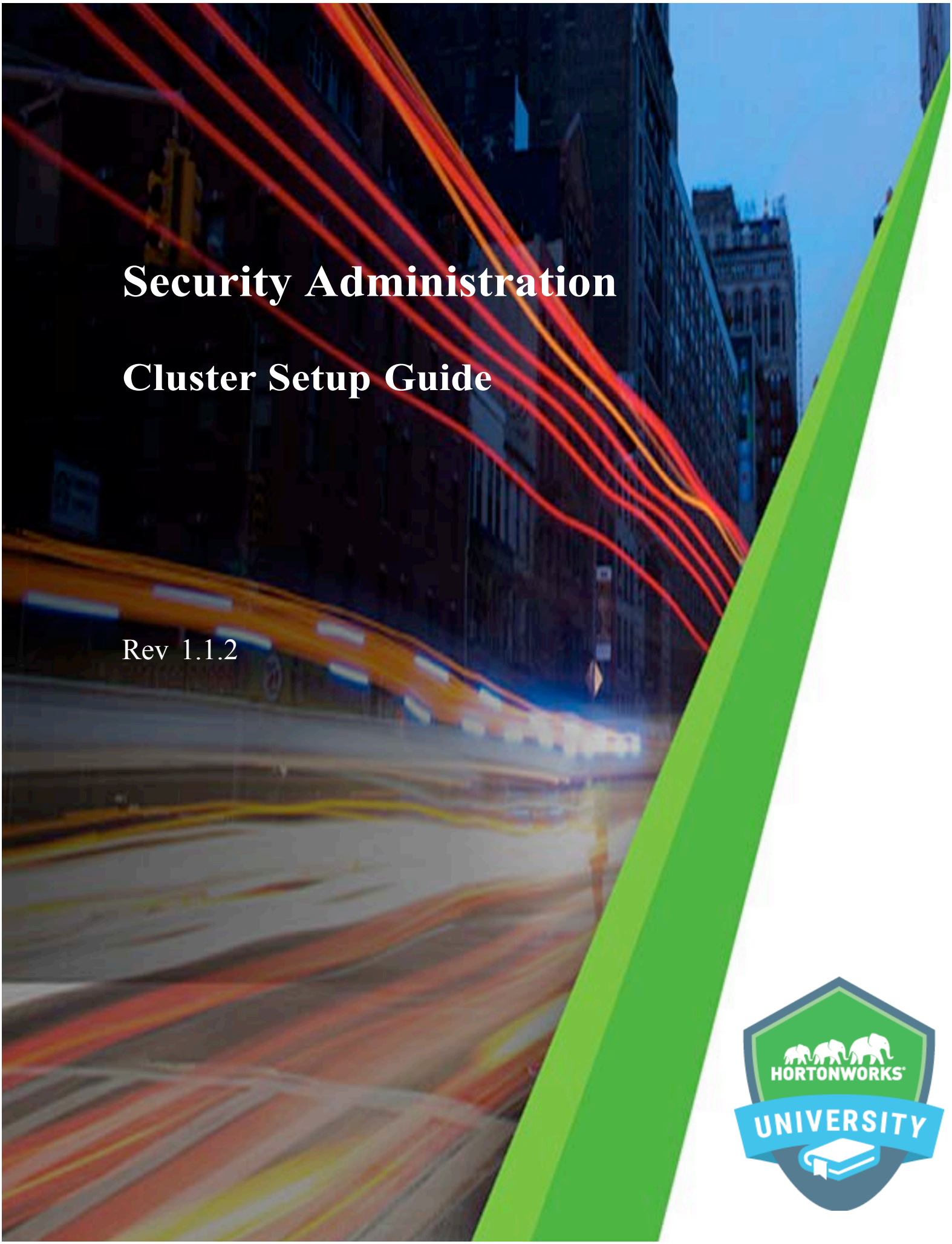

Security Administration
Cluster Setup Guide
Rev 1.1.2
HORTONWORKS
UNIVERSITY

Create the Security Clusters in AWS

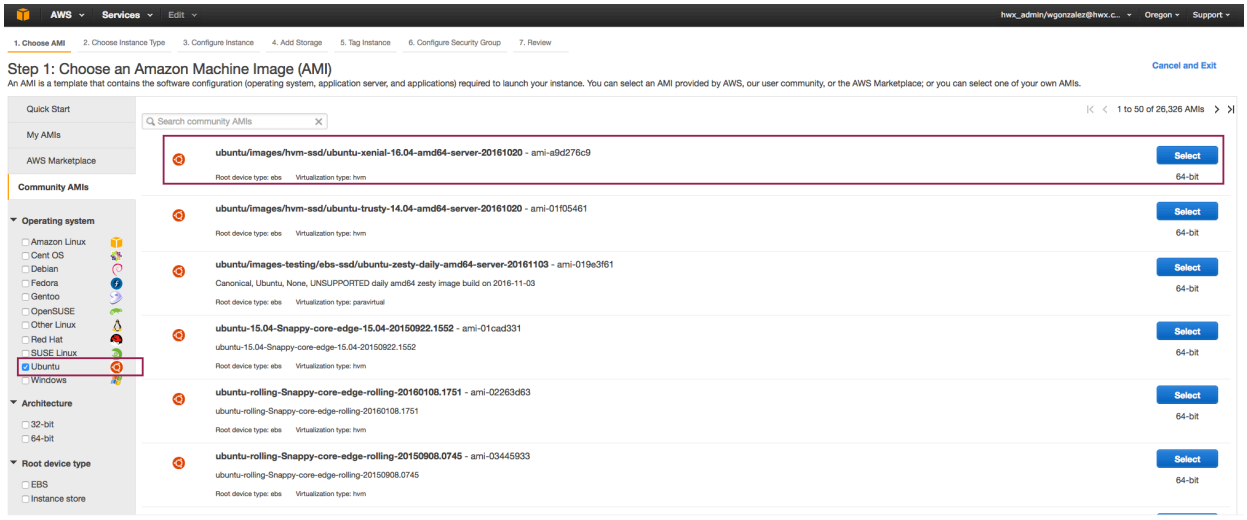
Objective:	This is an instruction guide to setup training Security Clusters
Before You Begin:	• You need to have an AWS account to launch clusters. • You need to have access to the AWS EC2 console • You need to ask Hortonworks to share the Windows Server AD Image with you (please contact hwuniversity@hortonworks.com) • You need to have the following ports open on your AWS firewall: 22, 80, 8080, 8443, 8444, 6080 & 6083.

