intel AI Academy

HANDS-ON:
TRAIN & INFERENCE CNN ON INTEL® AI DEVCLOUD

Chris Ma
Technical Solution Engineer

18 Mar 2019

AGENDA

- Introduction to Intel® DevCloud
- How to access to Intel® DevCloud
- How to run / track / check / delete a job on Intel® DevCloud
- Demo 1
  - Train & inference Cifar-10 dataset via QSUB in interactive mode (Sources by Prof. Aven)
- Demo 2
  - Train & inference breeds dataset via QSUB in normal mode

INTEL® AI DEVCLOUD

- Broad community of developers, data scientists, students, professors and startups
- Intel® Xeon® Scalable Processors(Intel® Xeon® Gold 6128 CPU @ 3.40GHz 24 cores with 2-way hyper-threading, 192 GB RAM (DDR4), 200 GB of file storage
- Intel® Neon™, Intel® TensorFlow*, Intel® Caffe*, Intel® Theano*, Intel® Python* 2.7 and 3.6 including NumPy, SciPy, pandas, scikit-learn, Jupyter, matplotlib, mpi4py, etc…
- **4 weeks of initial access, with extension based upon project needs**
- Technical support via Intel® AI Academy Support Community
- **Access Request**: https://software.intel.com/ai-academy/tools/devcloud

WELCOME TO DEVCLOUD

Dear Xiaowei Ma,

Please make a note of these new features of the Intel® AI Builder:

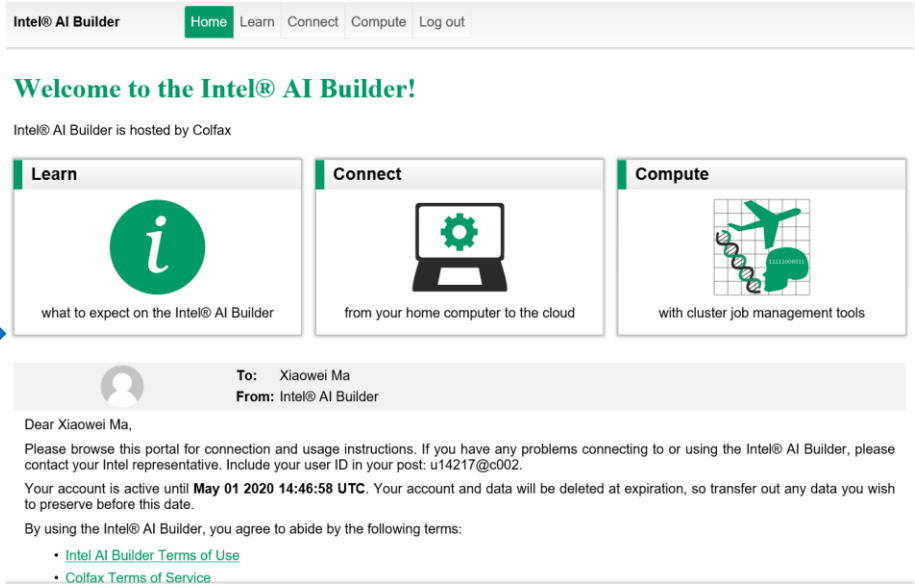
1) Now you can check the storage quota for your home directory using the command "getquota".

2) If you go over 90% of your storage quota, we will send you a courtesy notice.

