Cloudera's Enterprise Data Hub on the AWS Cloud

Quick Start Reference Deployment

Karthik Krishnan

October 2014
Last updated: August 2015 (revisions)

Contents

About This Guide

This Quick Start reference deployment guide includes architectural considerations and configuration steps for deploying Cloudera's Enterprise Data Hub (EDH) on the Amazon Web Services (AWS) cloud. It discusses best practices for deploying Cloudera's EDH on AWS using services such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Virtual Private Cloud (Amazon VPC). It also provide links to automated AWS CloudFormation templates that you can leverage for your deployment or launch directly into your AWS account.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their Cloudera EDH workloads on the AWS cloud.

Quick Starts are automated reference deployments for key enterprise workloads on the AWS cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

Overview

Cloudera EDH on AWS

Cloudera's Enterprise Data Hub (EDH) allows you to store your data with the flexibility to run a variety of enterprise workloads—including batch processing, interactive SQL, enterprise search, and advanced analytics—while utilizing robust security, governance, data protection, and management.

AWS provides customers with the ability to set up the infrastructure to support EDH in a flexible, scalable, and cost-effective manner. This reference deployment will assist you in building an EDH cluster on AWS by integrating Cloudera Director with an automated deployment initiated by AWS CloudFormation.

This guide is meant primarily for the deployment of the Cloudera's EDH cluster on AWS. For additional administration and support topics related to Cloudera's Enterprise Data Hub, visit Cloudera Support.

Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, configuration, network security, and other considerations discussed in this guide.

View template

Launch Quick Start

> **Note** The Launch link sets up a new Amazon VPC. To deploy Cloudera EDH into an existing Amazon VPC, see the Deployment section of this guide.

The template includes default settings that you can customize by following the instructions in this guide.

Time to deploy: Approximately 30 minutes

Cost and Licenses

This deployment uses Cloudera Director to deploy EDH automatically into a configuration of your choice. You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start. As of the date of publication, the cost for using the Quick Start for a **twelve-node cluster** ranges from approximately $12 to $82 per hour, depending on the instance type selected to meet your memory and compute requirements. The following table provides a cost estimate for twelve-node cluster.

Instance	VCPU	Memory (GiB)	Workload Type	HDFS Storage (TiB)	Storage Type	Cost/hr ($) **
m2.4xlarge	8	68.4	BALANCED	19.6875	MAGNETIC	11.76
c3.8xlarge	32	60.0	COMPUTE	7.5	SSD	20.16
i2.2xlarge	8	61.0	BALANCED	18.75	MAGNETIC	20.46
cc2.8xlarge	32	60.5	COMPUTE	38.90625	MAGNETIC	24
i2.4xlarge	16	122.0	MEMORY	37.5	SSD	40.92
hs1.8xlarge	16	117.0	BALANCED	562.5	MAGNETIC	55.2
i2.8xlarge	32	244.0	MEMORY	75	SSD	81.84

**Prices are subject to change. See the pricing pages for each AWS service you will be using or the AWS Simple Monthly Calculator for full details.

This deployment activates a 60-day trial of Cloudera Enterprise. To upgrade your version, see Managing Licenses on the Cloudera website.

AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the Getting Started section of the AWS documentation.)

- Amazon EC2 – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- Amazon VPC – The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- AWS CloudFormation – AWS CloudFormation gives you an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources that you want (like Amazon EC2 instances). You don't have to individually create and configure the resources or figure out dependencies; AWS CloudFormation handles all of that.
- IAM – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

Architecture Overview

AWS CloudFormation provides an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

The following components are deployed and configured as part of this reference deployment:

- An Amazon VPC configured with two subnets, one public and the other private.
- A NAT instance deployed into the public subnet and configured with an Elastic IP address (EIP) for outbound Internet connectivity and inbound SSH (Secure Shell)

access. The NAT instance is used for Internet access if any Amazon EC2 instances are launched within the private network.

> **Note** If you choose the option to create a new Amazon VPC, the Quick Start creates and configures the Amazon VPC, the two subnets, and the NAT instance for you. If you choose the option to deploy Cloudera EDH into an existing Amazon VPC, the Quick Start requires the described configuration.

- A Linux server instance deployed in the public subnet for downloading Cloudera Director and various configuration files and scripts
- An AWS Identity and Access Management (IAM) instance role with fine-grained permissions for access to AWS services necessary for the deployment process
- Security groups for each instance or function to restrict access to only necessary protocols and ports.
- A placement group to provide a logical grouping of instances and enable applications to participate in a low-latency, 10 Gbps network (optional)
- A fully customizable EDH cluster including worker nodes, edge nodes, and management nodes that you define based on your compute and storage requirements

In this reference architecture, we support two options for deploying Cloudera's Enterprise Data Hub within an Amazon VPC. One option is to launch all the nodes within a public subnet providing direct Internet access. The second option is to deploy all the nodes within a private subnet. The reference deployment builds both a public and private subnet, and the cluster can be deployed in either subnet using the configuration file.

EDH Cluster in a Public Subnet

This option builds the following environment in the AWS cloud.

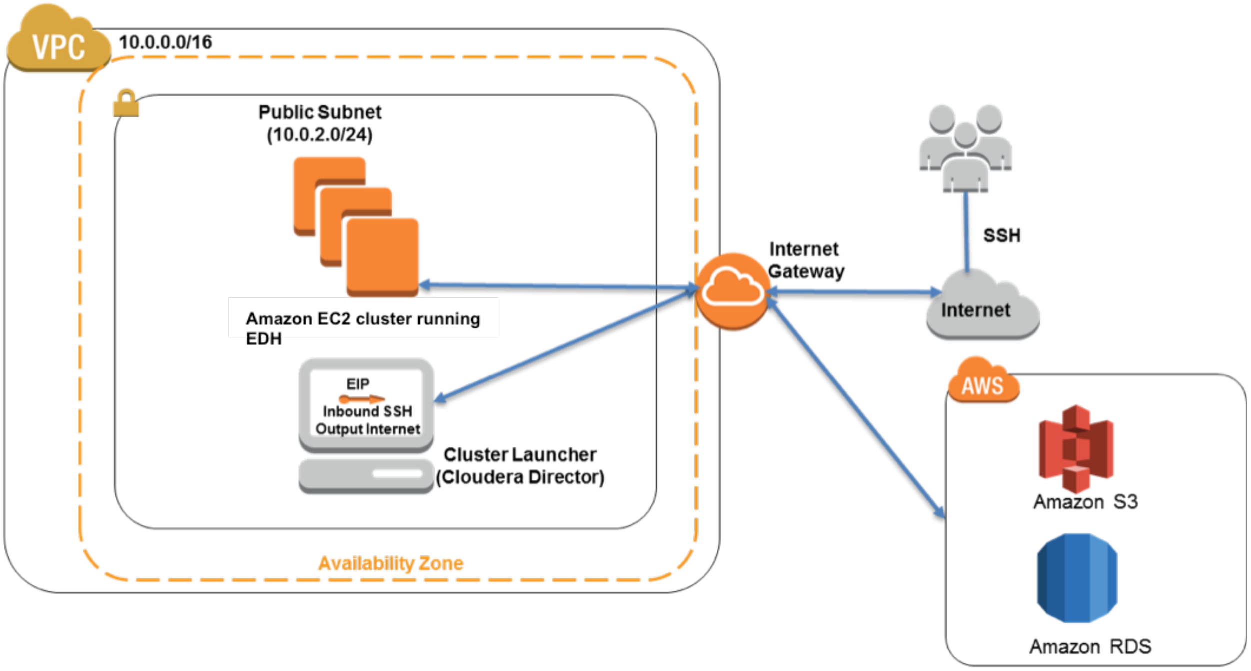


Figure 1: Public Subnet Topology

A public subnet cluster topology includes an Amazon EC2 instance (referred to as cluster launcher instance) which is launched within the public subnet. An Elastic IP Address (EIP) is assigned to the instance, and a security group allowing SSH access to the instance is created. The cluster launcher instance then builds the EDH cluster by launching all of the Hadoop related Amazon EC2 instances within the public subnet. In this topology, all the instances launched have direct access to the Internet and to any other AWS services that may be subsequently used such as Amazon S3, Amazon RDS, or others.