Step 1: Prepare an AWS CLI Server to Deploy AWS Clusters

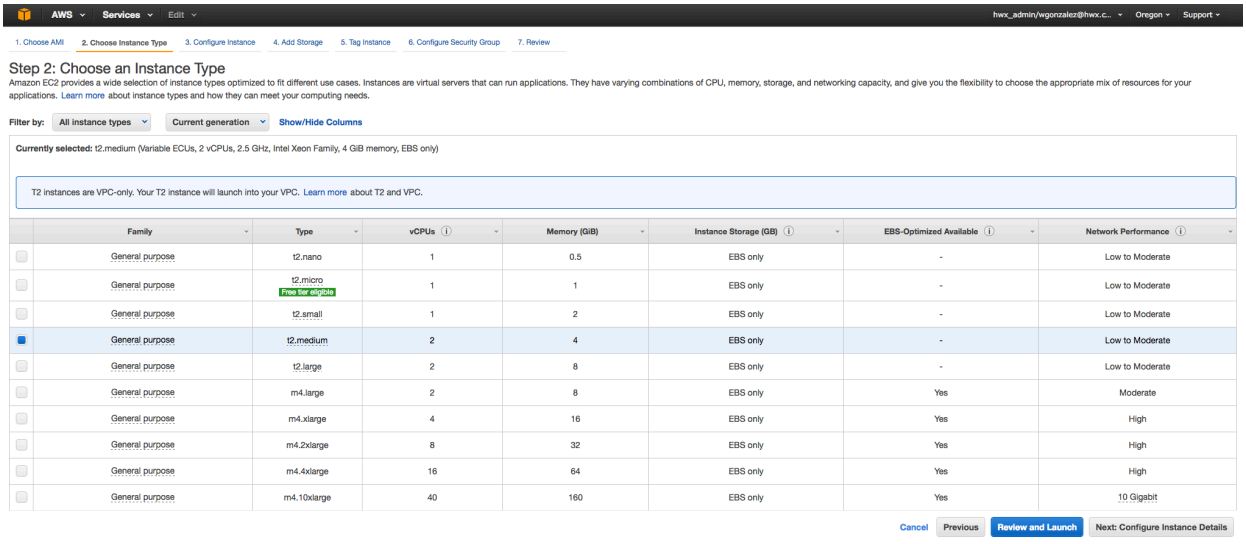
1.1. If you already have a server that has AWS CLI installed, you can jump to Step 2.

1.2. Launch an Ubuntu AWS image. This step can also be performed on other OS, although we are only giving Ubuntu as an example.

1.2.1. From your AWS console, choose an Ubuntu AMI in your zone similar to: Ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20161020:



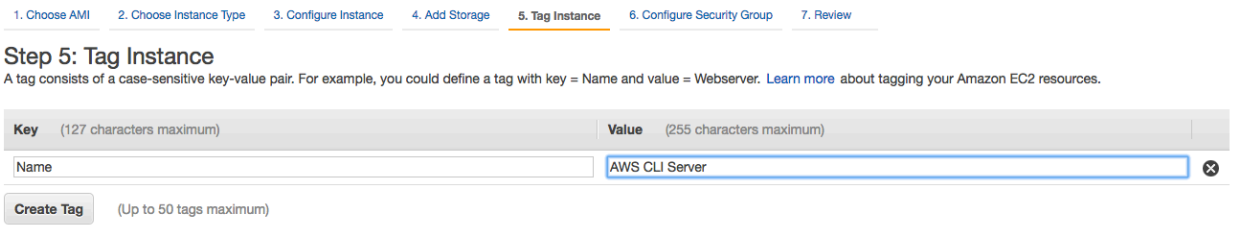
1.2.2. For the instance type, select t2.medium:



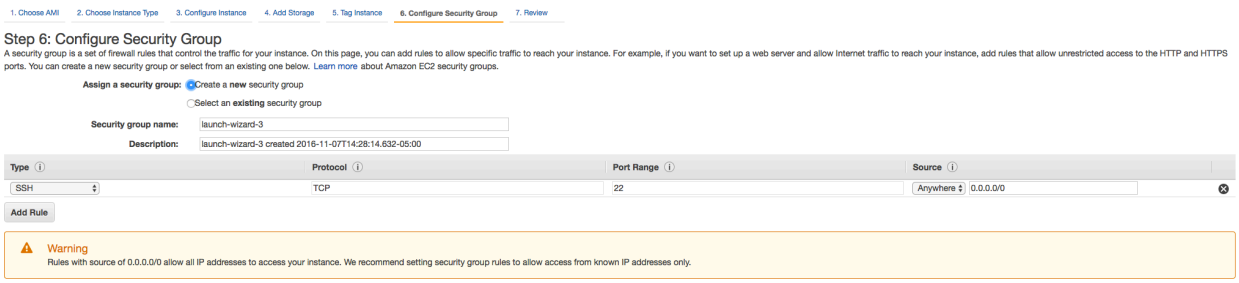
1.2.3. On the ‘Step 3 - Configure Instance’ steps you can take most of the defaults, just make sure you enable ‘Auto-assign Public IP’

1.2.4. Take defaults on ‘Step 4 – Add storage’

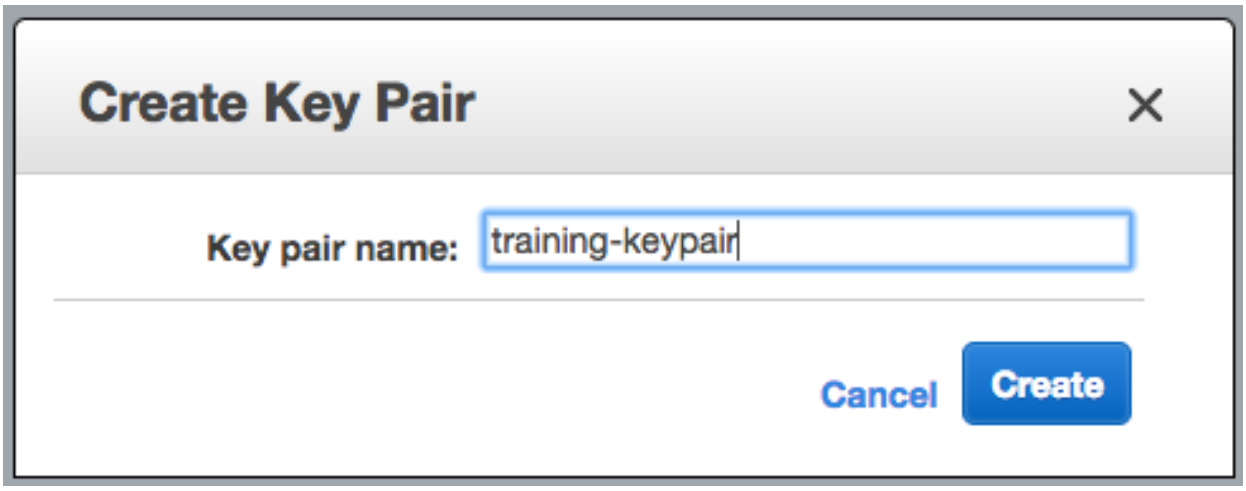
1.2.5. Name your instance appropriately on ‘Step 5 – Tag Instance’:



1.2.6. On ‘Step 6 – Configure Security Group’ create a new group with SSH access, then review and Launch:



1.2.7. Create a NEW key-pair for accessing your server. Name your keypair "training-keypair." The new keypair will be downloaded locally.