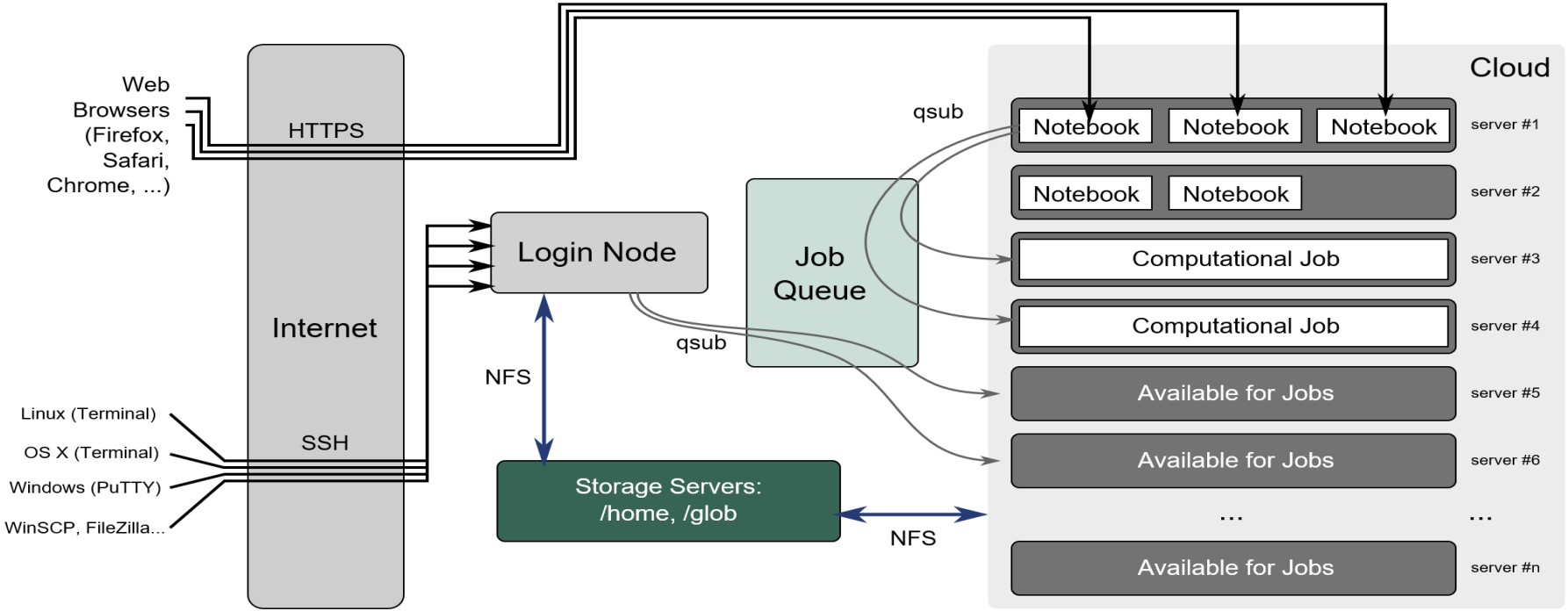
3) Your jobs can use scratch space on local SSDs of compute nodes. The location of the scratch directory indicated by the environment variable $PBS_SCRATCHDIR

You can find more information about these features in the access portal at the following URL:
https://access.colfaxresearch.com/?uuid=4733daa6-a2c1-459e-a580-7055a98f1145&p=learn#sec-dat

Click



INTRODUCTION TO DEVCLOUD ARCHITECTURE



ACCESS TO DEVCLOUD

- Linux / Mac / Linux on Windows
  - Download and save the Linux Access Key
  - Update ~/.ssh/config
  - Set correct restrictive permissions on the private SSH: $ chmod 600
- If you are using Putty From Windows
  - Download the ssh client PuTTY – make sure to use the 64bit MSI installer.
  - Download and save Windows access key
  - Right click on the downloaded key and choose “Load into Pageant”

ACCESS TO DEVCLOUD

- Linux / Mac / Linux on Windows
  - Download and save the Linux Access Key
  - Update ~/.ssh/config
  - Set correct restrictive permissions on the private SSH → chmod 600
- If you are using Putty From Windows
  - Download the ssh client PuTTY suite, 32 or 64 bit MSI installer.
  - Download and save Windows access key
  - Right click on the downloaded key and choose “Load into Pageant”

Quick steps:

1. Edit addkey.sh
2. Run addkey.sh

ADD KEY ACCESS TO DEVCLOUD

Step 1
Add key

1

Home | Learn | Connect | Compute | Log out

3. Connecting with Terminal (from Linux or a Mac)

If you are running Linux or an OS X operating system, then to access the cluster using a Secure Shell (SSH) client, you will need to set up SSH tunneling as described below.

3.1. Preparation

2

1. Download and save the Linux access key to the folder `~/Downloads/` on your computer
2. Add the following lines to file `~/.ssh/config`: (if you do not have this file, simply create one).

