



# Getting started on subMIT: Installing / Managing Software

Chad Freer

# Account Setup and Data Storage



## Creating Accounts:

1. Anyone with MIT ID can set up an account by simply uploading ssh keys
  - a. MIT accounts possible for external people collaborating with MIT groups
2. Account gets **automatically** created with all of the user spaces and permissions.

## User space:

1. 5 GB for User's Home → /home/submit/<user>
2. 50 GB for User's Work → /work/submit/<user>
3. 1 TB for User's Data → /data/submit/<user>

## NVME and additional space:

1. 30 TB of NVME scratch space for use by groups.
2. Additional Group space in /data as needed

## Support for using and developing many types of workflows:

1. Many setups are available directly out of box!
2. This talk focuses on what is available and how to **develop your own setup**
3. How each of these setups can be scaled in the batch system

# A First Word on Software Availability



## Native System:

1. Basic tools on the base machines: python, c++, Julia, Java, Matlab, etc.
2. Do not need to manage these but may not be flexible enough for common workflows

## Conda:

1. Personal package manager that you have control over
2. Easy to use and very flexible!

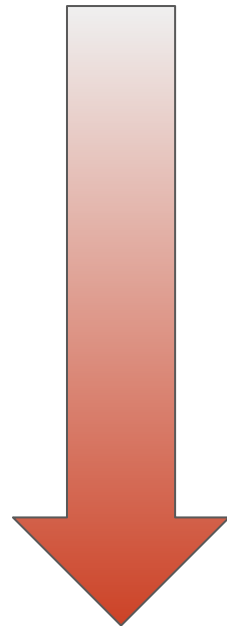
## CernVM File System (CVMFS):

1. CVMFS mounted on submit providing software and environments out of box
2. Easy to use if the software you want is already available!
3. Can also support your own environments through our MIT CVMFS instance
  - a. available on SubMIT, T2 and T3
  - b. This is a little more involved

## Containers:

1. Support for Docker/Podman and Singularity
2. Can be very involved if you are developing your own container

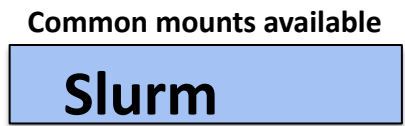
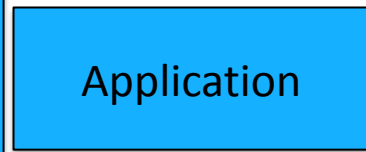
Simple



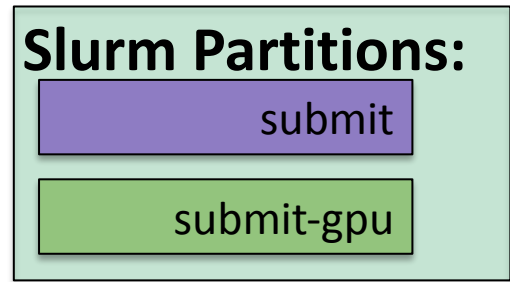
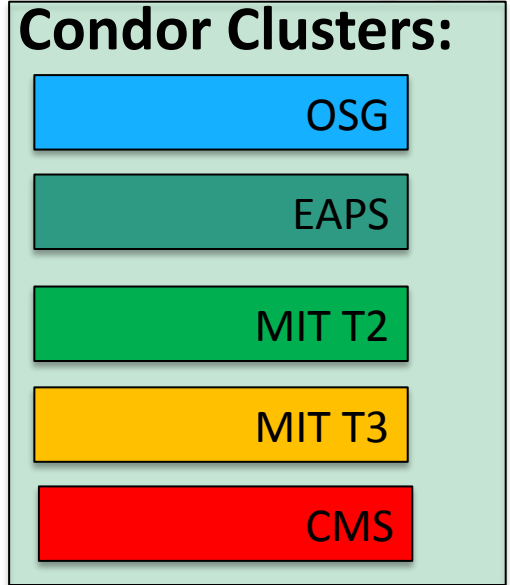
Complex  
to Develop



# Managing/Scaling Your Workflows



User specifies the location of output from both jobs:  
 /data/submit/<username>/<path>  
 /work/submit/<username>/<path>  
 /scratch/submit/<group>/<path>



# Conda

## User's guide link:

1. <https://submit.mit.edu/submit-users-guide/program.html#conda>

## Conda:

1. Personal package manager.
2. Download to your home/work space
3. Complete control over management, multiple environments
4. Widely supported and easy to use
5. Conda environments can often take up a lot of space (use /work)

## Environment:

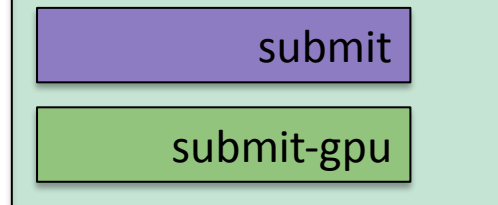
1. Environments live in your work/home areas
2. Only available on machines with those areas mounted (slurm)



