

# Security Administration

## Cluster Setup Guide

Rev 1.1.2



# Create the Security Clusters in AWS

<b>Objective:</b>	This is an instruction guide to setup training Security Clusters
<b>Before You Begin:</b>	<ul style="list-style-type: none"><li>• You need to have an AWS account to launch clusters.</li><li>• You need to have access to the AWS EC2 console</li><li>• You need to ask Hortonworks to share the Windows Server AD Image with you (please contact <a href="mailto:hwuniversity@hortonworks.com">hwuniversity@hortonworks.com</a>)</li><li>• You need to have the following ports open on your AWS firewall: 22, 80, 8080, 8443, 8444, 6080 &amp; 6083.</li></ul>

## 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](#):

The screenshot shows the AWS CloudFormation console with the 'Choose an Amazon Machine Image (AMI)' step selected. The search bar at the top contains the text 'ubuntu'. Below the search bar, a list of AMIs is shown, with the first item, 'ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20161020 - ami-a9d276c9', highlighted by a red box. To the right of this item is a 'Select' button, which is also highlighted with a red box. The rest of the list includes other Ubuntu and Snappy Core Edge images, each with a 'Select' button to its right. The left sidebar shows various operating systems and architectures available for selection.

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

The screenshot shows the 'Step 2: Choose an Instance Type' page. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type (which is selected), 3. Configure Instance, 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, and 7. Review. On the right, it shows the user's email (hwx\_admin/wgonzalez@hwx.c...), location (Oregon), and support status. Below the tabs, a note says: 'Step 2: Choose an Instance Type' and 'Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.' A filter bar at the top allows filtering by 'All instance types', 'Current generation', and 'Show/hide Columns'. A note below the filter says: 'Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.5 GHz, Intel Xeon Family, 4 GiB memory, EBS only)'. A message in a blue box says: 'T2 instances are VPC-only. Your T2 instance will launch into your VPC. Learn more about T2 and VPC.'. The main table lists various instance types with their details: Family, Type, vCPUs, Memory (GiB), Instance Storage (GiB), EBS-Optimized Available, and Network Performance. The 't2.medium' row is highlighted with a blue background. At the bottom right of the table are buttons: 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

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'

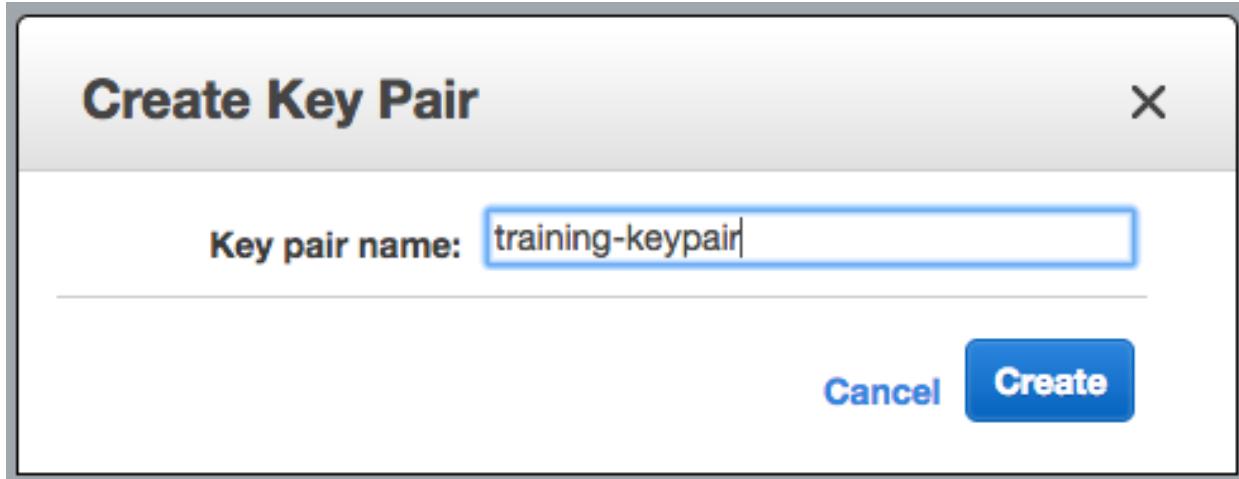
1.2.5. Name your instance appropriately on 'Step 5 – Tag Instance':

