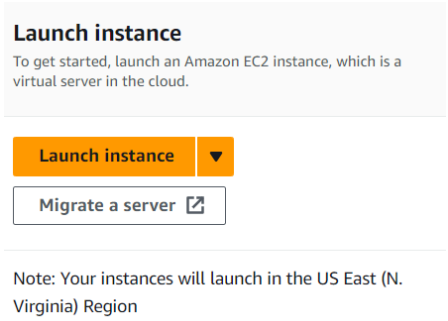
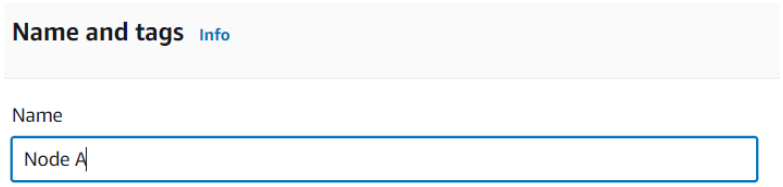
Provision and Configure Amazon Instances (Node A & Node B)

Node A:

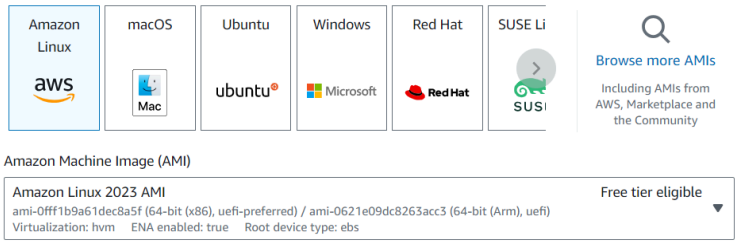
- Go to EC2 page on AWS console
- Click on “Launch Instance” button



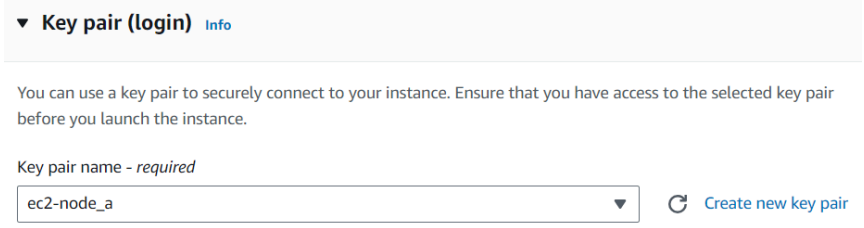
- Add “Name” - Node A



- Select “Amazon Linux 2023 AMI



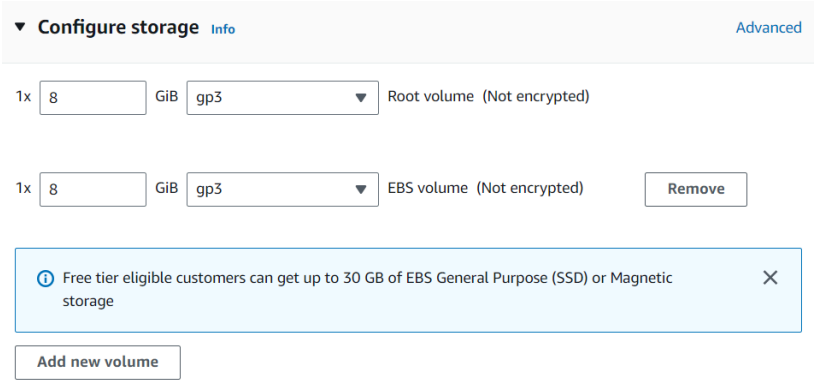
- Select or create a “Key Pair”



- Click on “Launch Instance”

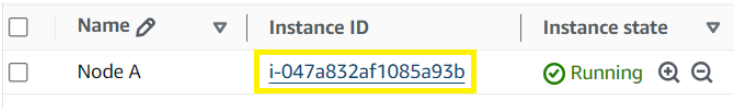
Node B:

