



Cloudera’s Enterprise Data Hub on the Amazon Web Services Cloud: Quick Start Reference Deployment

October 2014

Karthik Krishnan

Table of Contents

Table of Contents	2
Abstract	3
What We’ll Cover	4
Before You Get Started	4
Overview of Cloudera’s Enterprise Data Hub (EDH) on AWS	5
AWS Cluster Topology.....	6
Deployment.....	8
Step 1: Prepare an AWS Account.....	8
Step 2: Launch the Virtual Private Network and Configure AWS Services for EDH Deployment	9
Step 3: Configure Cluster and EDH Services	10
Step 4: Deploy the EDH cluster	13
Connect to Cloudera Director	16
Storage Configuration	17
Backup.....	18
Operating System and AMI	18
Security	18
AWS Identity and Access Management (IAM)	18
OS Security	18
Security Groups.....	18
Additional Information.....	19
Appendix A: Security Group Specifics	19

Abstract

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. We’ll discuss 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). We also provide links to automated AWS CloudFormation templates that you can leverage for your deployment or launch directly into your AWS account.

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 deployment method leverages Cloudera Director to deploy EDH automatically into a configuration of your choice. The cost for launching the reference deployment 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 a twelve-node cluster.

Instance	VCPU	Memory (GiB)	Workload Type	HDFS Storage (TB)	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 specific AWS services or the [AWS Simple Monthly Calculator](#) for full details.

What We’ll Cover

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. The following provides an outline of the steps involved in this deployment.

[Step 1: Prepare an AWS Account](#)

- Sign up for an AWS account
- Review default account limits for Amazon EC2 instances

[Step 2: Launch the Virtual Private Network and Configure AWS resources for EDH Deployment](#)

The following tasks are automated using AWS CloudFormation templates:

- Set up the Amazon VPC
- Create 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
- Start a cluster launcher Amazon EC2 instance. This instance will be used to deploy the EDH cluster using Cloudera Director
- Download Cloudera Director along with the necessary scripts and configuration files

[Step 3: Configure 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. In addition to these options, you can 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 have been modified to suit your compute and storage requirements, the EDH cluster can be launched using a simple command line executable.

Before You Get Started

If you are new to AWS, see the [Getting Started section](#) of the AWS documentation. In addition, familiarity with the following technologies is recommended:

- [Amazon EC2](#)
- [Amazon VPC](#)
- [AWS CloudFormation](#)
- [Amazon Identity and Access Management \(IAM\)](#)

Overview of Cloudera’s Enterprise Data Hub (EDH) on AWS

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
- 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

