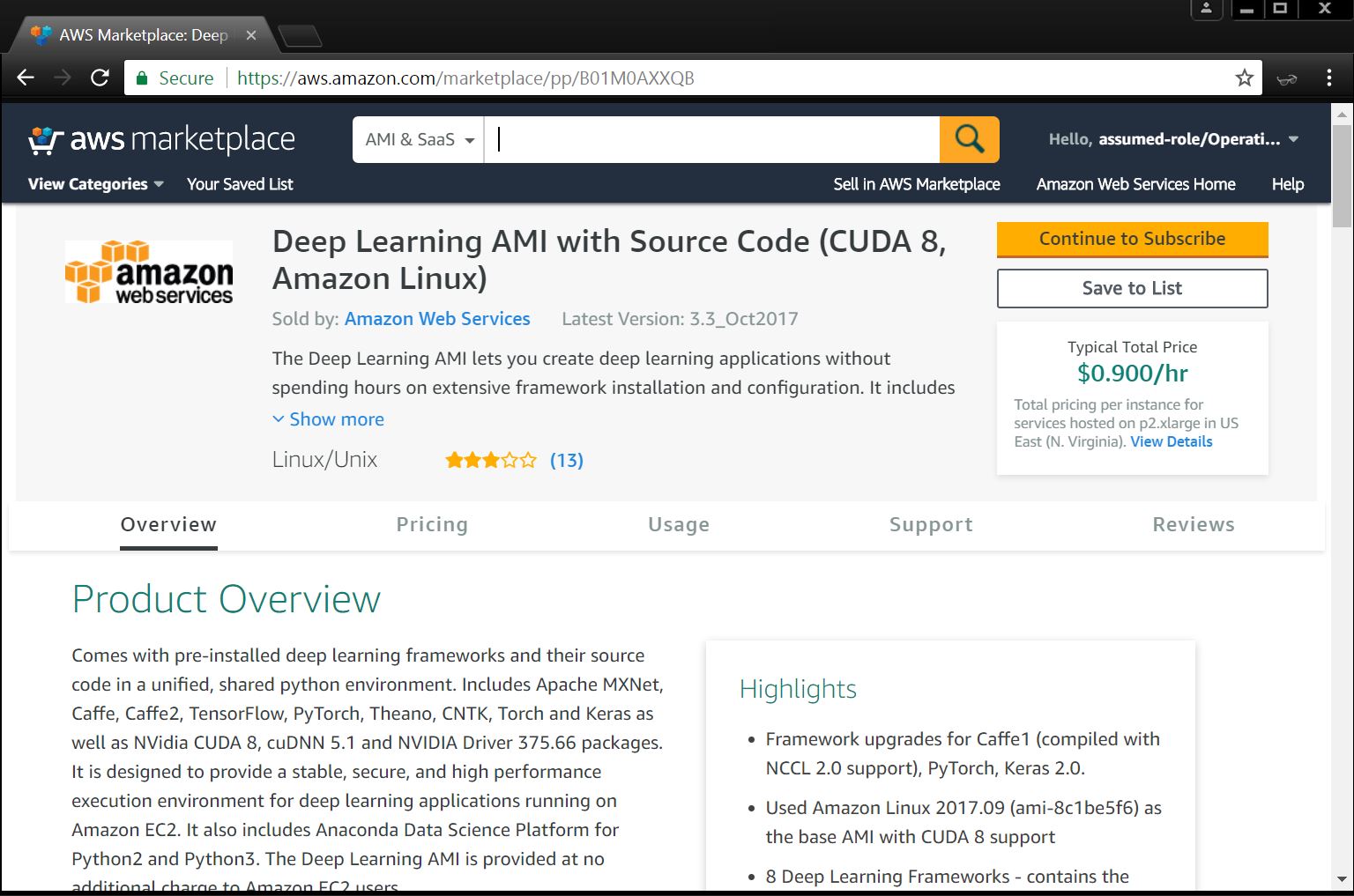
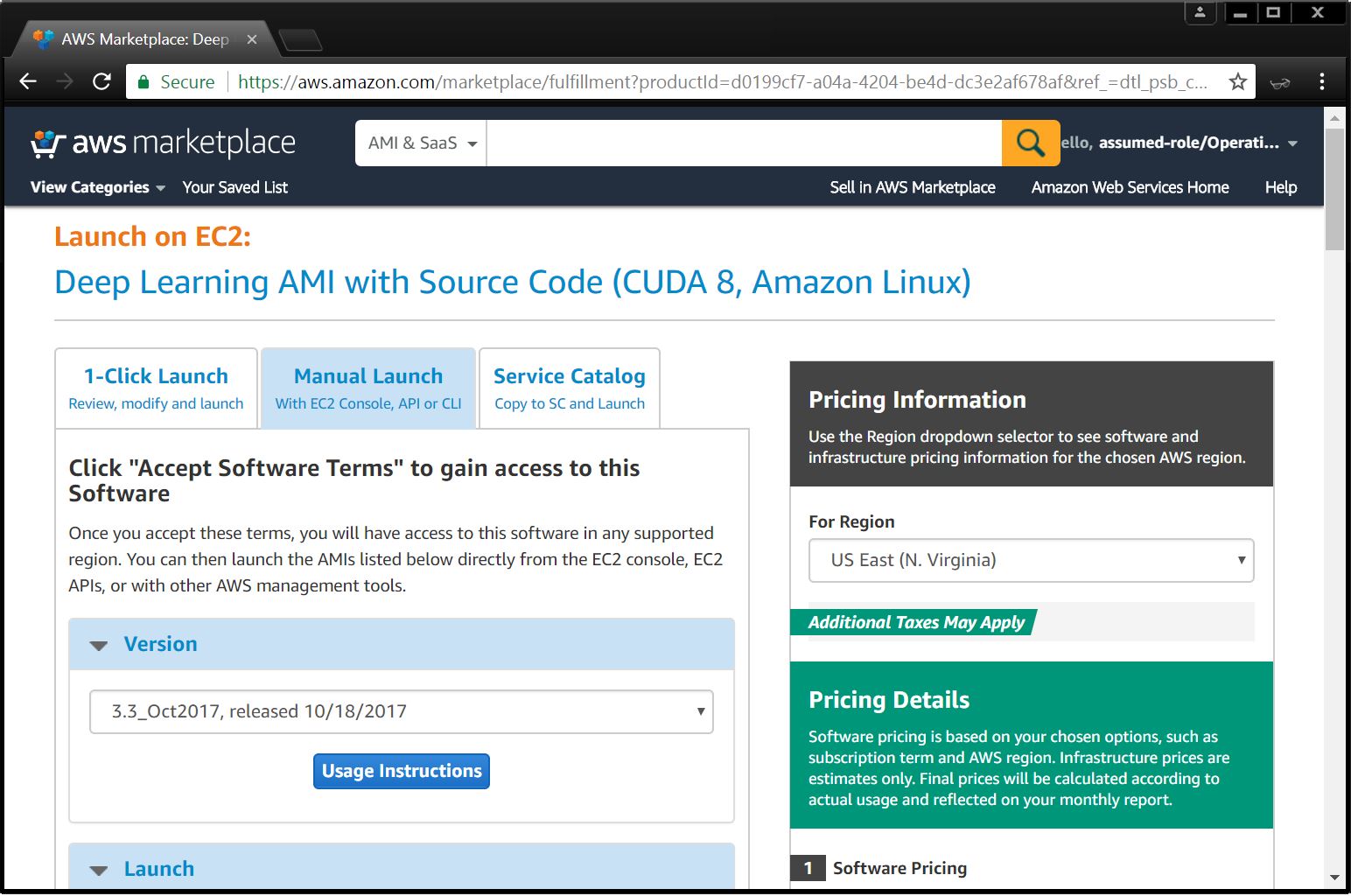
**Guide to Launch Intel Caffe Multi Nodes Training on AWS**