The screenshot shows the 'Step 5: Tag Instance' page. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Tag Instance (which is selected), 6. Configure Security Group, and 7. Review. A note above the form says: 'Step 5: Tag Instance' and 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.' The main form has two fields: 'Key' (127 characters maximum) and 'Value' (255 characters maximum). The 'Key' field contains 'Name' and the 'Value' field contains 'AWS CLI Server'. Below the fields is a 'Create Tag' button and a note '(Up to 50 tags maximum)'. At the bottom right are buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Configure Security Group'.

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

The screenshot shows the 'Step 6: Configure Security Group' page. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Tag Instance, 6. Configure Security Group (which is selected), and 7. Review. A note above the form says: 'Step 6: Configure Security Group' and 'A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.' The main form has a section 'Assign a security group:' with radio buttons for 'Create a new security group' (selected) and 'Select an existing security group'. Below this are fields for 'Security group name:' (set to 'launch-wizard-3') and 'Description:' (set to 'launch-wizard-3 created 2016-11-07T14:28:14.632-05:00'). Below these is a table for 'Add Rule' with columns: Type (SSH), Protocol (TCP), Port Range (22), and Source (Anywhere, 0.0.0.0/0). An 'Add Rule' button is at the bottom left. A warning message at the bottom says: 'Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

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 Mac OSX 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
$
```

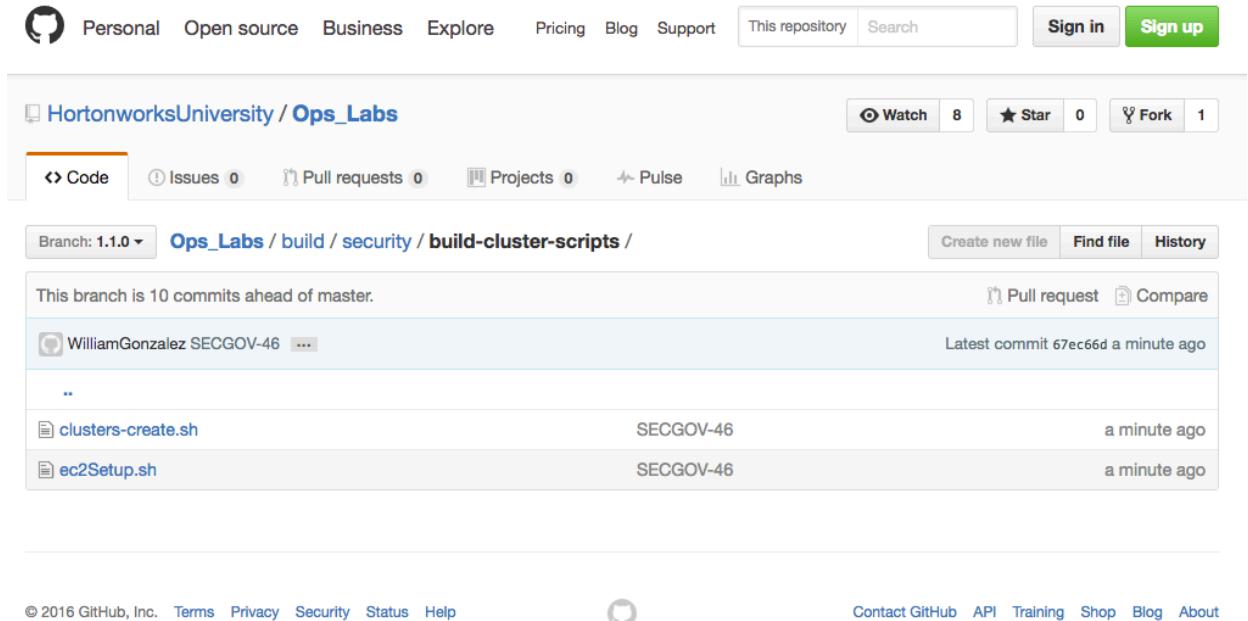
## Step 2: Download and Install Security Cluster Create scripts into AWS CLI Server

### 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](https://github.com/HortonworksUniversity/Ops_Labs/tree/1.1.2/build/security/build-cluster-scripts)

Download the 2 files found in that directory:



The screenshot shows a GitHub repository page for 'HortonworksUniversity / Ops\_Labs'. The 'Code' tab is selected. In the 'Branch: 1.1.0' dropdown, 'Ops\_Labs / build / security / build-cluster-scripts /' is chosen. The repository has 8 issues, 0 pull requests, 0 projects, and 1 fork. The latest commit was made a minute ago by WilliamGonzalez SEC GOV-46. The directory 'build-cluster-scripts' contains two files: 'clusters-create.sh' and 'ec2Setup.sh', both committed a minute ago.

