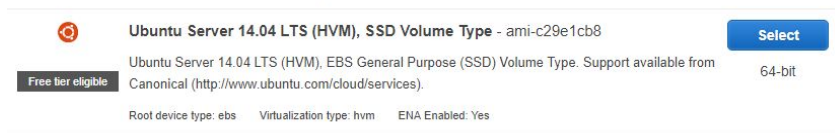# Deploy Ambari on AWS

1. Launch EC2 Ambari Server

1. In AWS console, select EC2 service. Click "Launch Instance" button to create a VM. Select Ubuntu AMI. Since we are going to deploy Ambary 2.2.2, use Ubuntu 14.04 image.

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

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.

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.

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	✕

Add Rule

⚠ **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.

2. 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.

3. 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 instance an access management role to the instance, and more.

Number of instances (i)   `4`   Launch into Auto Scaling Group (i)

	Name	▲	Instance ID	▼	Instance Type	▼	Availability Zone	▼	Instance State	▼	Status Checks	▼
☐	AmbariNode				m3.large		us-east-1b		● running		✓ 2/2 checks ...	
☐	AmbariNode				m3.large		us-east-1b		● running		✓ 2/2 checks ...	
☐	AmbariNode				m3.large		us-east-1b		● running		✓ 2/2 checks ...	
☐	AmbariNode				m3.large		us-east-1b		● running		✓ 2/2 checks ...	
☐	AmbariServer				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 (i)	Protocol (i)	Port Range (i)	Source (i)		Description (i)	
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.

4. Install and Start Ambari Server (v2.2.2)

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. Use default values for questions.

   ```
   sudo ambari-server setup
   ```

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

5. Start Ambari Server

   ```
   sudo ambari-server start
   ```

5. Deploy Hadoop Cluster

1. Open a web browser and connect to the DNS name of Ambari Server. Use port 8080. For example: http://<xx.xx.xx>.compute-1.amazonaws.com:8080.
2. Use user/password admin/admin.
3. Click on Launch Wizard. Give the name of your cluster. Click Next.
4. Select the stack version you would like to deploy. Click Next.

Stacks

- ◉ HDP 2.4
- ○ HDP 2.3
- ○ HDP 2.2

▸ Advanced Repository Options

[← Back]   [Next →]

5. Under Install Options page, enter all internal DNS of the AmbariNodes in the Target Hosts box. Upload your pem key for the nodes under Host Registration Information. Hit "Register and Confirm" button.

Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions

```
ip-172-31-10-123.ec2.internal
ip-172-31-13-240.ec2.internal
ip-172-31-8-137.ec2.internal
```

Host Registration Information

◉ Provide your SSH Private Key to automatically register hosts

[Choose File] ▬▬▬▬▬

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAyG6E/ocqwzUyF4ASsmLolukh/1JHeqfvMbE0SyJwXSBapZx
ClI3RkiBKQksX
```

SSH User Account   ubuntu

○ Perform manual registration on hosts and do not use SSH

[← Back]   [Register and Confirm →]

6. Wait until all nodes are installed successfully. Click Next.

Confirm Hosts

Registering your hosts.
Please confirm the host list and remove any hosts that you do not want to include in the cluster.

	Host	Progress	Status	Action
☐	ip-172-31-10-123.ec2.internal		Success	🗑 Remove
☐	ip-172-31-13-240.ec2.internal		Success	🗑 Remove
☐	ip-172-31-8-137.ec2.internal		Success	🗑 Remove

🗑 Remove Selected

Show: All (3) | Installing (0) | Registering (0) | Success (3) | Fail (0)

Show: 25 ▼   1 - 3 of 3   ⏮ ← → ⏭

All host checks passed on 3 registered hosts. Click here to see the check results.

← Back   Next →

7. Choose services under Services page. Click Next.
8. Choose host where you want the services to run on. Click Next.
9. Verify if all are okay. Click Next.
10. Enter passwords for database/services if asked. Click Next.
11. Finally, hit "Deploy" button. It may take some time to complete all installations.

Install, Start and Test

Please wait while the selected services are installed and started.

4 % overall

Show: All (3) | In Progress (3) | Warning (0) | Success (0) | Fail (0)

Host	Status	Message
ip-172-31-10-123.ec2.internal	4%	Installing DataNode
ip-172-31-13-240.ec2.internal	4%	Installing App Timeline Server
ip-172-31-8-137.ec2.internal	4%	Installing DataNode

3 of 3 hosts showing - Show All   Show: 25 ▼   1 - 3 of 3   ⏮ ← → ⏭

Install, Start and Test

Please wait while the selected services are installed and started.

```
████████████████████████████████████████████████  100 % overall
```

Show: **All (3)** | In Progress (0) | Warning (0) | Success (3) | Fail (0)

Host	Status	Message
ip-172-31-10-123.ec2.internal	████████ 100%	Success
ip-172-31-13-240.ec2.internal	████████ 100%	Success
ip-172-31-8-137.ec2.internal	████████ 100%	Success

3 of 3 hosts showing - Show All   Show: 25 ▾   1 - 3 of 3   ⏮ ← → ⏭

Successfully installed and started the services.

Next →

12. Congratulation, your Hadoop cluster is now ready!
    Don't forget to change the admin password.