/activate

#
# Move to the directory where the job was submitted from
# You could also 'cd' directly to your working directory
cd $SLURM_SUBMIT_DIR

#
# Run Python
#
python hello.py

```

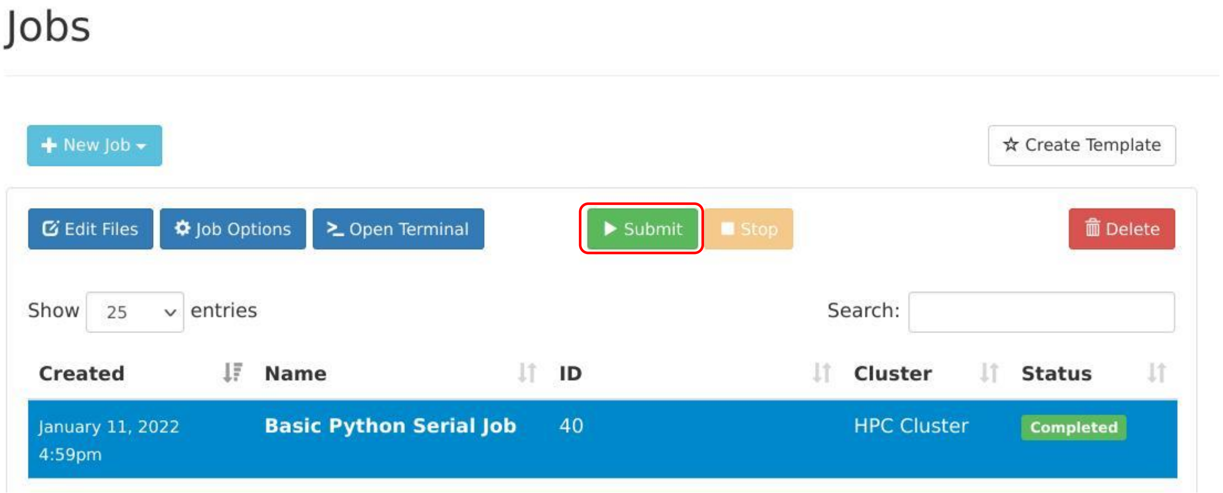
The follow is an example of `script.sh` using GPUs.

```
#!/bin/bash

#SBATCH --job-name=python_script
#SBATCH --time=01:00:00
#SBATCH -n 1
#SBATCH --gres=gpu:2

#   A Basic Python Serial Job
# The following lines set up the Python environment
source /usr/local/jupyter/2.1.4/bin/activate

# Move to the directory where the job was submitted from
# You could also 'cd' directly to your working directory
cd $SLURM_SUBMIT_DIR