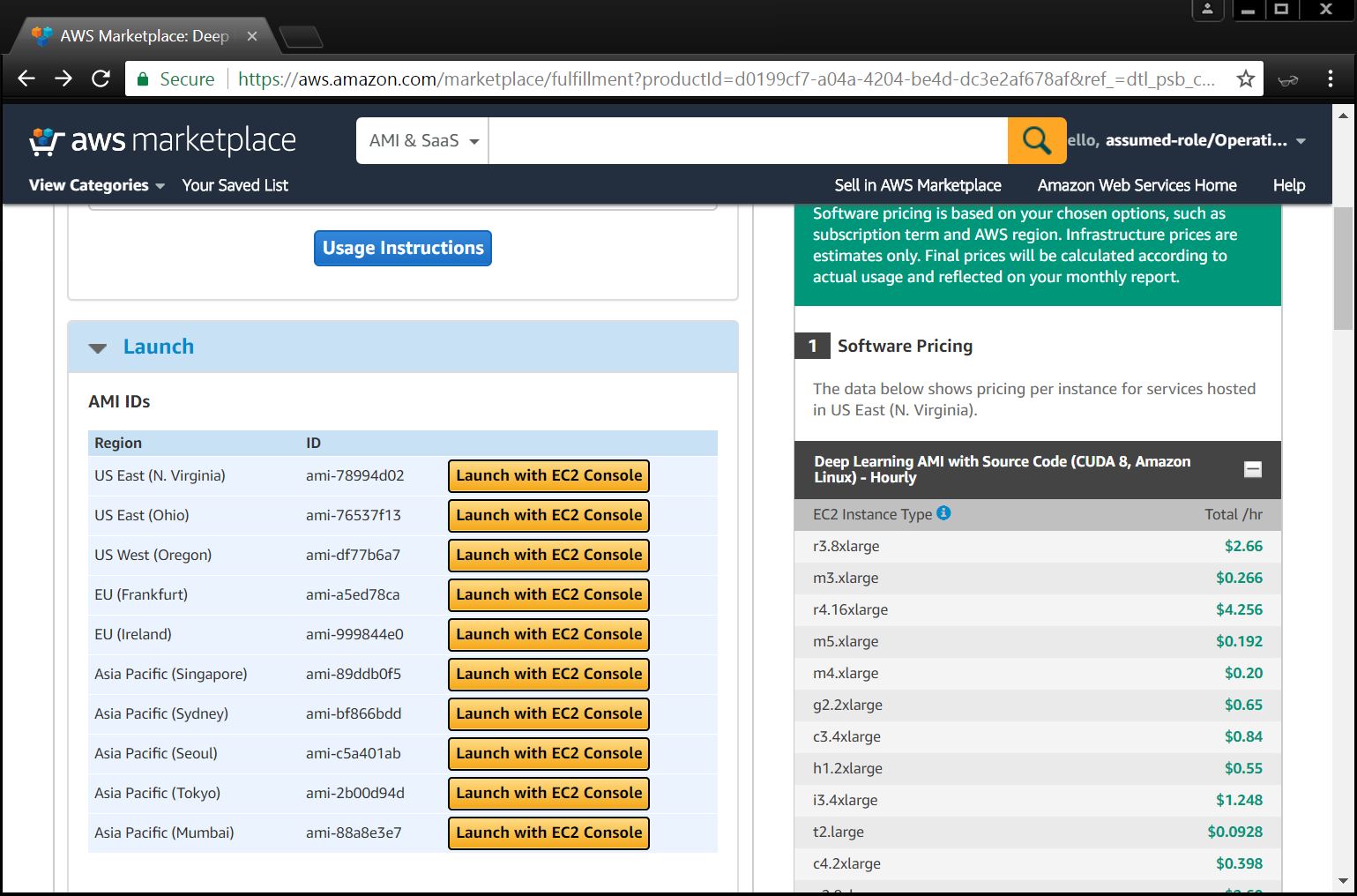
1. **Navigate to the** [**Deep Learning AMI**](https://aws.amazon.com/marketplace/pp/B01M0AXXQB) **offering in AWS marketplace and click “Continue to Subscribe”:** [**https://aws.amazon.com/marketplace/pp/B01M0AXXQB**](https://aws.amazon.com/marketplace/pp/B01M0AXXQB)