EDH Cluster in a Private Subnet

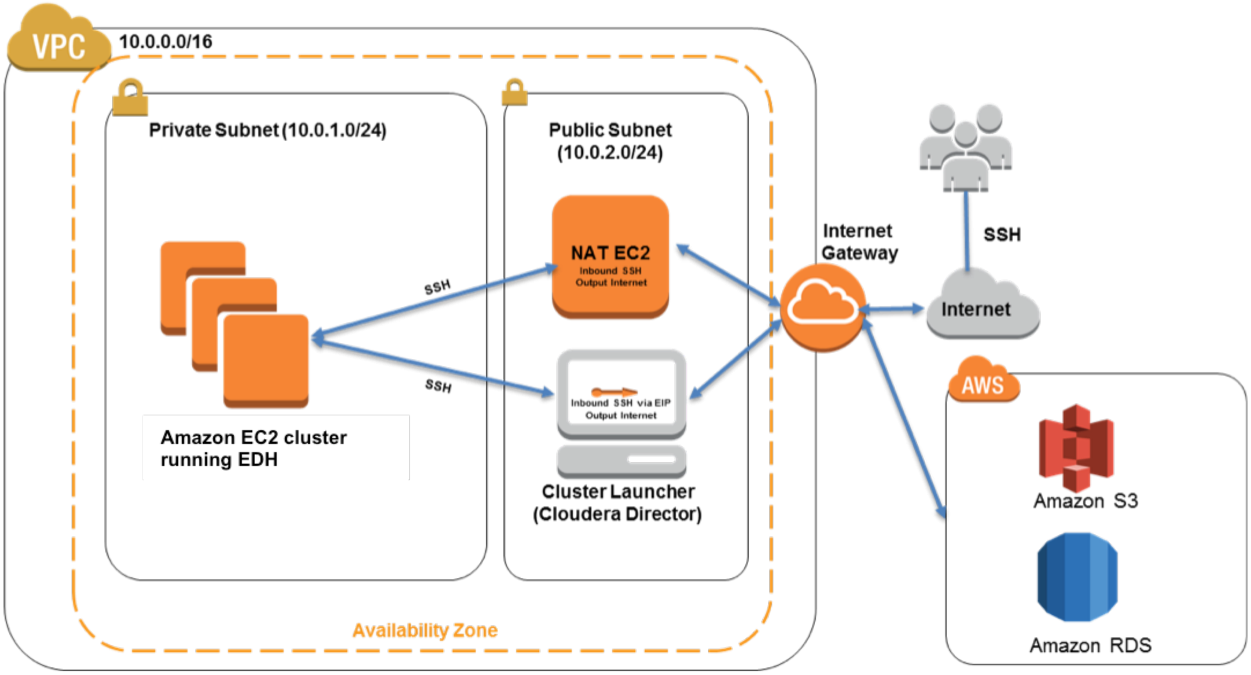


Figure 2: Private Subnet Topology

A private subnet cluster topology launches the cluster launcher instance within the public subnet. An Elastic IP Address (EIP) is assigned to the instance, and a security group allowing SSH access to the instance is created. All other Hadoop-related Amazon EC2 instances are created within the private subnet. In this topology, the Amazon EC2 instances within the EDH cluster do not have direct access to the Internet or to other AWS services. Instead, their access is routed through NAT instances residing in the public subnet. For more information about high availability for NAT instances, please see High Availability for Amazon VPC NAT Instances. This topology is more suitable if the EDH cluster doesn't require full external bandwidth to the Internet or to other AWS services such as Amazon RDS, Amazon S3, or others.

Deployment

Cloudera's Enterprise Data Hub is now easily deployable on the flexible AWS platform. This guide serves as a reference for customers who want to set up a fully customizable Hadoop cluster on demand. Building a scalable, on-demand infrastructure on AWS provides a cost-effective solution to handle large scale compute and storage requirements.

This reference deployment leverages Cloudera Director, which helps enable the delivery of an enterprise-class, elastic, self-service experience for the Enterprise Data Hub on cloud infrastructure. The flexible architecture allows you to choose the most appropriate network, compute, and storage infrastructure for your environment. You can deploy the Quick Start into an existing Amazon VPC or create a new Amazon VPC for the Cloudera EDH cluster.

What We'll Cover

The procedure for deploying Cloudera EDH on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Prepare an AWS account

- Sign up for an AWS account, if you don't already have one.
- Choose the region where you want to deploy the stack on AWS.
- Create a key pair in the region.
- Review account limits for Amazon EC2 instances, and request a limit increase, if needed.

Step 2 (option a, for a new Amazon VPC). Launch the Quick Start into your AWS account

When you launch the Quick Start using this option, the AWS CloudFormation template included with this Quick Start automates the following:

- Sets up the Amazon VPC.
- Creates various network resources needed during EDH deployment, including private and public subnets within an Amazon VPC, a NAT instance, security groups, and an IAM role.
- Starts a cluster launcher Amazon EC2 instance. This instance is used to deploy the EDH cluster using Cloudera Director.
- Downloads Cloudera Director along with the necessary scripts and configuration files.

Step 2 (option b, for an existing Amazon VPC). Launch the Quick Start into your AWS account

This option provides a separate template for launching the cluster into an existing Amazon VPC. The automation includes all the steps in option (a) except for the creation of a new Amazon VPC.

Step 3. Configure the cluster and EDH services

This step involves customizing the EDH deployment by choosing private or public subnets, Amazon EC2 instance types, the number of nodes in the cluster, and other parameters. Cloudera Director is used to configure various EDH services and their settings using a simple configuration file downloaded onto the cluster launcher Amazon EC2 instance created in Step 2. You can also choose a more complex setup involving multiple instance types, multiple security groups, a placement group, and other variables.

Step 4. Deploy the EDH cluster

After you have modified the configuration files to suit your compute and storage requirements, you can launch the EDH cluster using a simple command line executable.

Step 1. Prepare an AWS Account

1. If you don't already have an AWS account, create one at http://aws.amazon.com by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

2. Use the region selector in the navigation bar to choose the Amazon EC2 region where you want to deploy the EDH cluster on AWS.

   Amazon EC2 locations are composed of *regions* and *Availability Zones*. Regions are dispersed and located in separate geographic areas. All Amazon EC2 instances (except R3 instances) can be launched in any of the regions. R3 instances are currently available in all AWS regions except GovCloud (US), China (Beijing), and South America (São Paulo).

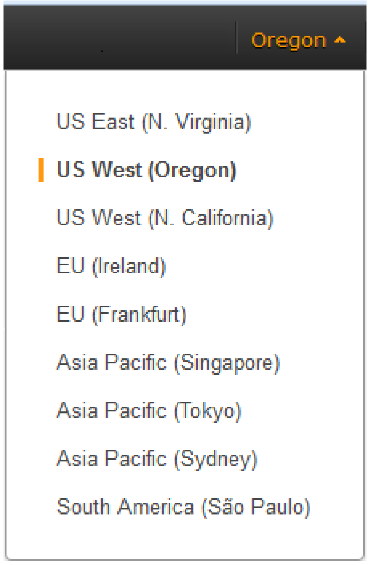


Figure 3: Choosing an Amazon EC2 Region

> **Tip** Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

3. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs**, **Create Key Pair**, type a name, and then choose **Create**.

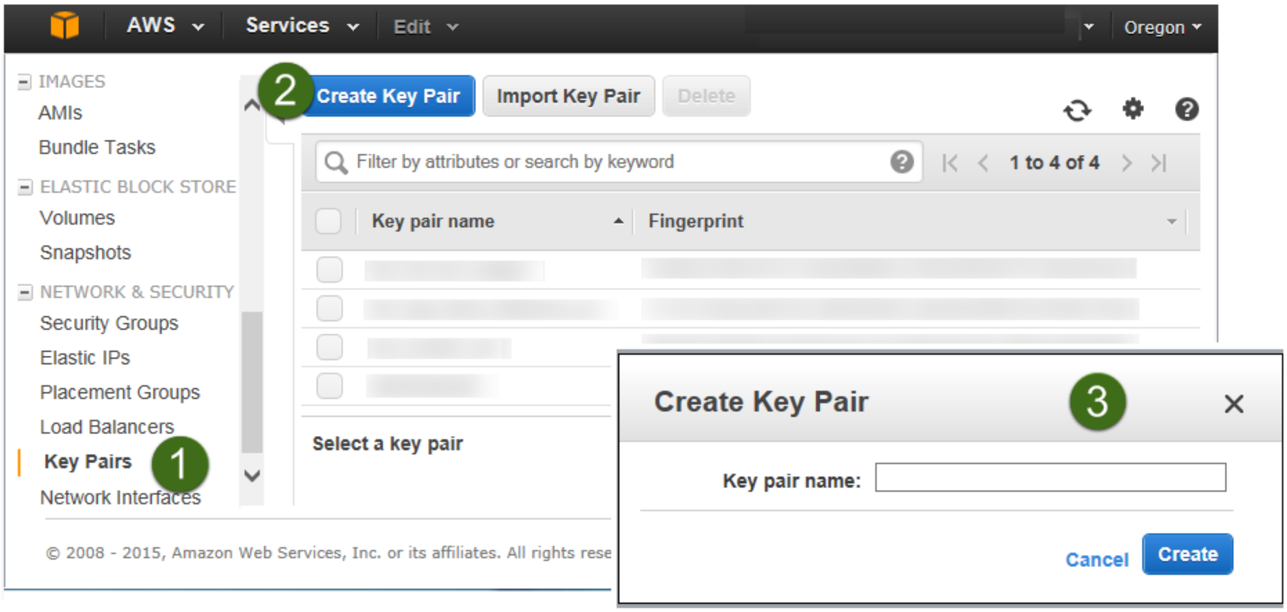
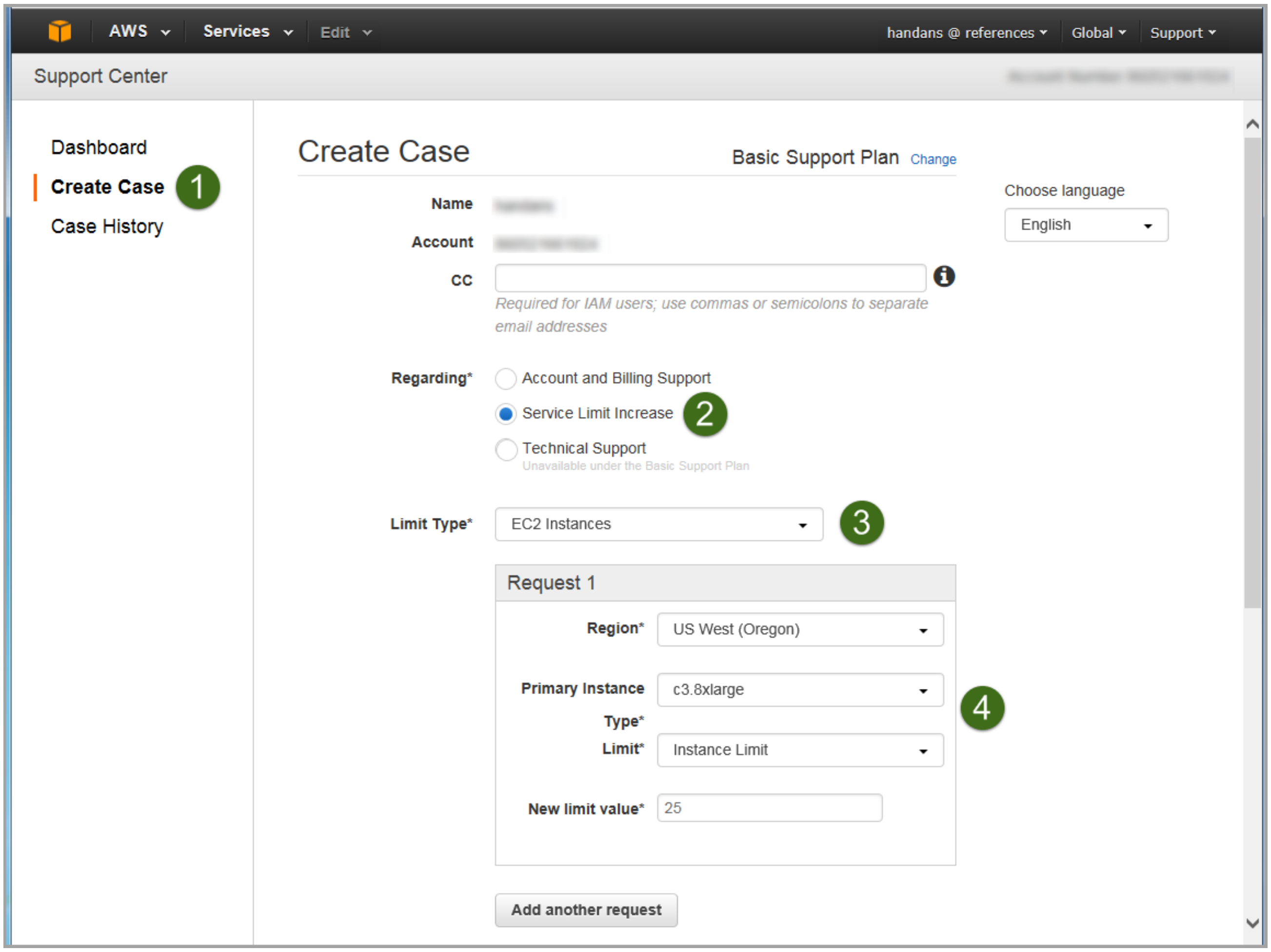


