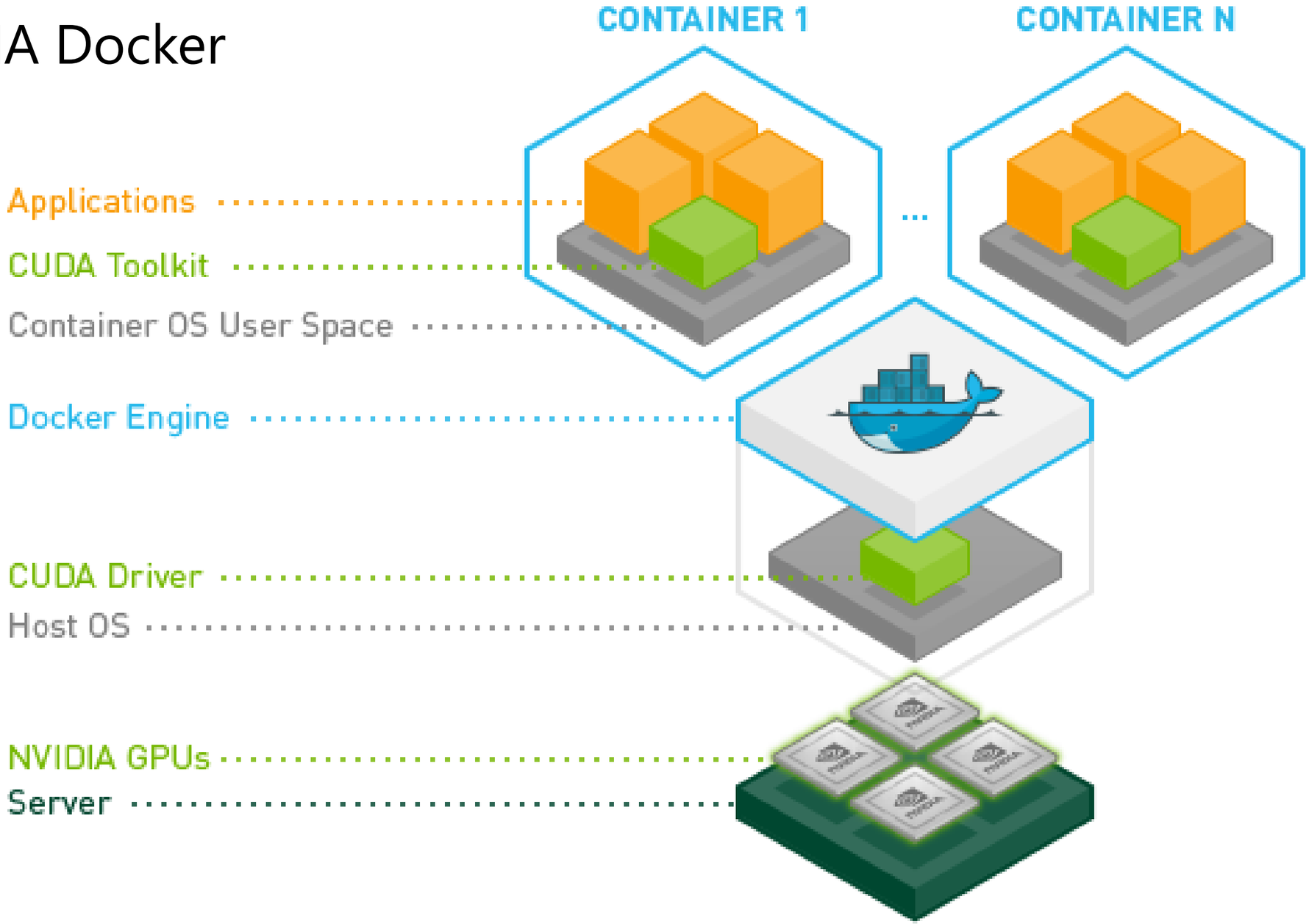
Introduced yourself to DGX1

Atul Yadav

NVIDIA Docker



Images vs Containers

Images

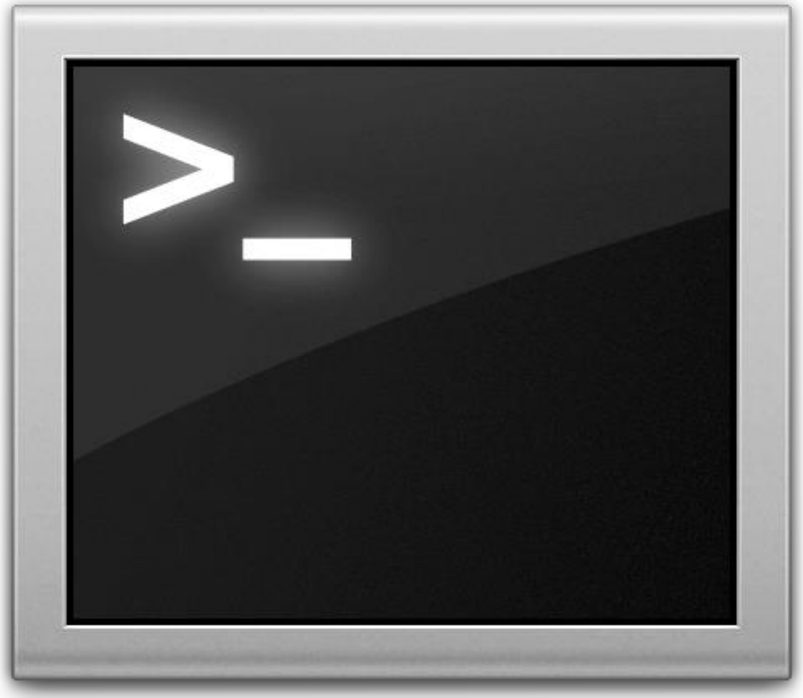
- Docker images are the basis of containers
- An image is an ordered collection of root file system
- An image does not have state and it never changes

Containers

- A container is a runtime instance of a docker images.
- A Container consists of a docker image and execution environment

Accessing the system

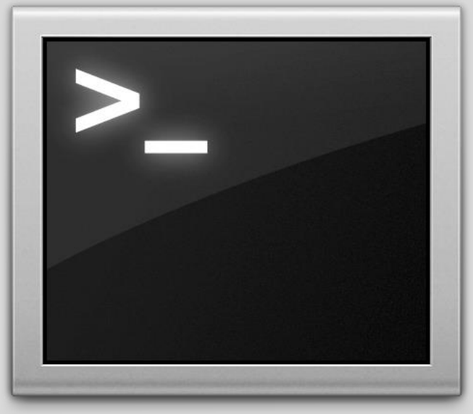
- The NVIDIA-DGX1 cluster has one node(Login & Compute), nvidia-dgx, through which the user can access the cluster and submit jobs.
- The machine is accessible for login using ssh from inside IISc network.
  - **ssh <computational_userid>@nvidia-dgx.serc.iisc.ac.in**
- The machine can be accessed after applying for basic HPC access, for which:
- Fill the online HPC application form here & submit at Room: 109, SERC.
- HPC Application form must be duly signed by your Advisor/Research Supervisor.



- Where do you start?

Cluster basics

Head node(s)