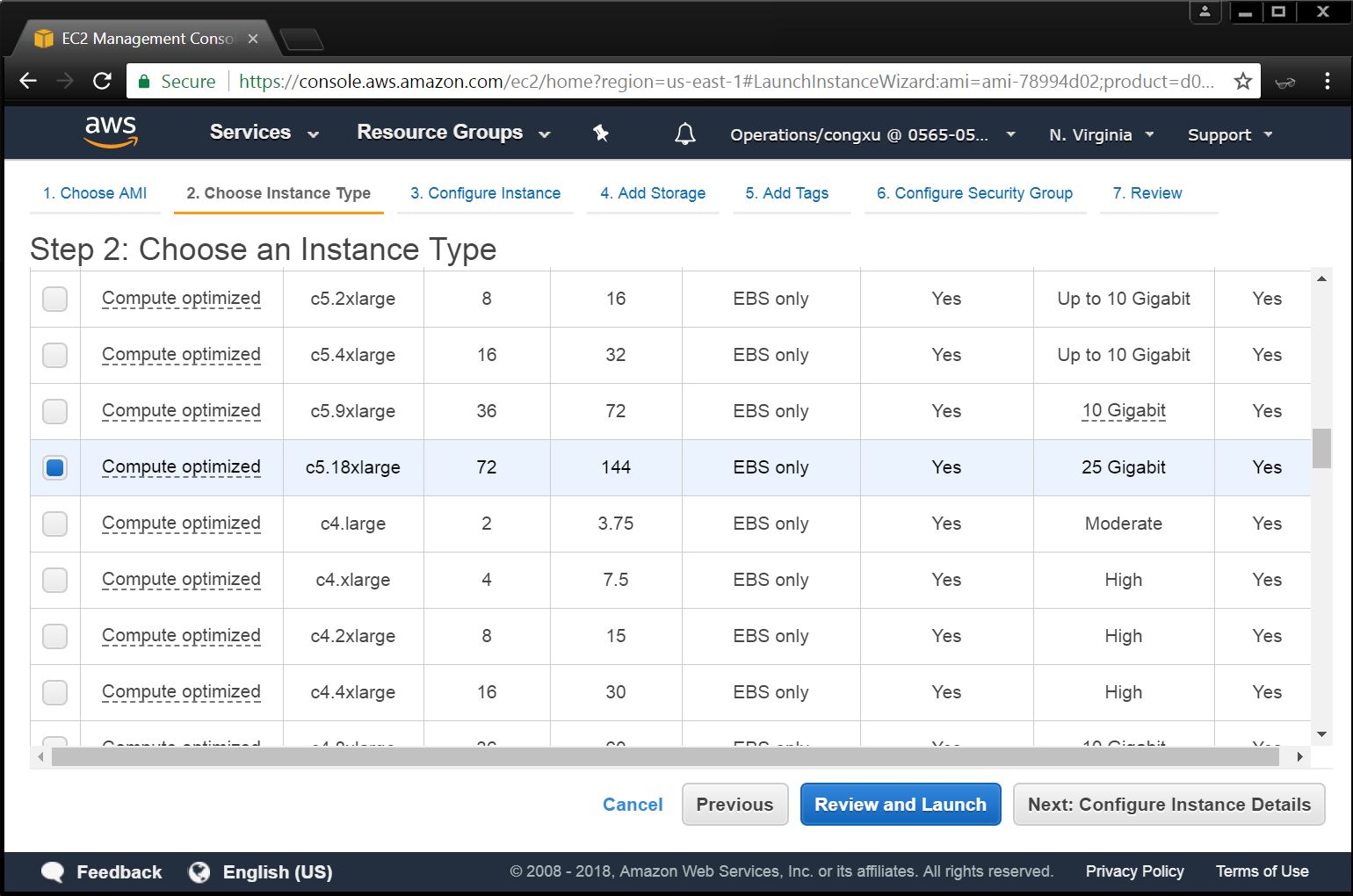
**2. One can either select 1-Click Launch or Manual Launch. Here we show steps by selecting “Manual Launch”. Select “Manual Launch” and select your desired region and click “Launch with EC2 Console”**