AWS Cluster Topology

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

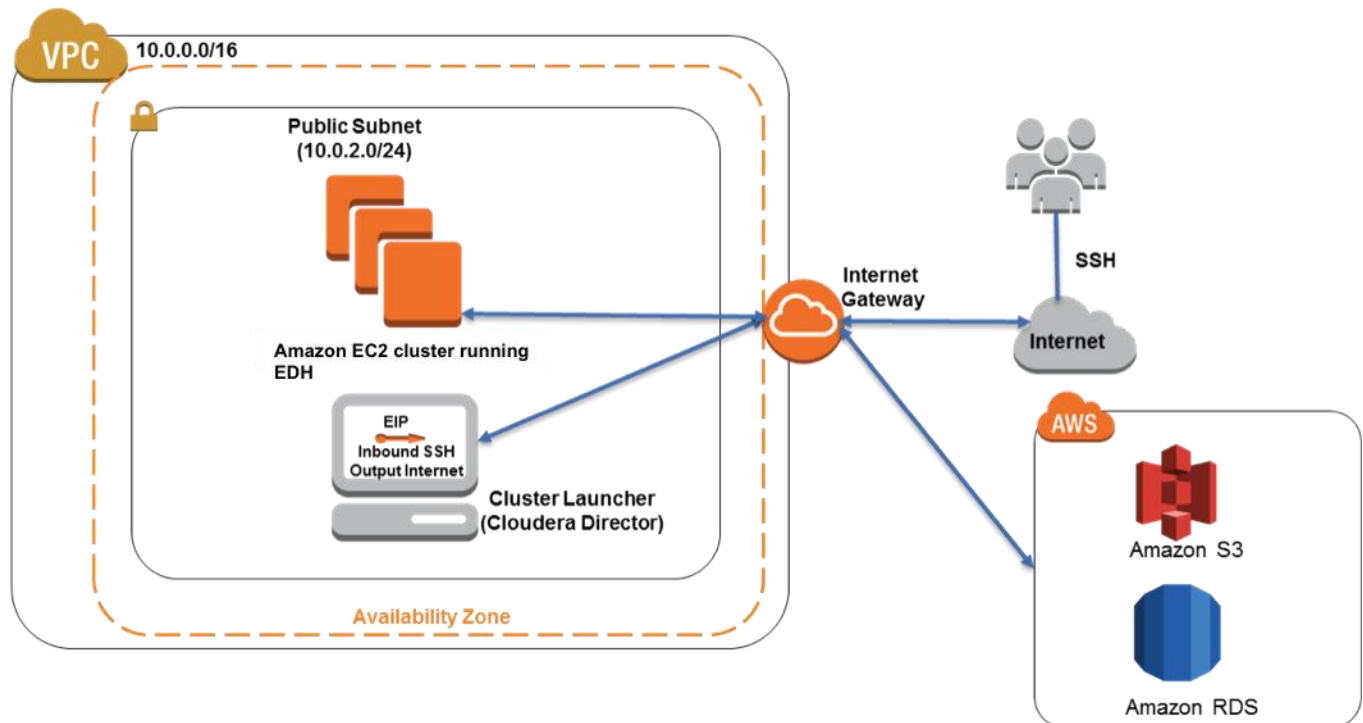


Figure 1: Public Subnet Topology

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

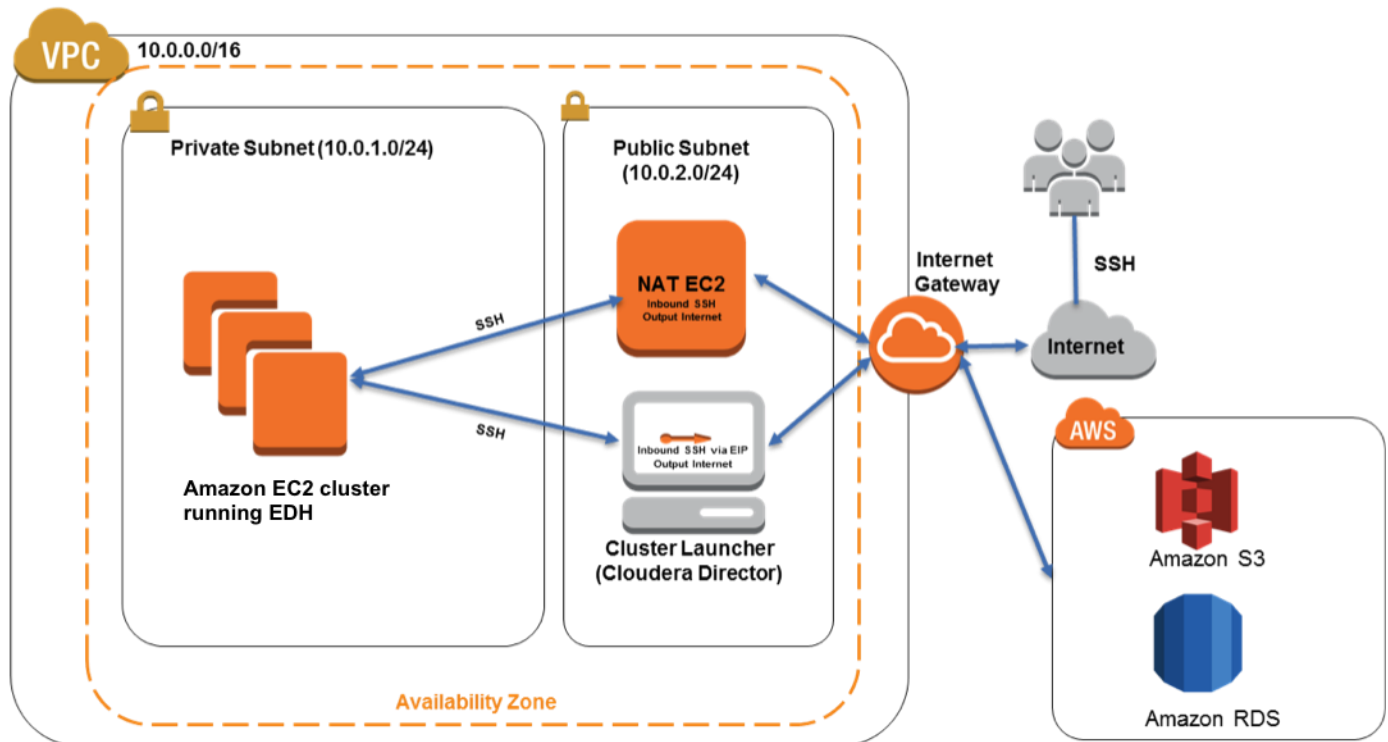


Figure 2: Private Subnet Topology

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

The following sections guide you through the deployment of an EDH Cluster on AWS.