Figure 4: Creating a Key Pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log into your instances, you must create a key pair. On Linux, we use the key pair to authenticate SSH login.

4. If necessary, request a service limit increase for the Amazon EC2 instance types that you intend to deploy. Depending on the instance type, the default limit for the number of instances that can be run varies from 2 to 20. You may check the default instance limits on the Amazon EC2 FAQ page. If you have existing deployments that leverage the instance type you need, or if you plan on exceeding this default with this reference deployment, you will need to request an Amazon Amazon EC2 instance service limit increase. It might take a few days for the new service limit to become effective. For more information, see Amazon EC2 Service Limits in the AWS documentation.



Support Center [redacted]

Create Case

Basic Support Plan Change

Choose language: English

Name: [redacted]

Account: [redacted]

CC:

Required for IAM users; use commas or semicolons to separate email addresses

Regarding*:
○ Account and Billing Support
● Service Limit Increase
○ Technical Support
Unavailable under the Basic Support Plan

Limit Type*: EC2 Instances

Request 1

Region*: US West (Oregon)

Primary Instance Type*: c3.8xlarge

Limit*: Instance Limit

New limit value*: 25

Figure 5: Requesting a Service Limit Increase

Step 2(a). Launch the Quick Start into Your AWS Account (New Amazon VPC)

In this step, you will launch an AWS CloudFormation template that automates the following:

- Configures the Amazon VPC that provides the base AWS network infrastructure for your EDH deployment.
- Creates the network resources needed for EDH deployment, including public and private subnets within the Amazon VPC, a NAT instance launched within the public subnet, security groups, and an IAM role.
- Starts an Amazon EC2 instance running Linux (RedHat) in the public subnet. This instance serves as a launcher node for the Cloudera cluster, and initiates cluster deployment.
- Downloads Cloudera Director along with the necessary scripts and configuration files. Cloudera Director is used to configure the EDH cluster.

All the steps here are fully automated by AWS CloudFormation; the only mandatory input expected by the template is *KeyName*, which is the name of the key pair you created in step 1, when you set up your AWS account.

1. Launch the AWS CloudFormation template into your AWS account.

   Launch
   (for new VPC)

   The template is launched in the US West (Oregon) region by default. You can change the region by using the region selector in the navigation bar.

   This stack takes approximately 30 minutes to create.

   > **Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. As of the date of publication, the cost for using the Quick Start for a twelve-node cluster ranges from approximately $12 to $82 an hour, depending on the instance type selected. See the Cost and Licenses section for cost estimates for different instance types. Prices are subject to change. See the pricing pages for each AWS service you will be using in this Quick Start for full details.

   You can also download the template to use it as a starting point for your own implementation.

2. On the **Select Template** page, keep the default settings for the stack name and template source, and then choose **Next**.

3. On the **Specify Parameters** page, review the parameters for the template. Provide a value for the *KeyName* parameter. You can also customize the following additional parameters. The AWS CloudFormation template uses these to generate a cluster configuration file. When you're done, choose **Next**.

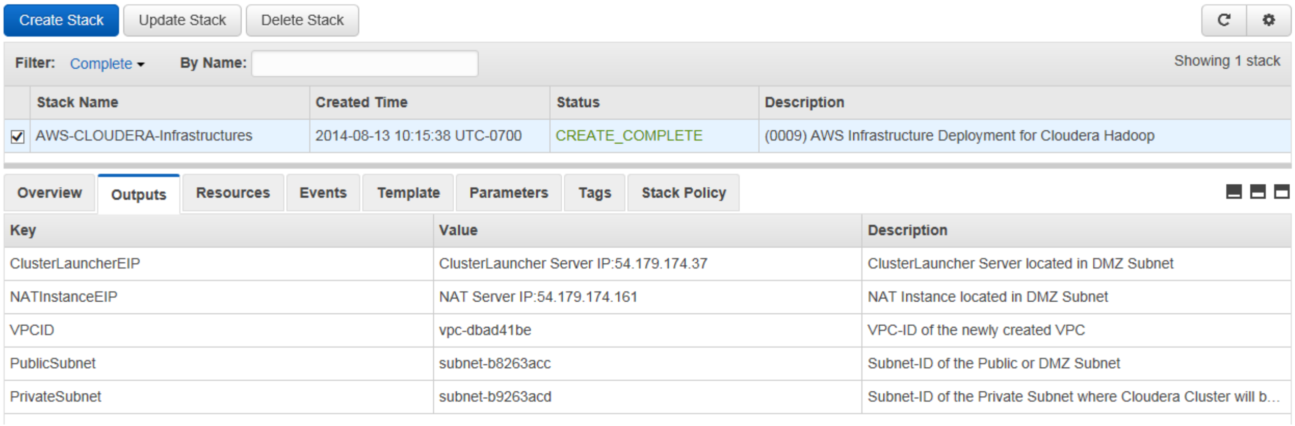
Parameter	**Default**	**Description**
VPCCIDR	10.0.0.0/16	CIDR block for the Amazon VPC you are creating.
DMZCIDR	10.0.2.0/24	CIDR block for the public DMZ subnet located in the new Amazon VPC.
PrivSubCIDR	10.0.1.0/24	CIDR block for private subnet where EDH will be deployed.
RemoteAccessCIDR	0.0.0.0/0	IP CIDR from which you are likely to SSH into the EDH launcher instance.
KeyName	*Requires input*	An existing public/private key pair, which allows you to connect securely to your instance after it launches. This is the key pair you created in Step 1, when you prepared your AWS account.
NATInstanceType	m1.small	Amazon EC2 instance type for the NAT instances.
ClusterLauncherType	t2.small	Amazon EC2 instance type for the EDH launcher instance.

After the cluster launcher instance is deployed, you can make additional changes to the EDH deployment by modifying the configuration file.

4. On the **Options** page, choose **Next**.

5. On the **Review** page, review and confirm the settings, and then choose **Create** to deploy the stack.

When the status field displays CREATE_COMPLETE and the launcher instance has been created successfully, as shown in Figure 6, you can continue to the next step to configure the cluster.