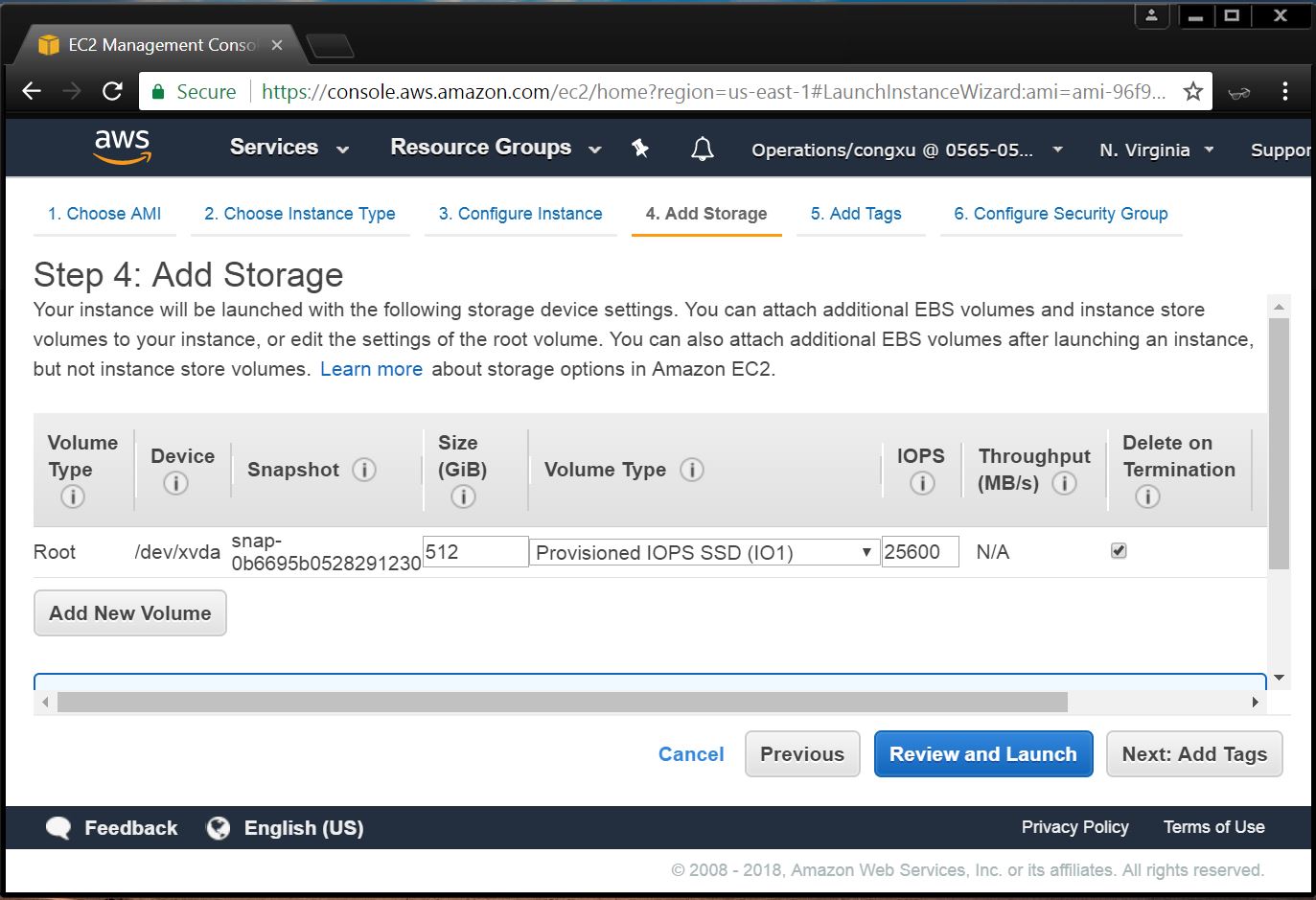
**3. Chose Region and Launch with EC2 Console**



