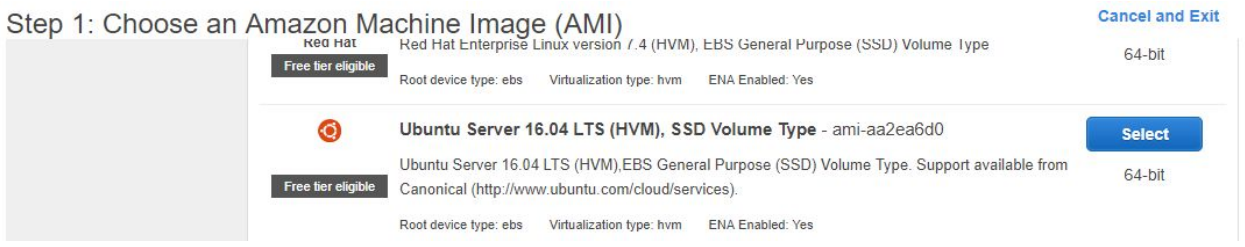
Deploy Ambari on AWS

Launch EC2 Ambari Server

1. In AWS console, select EC2 service. Click "Launch Instance" button to create a VM. Select Ubuntu AMI.



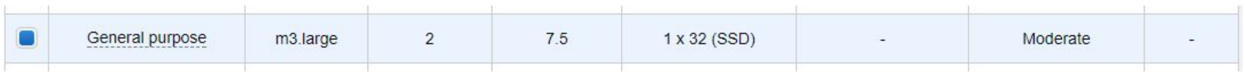
Step 1: Choose an Amazon Machine Image (AMI)

Red Hat Enterprise Linux version 7.4 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit

Free tier eligible

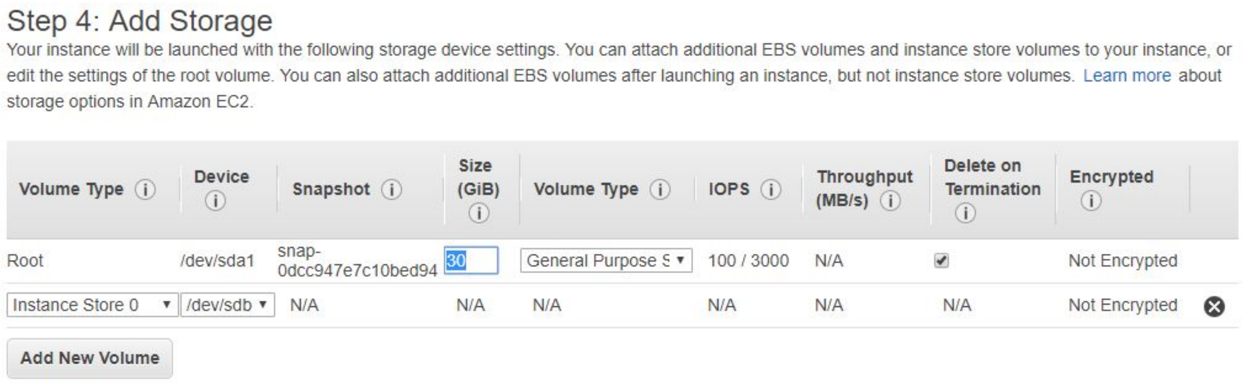
Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-aa2ea6d0
Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit

2. We will create a NameNode. Because this NameNode will be able to handle a relatively large cluster, choose m3.large instance. Click Next.



■	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate	-

3. On "Number of instances", use 1. Leave other options as default or change them to your configuration preference. Click Next.
4. Choose 30 GB of storage. Click Next.

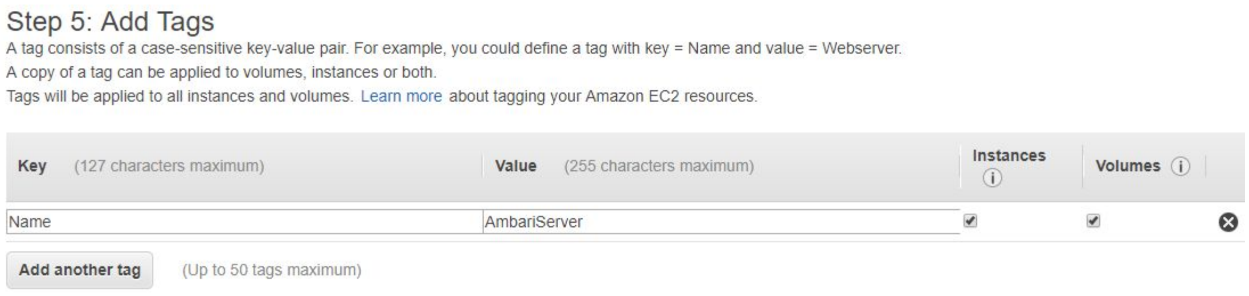


Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted	
Root	/dev/sda1	snap-0dcc947e7c10bed94	30	General Purpose S	100 / 3000	N/A	✓	Not Encrypted	
Instance Store 0	/dev/sdb	N/A	N/A	N/A	N/A	N/A	N/A	Not Encrypted	⊗

5. Add "Name" tag for this instance with value "Ambari Server". This is so that we can quickly recognize the instance from the console. Click Next.