	Stack Name	Created Time	Status	Description
☑	AWS-CLOUDERA-Infrastructures	2014-08-13 10:15:38 UTC-0700	CREATE_COMPLETE	(0009) AWS Infrastructure Deployment for Cloudera Hadoop

Key	Value	Description
ClusterLauncherEIP	ClusterLauncher Server IP:54.179.174.37	ClusterLauncher Server located in DMZ Subnet
NATInstanceEIP	NAT Server IP:54.179.174.161	NAT Instance located in DMZ Subnet
VPCID	vpc-dbad41be	VPC-ID of the newly created VPC
PublicSubnet	subnet-b8263acc	Subnet-ID of the Public or DMZ Subnet
PrivateSubnet	subnet-b9263acd	Subnet-ID of the Private Subnet where Cloudera Cluster will b...

Figure 6: Successful Creation of Launcher Instance

Step 2(b). Launch the Quick Start into Your AWS Account (Existing Amazon VPC)

If you have an Amazon VPC already constructed, you can still use this Quick Start to launch the cluster. The deployment steps are same as in step 2(a), except that you need to input the settings associated with your existing Amazon VPC—specifically, the Amazon VPC ID and the subnet IDs of the private and public subnets—during launch. All other options remain the same.

Launch (for existing VPC)

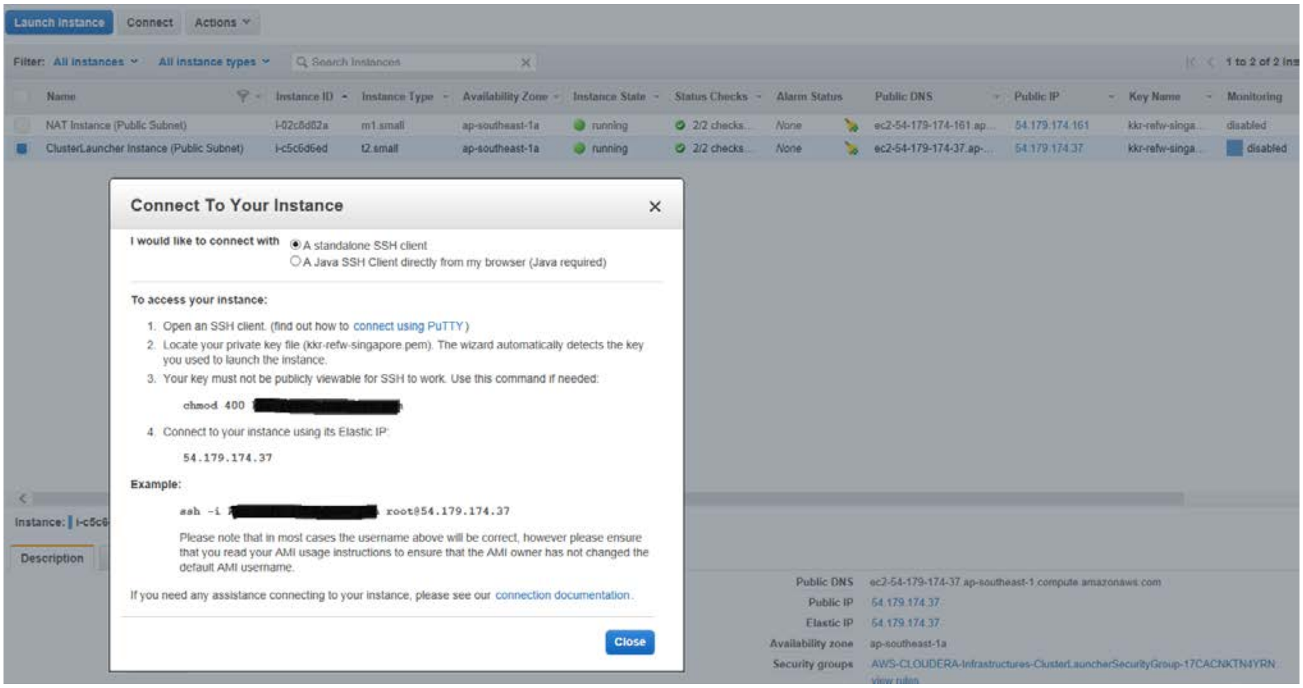
Parameter	Default	Description
PublicSubnet	None	ID of an existing public subnet in your Amazon VPC.
PrivateSubnet	None	ID of an existing private subnet where Cloudera nodes will be deployed.
VPC	None	The existing Amazon VPC where you want to deploy the Cloudera nodes.

Step 3. Configure the Cluster and EDH Services

In this step, you will use SSH to connect to the cluster launcher Amazon EC2 instance you created in step 2, and configure EDH services.

1. Connect to the cluster launcher instance.

   On the EC2 dashboard, click the **Connect** tab under **EC Instances**, as shown in Figure 7. You will need your private key to launch the instance.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
NAT Instance (Public Subnet)	i-02c8d82a	m1.small	ap-southeast-1a	running	2/2 checks...	None	ec2-54-179-174-161.ap...	54.179.174.161	kkr-refw-singa...	disabled
ClusterLauncher Instance (Public Subnet)	i-c5c6d6ed	t2.small	ap-southeast-1a	running	2/2 checks...	None	ec2-54-179-174-37.ap-...	54.179.174.37	kkr-refw-singa...	disabled

Connect To Your Instance

I would like to connect with ● A standalone SSH client ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (kkr-refw-singapore.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 [redacted]
```

4. Connect to your instance using its Elastic IP:

```
54.179.174.37
```

Example:

```
ssh -i [redacted] root@54.179.174.37
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

Close

Public DNS ec2-54-179-174-37.ap-southeast-1.compute.amazonaws.com
Public IP 54.179.174.37
Elastic IP 54.179.174.37
Availability zone ap-southeast-1a
Security groups AWS-CLOUDERA-Infrastructures-ClusterLauncherSecurityGroup-17CACNKTN4YRN view rules

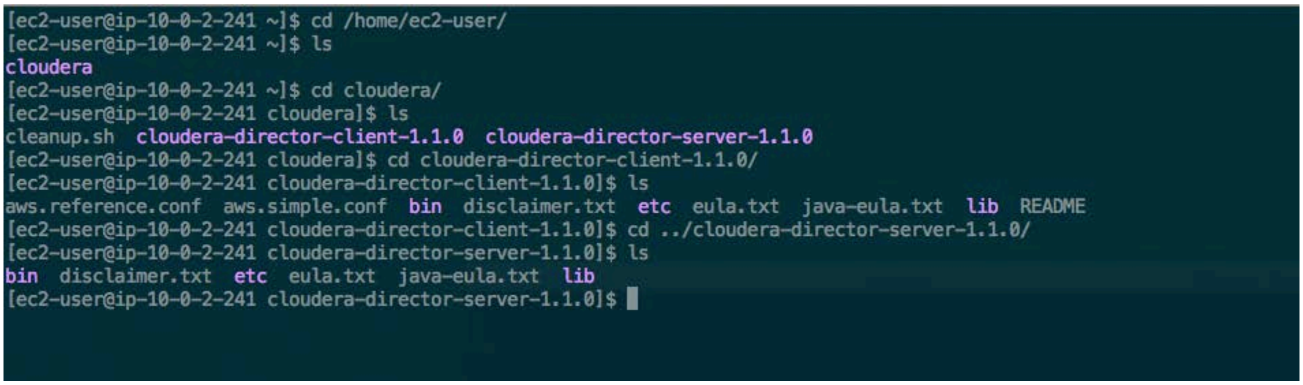
Figure 7: Connecting to the Cluster Launcher Using SSH

2. Configure EDH services.

   When you launch the cluster launcher instance, it will automatically download Cloudera Director and build a configuration file based on the resources created by the AWS CloudFormation template, such as Amazon VPC, private subnet, and public subnet. You can then modify the configuration file using the steps below to launch the most appropriate cluster for your scenario. The launcher instance is automatically assigned an Identity and Access Management (IAM) root role to grant access to all the AWS resources that may be needed by the default configuration created in step 1.

   Because the launcher instance is started with an IAM role, there is no need to distribute AWS credentials to deploy the EDH cluster. Because role credentials are temporary and rotated automatically, you don't have to manage credentials. For example, you don't have to worry about rotating credentials. For more detail about the benefits of the IAM role, see Using IAM Roles to Delegate Permissions to Applications that Run on Amazon EC2.