**4. Select the C5.18xlarge instance to launch**

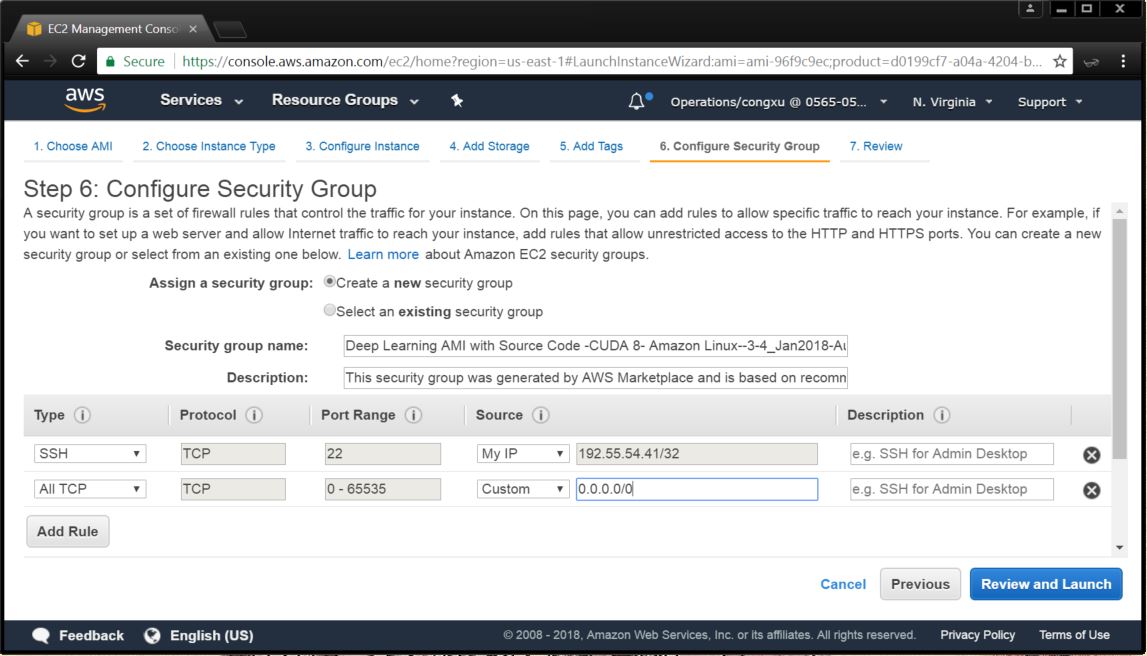


**5. Continue to configure the Instance, add storage and add tags.**

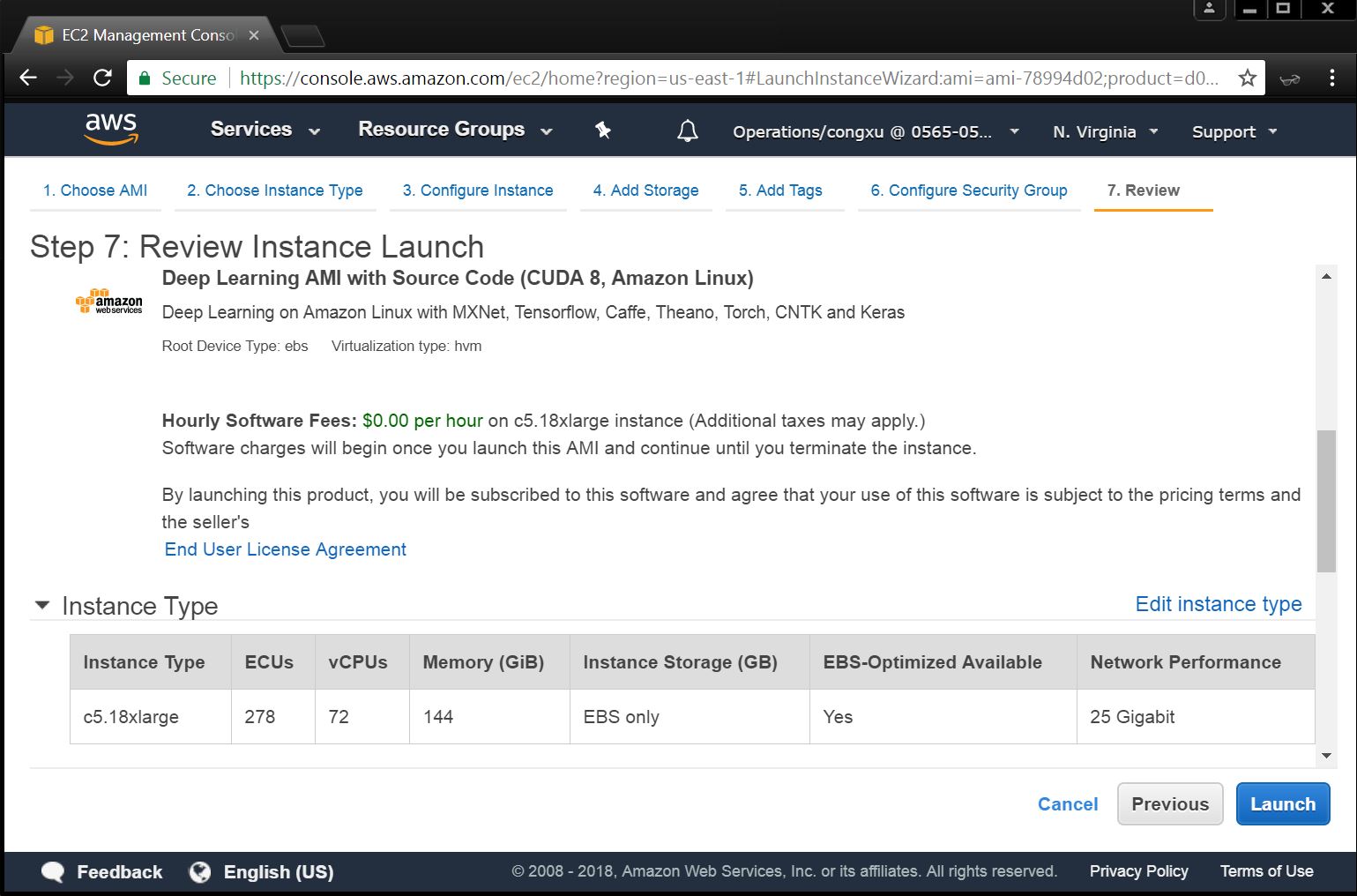