- Follow the same process as “Node A” (with 1 exception)
- Under “Configure Storage” - click on “Add new volume” button

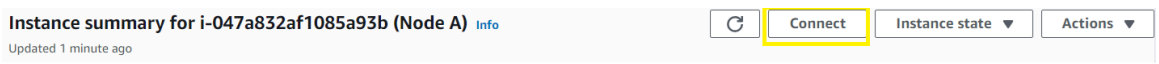


Connect to Node A and Node B via SSH

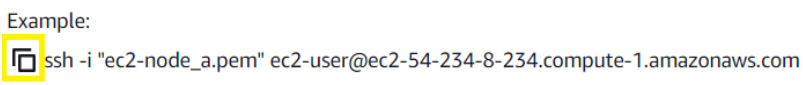
- Click on instance (Instance ID link)



- On upper right corner, click on “Connect” button



- Copy example “ssh” command and log into instance



- Follow the same process to log into “Node B”

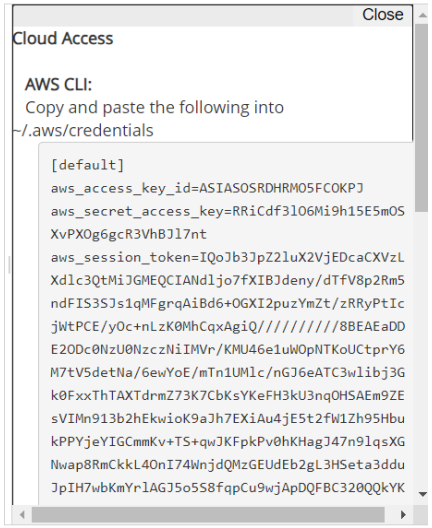
Configure security credentials for Node A & Node B

- Log into Node A & Node B and create credentials file

```
$ mkdir ~/.aws
$ cd ~/.aws
$ touch credentials
```

- In “AWS Academy Learner Lab” click on “AWS Details”

- Copy content of ~/.aws/credentials to credentials file on Node A & Node B



Install Software on Node A & Node B

- Install Java

```
// install java
sudo yum -y install java-17-amazon-corretto-devel
```

- Install Maven

```
// install maven
sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
sudo yum install -y apache-maven
mvn --version
```

- Install Git

```
// install git
sudo yum install git -y
git config --global user.name "Aryeh Golob"
git config --global user.email ag645@njit.edu
```

Configure EBS Mount Point on Node B

- Backup “fstab” file

```
sudo cp /etc/fstab /etc/fstab.orig
```

- Copy UUID for EBS volume (sudo blkid)

```
[ec2-user@ip-172-31-44-86 .aws]$ sudo blkid
/dev/xvda128: SEC_TYPE="msdos" UUID="A41B-D0D1" BLOCK_SIZE="512" TY
"
/dev/xvda127: PARTLABEL="BIOS Boot Partition" PARTUUID="d0854c70-9a
/dev/xvda1: LABEL="/" UUID="693eea79-11af-44b1-9c1e-01aced209966" B
```

- Open up fstab file (sudo vim /etc/fstab) and add /data directory mount entry

```
#
UUID=693eea79-11af-44b1-9c1e-01aced209966     /           xfs    defaults,noat
UUID=A41B-D0D1        /boot/efi       vfat    defaults,noatime,uid=0,gid=0,um
UUID=693eea79-11af-44b1-9c1e-01aced209966 /data xfs  defaults,nofail  0  2
```

- Grant “read & write” permission to “/data” directory

```
sudo chmod -R a+rw /data
```

- Reboot Node B

Clone Code From GIT Repository On Node A & Node B

- On both Node A & Node B

```
$ git clone https://github.com/qaz216/njitCloudComputing.git
```