### 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](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 used 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-f77fbef3"
        },
        "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”

The screenshot shows the AWS EC2 Dashboard for the EU West (Ireland) region. On the left, there's a sidebar with various navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, and more. The main content area has tabs for Resources, Account Attributes, and AWS Marketplace. Under Resources, it shows 4 Running Instances, 0 Dedicated Hosts, 17 Volumes, 2 Key Pairs, and 0 Placement Groups. Under Account Attributes, it lists Supported Platforms (EC2, VPC), Resource ID length management, and Additional Information (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us). Under AWS Marketplace, it lists products like Cisco Cloud Services Router (CSR 1000V - Direct Connect Multi-Gig) and VM-Series Next-Generation Firewall Bundle 2. A central box titled 'Create Instance' contains a 'Launch Instance' button, which is highlighted with a red box. Below the button, there's a note: 'Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking - for a low, predictable price. Try Amazon Lightsail for free.' At the bottom, there are 'Feedback' and 'English' buttons.

2.5.6 Select AWS Marketplace and search for this string “CentOS Linux 7 x86\_64 HVM EBS 1602”

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start  
My AMIs  
AWS Marketplace  
Community AMIs

- Categories**
  - All Categories
    - Software Infrastructure (707)
    - Developer Tools (104)
    - Business Software (389)
  - Operating System
    - Clear Filter
    - All Windows
      - Windows 2008 R2 (5)
      - Windows 2012 (1)
      - Windows 2012 R2 (21)
      - Windows 2016 (4)
    - All Linux/Unix
      - Amazon Linux (154)
      - Debian (295)
      - CentOS (5)
      - SUSE (13)

**CentOS**

**CentOS 7 (x86\_64) - with Updates HVM**  
★★★★★ (44) 1602 Sold by CentOS.org  
\$0.00/hr for software + AWS usage fees  
Linux/Unix, CentOS 7 | 64-bit Amazon Machine Image (AMI) | Updated: 2/25/16  
This is the official CentOS 7 x86\_64 HVM image that has been built with a minimal profile, suitable for use in HVM instance types only. The image contains just enough packages to ...  
More info **Select**