   Figure 8 lists the files that are downloaded automatically during launch.



```
[ec2-user@ip-10-0-2-241 ~]$ cd /home/ec2-user/
[ec2-user@ip-10-0-2-241 ~]$ ls
cloudera
[ec2-user@ip-10-0-2-241 ~]$ cd cloudera/
[ec2-user@ip-10-0-2-241 cloudera]$ ls
cleanup.sh  cloudera-director-client-1.1.0  cloudera-director-server-1.1.0
[ec2-user@ip-10-0-2-241 cloudera]$ cd cloudera-director-client-1.1.0/
[ec2-user@ip-10-0-2-241 cloudera-director-client-1.1.0]$ ls
aws.reference.conf  aws.simple.conf  bin  disclaimer.txt  etc  eula.txt  java-eula.txt  lib  README
[ec2-user@ip-10-0-2-241 cloudera-director-client-1.1.0]$ cd ../cloudera-director-server-1.1.0/
[ec2-user@ip-10-0-2-241 cloudera-director-server-1.1.0]$ ls
bin  disclaimer.txt  etc  eula.txt  java-eula.txt  lib
[ec2-user@ip-10-0-2-241 cloudera-director-server-1.1.0]$ 
```

Figure 8: Deployment Scripts and Configuration Files

> **Important** Before you begin deployment, copy the private keyfile (.pem) used to launch to the launcher instance. For example, you can copy the keyfile using the following command line:
>
> `scp –i mykey.pem mykey.pem ec2-user@cluster-launcher-public-ip:/home/ec2-user/mykey.pem`

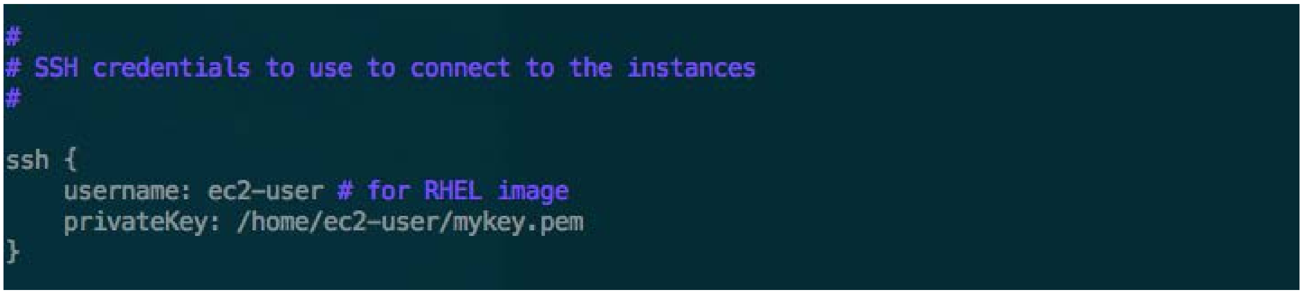
3. Modify the configuration file.

   There are two configuration files that are customizable during deployment:

   – aws.simple.conf for configuring simple clusters

   – aws.reference.conf for configuring complex clusters

   The only modification **required** for deployment is to replace `privateKey` with your private keyfile path.



```
#
# SSH credentials to use to connect to the instances
#

ssh {
    username: ec2-user # for RHEL image
    privateKey: /home/ec2-user/mykey.pem
}
```

Figure 9: Modifying the Private Keyfile Path in the Configuration File

You can make additional changes to the deployment configuration (for example, choosing instance type, node count, subnet type, EDH services, or installation versions) by further modifying the configuration file. The configuration files include baseline values based on the various resources (such as Amazon VPC ID and subnet ID) created during the launch of the AWS CloudFormation stack. By default, all Cloudera nodes are launched in the private subnet for security reasons. For more information about configuration parameters, see the Cloudera Director User Guide.

Step 4. Deploy the EDH Cluster

Cloudera Director supports two options for cluster deployment:

- Option 1: You can deploy using the CLI and manage the nodes manually.
- Option 2 (recommended): You can deploy using the Cloudera Director Server to manage multiple clusters. Cloudera Director provides a simple interface to deploy, scale, and terminate clusters, and helps you manage the cluster.

Option 1: Deploy Using the CLI, No Server

To deploy the EDH cluster, run the **cloudera-director** executable using one of the configuration files, as follows.

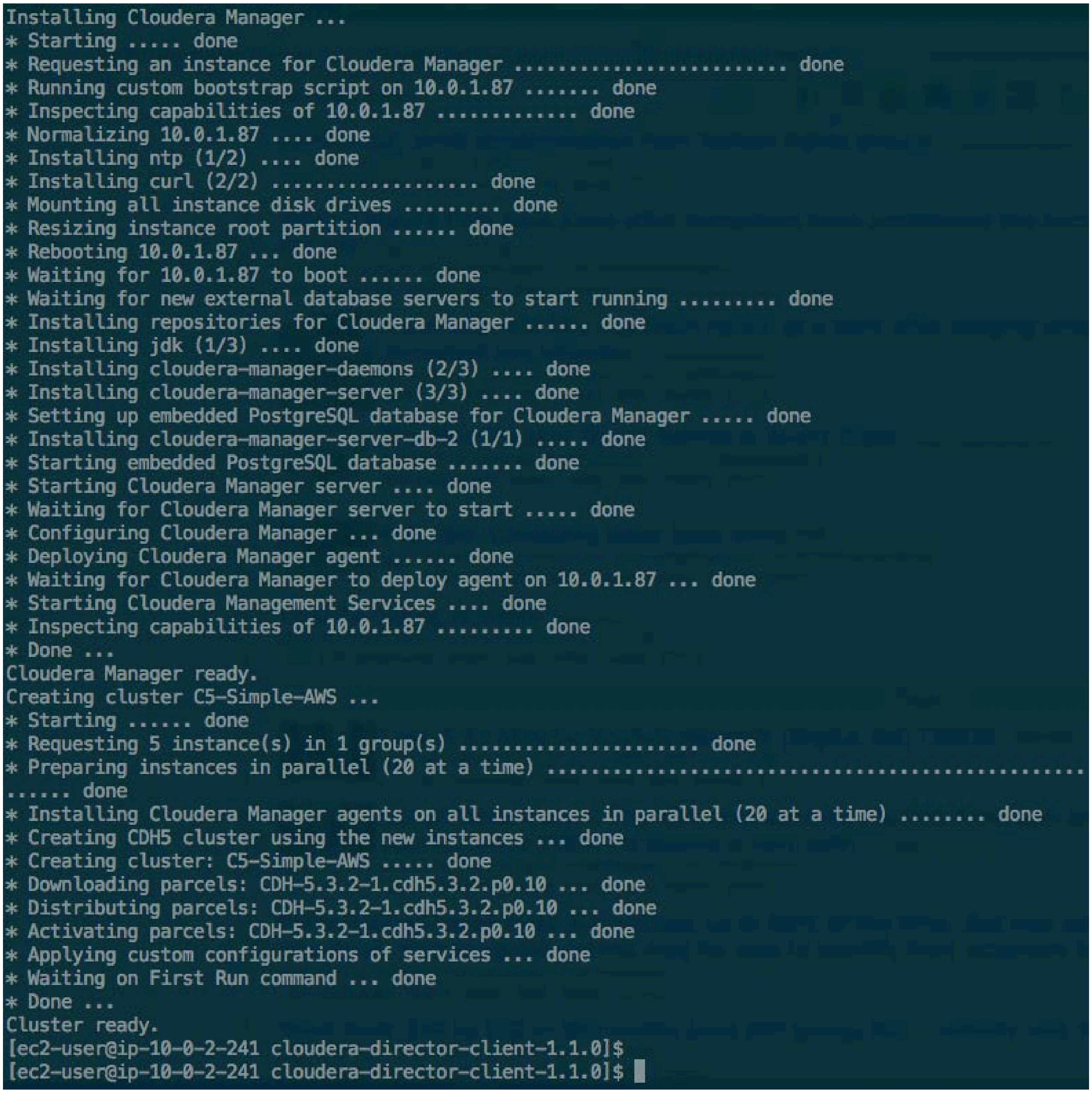
For a simple cluster:

```
./bin/cloudera-director bootstrap aws.simple.conf
```

For an advanced cluster:

```
./bin/cloudera-director bootstrap aws.reference.conf
```

Figure 10 shows a typical sequence of a completed EDH deployment using Cloudera Director.



```
Installing Cloudera Manager ...
* Starting ..... done
* Requesting an instance for Cloudera Manager .......................... done
* Running custom bootstrap script on 10.0.1.87 ....... done
* Inspecting capabilities of 10.0.1.87 ............. done
* Normalizing 10.0.1.87 .... done
* Installing ntp (1/2) .... done
* Installing curl (2/2) ................... done
* Mounting all instance disk drives ......... done
* Resizing instance root partition ...... done
* Rebooting 10.0.1.87 ... done
* Waiting for 10.0.1.87 to boot ...... done
* Waiting for new external database servers to start running ......... done
* Installing repositories for Cloudera Manager ...... done
* Installing jdk (1/3) .... done
* Installing cloudera-manager-daemons (2/3) .... done
* Installing cloudera-manager-server (3/3) .... done
* Setting up embedded PostgreSQL database for Cloudera Manager ..... done
* Installing cloudera-manager-server-db-2 (1/1) ..... done
* Starting embedded PostgreSQL database ....... done
* Starting Cloudera Manager server .... done
* Waiting for Cloudera Manager server to start ..... done
* Configuring Cloudera Manager ... done
* Deploying Cloudera Manager agent ...... done
* Waiting for Cloudera Manager to deploy agent on 10.0.1.87 ... done
* Starting Cloudera Management Services .... done
* Inspecting capabilities of 10.0.1.87 ......... done
* Done ...
Cloudera Manager ready.
Creating cluster C5-Simple-AWS ...
* Starting ...... done
* Requesting 5 instance(s) in 1 group(s) ...................... done
* Preparing instances in parallel (20 at a time) ...............................................
...... done
* Installing Cloudera Manager agents on all instances in parallel (20 at a time) ........ done
* Creating CDH5 cluster using the new instances ... done
* Creating cluster: C5-Simple-AWS ..... done
* Downloading parcels: CDH-5.3.2-1.cdh5.3.2.p0.10 ... done
* Distributing parcels: CDH-5.3.2-1.cdh5.3.2.p0.10 ... done
* Activating parcels: CDH-5.3.2-1.cdh5.3.2.p0.10 ... done
* Applying custom configurations of services ... done
* Waiting on First Run command ... done
* Done ...
Cluster ready.
[ec2-user@ip-10-0-2-241 cloudera-director-client-1.1.0]$
[ec2-user@ip-10-0-2-241 cloudera-director-client-1.1.0]$
```