****

**6. Configure the Security Group**

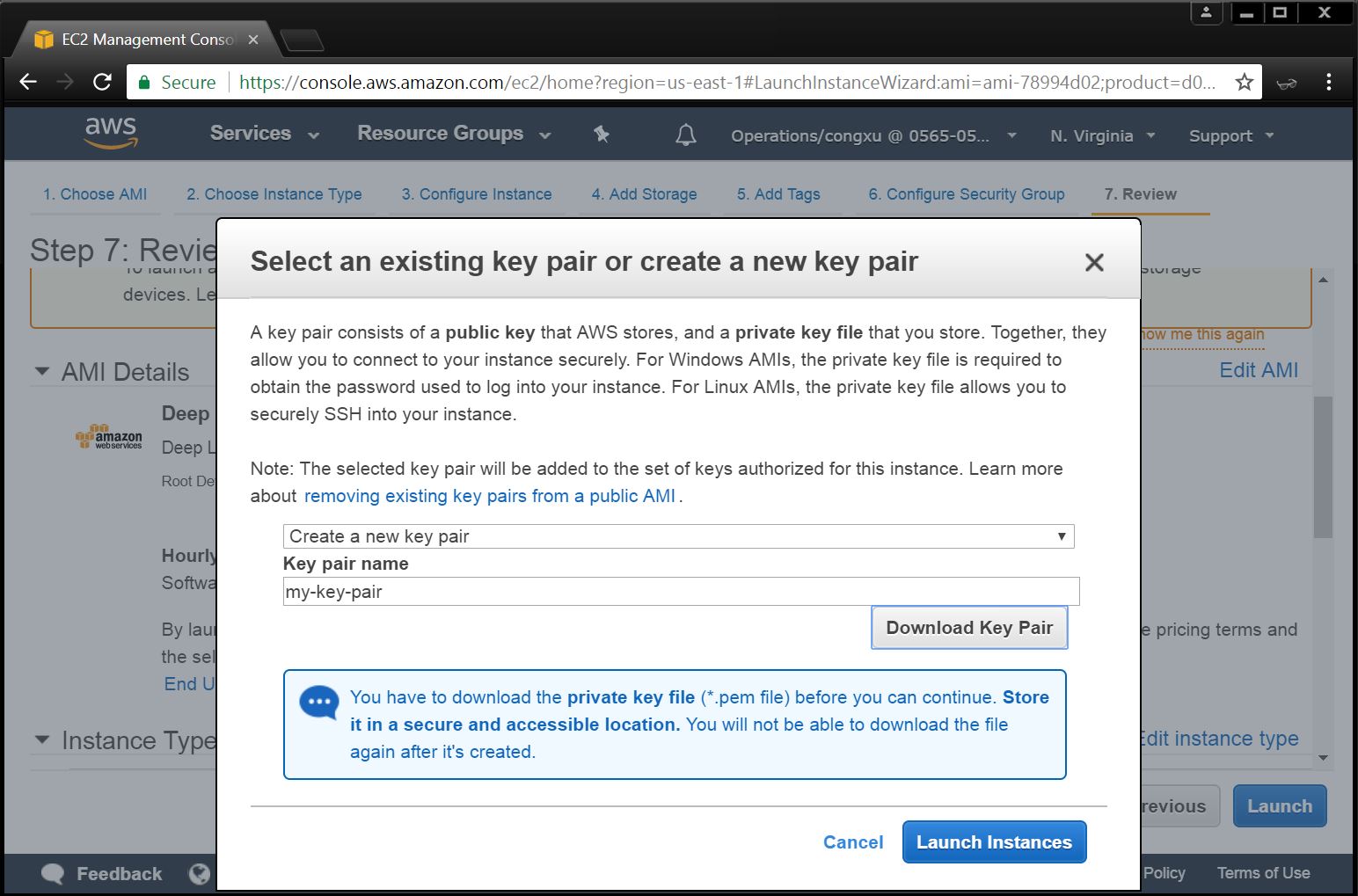
**Warning**: The configuration below should work, but may potentially cause security issues. To avoid those issues, change 0.0.0.0/0 to known IP addresses or Security Group.