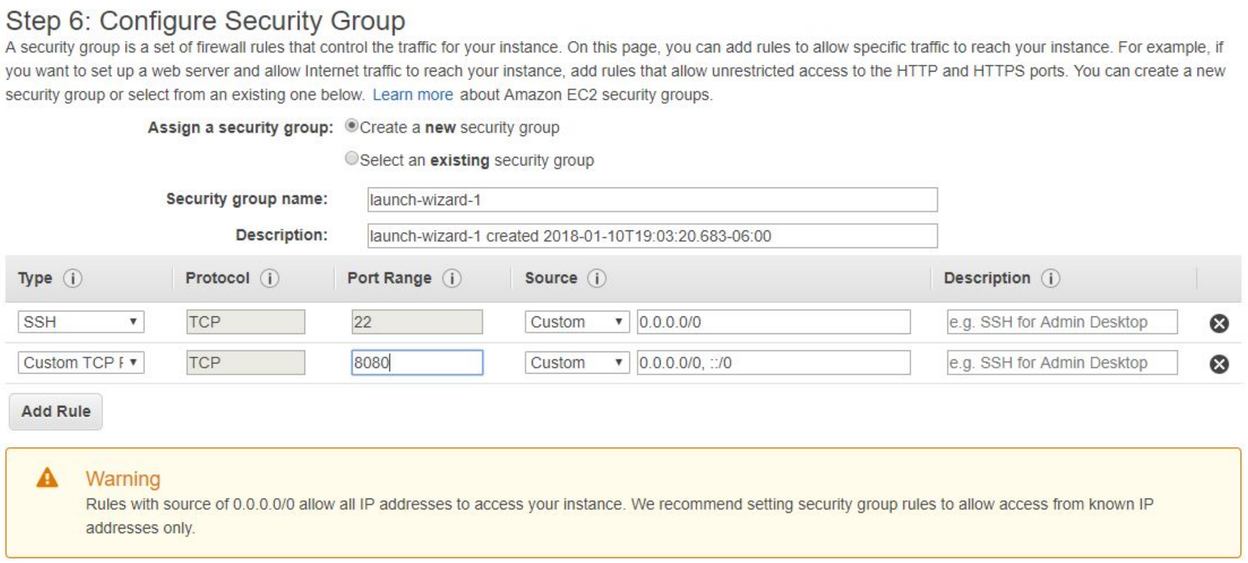


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes	
Name	AmbariServer	✓	✓	⊗

6. Add new rule as follow. Here we use 0.0.0.0/0 as Source which open connection from any IP address. Typically in production environment you will restrict the access to the specific IP or IP address ranges of your organization.



Step 6: Configure Security Group

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.

Assign a security group: [✓ Create a **new** security group] Select an **existing** security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2018-01-10T19:03:20.683-06:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

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.

7. Click “Review and Launch” button. Verify if everything is correct. Click “Launch” button. You can either create new key or use existing one.

Prepare the Instance for Ambari Deployment

1. Connect to the instance by ssh.
2. Turn off the Transparent Huge Pages. Add below lines in the /etc/rc.local file. You need to sudo the text edit command (e.g. sudo nano /etc/rc.local)

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi

if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

3. Install ntp package. The NTP will be used to synch the clock between the nodes in the cluster.

```
$ sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get install ntp
```

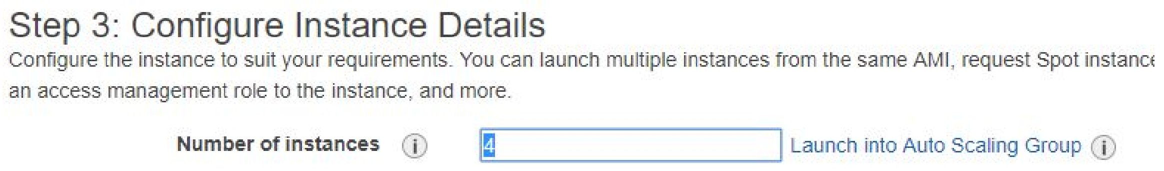
4. Check if ntp process is running.

```
$ sudo service ntp status
```

5. Now back to the EC2 Instance console. Right click on the Ambari server we just launched, Image, Create Image.
6. Name it as "AmbariNode", tick "No reboot" and click on "Create Image". We now have image ready to be used for us to launch the rest of Ambari nodes.