**CentOS 6 (x86\_64) - with Updates HVM**  
★★★★★ (23) 1602 Sold by CentOS.org  
\$0.00/hr for software + AWS usage fees  
Linux/Unix, CentOS 6 | 64-bit Amazon Machine Image (AMI) | Updated: 2/25/16  
This is the official CentOS 6 x86\_64 HVM image that has been built with a minimal profile. The image contains just enough packages to run within AWS, bring up an SSH Server and ...  
More info **Select**

**CentOS 6.5 (x86\_64) - Release Media**  
★★★★★ (55) 6.5 - 2013-12-01 Sold by CentOS.org  
\$0.00/hr for software + AWS usage fees  
Linux/Unix, CentOS 6.5 | 64-bit Amazon Machine Image (AMI) | Updated: 2/25/16  
This is the official CentOS 6.5 x86\_64 image that has been built with a minimal profile. The image contains just enough packages to run within AWS, bring up an SSH Server and allow ...  
More info **Select**

**CentOS 6 (x86\_64) - with Updates**  
★★★★★ (69) 6 - 2014-09-29 Sold by CentOS.org  
\$0.00/hr for software + AWS usage fees

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.

Step 2: Choose an Instance Type

Currently selected: t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Note: The vendor recommends using a t2.micro instance (or larger) for the best experience with this product.

Filter by: All instance types Current generation Show/Hide Columns

T2 instances are VPC-only. Your T2 instance will launch into your VPC. Learn more about T2 and VPC.

Family	Type	vCPUs
General purpose	t2.nano	1
General purpose	<b>t2.micro</b> Free tier eligible	1
General purpose	t2.small	1
General purpose	t2.medium	2
General purpose	t2.large	2
General purpose	t2.xlarge	4
General purpose	t2.2xlarge	8
General purpose	m4.large	2
General purpose	m4.xlarge	4
General purpose	m4.2xlarge	8

**Boot from General Purpose (SSD)**

General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB.

Make General Purpose (SSD) the default boot volume for all instance launches from the console going forward (recommended).

Make General Purpose (SSD) the boot volume for this instance.

Continue with Magnetic as the boot volume for this instance.

Free tier eligible customers can get up to 30GB of General Purpose (SSD) storage.

**Next**

Optimized Available Network Performance IPv6 Support

- Low to Moderate Yes

- High Yes

- High Yes

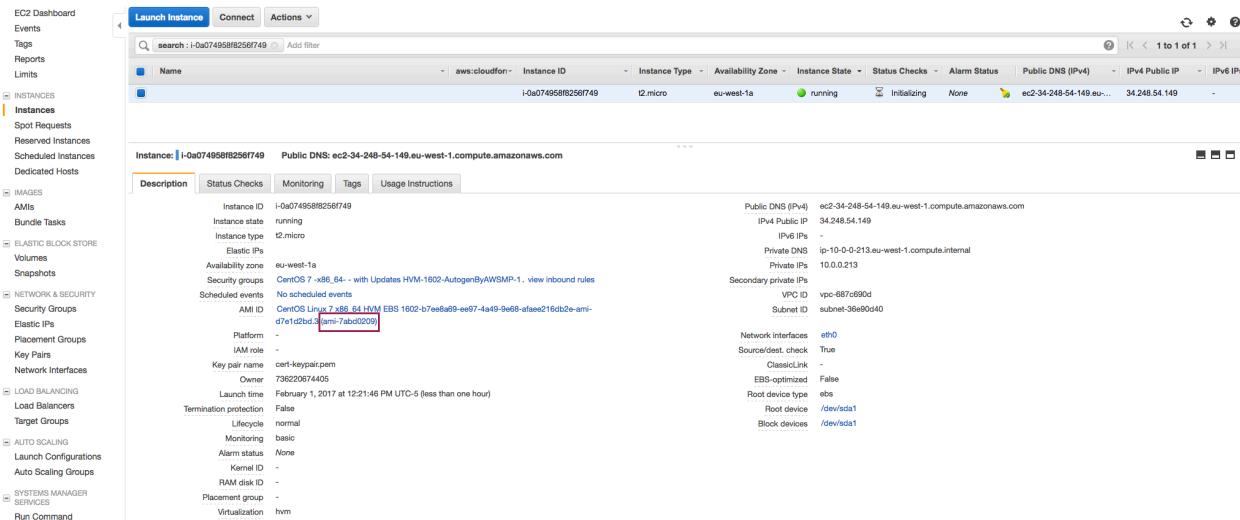
- High Yes

**Cancel Previous Review and Launch Next: Configure Instance Details**

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:

```

    "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-61bbbf104"
        },
        "us-west-1": {
            "AMI": "ami-f77fbbeb3"
        },
        "us-west-2": {
            "AMI": "ami-d440a6e7"
        }
    }
},

```

## 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:

```
1 #!/bin/bash
2
3 #####
4 # CREATE SECURITY CLUSTERS
5 #####
6
7
8 # 1. DEFINE VARIABLES
9 # 1.1 global vars
10 export AWS_CLI_HOME=/usr/local/aws
11 PATH=$PATH:$AWS_CLI_HOME/bin
12 export AWS_ACCESS_KEY_ID=<your aws key id>
13 export AWS_SECRET_ACCESS_KEY=<your aws secret access key>
14
15 # 1.2 job specific variables -- change these as required
16 export AWS_DEFAULT_REGION=us-west-2
17 export EC2_URL=https://ec2.us-west-2.amazonaws.com
18 export SEC_GROUP=sg-a02d17c4
19 export AMI=ami-0e3fde6e
20 export SUBNET=subnet-02edac67
21 export TRAINING_NAME=HDP25-Security-Rev1-1-0-Willie-Test
22 export FIRST_CLUSTER_LABEL=100
23 export NO_OF_VMs=2
24 export NO_OF_ADDTL_NODES=3
25 export INSTANCE_TYPE="m4.large"
26 export ADD_AD_SERVER=true
27
28 # 1.3 Cloudformation variables -- do not change
29 export lab_prefix=$TRAINING_NAME"-"
30 export lab_first=$FIRST_CLUSTER_LABEL
31 export lab_count=$NO_OF_VMs
32 export lab_batch=$NO_OF_ADDTL_NODES
33 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](mailto: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](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

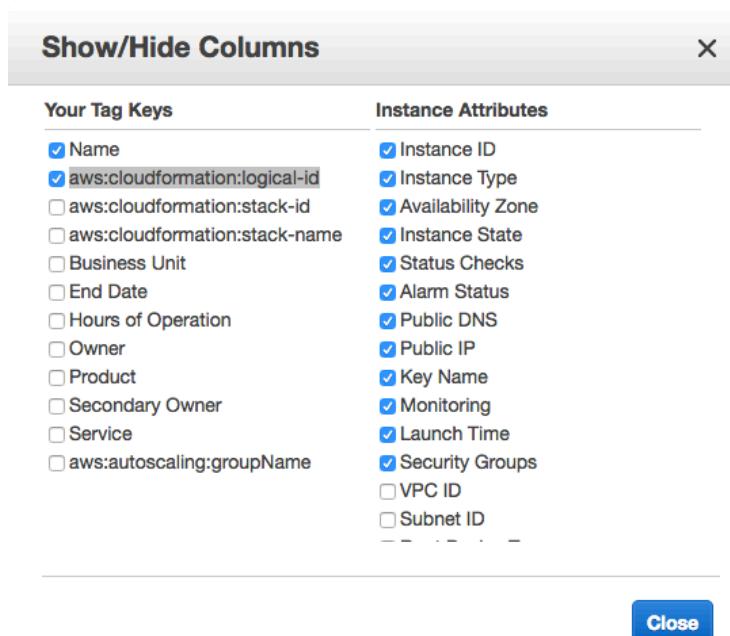
3. Final List of Hosts:

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:

The screenshot shows the AWS CloudFormation console. At the top, there is a search bar with the query "search : My-Cluster". Below the search bar is a table listing resources. One row is selected, showing details for an instance named "My-Cluster-100". The table columns include Name, aws.cloudformation:logical-id, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, Public IP, Key Name, and Monitoring.

**Selected Resource Details:**

- Instance ID:** i-92750f8a (My-Cluster-100)
- Public DNS:** ec2-35-162-244-231.us-west-2.compute.amazonaws.com
- Description:** AdditionalNodes
- Status Checks:** 2/2 checks ... None
- Alarm Status:** None
- Public DNS:** ec2-35-162-244-231.us-west-2.compute.amazonaws.com
- Public IP:** 35.162.244.231
- Key Name:** training-keypair (enabled)
- Monitoring:** Enabled