Figure 10: EDH Deployment Sequence

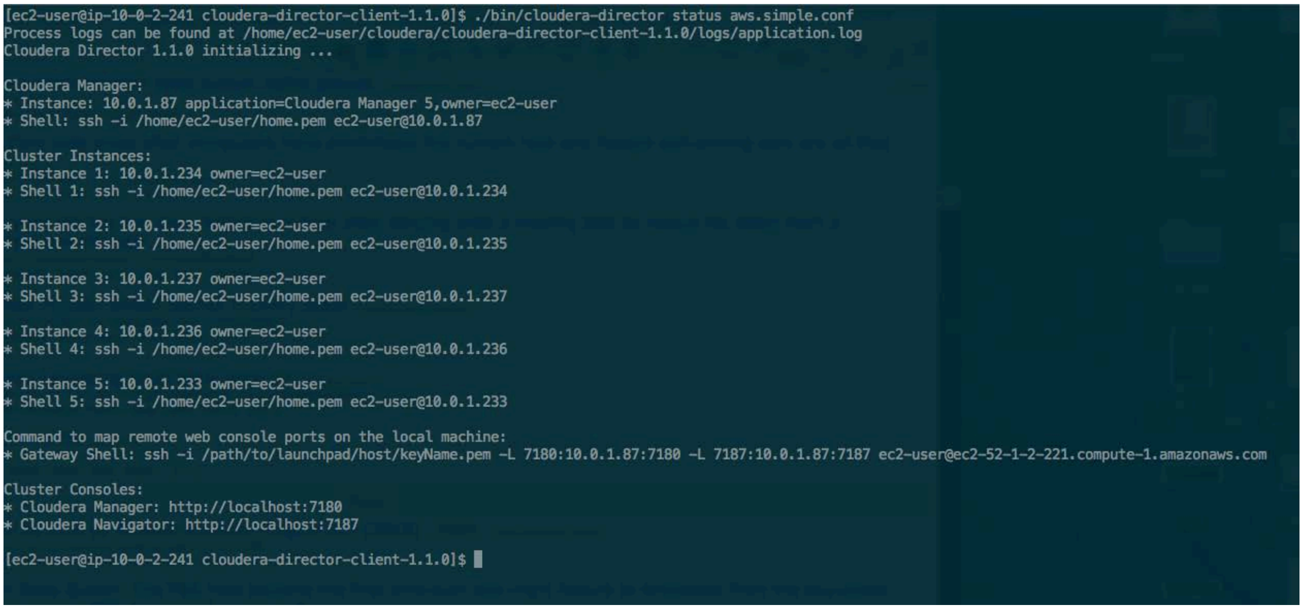
Cloudera Director also supports other command arguments, such as terminate and status query.

For example, for a simple cluster:

```
./bin/cloudera-director status aws.simple.conf
```

For an advanced cluster:

```
./bin/cloudera-director status aws.reference.conf
```



```
[ec2-user@ip-10-0-2-241 cloudera-director-client-1.1.0]$ ./bin/cloudera-director status aws.simple.conf
Process logs can be found at /home/ec2-user/cloudera/cloudera-director-client-1.1.0/logs/application.log
Cloudera Director 1.1.0 initializing ...

Cloudera Manager:
* Instance: 10.0.1.87 application=Cloudera Manager 5,owner=ec2-user
* Shell: ssh -i /home/ec2-user/home.pem ec2-user@10.0.1.87

Cluster Instances:
* Instance 1: 10.0.1.234 owner=ec2-user
* Shell 1: ssh -i /home/ec2-user/home.pem ec2-user@10.0.1.234

* Instance 2: 10.0.1.235 owner=ec2-user
* Shell 2: ssh -i /home/ec2-user/home.pem ec2-user@10.0.1.235

* Instance 3: 10.0.1.237 owner=ec2-user
* Shell 3: ssh -i /home/ec2-user/home.pem ec2-user@10.0.1.237

* Instance 4: 10.0.1.236 owner=ec2-user
* Shell 4: ssh -i /home/ec2-user/home.pem ec2-user@10.0.1.236

* Instance 5: 10.0.1.233 owner=ec2-user
* Shell 5: ssh -i /home/ec2-user/home.pem ec2-user@10.0.1.233

Command to map remote web console ports on the local machine:
* Gateway Shell: ssh -i /path/to/launchpad/host/keyName.pem -L 7180:10.0.1.87:7180 -L 7187:10.0.1.87:7187 ec2-user@ec2-52-1-2-221.compute-1.amazonaws.com

Cluster Consoles:
* Cloudera Manager: http://localhost:7180
* Cloudera Navigator: http://localhost:7187

[ec2-user@ip-10-0-2-241 cloudera-director-client-1.1.0]$
```

Figure 11: EDH Deployment Sequence with Status Query

Option 2: Using Cloudera Director Server

The Cloudera Director Server deployment option is more suitable if you want to deploy multiple clusters and want to manage them through a server.

1. Start the Cloudera Director Server from the server directory using the following command:

   ```
   ./bin/cloudera-director-server
   ```

   This command starts the server on port 7189 (default) of the cluster launcher instance and runs in the foreground. You can optionally use the –port=*port* argument to specify a different port. Alternatively, to start Cloudera Director in the background, use a bin/start script.

2. Deploy the cluster using one of the following commands.

   For a simple cluster:

   ```
   ./bin/cloudera-director bootstrap-remote aws.simple.conf \
   --lp.remote.hostAndPort=127.0.0.1:7189 \
   --lp.remote.username=admin \
   ```

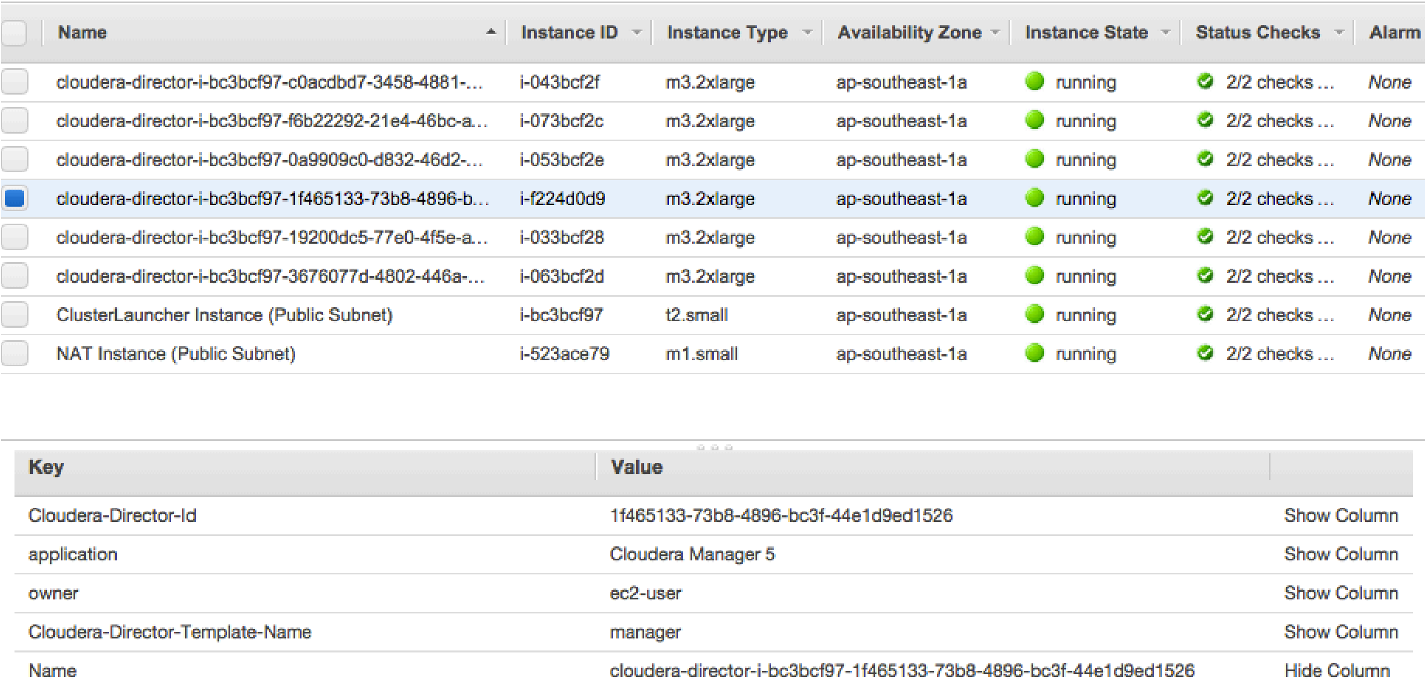
```
--lp.remote.password=admin
```

For an advanced cluster:

```
./bin/cloudera-director bootstrap-remote aws.reference.conf \
--lp.remote.hostAndPort=127.0.0.1:7189 \
--lp.remote.username=admin \
--lp.remote.password=admin
```

Accessing the Cluster with Cloudera Manager

Once the EDH cluster has been launched, you can connect to Cloudera Manager to access the cluster and add any additional services or other maintenance operations. You can connect to Cloudera Manager from a local host by forwarding the local port to the remote IP/Port where Cloudera Manager is running. The instances are associated with various Tags, which can be used to find more information about individual nodes. For example, Figure 12 shows the node where the Cloudera Manager application is running.