## Condor Clusters:



## Slurm Partitions:



# Jupyterhub and Environments



## Jupyterhub Options and GPUs:

1. <https://submit.mit.edu/jupyter/>
2. Jupyterhub uses Slurm to spawn jobs
3. Access to GPUs



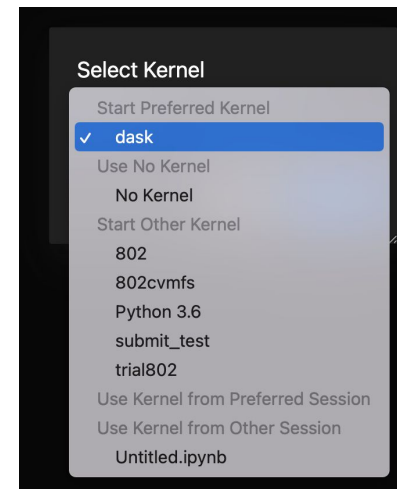
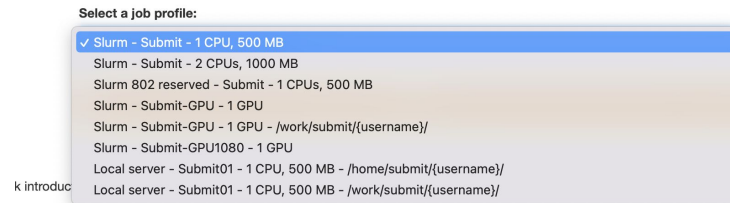
jupyterhub

## Connection with User's Conda environments:

1. ***Jupyterhub loads the user's conda environments!***
2. Central conda environments are also available
3. Provides a lot of flexibility for workflows

## Dask:

1. Dask can be run through jupyterhub
2. Can access conda environment



# CVMFS



## User's guide link:

1. <https://submit.mit.edu/submit-users-guide/program.html#cvmfs>

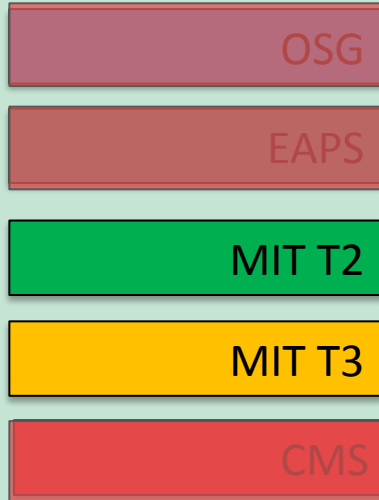
## CVMFS:

1. Scalable and reliable software distribution service
2. Access to a wide array of software and environments
  - a. **Often what you need is already out there!**
  - b. OS support for cmssw-slc6/cmssw-cc7/cmssw-el8/cmssw-el9  
<https://submit.mit.edu/submit-users-guide/program.html#additional-operating-systems-cms-specific>
  - c. /cvmfs/unpacked.cern.ch is available for a range of other images
3. Well supported and available on resources across the world
4. Good for scaling to many thousands of jobs
5. Lives in /cvmfs

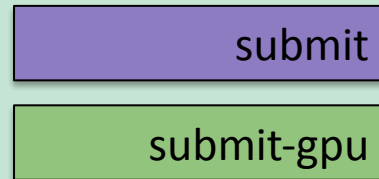
## MIT cvmfs instance: *\*shown right*

1. Users can create their own environments in /work
2. Available on MIT resources (**submit, T2 and T3**)
3. /cvmfs/cvmfs.cmsaf.mit.edu/submit/work

## Condor Clusters:



## Slurm Partitions:



# Containers

## Containers:

1. Standard unit of software that packages up code + dependencies
2. Lightweight, executable, contain everything you need
3. You can also develop your own containers (more complex)
4. <https://submit.mit.edu/submit-users-guide/program.html#containers>

## Docker/Podman:

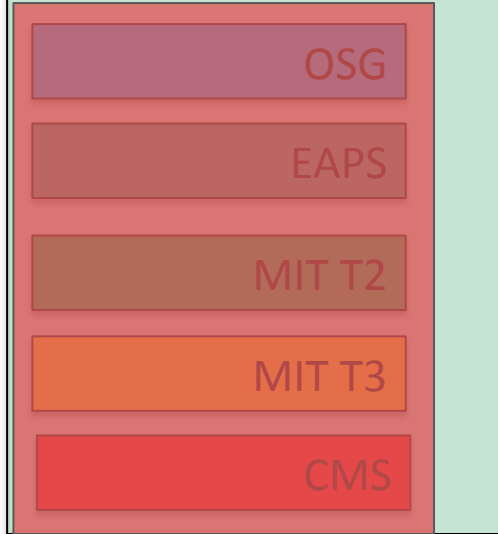
1. All users have access to Docker (Podman for almalinux)
2. Can build your own images or use premade through Dockerhub
3. Building your own dockers takes some overhead but is very flexible!
4. Dockerhub has wide support from community and is easy to use