Step 1: Prepare an AWS Account

This section describes preparation steps that may be necessary for this reference deployment. Prerequisites for deployment include creating a key pair for deployment and requesting Amazon EC2 limit increases if applicable.

1. If needed, create an AWS account at <http://aws.amazon.com>. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.
2. 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).

Tip

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

3. Create a [key pair](#) in your preferred region.
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 limit increase for the Amazon EC2 instance type(s) that you intend to deploy. Depending on the instance type, the default limit for the number of instances that can be run varies from two 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 [EC2 Instance service limit increase](#).

The screenshot shows the Amazon Support Center interface. At the top, there is a navigation bar with 'Support Center' on the left and 'Welcome [Account] | Sign out' on the right. Below the navigation bar, the page title is 'Home > Open a new case'. On the right side, there is a section for 'Frequently Asked Service Limit Questions' with a link 'What are the default service limits?'. The main form is titled 'Regarding *' and has three radio button options: 'Account and Billing Support', 'Service Limit Increase' (which is selected), and 'Technical Support'. Below this, there are several dropdown menus: 'Limit Type*' set to 'EC2 Instances', 'EC2 Region*' set to 'US East (Northern Virginia)', 'Operating System*' set to 'Linux/OpenSolaris', 'Primary Instance Type*' set to 'c3.8xlarge', and 'Frequency of Usage*' set to 'Always On'. At the bottom, there is a text area for 'Use Case Description *' containing the text 'Need to deploy CDH Cluster supporting 25 nodes'.

Figure 3: Sample Amazon EC2 Limit Increase Request

Step 2: Launch the Virtual Private Network and Configure AWS Services for EDH Deployment

In this step, you will launch an AWS CloudFormation template that configures the Virtual Private Network (VPN) that provides the base AWS network infrastructure for your EDH deployment. It also builds public and private subnets along with a NAT instance launched within the public subnet. An Amazon EC2 instance running Linux (RedHat) is launched in the public subnet and serves as a launcher node for the Cloudera cluster. The cluster deployment is initiated by the launcher node. 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.

The parameters in the following table are used by the AWS CloudFormation template to generate a cluster configuration file. After the cluster launcher instance is deployed, you can make additional changes to the EDH deployment by modifying the configuration file.

The following table lists the customizable parameters used in the AWS CloudFormation template associated with this Quick Start Reference Deployment.

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	<User Provided>	Name of an existing Amazon EC2 Key Pair
NATInstanceType	m1.small	Amazon EC2 instance type for the NAT Instances
ClusterLauncherType	t2.small	Amazon EC2 instance type for the EDH launcher Instance

Launch the VPN template into your AWS account using AWS CloudFormation: [Launch Stack](#)

When the AWS CloudFormation status indicates “CREATE_COMPLETE” and the Launcher Instance has been created successfully as shown below, you can continue to the next step.

The screenshot shows the AWS CloudFormation console interface. At the top, there are buttons for 'Create Stack', 'Update Stack', and 'Delete Stack'. Below these, there's a filter section with 'Filter: Complete' and a search box. The main area displays a table of stacks with one stack selected: 'AWS-CLOUDERA-Infrastructures' with a status of 'CREATE_COMPLETE'. Below the stack list, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', and 'Stack Policy'. The 'Outputs' tab is active, showing a table of stack outputs.

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 4: Template 1 Complete Example

Step 3: Configure Cluster and EDH Services

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

Connect to the Cluster Launcher Instance

Connect to the cluster launcher instance by clicking the **Connect** tab under **EC2 Instances** as below. You will need your private key to launch the instance.

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 (kkf-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](#)

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	kkf-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	kkf-refw-singa...	disabled

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-17CACNKT4YRN.
[view rules](#)