**7. Review your instance, and Launch**

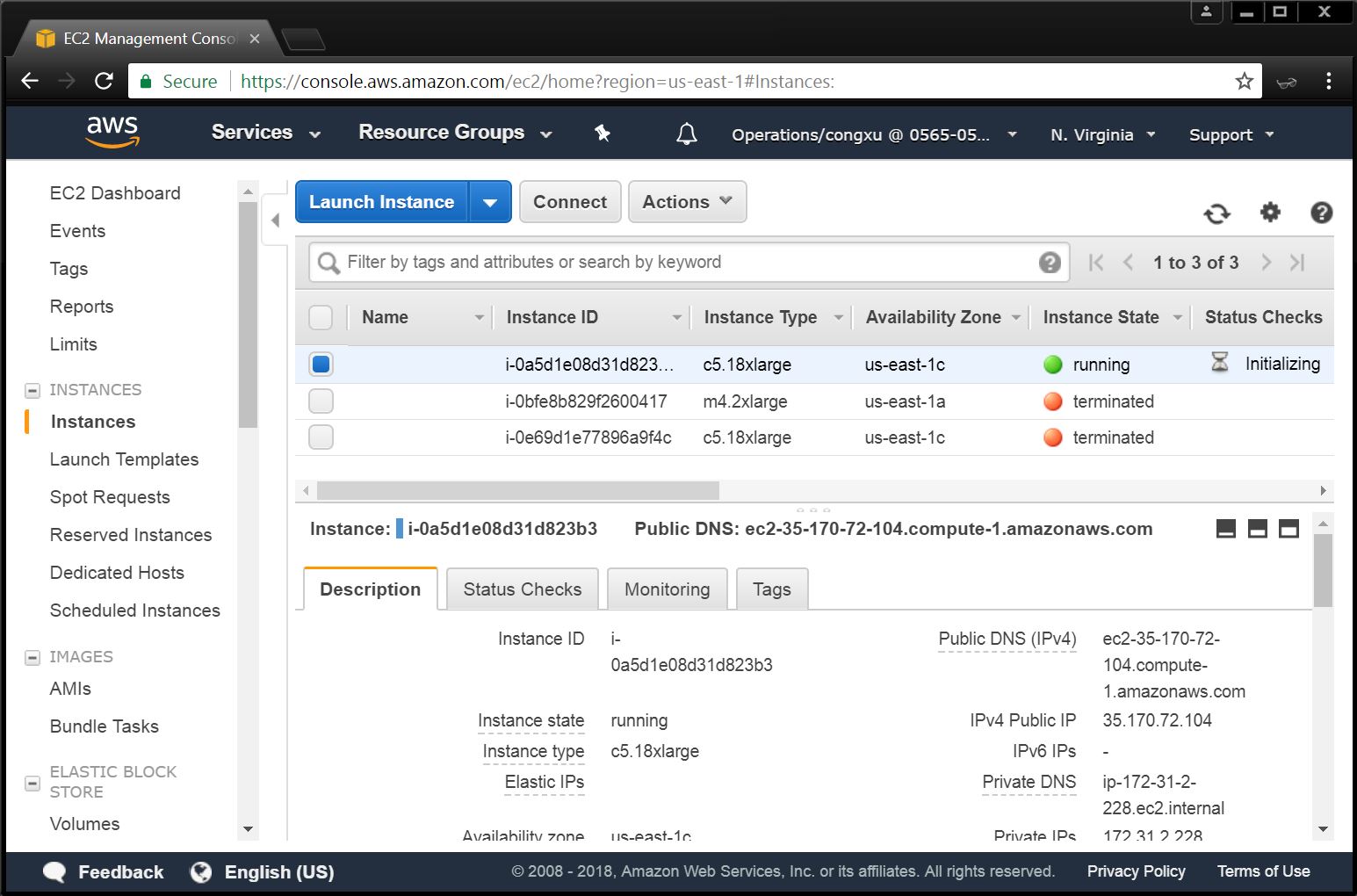


**8. Select or create a new private key file for secure access, then launch instances**



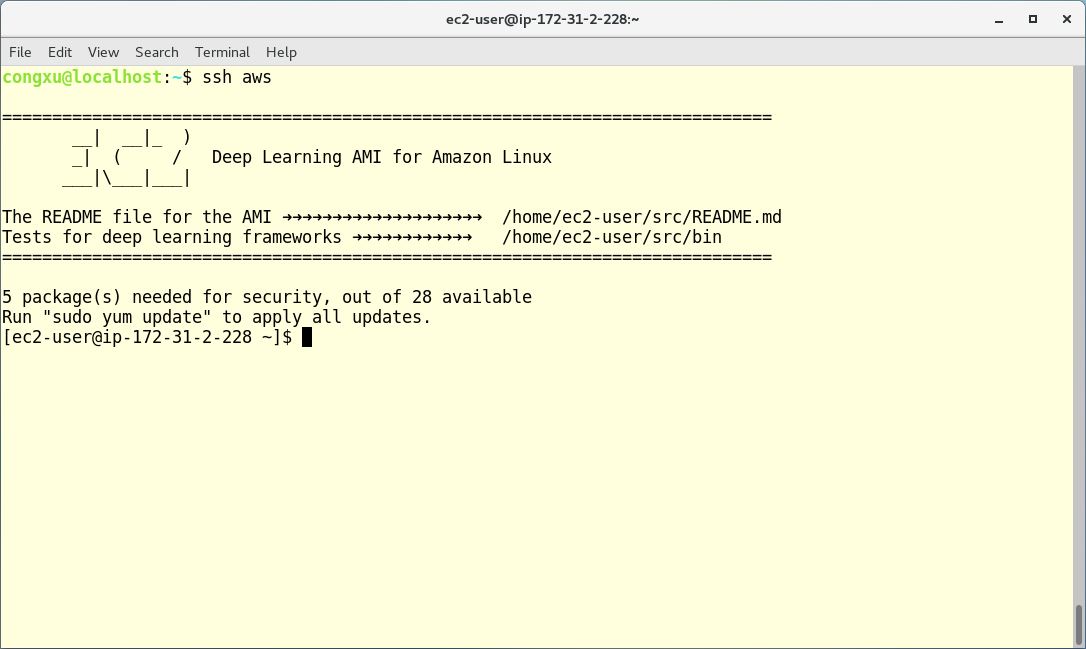
**Access Your Instance**

**1. Find your instance DNS**



**2. SSH into your instance**

Modify ~/.ssh/config file and connect to the instance via SSH



**Install Intel Caffe and Run Tests**

**1. Setting up Intel ICC environment**

**a. Download Intel Parallel Studio XE (Intel ICC and Intel MPI is included)**

(1) Go to following webpage

https://software.intel.com/en-us/parallel-studio-xe/choose-download#buynow

(2) Choose “Linux” in “Try It Now” Section

(3) Choose “Free Trial” in “Start Your Free Trial” Section

(4) Register and submit

You will receive an email containing instructions for how to download “Intel® Parallel Studio XE Cluster Edition for Linux\*”

(5) Get initial download package

Open your email, then “Download” -> “Customizable Package”

**b. Install Intel Parallel Studio XE**

(1) Upload Intel Parallel Studio XE installation package to Amazon DL AMI instance

(2) Login Amazon DL AMI instance

(3) Unzip Intel Parallel Studio XE tar file and install

# cd /PATH/TO/parallel\_studio\_xe\_online

# ./install.sh

\*Note: You will need “Serial Number” received in email to install the software

**c. Configure environment variables**

(1) Add following lines to ~/.bashrc

source /PATH/TO/intel/compilers\_and\_libraries/linux/bin/compilervars.sh intel64

export KMP\_HW\_SUBSET=1T

export KMP\_AFFINITY="granularity=fine,compact"

(2) To avoid MPI libraries conflicts, comment lines below in ~/.bashrc, and “unset LD\_PRELOAD”