Launch the Nodes

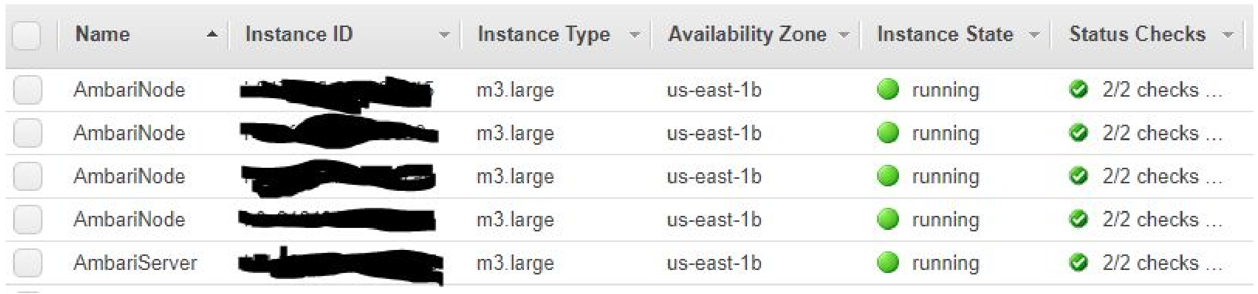
1. Now it is time to use the image "AmbariNode" that we have created above to launch the nodes. Same as before, use m3.large instance with 30 GB storage. Give the intances' tag name as "AmbariNode".
2. In this example, 4 nodes are created.



Step 3: Configure Instance Details

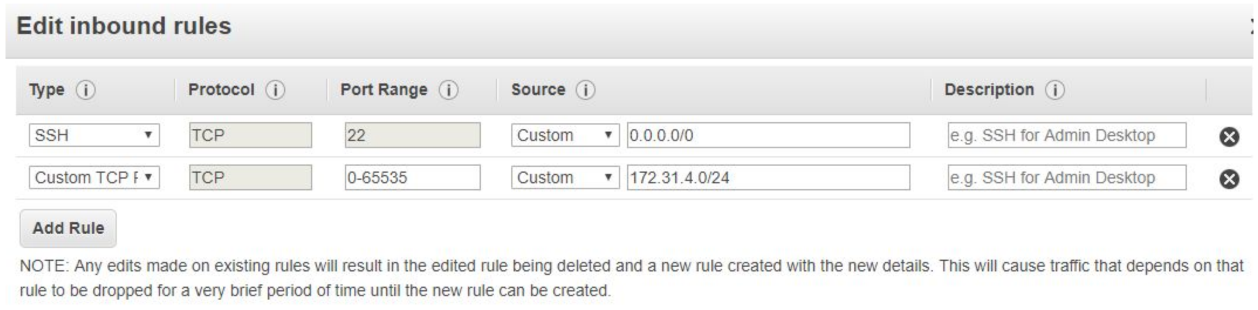
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instanc an access management role to the instance, and more.

Number of instances: 4 — Launch into Auto Scaling Group



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	AmbariNode	[redacted]	m3.large	us-east-1b	● running	✓ 2/2 checks ...
	AmbariNode	[redacted]	m3.large	us-east-1b	● running	✓ 2/2 checks ...
	AmbariNode	[redacted]	m3.large	us-east-1b	● running	✓ 2/2 checks ...
	AmbariNode	[redacted]	m3.large	us-east-1b	● running	✓ 2/2 checks ...
	AmbariServer	[redacted]	m3.large	us-east-1b	● running	✓ 2/2 checks ...

3. Configure the security group of each Ambari Node and Server so that they can communicate to each other through the private IP address. Let's say the IP addresses are between 172.31.4.10 and 172.31.4.14, you can add the create a security group with Inbound Rules that reflect this.



Edit inbound rules

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	0-65535	Custom	172.31.4.0/24	e.g. SSH for Admin Desktop

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Install and Start Ambari Server

1. Copy the private key (.pem) file from your PC to the Ambari Server. Save the pem file as id_rsa under directory /home/<user>/.ssh/ in the Ambari server.
2. SSH to the Ambari server. Add the ambari repository into the repository sources. The example below uses version 2.2.2.

```
cd /etc/apt/sources.list.d
sudo wget
```

http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.2.2.0/ambari.list

3. Install Ambari Server from the public Ambari repository:

```
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
B9733A7A07513CAD
sudo apt-get update
sudo apt-get install ambari-server
```

4. Run Setup. You can use default parameter here.

```
sudo ambari-server setup
```

`Note: for MySQL, the JDBC driver can be downloaded here.`

5. Start Ambari Server

```
sudo ambari-server start
```