1.3. Install AWS CLI on the newly provisioned AWS CLI Server

1.3.1. After the instance finishes launching (check 'Instance State' and 'Status check' columns on the EC2 Console), try to ssh to the instance using a terminal window (if connecting from MacOS) or puTTY (if connecting from Windows). For instance from a MacOSX terminal, type (substitute < … > for the actual values):

```
$ ssh -i /<directory where key-pair was installed>/training-
keypair.pem ubuntu@<instance IP>
```

1.3.2. Install the AWS CLI as described here: http://docs.aws.amazon.com/cli/latest/userguide/installing.html#install-bundle-other-os

1.3.4. After installing the AWS CLI, make sure the ‘aws’ command works:

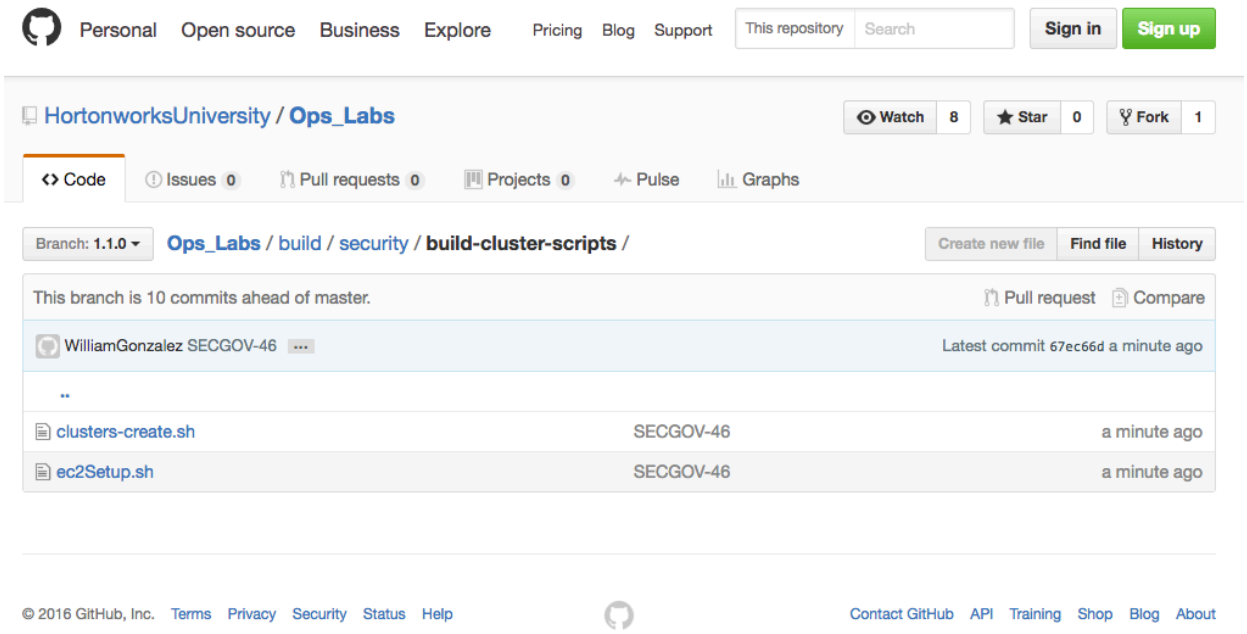
```
$ aws
usage: aws [options] <command> <subcommand> [<subcommand> ...]
[parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: the following arguments are required: command
$
```

Step 2: Download and Install Security Cluster Create scripts into AWS CLI Server

2.1. Download scripts from the relevant GitHub branch for this guide.

2.1.1. For Security 1.1.2, please see https://github.com/HortonworksUniversity/Ops_Labs/tree/1.1.2/build/security/build-cluster-scripts

Download the 2 files found in that directory:



2.2. Upload the 3 files to the AWS CLI Server (saved here to /home/ubuntu):

2.2.1 Copy or Move the training-keypair.pem to the same directory where the other 2 files from GitHub were saved.

2.2.2 If you are using a Windows machine to connect to the AWS instance:

```
$ c:\>"c:\Program Files (x86)\PuTTY\pscp.exe" -l ubuntu scp
c:\Users\myUserName\Documents\*.* <IP>:
```

2.2.3 If you are using a MacOS machine to connect to the AWS instance:

```
$ scp -i ./NEW-AWS-CLI-Server-keypair.pem <directory where you
downloaded github script files>/* ubuntu@<IP>:~
```

2.3 Login to the AWS CLI Server again and ensure the 3 files got copied to AWS CLI Server appropriately (this includes the keypair not shown in screenshot below):

```
$ $ ls -al /home/ubuntu/
total 44
drwxr-xr-x 4 ubuntu ubuntu 4096 Nov  7 20:59 .
drwxr-xr-x 3 root   root   4096 Nov  7 19:34 ..
-rw------- 1 ubuntu ubuntu   34 Nov  7 20:57 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Aug 31  2015 .bashrc
drwx------ 2 ubuntu ubuntu 4096 Nov  7 20:00 .cache
-rwx------ 1 ubuntu ubuntu 1527 Nov  7 20:59 clusters-create.sh
-rwx------ 1 ubuntu ubuntu 4958 Nov  7 20:59 ec2Setup.sh
-rw-r--r-- 1 ubuntu ubuntu  655 Jun 24 15:44 .profile
drwx------ 2 ubuntu ubuntu 4096 Nov  7 19:34 .ssh
```