# if [ -e /usr/local/mpi/lib/libmpi.so ]; then

# export LD\_PRELOAD=/usr/local/mpi/lib/libmpi.so

# fi

(3) Reload ~/.bashrc

# source ~/.bashrc

**2. Download and build Intel Caffe**

**a. Check out Intel Caffe**

# cd /home/ec2-user/src

# git clone <https://github.com/intel/caffe.git> intelcaffe

# cd /home/ec2-user/src/intelcaffe

# git checkout DAWN\_Bench

**b. Build Intel Caffe with Intel ICC**

# cd /home/ec2-user/src/intelcaffe

# ./scripts/build\_intelcaffe.sh

**3. Prepare for multi nodes training**

**a. Configure password less SSH**

(1) Generate ssh key and append public key to authorized\_keys

# ssh-keygen -t rsa

# cat /home/ec2-user/.ssh/id\_rsa.pub >> authorized\_keys

(2) Disable strict key checking

# sudo vim /etc/ssh/ssh\_config

Add “StrictHostKeyChecking no” to ssh\_config file

**b. Prepare ImageNet dataset for Intel Caffe**

Create ImageNet LMDB based on the instructions in <https://github.com/intel/caffe/wiki/How-to-create-ImageNet-LMDB>

\* Note: In our test ImageNet LMDB was created with encoding but without resizing, thus make sure setting “ENCODE=true” and “RESIZE=false” in create\_imagenet.sh script

**c. Prepare solver.prototxt and train\_val.prototxt**

Update the path to ImageNet LMDB datasets in train\_val.prototxt

**d. Save the AMI**

Right click the instance -> Image -> Create Image

**4. Run Intel Caffe Multi Nodes Training**

**a. Use AWS CLI to launch multi nodes on AWS**

* **resnet\_50\_64\_nodes\_2k\_batch:**

# aws ec2 run-instances --image-id <ami-id> --count 64 --instance-type c5.18xlarge --key-name <ssh-key> --security-group-ids <security-group-ids> --associate-public-ip-address --ebs-optimized --placement GroupName=<placement-group-name>,Tenancy=default

\* Note: <ami-id> is the image id we created in step 3.d

* **resnet\_56\_128\_nodes\_4k\_batch**

# aws ec2 run-instances --image-id <ami-id> --count 128 --instance-type c5.18xlarge --key-name <ssh-key> --security-group-ids <security-group-ids> --associate-public-ip-address --ebs-optimized --placement GroupName=<placement-group-name>,Tenancy=default

\* Note: <ami-id> is the image id we created in step 3.d

* **resnet\_50\_128\_nodes\_4k\_batch**

# aws ec2 run-instances --image-id <ami-id> --count 128 --instance-type c5.18xlarge --key-name <ssh-key> --security-group-ids <security-group-ids> --associate-public-ip-address --ebs-optimized --placement GroupName=<placement-group-name>,Tenancy=default

**b. Prepare the hostfile**

(1) Use AWS CLI to obtain host names

# aws ec2 describe-instances --output text --filters "Name=instance-state-name,Values=running" "Name=placement-group-name,Values=<placement-group-name>" --query Reservations[\*].Instances[\*].PrivateDnsName

(2) Put the returned host names in a temp file, then sort the host names

# sort -V temp > hostfile

(3) Upload hostfile to master node

**c. Login master node and run Intel Caffe multi node training**

# cd /home/ec2-user/src/intelcaffe

* **resnet\_50\_64\_nodes\_2k\_batch**

# ./scripts/run\_intelcaffe.sh --hostfile /home/ec2-user/hostfile --mode train --debug off --network tcp --netmask eth0 --num\_mlsl\_servers -1 --engine MKLDNN --num\_omp\_threads 0 --solver /home/ec2-user/src/intelcaffe/models/intel\_optimized\_models/multinode/resnet\_50\_64\_nodes\_2k\_batch/solver.prototxt --output intelcaffe\_workspace --benchmark none

* **resnet\_56\_128\_nodes\_4k\_batch**

./scripts/run\_intelcaffe.sh --hostfile /home/ec2-user/hostfile --mode train --debug off --network tcp --netmask eth0 --num\_mlsl\_servers -1 --engine MKLDNN --num\_omp\_threads 0 --solver models/intel\_optimized\_models/multinode/resnet\_56\_128\_nodes\_4k\_batch/solver\_resnet56.prototxt --output intelcaffe\_workspace --benchmark none

* **resnet\_50\_128\_nodes\_4k\_batch**

# ./scripts/run\_intelcaffe.sh --hostfile /home/ec2-user/hostfile --mode train --debug off --network tcp --netmask eth0 --num\_mlsl\_servers -1 --engine MKLDNN --num\_omp\_threads 0 --solver /home/ec2-user/src/intelcaffe/models/intel\_optimized\_models/multinode/resnet\_50\_128\_nodes\_4k\_batch/solver.prototxt --output intelcaffe\_workspace --benchmark none

**d. Evaluate the accuracy of the model**

# cd /home/ec2-user/src/intelcaffe

* **resnet\_50\_64\_nodes\_2k\_batch**

# ./build/tools/caffe test --model /home/ec2-user/src/intelcaffe/models/intel\_optimized\_models/multinode/resnet\_50\_64\_nodes\_2k\_batch/train\_val.prototxt --iterations <test\_iter same as solver> --weights <path to .caffemodel>

* **resnet\_56\_128\_nodes\_4k\_batch**

# ./build/tools/caffe test --model /home/ec2-user/src/intelcaffe/models/intel\_optimized\_models/multinode/resnet\_56\_128\_nodes\_4k\_batch/train\_val.prototxt --iterations <test\_iter same as solver> --weights <path to .caffemodel>

* **resnet\_50\_128\_nodes\_4k\_batch**

# ./build/tools/caffe test --model /home/ec2-user/src/intelcaffe/models/intel\_optimized\_models/multinode/resnet\_50\_128\_nodes\_4k\_batch/train\_val.prototxt --iterations <test\_iter same as solver> --weights <path to .caffemodel>