# Run Python
python /home/hpcadmin/ondemand/data/sys/myjobs/projects/default/2/hello.py
```

In this case, an environment jupyter is activated on the node to run the python code in `hello.py`.

Finally, save the changes, close Edit-page and “`Submit`” the job.

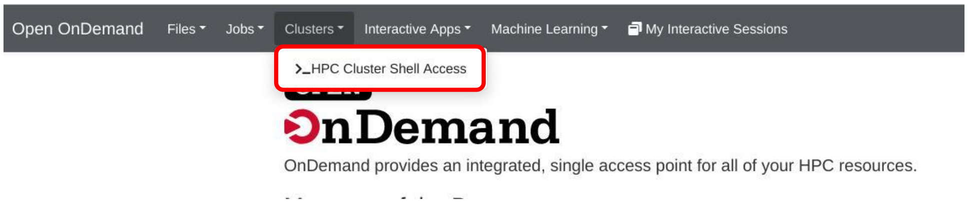
When the job end, you can select again “`Open Dir`” and look at the job output in the ***slurm-40.out*** file created.



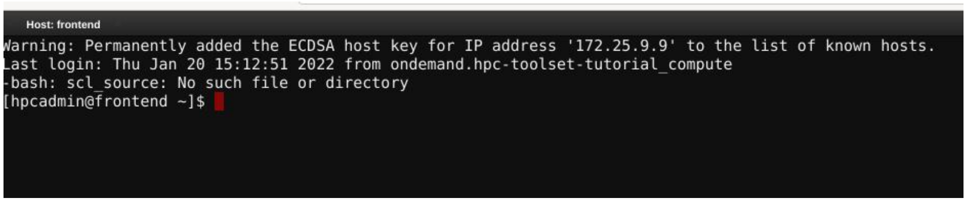
Created	Name	ID	Cluster	Status
January 11, 2022 4:59pm	Basic Python Serial Job	40	HPC Cluster	Completed

1.2.2 Launch jobs from Cluster Shell

Furthermore, you can use Cluster-Shell to launch jobs and to run code.



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

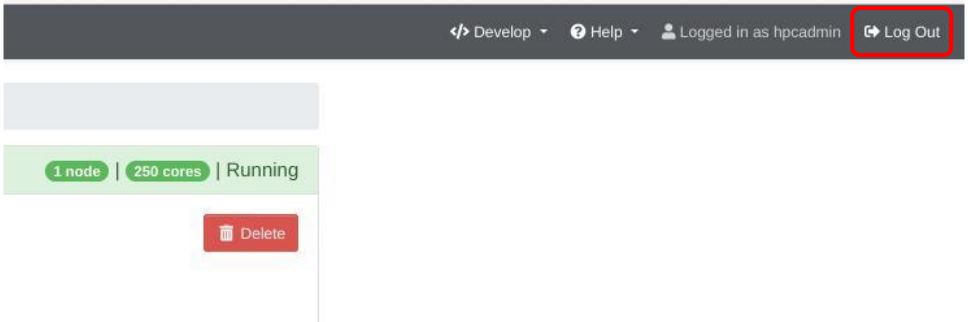


```
Warning: Permanently added the ECDSA host key for IP address '172.25.9.9' to the list of known hosts.
Last login: Thu Jan 20 15:12:51 2022 from ondemand.hpc-toolset-tutorial_compute
-bash: scl_source: No such file or directory
[hpcadmin@frontend ~]$
```

The following example executes the above `script.sh`.

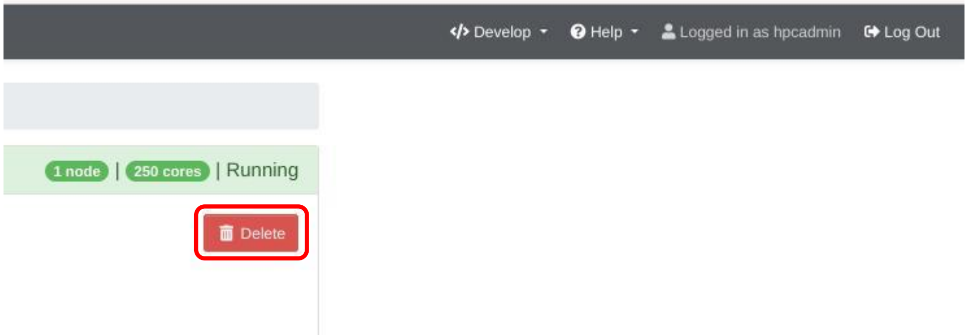
```
[hpcadmin@frontend]$/home/hpcadmin/ondemand/data/sys/myjobs/projects/default/2/script_
gpu.sh
```

1.3 Logout button



If you click the `Log Out` button, it will disconnect your session with the server, but the server will continue running. You can reconnect with the session later from the “`My Interactive Sessions`” menu in the main top menu bar of the `OnDemand` dashboard.

Note: Your allocation account will be charged as long as the job is running. So make sure you use the `Delete` if you wish to finish the job. You can verify if you still have any interactive sessions running from the “`My Interactive Sessions`” option on the main top menu bar in the `OnDemand` dashboard.



2. Resource Slurm Limits

It is important to note, when a job is launched, the job follows a hierarchy Slurm limitations:

1. Partition QOS limit (Quality of Service)
2. Job QOS limit
3. User association
4. Account association(s), ascending the hierarchy
5. Root/Cluster association
6. Partition limit
7. None