## Singularity:

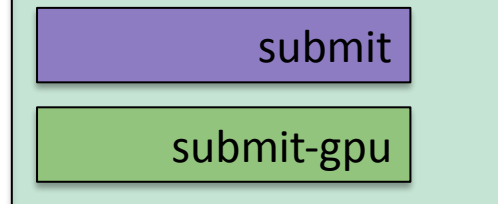
1. Singularity is designed for ease of use on shared resources in HPCs
2. Can interplay with Docker and be made available on MIT CVMFS



## Condor Clusters:



## Slurm Partitions:





## **SubMIT allows users to develop/use their software in ways best suited to their needs**

**Many simple solutions are available to you out of box**

1. native system has basic tools
2. cvmfs and docker are well supported and have

**Users can create their own setups to fit their specific workflow**

1. conda allows users great control over their packages and is easy to use
2. docker/singularity support means that users can create more complex setups

**Submit is connected to many resources and allows users to scale their projects**

1. Slurm can be used easily with most setups and includes GPU support
2. HTCondor is also available but may be limited in scope due to not having mounted directories
  - a. There still remains a lot that can be done

# Backup





# SubMIT



An MIT Physics Department analysis facility.

→ provide ecosystems to many research areas

<https://submit.mit.edu>

subMIT system provides an interactive login pool + scale-out to batch resources

- Home and Work directories
- SSH or Jupyterhub access
- Convenient software environment (CentOS7 native (moving to Alma Linux 9), docker/singularity images, conda)
- Local batch system with O(1000) cores, >50 GPU's 8 A30s and ~45 1080s (more being integrated)
- Local storage (1TB/user), 10's of TB for larger group datasets (gluster distributed file system with ~400TB of spinning disks)
- Fast networking: 100 Gbps ethernet
- Convenient access to larger external resources (OSG, CMS Tier-2 and Tier-3, LQCD Cluster, EAPS)
- Strong focus on user support (ticketing system, AI-based chatbot in development)

# Examples of workflows on subMIT from LHC/CMS



Very different analysis requirement

1. Search for **rare decay of the Higgs Boson (CADI HIG-23-005)**:
  - a. largely profit of event size reduction, simple calculations and almost interactive analysis
  - b. small final dataset for ML inference, GPU used for training
  - c. use of SSD disk, RootData Frame, conda, correctionlib for syst evaluation
2. Search for **“Soft Unclustered Energy Patterns (SUEPs)” (CADI EXO-23-002)**:
  - a. real time analysis reclustering the “jets”, select SUEP candidates and boost in that frame
  - b. heavily relying in the parallelization (batching HTcondor)
  - c. Coffea SW, conda, Dask used for several studies.
3. Measurement of the **W boson mass (CADI SMP-23-002)**:
  - a. challenge in bookkeeping of templates for systematics variation of uncertainty weights for both background and signal, i.e. build  $O(10^3)$  replicas of the final histograms
  - b. need multithreading and memory-based challenges
  - c. need a big machine for now
  - d. tensorflow used for final fit
  - e. RootDataFrame, singularity image

Common feature: use the nanoAOD simplified data format as input

# Batch Computing Systems



## Batch Computing:

1. Computerized batch processing is a method of running software programs in batches automatically.
2. Users are required to submit the jobs, but no other interaction is required

## Slurm vs HTCondor:

### 1. High Throughput or HTCondor:

- a. high-throughput computing software framework for coarse-grained distributed parallelization of computationally intensive tasks.
- b. It can be used to manage workload on a dedicated cluster of computers (No Common Mount points/filesystems)

### 2. Simple Linux Utility for Resource Management or Slurm:

- a. allocating access to resources (computer nodes) to users for some duration of time so they can perform work
- b. framework for parallel jobs such as Message Passing Interface (MPI) on a set of allocated nodes
- c. Home and other mount point available on all nodes



# Slurm Partitions on SubMIT

1. Main partition (“submit”):
  - a. Main partition for SubMIT
  - b. O(2000) CPU cores available to users spread across 19 machines
  
2. GPU partition (“submit-gpu”):
  - a. Main submit GPU partition
  - b. 4 servers with 48 CPUs and 2 A30 GPUs each
  
3. Additional GPU resources (“submit-gpu1080”):
  - a. Secondary GPU partition
  - b. 12 servers with 24 CPUs and 4 1080 GPUs each
  
4. AlmaLinux test partition (“submit-alma9”):
  - a. Test partition for AlmaLinux
  - b. Currently only 2 new servers but will be expanded as we upgrade servers
    - i. Will continue to provide support for CentOS through singularity