2.4 Make the scripts executable:

```
$ chmod 755 /home/ubuntu/*.sh
```

2.5 Add Json template to the current directory

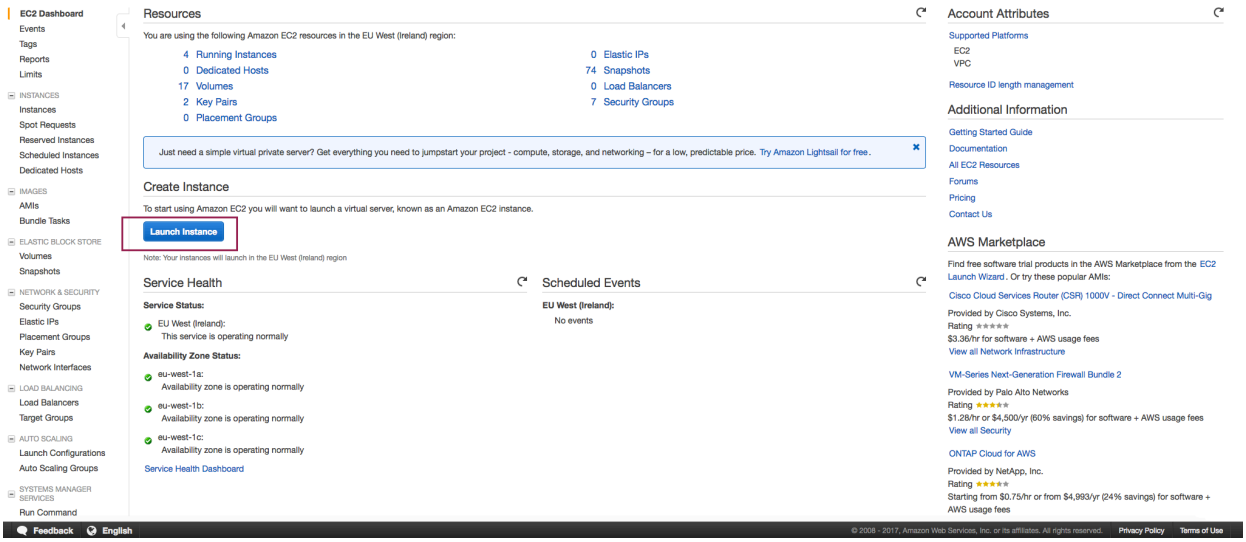
2.5.2 Copy the file found in https://github.com/HortonworksUniversity/Ops_Labs/blob/1.1.2/build/security/cloudformation.json into the directory above

2.5.3 The json file defines, among other things the AWS AMIs that will be use as base images for the nodes.

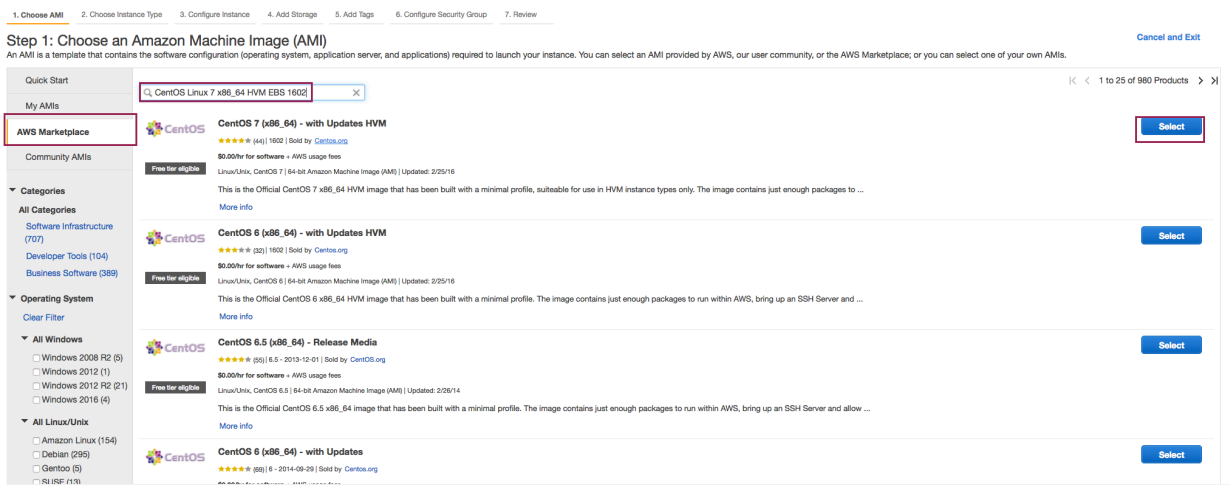
```
    },
    "CENTOS7": {
        "ap-northeast-1": {
            "AMI": "ami-b80b6db8"
        },
        "ap-northeast-2": {
            "AMI": "ami-c74789a9"
        },
        "ap-southeast-1": {
            "AMI": "ami-2a7b6b78"
        },
        "ap-southeast-2": {
            "AMI": "ami-d38dc6e9"
        },
        "eu-central-1": {
            "AMI": "ami-e68f82fb"
        },
        "eu-west-1": {
            "AMI": "ami-7abd0209"
        },
        "sa-east-1": {
            "AMI": "ami-fd0197e0"
        },
        "us-east-1": {
            "AMI": "ami-61bbf104"
        },
        "us-west-1": {
            "AMI": "ami-f77fbeb3"
        },
        "us-west-2": {
            "AMI": "ami-d440a6e7"
        }
    }
},
```


2.5.4 These AMIs are HWX Private images and cannot be shared. You will have to change to change these AMIs into free “AWS Marketplace” AMIs for each region that you want to launch a cluster in. Subsequent steps 2.5.4–2.5.10 describe how.

2.5.5 For this example we will use Ireland AWS Region. You will have to replicate these steps for each region that you want to launch clusters in. In Ireland region EC2 dashboard, select “Launch Instances”

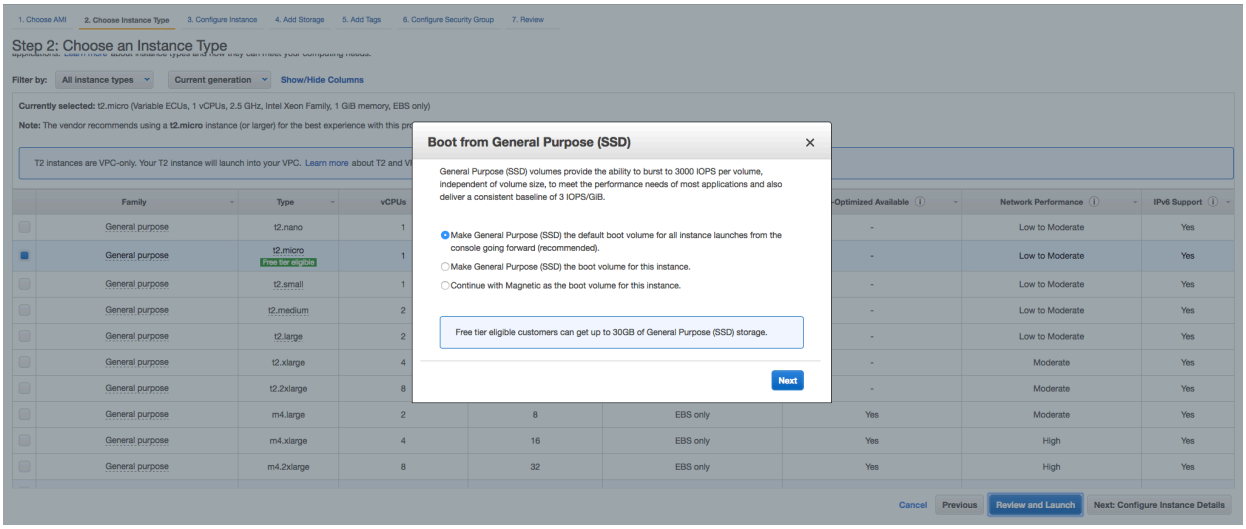


2.5.6 Select AWS Marketplace and search for this string “CentOS Linux 7 x86_64 HVM EBS 1602”



2.5.7 Unfortunately the AMI# is not provided in the previous selection. Also, if it is your first time selecting this AMI, AWS likes to approve the usage of this beforehand (takes a few minutes). The best thing to do is to launch a very small instance to a: obtain AMI# and b: get the Centos base image approved in your account. Next is how to do that. Click on the “Select” button to advance.