Configure Both Node A & Node B From Config File

- Configuration file location

```
./cloud-app/src/main/resources/application.properties
```

- Config File Content

```
# application mode
#app.mode=car_recognition
app.mode=text_recognition

#bucket name
app.bucket=njit-cs-643

# sql queue info
app.queue.name=car-reco-queue.fifo
app.queue.group.id=image-processing

# processing delay for node A (car recognition)
app.car.recognition.delay=5

# result file
app.file.location=/data/result_file.txt
```

- Set Mode for Node A (Car Recognition)

```
# application mode
app.mode=car_recognition
#app.mode=text_recognition
```

- Set Mode for Node B (Text Recognition)

```
# application mode
#app.mode=car_recognition
app.mode=text_recognition
```

Compile Code On Node A & Node B

- Compile Using Maven

```
$ mvn compile
```

Run Application First on Node A then on Node B

- Use Maven to run application

```
$ mvn exec:java -Dexec.mainClass="com.njit.aryeh.RecognitionApp" -Dexec.cleanupDaemonThreads=false
```

- Verify console output for Node A

```
mode = car_recognition
running car recognition node ...
Creating queue: car-reco-queue1.fifo
image name: 1.jpg - label: car - confidence: 99.94897
image name: 2.jpg - label: car - confidence: 99.703125
image name: 4.jpg - label: car - confidence: 99.48165
image name: 5.jpg - label: car - confidence: 99.52181
image name: 6.jpg - label: car - confidence: 98.753235
image name: 7.jpg - label: car - confidence: 99.999916
sending -1
```

- Verify console output for Node B

```
mode = text_recognition
running text recognition node ...
Text detected for image: 1.jpg - text: $ BR8167 - confidence: 91.88689
Text detected for image: 1.jpg - text: $ - confidence: 93.11517
Text detected for image: 1.jpg - text: BR8167 - confidence: 90.65859
Text detected for image: 4.jpg - text: YHI9 OTZ - confidence: 99.27241
Text detected for image: 4.jpg - text: YHI9 - confidence: 99.11218
Text detected for image: 4.jpg - text: OTZ - confidence: 99.43263
Text detected for image: 7.jpg - text: Lamborghini - confidence: 97.241585
Text detected for image: 7.jpg - text: LP 610 LB - confidence: 95.650894
Text detected for image: 7.jpg - text: BO - confidence: 76.901344
Text detected for image: 7.jpg - text: BWW - confidence: 17.247715
Text detected for image: 7.jpg - text: Lamborghini - confidence: 97.241585
Text detected for image: 7.jpg - text: LP - confidence: 99.60253
Text detected for image: 7.jpg - text: 610 LB - confidence: 91.69925
Text detected for image: 7.jpg - text: BO - confidence: 76.901344
Text detected for image: 7.jpg - text: BWW - confidence: 17.247715
-1 received ... exiting
```

- Verify data written to file on EBS partition (/data/result_file.txt)

```
[ec2-user@ip-172-31-44-86 cloud-app]$ cat /data/result_file.txt
1.jpg - text: $ BR8167 - confidence: 91.88689
1.jpg - text: $ - confidence: 93.11517
1.jpg - text: BR8167 - confidence: 90.65859
4.jpg - text: YHI9 OTZ - confidence: 99.27241
4.jpg - text: YHI9 - confidence: 99.11218
4.jpg - text: OTZ - confidence: 99.43263
7.jpg - text: Lamborghini - confidence: 97.241585
7.jpg - text: LP 610 LB - confidence: 95.650894
7.jpg - text: BO - confidence: 76.901344
7.jpg - text: BWW - confidence: 17.247715
7.jpg - text: Lamborghini - confidence: 97.241585
7.jpg - text: LP - confidence: 99.60253
7.jpg - text: 610 LB - confidence: 91.69925
7.jpg - text: BO - confidence: 76.901344
7.jpg - text: BWW - confidence: 17.247715
```