**Resource Properties:**

- Instance ID: i-92750f8a
- Instance state: running
- Instance type: m4.large
- Private DNS: ip-172-30-0-36.us-west-2.compute.internal
- Private IPs: 172.30.0.36
- Secondary private IPs:
- VPC ID: vpc-442e5f21
- Subnet ID: subnet-02edac67
- Network interfaces: eth0
- Source/dest. check: True
- ClassicLink: -
- EBS-optimized: False
- Root device type: ebs
- Root device: /dev/sda1
- Block devices: /dev/sda1

**Configuration:**

- Public DNS: ec2-35-162-244-231.us-west-2.compute.amazonaws.com
- Public IP: 35.162.244.231
- Elastic IPs:
- Availability zone: us-west-2a
- Security groups: hdp-training , view rules
- Scheduled events: No scheduled events
- AMI ID: CentOS Linux 7 x86\_64 HVM EBS 20150928\_01-b7ee8a69-ee97-4a49-9e68-afaae216db2e-ami-69327abc.2 (ami-d440a6e7)
- Platform: -
- IAM role: -
- Key pair name: training-keypair
- Owner: 736220674405
- Launch time: November 7, 2016 at 5:20:16 PM UTC-5 (less than one hour)
- Termination protection: False
- Lifecycle: normal
- Monitoring: detailed
- Alarm status: None

### 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:

The screenshot shows the AWS CloudFormation Administration console. The URL is [https://console.aws.amazon.com/cloudformation/home?region=us-east-1#stacks:sort\\_by=creationTime:desc](#). The page displays the CloudFormation service page under the Amazon Web Services category.

**CloudFormation:** Create and Manage Resources with Templates (highlighted with a red box).

**Other Services:**

- Compute:**
  - EC2: Virtual Servers in the Cloud
  - EC2 Container Service: Run and Manage Docker Containers
  - Elastic Beanstalk: Run and Manage Web Apps
  - Lambda: Run Code without Thinking about Servers
  - Server Migration: Migrate on-premises servers to AWS
- Storage & Content Delivery:**
  - S3: Scalable Storage in the Cloud
  - CloudFront: Global Content Delivery Network
  - Elastic File System: Fully Managed File System for EC2
  - Glacier: Archive Storage in the Cloud
  - Snowball: Large Scale Data Transport
  - Storage Gateway: Hybrid Storage Integration
- Database:**
  - RDS: Managed Relational Database Service
  - DynamoDB: Managed NoSQL Database
  - ElastiCache: In-Memory Cache
  - Redshift: Fast, Simple, Cost-Effective Data Warehousing
  - DMS: Managed Database Migration Service
- Networking:**
  - VPC: Isolated Cloud Resources
  - Direct Connect: Dedicated Network Connection to AWS
  - Route 53:
- Developer Tools:**
  - CodeCommit: Store Code in Private Git Repositories
  - CodeDeploy: Automate Code Deployments
  - CodePipeline: Release Software using Continuous Delivery
- Management Tools:**
  - CloudWatch: Monitor Resources and Applications
  - CloudFormation: Create and Manage Resources with Templates (highlighted with a red box)
  - CloudTrail: Track User Activity and API Usage
  - Config: Track Resource Inventory and Changes
  - OpsWorks: Automate Operations with Chef
  - Service Catalog: Create and Use Standardized Products
  - Trusted Advisor: Optimize Performance and Security
- Security & Identity:**
  - Identity & Access Management: Manage User Access and Encryption Keys
  - Directory Service: Host and Manage Active Directory
  - Inspector: Analyze Application Security
  - WAF: Filter Malicious Web Traffic
  - Certificate Manager: Provision, Manage, and Deploy SSL/TLS Certificates
- Analytics:**
  - EMR: Managed Hadoop Framework
  - Data Pipeline: Orchestration for Data-Driven Workflows
  - Elasticsearch Service: Run and Scale Elasticsearch Clusters
  - Kinesis:
- Internet of Things:**
  - AWS IoT: Connect Devices to the Cloud
- Game Development:**
  - GameLift: Deploy and Scale Session-based Multiplayer Games
- Mobile Services:**
  - Mobile Hub: Build, Test, and Monitor Mobile Apps
  - Cognito: User Identity and App Data Synchronization
  - Device Farm: Test Android, iOS, and Web Apps on Real Devices in the Cloud
  - Mobile Analytics: Collect, View and Export App Analytics
  - SNS: Push Notification Service
- Application Services:**
  - API Gateway: Build, Deploy and Manage APIs
  - AppStream: Low Latency Application Streaming
  - CloudSearch: Managed Search Service
  - Elastic Transcoder: Easy-to-Use Scalable Media Transcoding
  - SES: Email Sending and Receiving Service
  - SQS: Message Queue Service
  - SWF: Workflow Service for Coordinating Application Components
- Enterprise Applications:**
  - WorkSpaces: Desktops in the Cloud
  - WorkDocs: Secure Enterprise Storage and Sharing Service
  - WorkMail: Secure Email and Calendaring Service