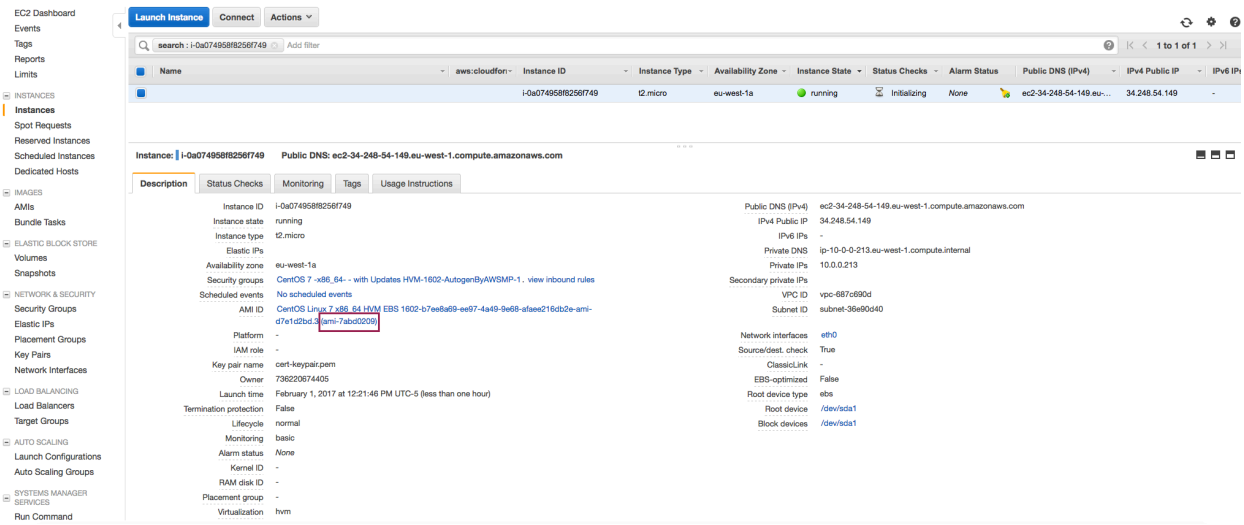
2.5.8 Since this is just a test instance to obtain the AMI and get the Marketplace Centos base image approved with your account, it does not matter what size we use, for this example, I am using t2.micro. This only needs to run for a couple minutes so charges are marginal. Click on Review and Launch and if you get a pop-out like below, just click the second choice and next.



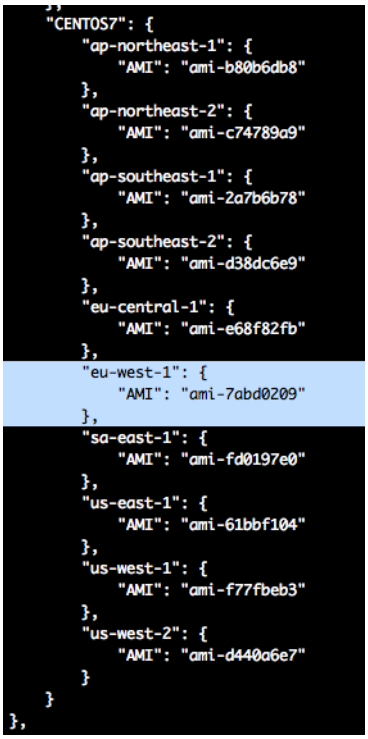
2.5.9 You will then get a confirmation screen, click launch and either create a new key-pair or use an existing one if you have one, it really does not matter because we wont access this and kill it in a few minutes.

2.5.10 The instance will launch and for the first time you use it you will need to be authorized to use it. It normally takes a few minutes to get an email saying that your account was authorized to use this image and the instance is launched.

2.5.11 After that you can go to the console and see the image running. The most important thing here is that your AWS account is now authorized to use this marketplace centos image and you can see the AMI ID. The AMI for Ireland is “ami-7abd0209” as shown here, but each AWS region will have a different id. You have to repeat these for each region so that you can get the AMI authorized and get the AMI id. After you write down the AMI id you can terminate this test instance.



2.6 Go back and edit the cloudformation.json file copied from github earlier and enter the correct AMI id for your region as obtained from the step above:



2.7 Update the clusters-create.sh script

2.7.2 Make a backup copy of the script first

2.7.3 There is a line that points to the github copy of the json file that needs to be changed to point to the local json file updated in the last step:

Change:

```
    --template-body
https://raw.githubusercontent.com/HortonworksUniversity/Ops_Labs
/1.1.2/build/security/cloudformation.json \
```

to:

```
    --template-body
file:////var//lib//jenkins//workspace//HDP2.x-Security-
Rev1.1.2//cloudformation.json \
```

2.7.4 In this example, my json file is in "/var/lib/jenkins/workspace/HDP2.x-Security-Rev1.1.2/cloudformation.json" Notice the double slash syntax for directories.

2.7.5 You are now ready to launch the security clusters.

Step 3: Launch Security Cluster(s)

3.1 Please review and update ec2Setup.sh. The next several steps describe how.

3.1.1 Edit lines 10-26 in the ec2Setup.sh script

```
$ vi /home/ubuntu/ec2Setup.sh
```

3.1.2 Here is a screenshot of the first few lines of the script:

```
#!/bin/bash

############################
# CREATE SECURITY CLUSTERS
############################


# 1. DEFINE VARIABLES
# 1.1 global vars
export AWS_CLI_HOME=/usr/local/aws
PATH=$PATH:$AWS_CLI_HOME/bin
export AWS_ACCESS_KEY_ID=<your aws key id>
export AWS_SECRET_ACCESS_KEY=<your aws secret access key>

# 1.2 job specific variables -- change these as required
export AWS_DEFAULT_REGION=us-west-2
export EC2_URL=https://ec2.us-west-2.amazonaws.com
export SEC_GROUP=sg-a02d17c4
export AMI=ami-0e3fde6e
export SUBNET=subnet-02edac67
export TRAINING_NAME=HDP25-Security-Rev1-1-0-Willie-Test
export FIRST_CLUSTER_LABEL=100
export NO_OF_VMs=2
export NO_OF_ADDTL_NODES=3
export INSTANCE_TYPE="m4.large"
export ADD_AD_SERVER=true

# 1.3 Cloudformation variables -- do not change
export lab_prefix=$TRAINING_NAME"-"
export lab_first=$FIRST_CLUSTER_LABEL
export lab_count=$NO_OF_VMs
export lab_batch=$NO_OF_ADDTL_NODES
export cfn_parameters='
```