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
□	cloudera-director-i-bc3bcf97-c0acdbd7-3458-4881-...	i-043bcf2f	m3.2xlarge	ap-southeast-1a	● running	✓ 2/2 checks ...	None
□	cloudera-director-i-bc3bcf97-f6b22292-21e4-46bc-a...	i-073bcf2c	m3.2xlarge	ap-southeast-1a	● running	✓ 2/2 checks ...	None
□	cloudera-director-i-bc3bcf97-0a9909c0-d832-46d2-...	i-053bcf2e	m3.2xlarge	ap-southeast-1a	● running	✓ 2/2 checks ...	None
☑	cloudera-director-i-bc3bcf97-1f465133-73b8-4896-b...	i-f224d0d9	m3.2xlarge	ap-southeast-1a	● running	✓ 2/2 checks ...	None
□	cloudera-director-i-bc3bcf97-19200dc5-77e0-4f5e-a...	i-033bcf28	m3.2xlarge	ap-southeast-1a	● running	✓ 2/2 checks ...	None
□	cloudera-director-i-bc3bcf97-3676077d-4802-446a-...	i-063bcf2d	m3.2xlarge	ap-southeast-1a	● running	✓ 2/2 checks ...	None
□	ClusterLauncher Instance (Public Subnet)	i-bc3bcf97	t2.small	ap-southeast-1a	● running	✓ 2/2 checks ...	None
□	NAT Instance (Public Subnet)	i-523ace79	m1.small	ap-southeast-1a	● running	✓ 2/2 checks ...	None

Key	Value	
Cloudera-Director-Id	1f465133-73b8-4896-bc3f-44e1d9ed1526	Show Column
application	Cloudera Manager 5	Show Column
owner	ec2-user	Show Column
Cloudera-Director-Template-Name	manager	Show Column
Name	cloudera-director-i-bc3bcf97-1f465133-73b8-4896-bc3f-44e1d9ed1526	Hide Column

Figure 12: Using Instance Tags

In Figure 12, Cloudera Manager is running on the instance with private IP 10.0.1.224 on port 7180. We can forward localhost:7180 to Cloudera Manager using its public IP with the following command:

```
ssh -i mykey.pem -L 7180:10.0.1.224:7180 \
-L 7187:10.0.1.224:7187 ec2-user@cluster-launcher-public-ip
```

When port forwarding is complete, open the browser on the local host, go to http://localhost:7180 and log in with admin/admin.

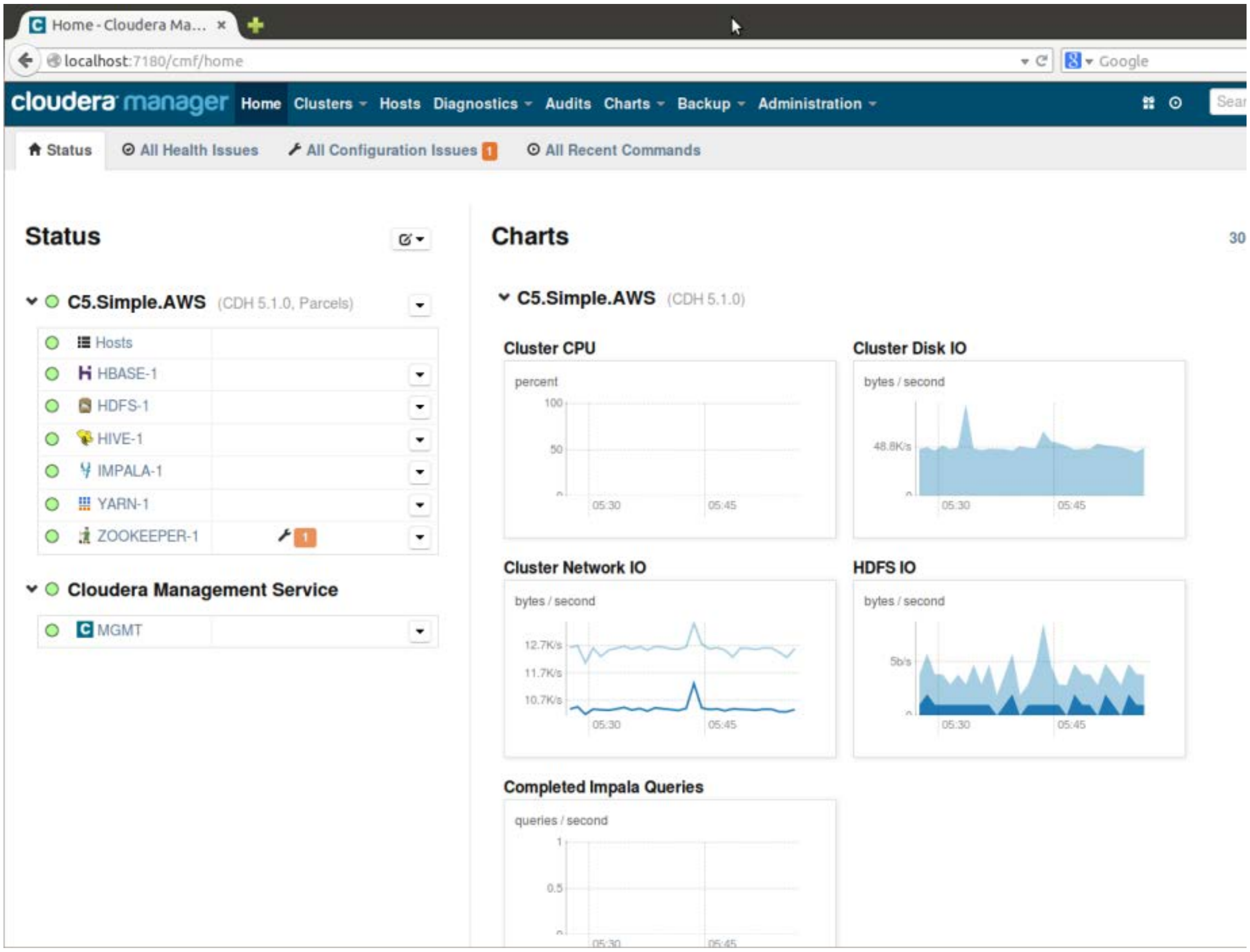
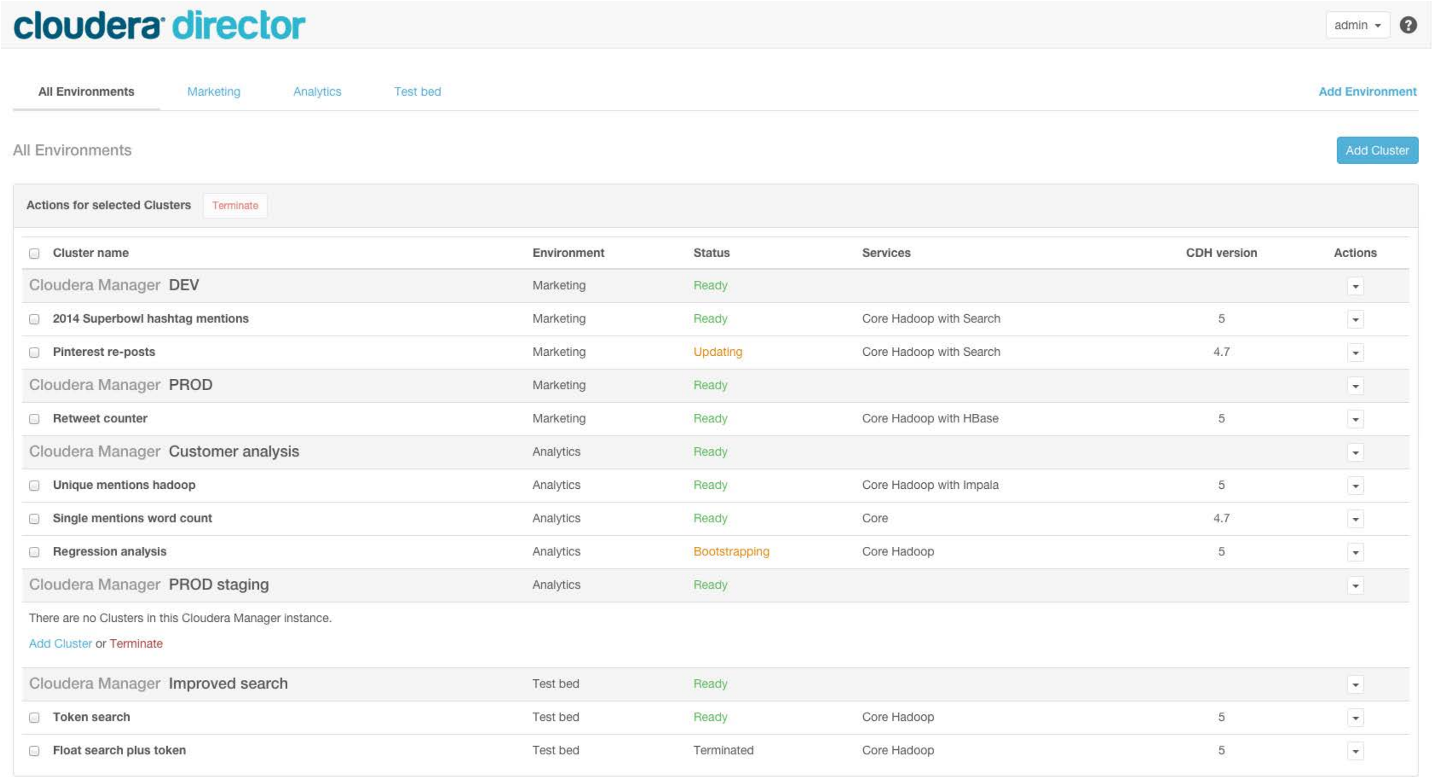


Figure 13: Connecting to Cloudera Manager

Managing the Cluster with Cloudera Director

For ongoing management of the cluster or to launch additional clusters, you can use Cloudera Director's web interface. To connect to Cloudera Director, you need to set up a SOCKS proxy for security purposes. For more information, see the SOCKS proxy documentation on the Cloudera website.