The screenshot shows the AWS CloudFormation console with the 'My-Cluster' stack selected. The 'Resources' tab is active, displaying the following resource details:

Logical ID	Physical ID	Type	Status	Status Reason
AdditionalNodeLaunchConfig	My-Cluster-100-AdditionalNodeLaunchConfig-1GBXE0AK168XN	AWS::AutoScaling::LaunchConfiguration	CREATE_COMPLETE	
AdditionalNodes	My-Cluster-100-AdditionalNodes-V4959I2XH1Z	AWS::AutoScaling::AutoScalingGroup	CREATE_COMPLETE	
AmbariNode	i-037770d1b	AWS::EC2::Instance	CREATE_COMPLETE	
waitConditionAmbari	arn:aws:cloudformation:us-west-2:736222674405:stack/My-Cluster-100/erb63960-a537-11e6-ae04-50a68a0e32f2/waitHandleAmbari	AWS::CloudFormation::WaitCondition	CREATE_COMPLETE	
waitHandleAmbari	https://cloudformation-waitcondition-us-west-2.s3-us-west-2.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-west-2%3A736222674405%3Astack/My-Cluster-100/erb63960-a537-11e6-ae04-50a68a0e32f2/waitHandleAmbari?AWSAccessKeyId=AKIAVY5RM4D960-a537-11e6-ae04-50a68a0e32f2&Signature=wjC59h1BqOj3l7AFCeINV9G%3D	AWS::CloudFormation::WaitConditionHandle	CREATE_COMPLETE	

**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:

The screenshot shows the AWS EC2 Instances page. There are five instances listed, all belonging to the 'My-Cluster-100' stack. The instances are: i-92750f8a, i-93750f8b, i-94750f8c, i-03770d1b, and i-04770d1c. The 'Actions' dropdown menu is open over the first instance, with 'Terminate' selected. The status of all instances is 'running'. The 'Monitoring' column indicates that four instances have monitoring enabled (blue) and one is disabled (green).

5.3 In addition, go to the CloudFormation Console and select the clusters to be deleted:

The screenshot shows the AWS CloudFormation Stacks page. It displays a single stack named 'HDP-Ops-Admin' with a status of 'CREATE\_COMPLETE'. The 'Actions' dropdown menu is open over the stack, with 'Delete Stack' selected. Other options in the menu include 'Create Change Set For Current Stack', 'Update Stack', and 'View/Edit template in Designer'.

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

The screenshot shows a modal dialog box titled 'Delete Stack'. It contains the message 'Are you sure you want to delete this stack?'. Below this is a text input field containing 'HDP-Ops-Admin'. A note at the bottom states 'Deleting a stack deletes all stack resources.' At the bottom right are two buttons: 'Cancel' and 'Yes, Delete'.

# 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](#)



Copyright © 2012 - 2016 Hortonworks, Inc. All rights reserved.

The contents of this course and all its lessons and related materials, including handouts to audience members, are Copyright © 2012 - 2015 Hortonworks, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Hortonworks, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Hortonworks, Inc. Hortonworks, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

All other trademarks are the property of their respective owners.