3.1.3 Lines 10-11 are normally not need to change, but verify that the aws command is at /usr/local/aws.

3.1.4 Change lines 12-13 to reflect your AWS account key id and secret key

3.1.5 Change lines 16-17 to reflect the AWS region where you want the clusters to be created.

3.1.6 Line 18 represents a AWS Security group that you need to create that exposes ports for the cluster. This Sec-Group will be used for all cluster nodes as well as the WIN-AD server. To setup the AWS Sec Group ports, please see table in step 3.1.17.

3.1.7 Line 19 is the AMI ID of the Windows AD Server that was shared with you from Hortonworks in your default region. If you do not have an AMI shared with you, please contact hwuniversity@hortonworks.com.

3.1.8 Line 20 is a VPC subnet that needs to be created in your AWS region. For info on how to create a VPC Subnet, please refer to: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html

3.1.9 Line 21 is the Basic Name of the clusters (one for each class). Since this setup creates a multi-node clusters, this line represents the cluster PREFIX. Normally clusters will be appended a NUMBER for each student after this basic name, for instance: My-Cluster-100, My-Cluster-101, etc., where the postfix 100 & 101 refer to separate clusters/students. For instance there will be several instances named "My-Cluster-100" that all refer to a single student's cluster.

3.1.10 Line 22 is the numeric POSTFIX that will be appended to the Cluster Name Prefix described in the previous step. In the screenshot, the postfix for the first cluster, for instance, is "100" and will be incremented by 1 for each cluster, thus you can end up with clusters each named "My-Cluster-100," "My-Cluster-101," "My-Cluster-102" etc.

3.1.11 Line 23 is the number of Clusters (one per student) to be created. Each cluster can have 2 or more nodes.

3.1.12 Line 24 is the number of "additional" nodes besides Ambari to be created per cluster. For the Security class, this is either 2 or 3. As per screenshot, this default is "3" and this will create an Ambari node + 3 Additional nodes (total 4 nodes) per student.

3.1.13 Line 25 is the AWS instance type. We have found that "m4.large" is acceptable for this class.

3.1.14 Line 26 is the flag whether to create an AD Server or not. For the Security class this is required. Leave as "true." Please note that only 1 AD server is created/needed per class

3.1.15 Finally, line 60 is defining the WIN-AD server as AWS instance type "m3.xlarge." Some AWS accounts/regions do not support this type. However, you can use an instance type that is similar. To see what your region/aws account supports, just follow the same steps as in 1.2.1~1.2.2 to see list of available instances and use something similar.

3.1.16 After making the appropriate changes, save the ec2Setup.sh script.

3.1.17 AWS Security Group Port list

Type	Protocol	Port	Source
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
LDAP	TCP	389	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
Custom TCP Rule	TCP	3000	0.0.0.0/0
Custom TCP Rule	TCP	3359	0.0.0.0/0
RDP	TCP	3389	0.0.0.0/0
Custom TCP Rule	TCP	4000	0.0.0.0/0
Custom TCP Rule	TCP	6080	0.0.0.0/0
Custom TCP Rule	TCP	6083	0.0.0.0/0
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	8081	0.0.0.0/0
Custom TCP Rule	TCP	8088	0.0.0.0/0
Custom TCP Rule	TCP	8443	0.0.0.0/0
Custom TCP Rule	TCP	8444	0.0.0.0/0
Custom TCP Rule	TCP	8983	0.0.0.0/0
Custom TCP Rule	TCP	9995	0.0.0.0/0
Custom TCP Rule	TCP	11000	0.0.0.0/0
Custom TCP Rule	TCP	15000	0.0.0.0/0
Custom TCP Rule	TCP	19888	0.0.0.0/0
Custom TCP Rule	TCP	50070	0.0.0.0/0
Custom TCP Rule	TCP	50075	0.0.0.0/0
Custom TCP Rule	TCP	59100	0.0.0.0/0
Custom TCP Rule	TCP	61888	0.0.0.0/0
All traffic	All	All	sg-xxxxxxxx (THIS security group)
All ICMP - IPv4	All	N/A	0.0.0.0/0

3.2 Run the ec2Setup.sh script:

```
$ /home/ubuntu/ec2Setup.sh
```

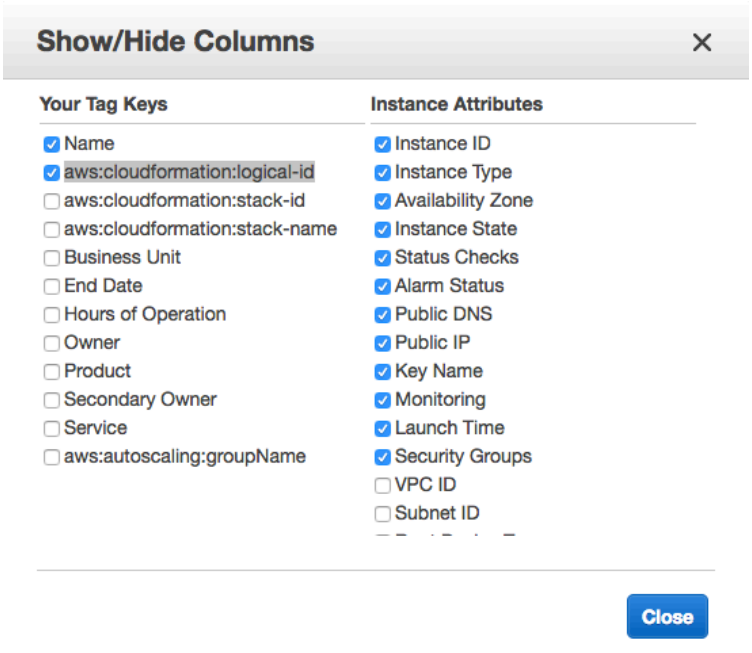
3.3 This will launch an AWS CloudFormation job that takes several minutes, please be patient (it may take up to 5 minutes per cluster to be build). You will eventually see this output:

```
$ /home/ubuntu/ec2Setup.sh
1. Creating new clusters in region us-west-2...
   This may take several minutes, please wait.
   Creating Win-AD Instance

2. List of newly created Instances:
i-03770d1b i-92750f8a i-93750f8b i-94750f8c

3. Final List of Hosts:

 My-Cluster-100 : AdditionalNodes : 35.162.214.71/172.30.0.34
 My-Cluster-100 : AdditionalNodes : 35.162.244.231/172.30.0.36
 My-Cluster-100 : AdditionalNodes : 35.163.20.252/172.30.0.35
 My-Cluster-100 : AmbariNode----- : 35.163.20.215/172.30.0.187
 My-Cluster     : WIN AD SERVER   : 35.161.150.18/172.30.0.131
```