From Cloudera Director's web interface you can clone the cluster you just created, dynamically scale the cluster, or launch new clusters. You can also view all of your clusters from a centralized dashboard.



☐ Cluster name	Environment	Status	Services	CDH version	Actions
Cloudera Manager DEV	Marketing	Ready			▾
☐ 2014 Superbowl hashtag mentions	Marketing	Ready	Core Hadoop with Search	5	▾
☐ Pinterest re-posts	Marketing	Updating	Core Hadoop with Search	4.7	▾
Cloudera Manager PROD	Marketing	Ready			▾
☐ Retweet counter	Marketing	Ready	Core Hadoop with HBase	5	▾
Cloudera Manager Customer analysis	Analytics	Ready			▾
☐ Unique mentions hadoop	Analytics	Ready	Core Hadoop with Impala	5	▾
☐ Single mentions word count	Analytics	Ready	Core	4.7	▾
☐ Regression analysis	Analytics	Bootstrapping	Core Hadoop	5	▾
Cloudera Manager PROD staging	Analytics	Ready			▾
There are no Clusters in this Cloudera Manager instance. Add Cluster or Terminate					
Cloudera Manager Improved search	Test bed	Ready			▾
☐ Token search	Test bed	Ready	Core Hadoop	5	▾
☐ Float search plus token	Test bed	Terminated	Core Hadoop	5	▾

Figure 14: Cloudera Director

Storage Configuration

This deployment uses Amazon EC2 instance stores as the primary storage for HDFS data. This disk storage is attached to the instance and provides a temporary block-level storage for use with an instance. The size of an instance store ranges from 900 MiB to up to 48 TiB and varies by instance type according to the following table.

Instance Type	Instance Store Volumes
m2.4xlarge	2 x 840 GiB (1680 GiB)
c3.8xlarge	2 x 320 GiB SSD (640 GiB)
i2.2xlarge	2 x 800 GiB SSD (1600 GiB)
cc2.8xlarge	4 x 840 GiB (3360 GiB)
r3.8xlarge	2 X 320 GiB (640 GiB)

Instance Type	Instance Store Volumes
i2.4xlarge	4 x 800 GiB SSD (3200 GiB)
hs1.8xlarge	24 x 2048 GiB (48 TiB)
i2.8xlarge	8 x 800 GiB SSD (6400 GiB)

Instance store volumes are usable only from a single instance during its lifetime; they can't be detached and then attached to another instance. However they persist during restarts. Since these are local stores, they carry performance benefits during I/O operations since data doesn't have to be shipped over the network. For more information about instance stores, see the Amazon EC2 documentation.

Backup

For backup purpose, we recommend using Amazon S3 to keep a copy of HDFS data from instance stores. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Operating System and AMI

Launchpad supports RedHat version 6.4. A default 64-bit AMI is chosen in the configuration file to be installed on the instance. If you need to install other versions, please refer to Launchpad document on OS support and customize the AMI. For a list of different AMIs across regions, visit Red Hat and Amazon Web Services.

Security

The AWS cloud provides a scalable, highly reliable platform that helps enable customers to deploy applications and data quickly and securely.

When you build systems on the AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. In turn, you assume responsibility and management of the guest operating system (including updates and security patches), other associated applications, as well as the configuration of the AWS-provided security group firewall. For more information about security on AWS, visit the AWS Security Center.

AWS Identity and Access Management (IAM)

This solution leverages an IAM role with least privileged access. It is not necessary or recommended to store SSH keys or secret keys or access keys on the provisioned instances.

OS Security

The root user on cluster nodes can only be accessed using the SSH key specified during the deployment process. Amazon Web Services does not store these SSH keys, so if you lose your SSH key you can lose access to these instances.

Operating system patches are your responsibility and should be performed on a periodic basis.

Security Groups

A *security group* acts as a firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. The new rules are automatically applied to all instances that are associated with the security group.

The security groups created and assigned to the individual instances as part of this solution are restricted as much as possible while allowing access to the various functions needed by Hadoop. We recommend reviewing security groups to further restrict access as needed once the EDH cluster is up and running.

Additional Resources

AWS services

- Getting Started
  http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/intro.html
- AWS CloudFormation
  http://aws.amazon.com/documentation/cloudformation/
- Amazon EC2
  - User's guide:
    http://docs.aws.amazon.com/ec2/
  - Regions and Availability Zones:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html
  - Key pairs:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html
  - Instance stores:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#instance-storage-concepts
  - FAQ:
    http://aws.amazon.com/ec2/faqs
- Amazon Identity and Access Management
  - User's guide:
    http://aws.amazon.com/documentation/iam/
  - Benefits of the IAM role:
    http://docs.aws.amazon.com/IAM/latest/UserGuide/role-usecase-ec2app.html
- Amazon VPC
  - Documentation:
    http://aws.amazon.com/documentation/vpc/
  - High availability for NAT instances
    https://aws.amazon.com/articles/2781451301784570
- AWS Security Center
  http://aws.amazon.com/security/

- Red Hat and AWS
  http://aws.amazon.com/partners/redhat/

Cloudera

- Cloudera website
  http://www.cloudera.com
- Cloudera documentation
  http://www.cloudera.com/content/cloudera/en/documentation.html
- Cloudera Director
  http://www.cloudera.com/content/cloudera/en/documentation/cloudera-director/latest/PDF/cloudera-director.pdf
- Cloudera Support
  http://www.cloudera.com/content/cloudera/en/products-and-services/cloudera-support.html
- Managing licenses
  http://www.cloudera.com/content/cloudera/en/documentation/cloudera-manager/v4-latest/Cloudera-Manager-Administration-Guide/cmag_licenses.html

Additional Quick Start Reference Deployments

- https://aws.amazon.com/quickstart/

Appendix: Security Group Specifics

The following are the configured inbound and outbound protocols and ports allowed for the various instances deployed as part of this solution:

Cluster Launcher Instance Security Group			
Inbound			
Source	Protocol	Port Range (Service)	Comments
Restricted to CIDR block specified during the deployment process	TCP	22 (SSH)	Allow inbound SSH access to Linux instance from your network (over the Internet gateway)
Custom TCP rule	TCP	1-65535	10.0.1.0/24 (private subnet within the Amazon VPC)
Custom TCP rule	TCP	1-65535	10.0.2.0/24 (public subnet within the Amazon VPC)
Outbound			
Destination	Protocol	Port Range	Comments
0.0.0.0/0	TCP	1-65535	Allow outbound access from cluster launcher instance to anywhere

NAT Security Group			
Inbound			
Source	Protocol	Port Range (Service)	Comments
Restricted to CIDR block specified during the deployment process	TCP	22 (SSH)	Allow inbound SSH access to Linux instance from your network (over the internet gateway)
10.0.0.0/16	TCP	80 (HTTP)	Allow inbound HTTP access only from instances deployed in the Amazon VPC
10.0.0.0/16	TCP	443 (HTTPS)	Allow inbound HTTPS access only from instances deployed in the Amazon VPC

NAT Security Group			
Outbound			
Destination	Protocol	Port Range	Comments
10.0.1.0/24	TCP	22 (SSH)	Allow SSH access from NAT instance to 10.0.1.0 subnet
0.0.0.0/0	TCP	80 (HTTP)	Allow outbound HTTP access from instances deployed in the Amazon VPC to anywhere
0.0.0.0/0	TCP	443 (HTTPS)	Allow outbound HTTPS access from instances deployed in the Amazon VPC to anywhere

EDH Cluster Nodes			
Inbound			
Source	Protocol	Port Range (Service)	Comments
Inbound			
Restricted to CIDR block specified during the deployment process	TCP	22 (SSH)	Allow inbound SSH access to Linux instance from your network (over the Internet gateway)
Custom TCP rule	TCP	1-65535	10.0.1.0/24 (private subnet within the Amazon VPC)
Custom TCP rule	TCP	1-65535	10.0.2.0/24 (public subnet within the Amazon VPC)
Outbound			
0.0.0.0/0	TCP	1-65535	Outbound access from all the cluster nodes allowed to anywhere

Send Us Feedback

We welcome your questions and comments. Please post your feedback on the AWS Quick Start Discussion Forum.

You can visit our GitHub repository to download the templates and scripts for this Quick Start, and to share your customizations with others.

Document Revisions

Date	Change	Location
August 2015	Added support for Cloudera Director 1.5.0	AWS CloudFormation templates; no documentation changes
May 2015	Added option to deploy Quick Start into an existing Amazon VPC.	Step 2(b)
March 2015	Updated examples to reflect changes in Cloudera Director 1.1.	Figures 8-11
October 2014	Initial publication	

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.