ADD KEY ACCESS TO DEVCLOUD

<file addkey.sh> addkey.sh

```
mkdir ~/.ssh
touch ~/.ssh/config
echo "Host colfax" > ~/.ssh/config
echo "User uxxxx" >> ~/.ssh/config
echo "IdentityFile ~/Downloads/colfax-access-key-xxxx" >> ~/.ssh/config
echo "ProxyCommand ssh -T -i ~/Downloads/colfax-access-key-xxxx guest@cluster.colfaxresearch.com" >> ~/.ssh/config
chmod 600 ~/Downloads/colfax-access-key-xxxx
chmod 600 ~/.ssh/config
```

3 **Replace xxxx with your access key number**

Make sure your key is under ~/Download/

4 **Run addkey.sh script**

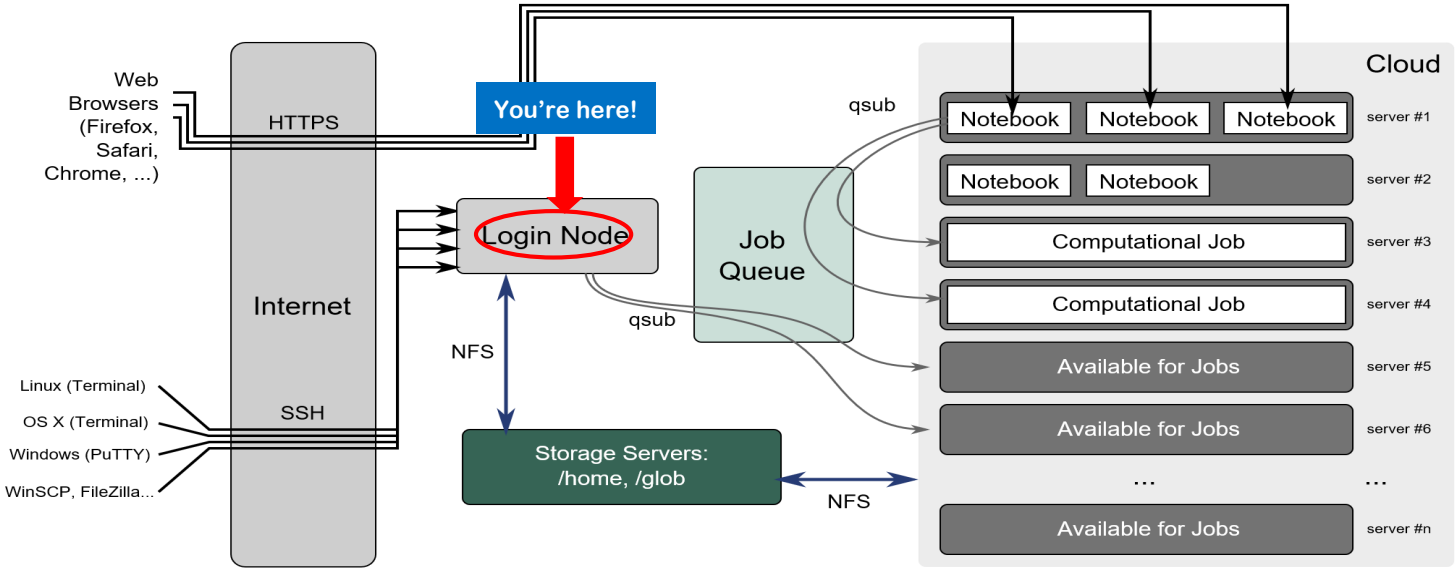
```
$ sh addkey.sh
```

ACCESS TO DEVCLOUD

- Connect to the login node

  $ ssh colfax # Notice it is **NOT** compute node

Step 2
Connect



LOG IN COMPUTE NODE

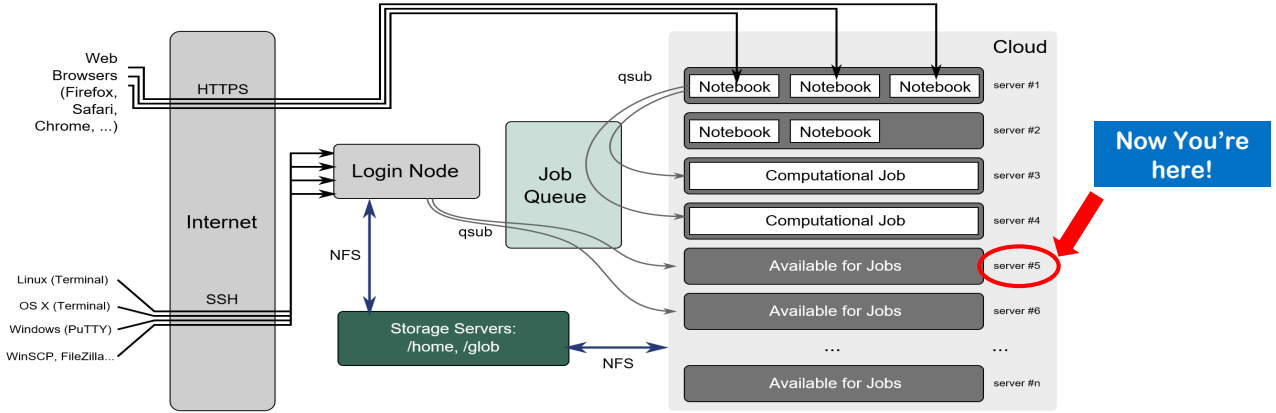
- Use **qsub -I** to log in the compute node in interactive mode

Step 3
Login

$ **qsub -l walltime=24:00:00 -I**

```
[u5662@c009 ~]$
```

```
[u5662@c009-n014 ~]$
```



INSTALL DEPENDENCIES

Step 4
Setup environment

- Virtual environment