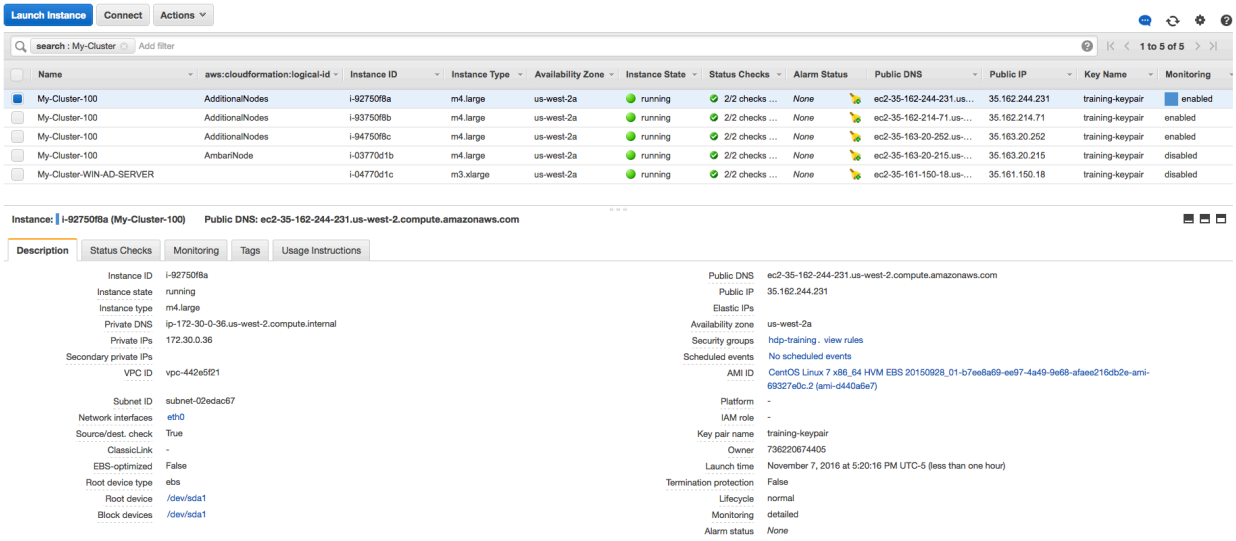
3.4 In this example I created 1 cluster named “My-Cluster” with an Ambari server, plus 3 additional nodes + Win AD server. In this output, each node has external/internal IPs. The Ambari node, for instance, has an external IP: 35.163.20.215 and an internal IP: 172.30.0.187.