Login to head node

Scheduler

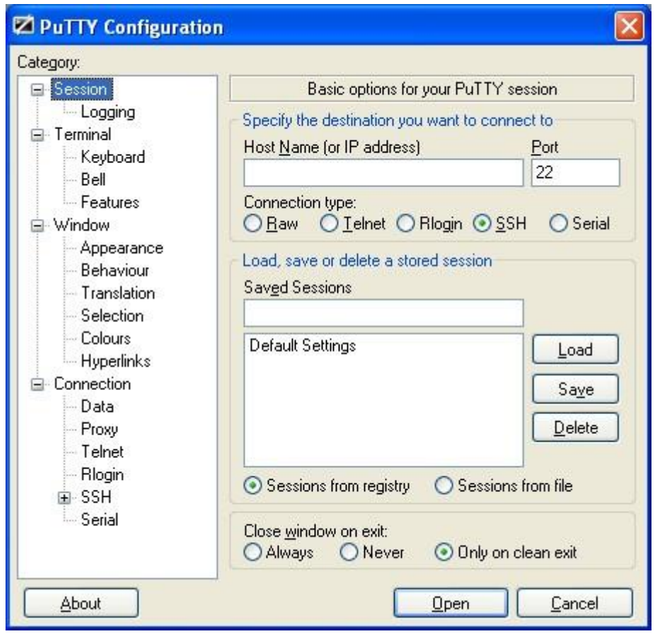
Tell the scheduler what you want to do

Compute resources

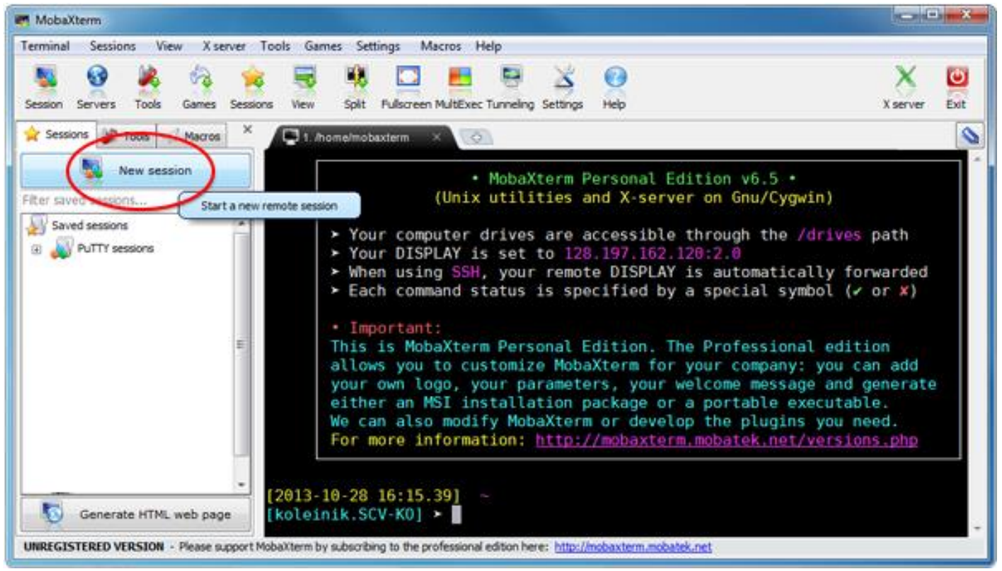
Your job runs on the cluster

Logging in

PuTTY



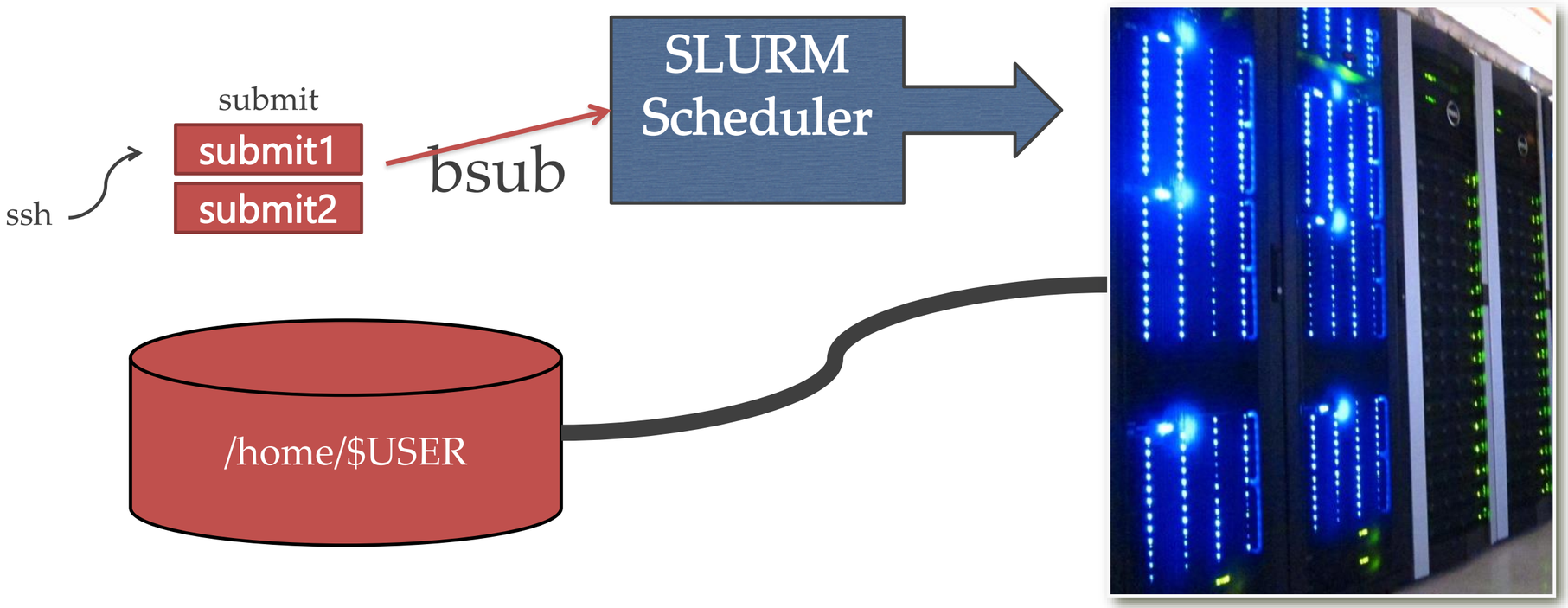
MobaXterm



Scheduling a job

- Need to tell scheduler what you want to do
  - Information about **how long** your job will run
  - How many **GPUs** you want and how you want them grouped
  - How much **RAM** your job will use
  - The **commands** that will be run

General Cluster Schematic



- Submission Script

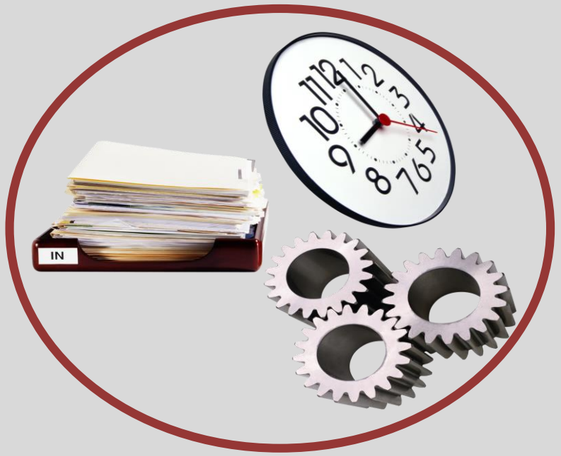
```
#!/bin/bash
#
#SBATCH --job-name=serial_job_test    # Job name
#SBATCH --ntasks=1                    # Run on a single CPU
#SBATCH --time=02:00:00               # Time limit hrs:min:sec
#SBATCH --output=serial_test_%j.out   # Standard output and error
log
#SBATCH --cpus-per-task=1
#SBATCH --gres=gpu:1
#SBATCH --mem=12GB
#SBATCH --partition=qreserve

pwd; hostname; date |tee result
echo "Running program on $SLURM_CPUS_ON_NODE CPU cores"
echo "Running program on $CUDA_VISIBLE_DEVICES GPU Devices"
nvidia-smi
```

Compute resources

Your job runs on the cluster

Scheduler



Tell the scheduler what you want to do

Running Jobs on the Cluster

- You must make reservations!
  - Cluster is a shared resource, so you must ask for exclusive use of nodes and cores
  - The job request goes into a queue, and is granted when resources are available
  - How to do this? bsub!