$ **conda create -n** *<tf>* **-c intel** *pip numpy …*	# Create virtual environment (VE)
$ **source activate tf**	# Activate VE
(tf) **conda install -c intel** *tensorflow-mkl*	# Install Intel Tensorflow in VE
(tf) **pip install** *keras*	# Install Keras in VE
(tf) **source deactivate**	# Deactivate VE

- Local environment

$ **pip install --user** *tensorflow*	# Install Tensorflow in local
$ **pip install --user** *keras*	# Install Keras in local

UPLOAD SCRIPTS & DATA SET

Step 5
Upload

- Linux or Mac
  - Use **scp** command to transfer files → $ **scp -r** *<src>* **colfax:***<des>*

    E.g.: $ scp –r data/breeds colfax:~/data/
- Windows
  - Use WinSCP tool to transfer files

! Note that scripts and data set need **uploaded from local** instead of login node

DEMO 1

- Train & inference Cifar-10 via QSUB in interactive mode (Sources by Prof. Aven)

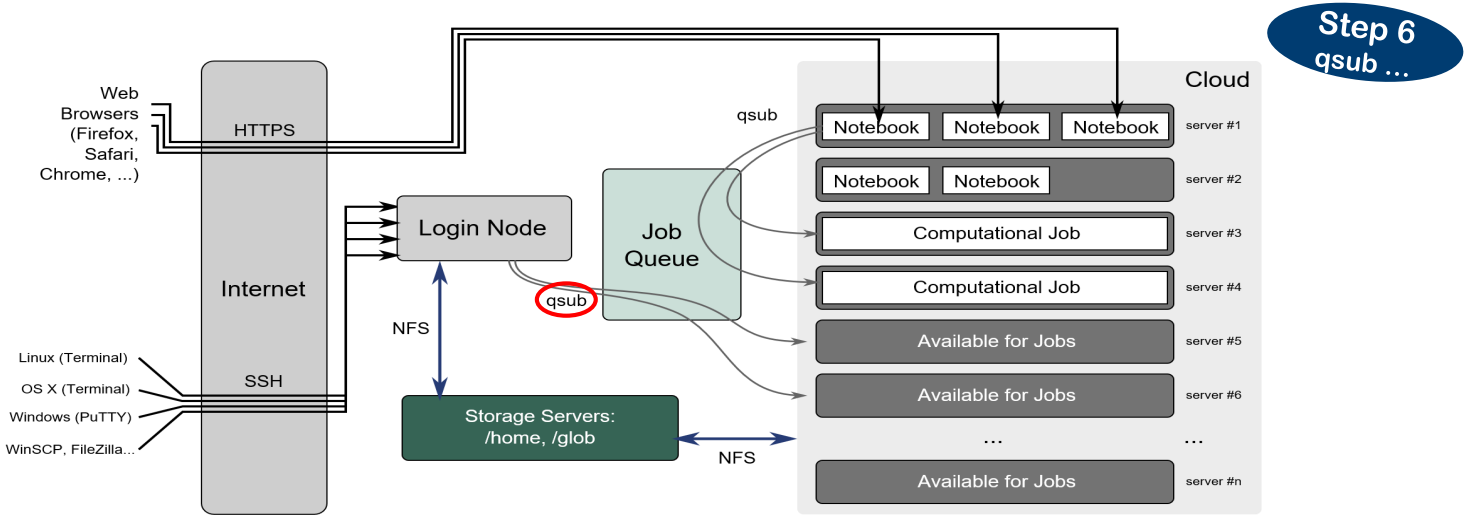
RUN A JOB VIA QSUB IN NORMAL MODE

- Use **qsub -l** *<param1>* **-l** *<param2> <script file>* to launch a queued job