3.5 You can also see this in the EC2 Console:

3.5.1 In the EC2 console add a column for aws:cloudformation:logical-id:

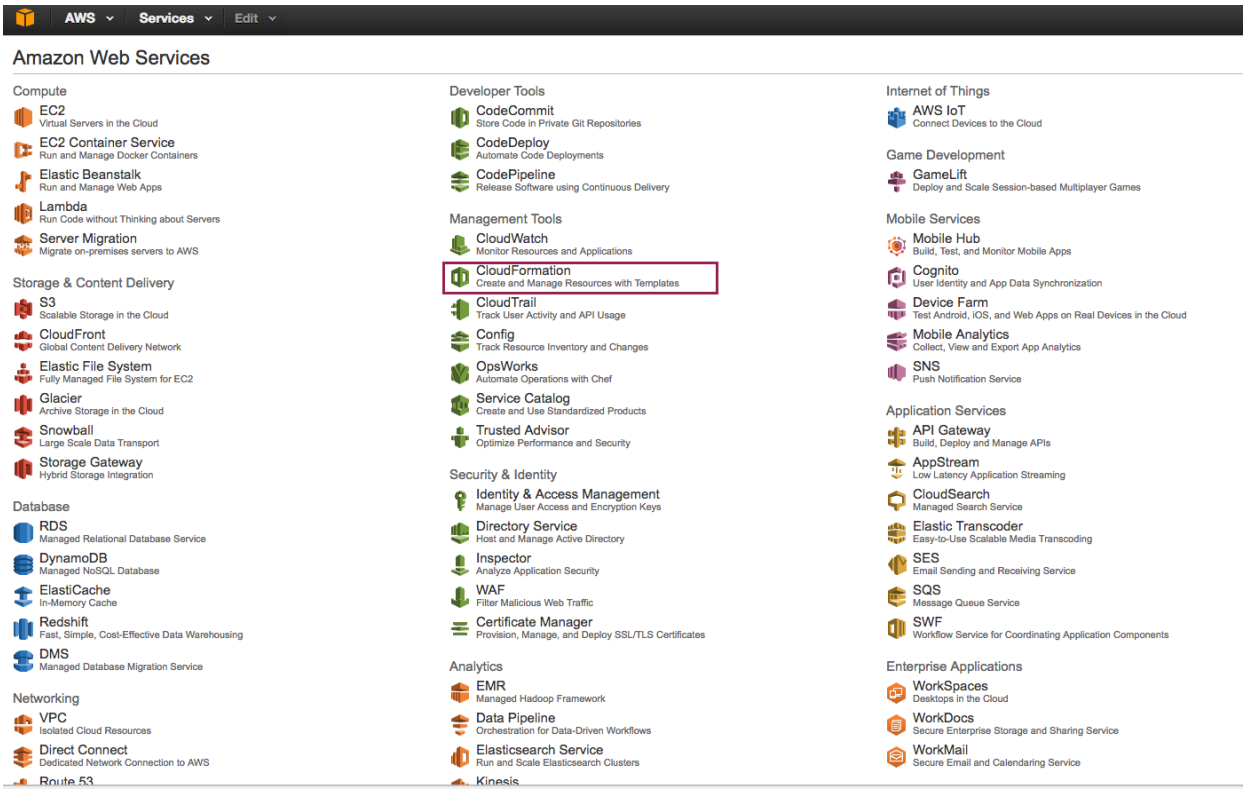


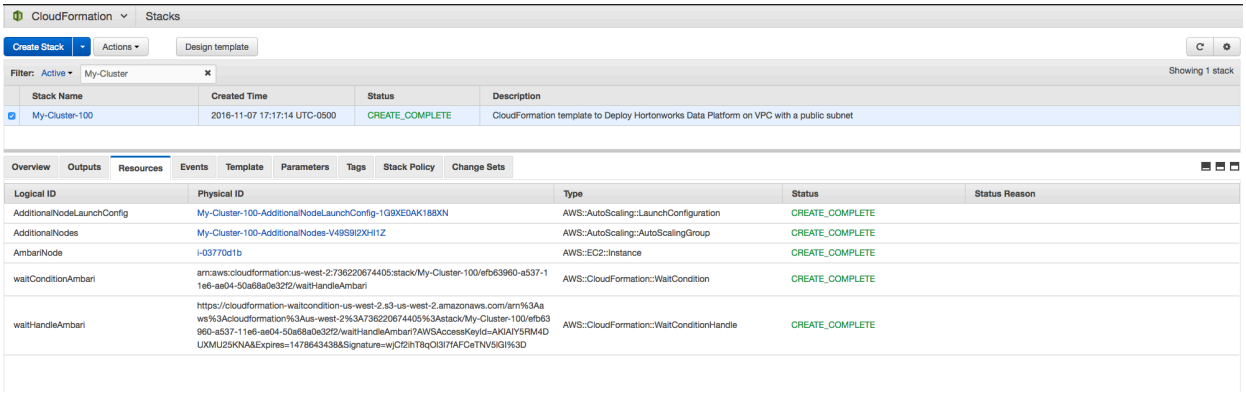
3.5.2 Search for “My-Cluster” and you will see the created nodes:



3.5.3 In this screenshot you can see One cluster, My-Cluster-100 (for one student), that has 1 Ambari node + 3 Additional Nodes + 1 AD Server.

3.6 Also, you can check the additional CloudFormation Administration console:





3.7 At this point you have a list of all clusters, all nodes, and their external/internal IPs from the ec2Script.sh output. You are now ready to start the labs.

3.8 After your clusters have been successfully created, you can Stop (but not terminate) the AWS CLI Server to limit your AWS costs.

Step 4: Test cluster connectivity

4.1 Ssh to each cluster node via Terminal window (if MacOS) or puTTY (if Windows) using the previously downloaded training-keypair. Make sure that you can ssh and that nodes are responding to ping between each other.

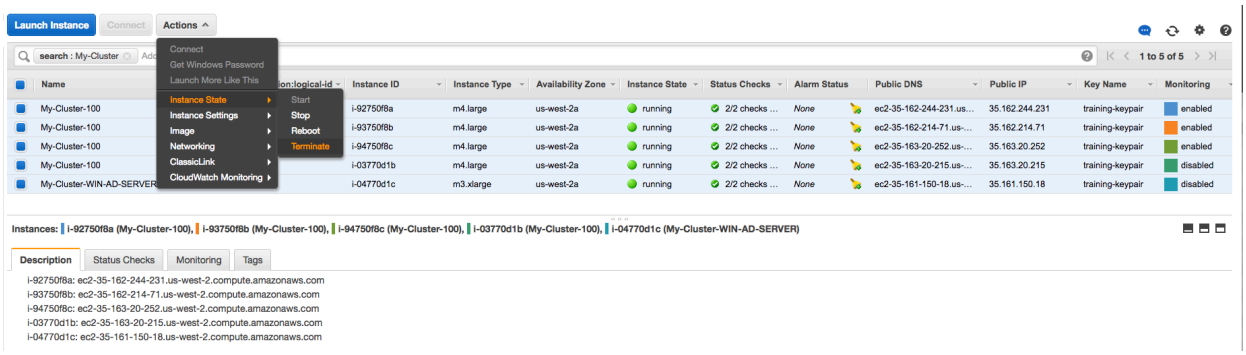
4.2 To access the WIN-AD server, you will need the Remote Desktop application (Windows) or Microsoft Remote Desktop.app (Mac OS). The credentials for the server are user/password = Administrator/pqysjWG9V(F

Shut Down Clusters after class completion

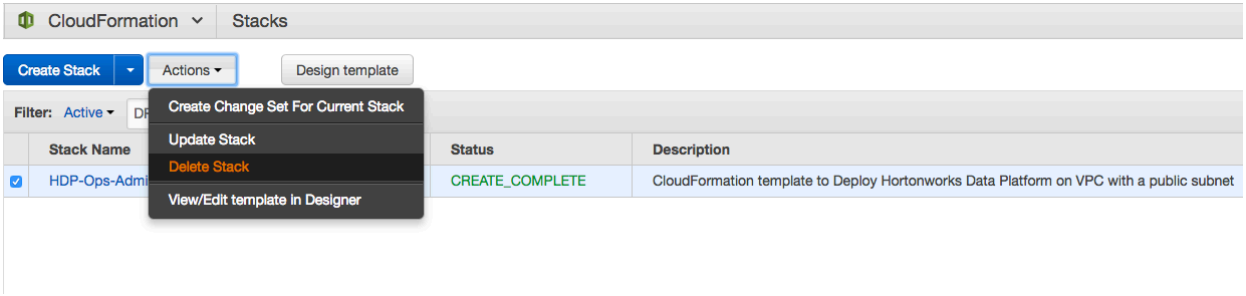
Step 5: How to shut down the clusters on the last day after the end of class.

5.1 Normally you should terminate instances via the EC2 console and selecting "Terminate." However, these clusters were created with AWS CloudFormation, which has a feature to "RELAUNCH" terminated/stopped instances. To avoid the automatic re-launching of instances you should perform the following steps.

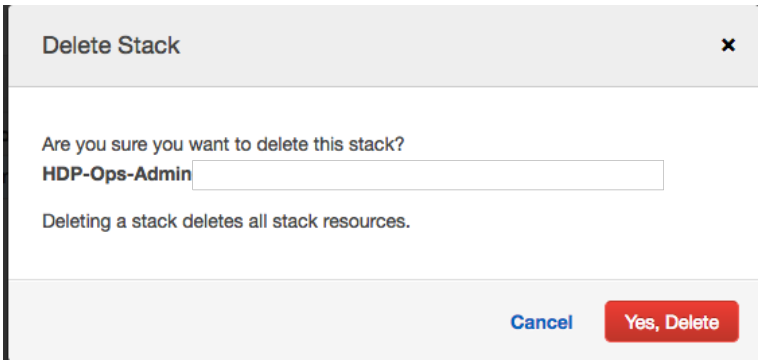
5.2 In the EC2 Console select all the nodes from all the class clusters to be deleted and "Terminate" them:



5.3 <u>In addition</u>, go to the CloudFormation Console and select the clusters to be deleted:



5.4 Confirm delete. After a few minutes the Clusters will be gone from both AWS and CloudFormation Consoles.



Classes Available Worldwide Through Our Partners

Study Options Worldwide

In combination with our partner providers, classes are often available in numerous locations across the world.

Private On-site Training

Hortonworks training in-house covers all of our basic coursework, and provides a more intimate setting for 6 or more students.
Contact us for more details