Environment

- Resource Manager
  - Responsible to allocate resources within a cluster
    - What are the resources?
      - CPUs
      - Memory
      - Time
      - GPUs
  - Scheduler (limited resource for lot of work)
  - Manages queues

My First Program

- **nvidia-docker run -t ${USE_TTY} --name $SLURM_JOB_ID --user $(id -u $USER):$(id -g $USER) --rm -v /home/proj/18/secguest1/atul:/workspace -v /etc/passwd:/etc/passwd -v /etc/group:/etc/group -v /etc/shadow:/etc/shadow nvcr.io/nvidia/tensorflow:18.09-py3 python -c 'import tensorflow as tf; print (tf.__version__);'**

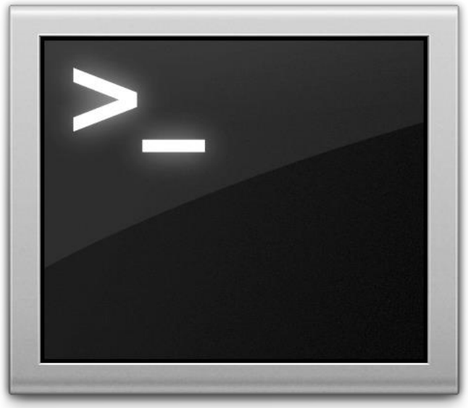
My First Program

- nvidia-docker run
  - Syntax for calling docker for running the deep learning framework
- -t ${USE_TTY}
  - Terminal
- --name $name
  - Name Of the Images
- --rm -p 9999:8888
  - Setting up the port
- --user $(id -u $USER):$(id -g $USER)
- -v /etc/passwd:/etc/passwd
- -v /etc/group:/etc/group
- -v /etc/shadow:/etc/shadow
  - Mounting the User Variable
- -v /home/$USER:/workspace
- -v /localscratch/demo:/workspace
  - Mounting the Working Directory
- nvcr.io/nvidia/tensorflow:18.09-py3 python -c 'import tensorflow as tf; print (tf.__version__);'
  - Framework'

Helpful command

- #sinfo
  - reports the state of partitions and nodes
- #sbatch
  - submits a job script
- #squeue
  - reports the state of jobs in the batch queue
- #scancel
  - cancels a pending or running job jobid

Linux Command Line



- Lots of online resources
  - Google: cheat sheet
- User manuals for applications
- The faster you can type, the faster you will be done

The Shell

- When you type commands and run programs, you are actually running a program called a **shell**
  - Designed to take user input, run programs and display output
  - Started automatically when Terminal app started or when you log into a computer
  - Linux runs the **bash** shell, by default
- Maintains useful **environment variables**
  - **$PWD**, which holds your **current working directory** path
  - **$HOME** or ~, which holds your home directory path
  - **$PATH**, which holds locations of programs
- Powerful tool for organizing and executing commands
  - Useful to combine programs or redirect inputs and outputs, without having to write a program to do that
  - Full-fledged programming language, used to write **shell scripts** to run sets of commands

Key Points

- Submit all jobs from non-root user only
- ssh should be password-free from master node to other compute nodes particularly non-root users
- Job submission Directory/Home Directory should be shared across the nodes
- Applications Directory should be shared / stored same location across the nodes
- After kill job it is better to check that whether job is properly killed or not.
- Check availability of licenses before submitting the job if you are running licensed application
- Always check the queue status before submitting the job.

What do you need to know how to do to “survive”?

- How to get into the cluster, and back out again.
- How to run commands in the shell.
- How to navigate around the directories (and make and remove them).
- How to create, look at and edit text files.
- How to write scripts to do the computations you need to do.
- How to submit jobs, to run things on the nodes.

Helpful Tips

- Ask these questions to keep you oriented
  - What computer am I on?
    - Look at the prompt, ‘hostname’
  - What directory am I in?
    - Look at the prompt and window top
    - ‘pwd’, ‘cd’
  - Where are the files for my analysis?
    - ‘ls’
    - ‘mkdir’, ‘rm’, ‘rmdir’
    - ‘more’ or ‘less’, ‘head’, ‘tail’
  - What program(s) do I have running?
    - ‘ps’, ‘top’, ‘screen’
  - What jobs do I have running?
    - ‘bsub’

- Help and Support
  - We are here to help!