$ **qsub -l nodes=***<number of required nodes>***:ppn=***<number of required CPU cores>* **-l walltime=***<xx:xx:xx>*

<script file> **-k oe** → **Output and error file**

Eg.: $ qsub -l nodes=1:ppn=1 -l walltime=24:00:00 main -k oe



RUN DEMO 2

- Prepare environment and submit a job

  $ sh prepare_env_tf

  $ sh qsub_tf

- Output files under **HOME** user path

Step 7
Final check!

<file qsub_tf>

qsub -l nodes=1:ppn=1
script_PetsHandsOnExcercise -k oe

$ sh qsub_tf

<file
script_PetsHandsOn
Excercise.o2019>

output file

Check:
$ **view** script_PetsHandsOnExcercise.o2019

<file
script_PetsHandsOn
Excercise.e2019>

error file

Check:
$ **view** script_PetsHandsOnExcercise.e2019

HOW TO TRACK STATUS...

- Use **qstat** to check status of submitted job

```
[u5644@c009 ~]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
551.c009                   my1stjob         u5644             00:00:00 R batch
552.c009                   my2ndjob         u5644             00:00:00 R batch
557.c009                   mylargejob       u5644                    0 Q batch
```

Q: Queued
R: Run
C: Completed*
...

DELETE JOBS

- Use **qdel <*job id*>** to remove submitted job

```
[u5644@c009 ~]$ qdel 549
```

- Use **qdel all** to cancel all running and queued jobs

```
[u5644@c009 ~]$ qdel all
```

TRAINING & TEST BREEDS WORKFLOW

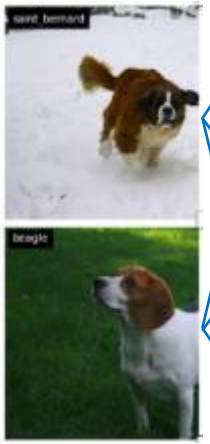
Problem Statement
- State an issue to resolve

Get Data
- Adopt dataset

Clean Data
- Clean, normalize, split, augment, etc....

Train
- 37 breeds — learn to tell them apart

Test
- Test local sample



British Shorthair

Wheaten Terrier

?

Picture source: Stanford ImageNetDogs dataset
http://vision.stanford.edu/aditya86/ImageNetDogs/main.html

TRAINING & TEST BREEDS WORKFLOW

Problem Statement
- State an issue to resolve

Get Data
- Adopt dataset

Clean Data
- Clean, normalize, split, augment, etc....

Train
- 37 breeds — learn to tell them apart

Test
- Test local sample

prepare_env_tf

```
cp /data/breeds ~/
...
```

TRAINING & TEST BREEDS WORKFLOW

Problem Statement
- State an issue to resolve

Get Data
- Adopt dataset

Clean Data
- Clean, normalize, split, augment, etc....

Train
- 37 breeds — learn to tell them apart

Test
- Test local sample

prepare_env_tf

clean_data.py

TRAINING & TEST BREEDS WORKFLOW

Problem Statement
- State an issue to resolve

Get Data
- Adopt dataset

Clean Data
- Clean, normalize, split, augment, etc….

Train
- 37 breeds — learn to tell them apart

Test
- Test local sample

script_PetsHandsOnExcercise

```
$ train_image_classifier.py
--train_dir=...
```

TRAINING & TEST BREEDS WORKFLOW

Problem Statement
- State an issue to resolve

Get Data
- Adopt dataset

Clean Data
- Clean, normalize, split, augment, etc….

Train
- 37 breeds — learn to tell them apart

Test
- Test local sample

script_PetsHandsOnExcercise

```
$ eval_image_classifier.py
--dataset_dir=…
```

DEVCLOUD FAQ

The bottom of here

THANKS!!
Q&A