CS 470 Amazon AWS Activity

Part I: Setup

1. Log into Amazon AWS (aws.amazon.com).

2. Go to the "Credits" section in account information and verify that you have credit remaining so that you are not charged for any compute time during this exercise. Alternatively, you can stick to the "t2.micro" machine type, which is free for up to 750 hours of compute time if your account is less than 12 months old.

3. Make sure you're in the N. Virginia region.

4. Create a new key pair and download the .pem file. If it downloads as .pem.txt, rename it to just .pem.

Part II: Single Instance – Apache Web Server

1. Start a single EC2 instance:

AMI: **Amazon Linux**
Machine type: **any**
Security: enable **HTTP** from **anywhere** (make sure you leave SSH enabled as well; do not replace it)
Key pair: **use existing**

Hint: Don't click "Review and Launch" until you set up the HTTP firewall rule; the key pair option is afterward.
Hint: To remove the security warning, set the SSH access to only your IP address (add a "/32" CIDR denotation).

2. Change instance name to "apache".

TO SUBMIT: Take a screenshot of the instance running in the dashboard, with your name as the account owner and the instance's public DNS clearly visible.

3. Find instance DNS in dashboard and SSH into the instance:

```
$ ssh -i <pem-file> ec2-user@<instance-dns>
```

4. Install Apache and set up a "www" group:

```
$ sudo yum install -y httpd24
$ sudo service httpd start
$ sudo groupadd www
$ sudo usermod -a -G www ec2-user
```

5. Log out and back in. Fix permissions for Apache (it's probably easiest to just copy/paste these commands):

```
$ sudo chown -R root:www /var/www
$ sudo chmod 2775 /var/www
$ find /var/www -type d -exec sudo chmod 2775 {} \;
$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

6. Edit index.html in /var/www/html. Browse to public DNS to view.

TO SUBMIT: Add your eID to the website and take a screenshot of it in the browser with the public DNS name (same as the previous screenshot) clearly visible in the browser navigation field.

7. Feel free to experiment further. Terminate the instance when you are done with it.

Part III: Multiple Instances – Virtual 4-Node Cluster

1. Start four EC2 instances:

AMI: **Amazon Linux**
Machine type: **any**
Number of instances: **4**
Security: enable **all TCP** traffic from **anywhere** (make sure you leave SSH enabled as well; do not replace it)
Key pair: **use existing**

2. Change instance names to "node1" through "node4".

TO SUBMIT: Take a screenshot of the instances running in the dashboard, with your name and the private IP of "node1" clearly visible.

3. Find the public instance DNS for node 1 and SSH into the instance:

```
$ ssh -i <pem-file> ec2-user@<instance-dns>
```

4. Edit /etc/hosts using superuser permissions, and add node2, node3, and node4 with the public IPs from the dashboard:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
<x.x.x.x> node2
<x.x.x.x> node3
<x.x.x.x> node4
```

5. Log out of instance.

6. Copy .pem file from local system to /home/ec2-user/.ssh/id_rsa on node1:

```
$ scp -i <pem-file> <pem-file> ec2-user@<instance-dns>:~/.ssh/id_rsa
```

7. Log back into node1 instance and fix permissions:

```
$ chmod 400 ~/.ssh/id_rsa
```

8. Download "run" and "copy" scripts, make them both executable, and test the run script:

```
$ wget w3.cs.jmu.edu/lam2mo/files/run.sh
$ wget w3.cs.jmu.edu/lam2mo/files/copy.sh
$ chmod +x run.sh copy.sh
$ ./run.sh hostname
```

The first time you run the last command, you should be prompted to add each node to the list of known hosts. Verify that this worked by running that last command a second time. You should not be prompted again.

Take a minute to look over run.sh and copy.sh. These control scripts will help you keep all of the nodes in sync with respect to the commands you run and the files you are using. On our cluster, all nodes mount your home folder from a NFS server, but that is not the case in your "mini-cluster" on AWS.

Part IV: Multiple Instances – MPI Cluster

1. Make sure you're still logged into the master node from Part III. From there, install MPI on all nodes:

```
$ ./run.sh sudo yum install -y openmpi-devel
```

2. Load MPI in your environment by adding the following to ~/.bashrc and copying it to all nodes:

```
export PATH=/usr/lib64/openmpi/bin:$PATH
export LD_LIBRARY_PATH=/usr/lib64/openmpi/lib

$ ./copy.sh ~/.bashrc
```

3. Log out and back in, or "source ~/.bashrc". Check to make sure MPI is available using "which mpicc".

4. Create ~/hostfile with the following node list:

```
localhost
node2
node3
node4
```

5. Download hello world program, compile it, and copy the resulting executable to all nodes:

```
$ wget w3.cs.jmu.edu/lam2mo/files/mpi_hello.c
$ mpicc -o hello mpi_hello.c
$ ./copy.sh ~/hello
```

6. Run the program. Note the inclusion of the hostfile parameter, which was not necessary on our cluster because SLURM managed specific node allocations. What happens when you increase the number of MPI processes to more than four?

```
$ mpirun -np 4 -hostfile ~/hostfile ~/hello
```

TO SUBMIT: Take a screenshot of the resulting execution, with the above mpirun command and all output (including hostnames) clearly visible.

7. Feel free to experiment further. Make sure you terminate all four instances when you are done with them.

BONUS: Get your P2 solution running on this virtual 4-node "mini-cluster." How does the performance and scaling behavior compare with our JMU cluster? Try building a larger cluster or use stronger nodes ("t2.2xlarge" or "m4.2xlarge" are probably the closest to the configuration of the JMU compute nodes).

2nd BONUS: Set up and use the AWS command-line interface to be able to control your instances from the terminal. Here is the website with instructions: `https://aws.amazon.com/cli/`

3rd BONUS: Set up and use Kubernetes, which is a tool for automatically deploying and managing clusters on AWS. Here is the website with instructions: `https://kubernetes.io/docs/getting-started-guides/kops/`

Submission

Submit all four screenshots on Canvas as .png files by the due date. Submit them as individual files; do not create a zip file. If you wish to consolidate related screenshots into a single image before submitting you may do so. If you wish to submit evidence of completing the bonus objectives, send me an email with the subject line "CS 470 AWS Bonus" and include any relevant screenshots or log dumps.