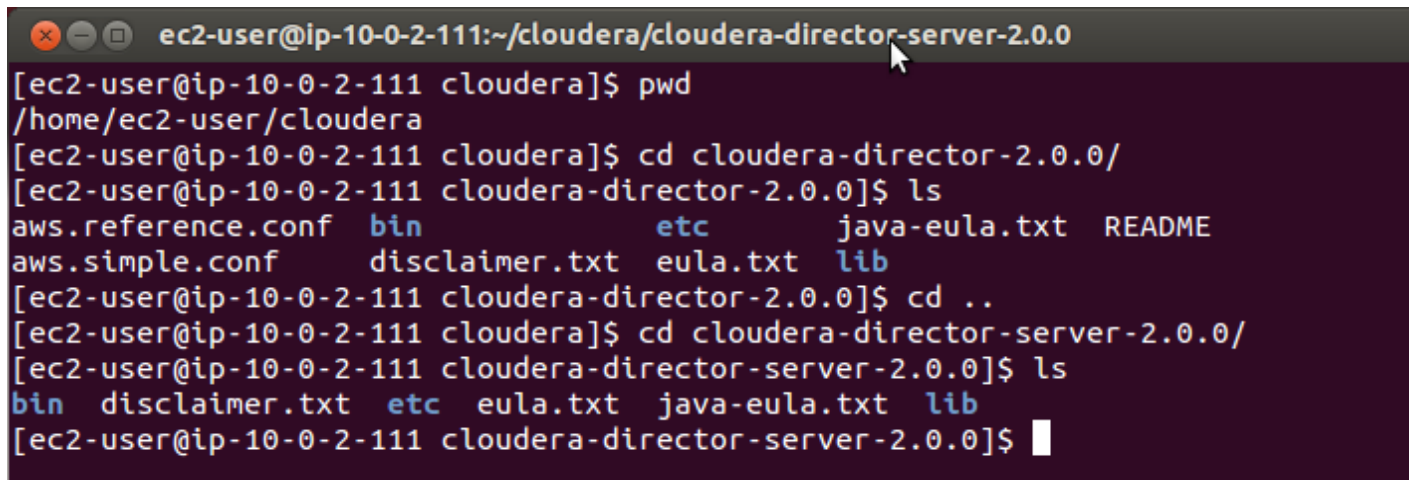
Figure 5: Connect to Cluster Launcher using SSH

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 [Granting Applications that Run on Amazon EC2 Instances Access to AWS Resources](#).

Figure 6 lists the files that are downloaded automatically during launch.



```
ec2-user@ip-10-0-2-111:~/cloudera/cloudera-director-server-2.0.0
[ec2-user@ip-10-0-2-111 cloudera]$ pwd
/home/ec2-user/cloudera
[ec2-user@ip-10-0-2-111 cloudera]$ cd cloudera-director-2.0.0/
[ec2-user@ip-10-0-2-111 cloudera-director-2.0.0]$ ls
aws.reference.conf  bin          etc          java-eula.txt  README
aws.simple.conf    disclaimer.txt  eula.txt    lib
[ec2-user@ip-10-0-2-111 cloudera-director-2.0.0]$ cd ..
[ec2-user@ip-10-0-2-111 cloudera]$ cd cloudera-director-server-2.0.0/
[ec2-user@ip-10-0-2-111 cloudera-director-server-2.0.0]$ ls
bin  disclaimer.txt  etc  eula.txt  java-eula.txt  lib
[ec2-user@ip-10-0-2-111 cloudera-director-server-2.0.0]$
```

Figure 6: Deployment Scripts and Configuration Files

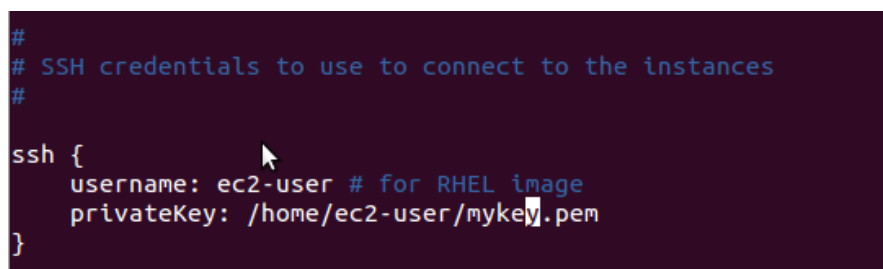
Important

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
```

Modify the configuration file

There are two configuration files that are customizable during deployment: `aws.simple.conf` for configuring simple clusters and `aws.reference.conf` for configuring multiple clusters. The only instance modification *required* for deployment is replacing the `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 7: 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. For more information about configuration parameters, please see the [Cloudera Director User Guide](#).

Step 4: Deploy the EDH cluster

Cloudera Director supports two options for cluster deployment: you can deploy using the CLI and manage the nodes manually or deploy using the Cloudera Director Server to manage multiple clusters (recommended).

Deploy Using the CLI, No Server (Option 1)

To deploy the EDH cluster, run `cloudera-director` executable using one of the configuration file as below:

```
./bin/cloudera-director bootstrap aws.simple.conf (simple cluster)
-OR-
./bin/cloudera-director bootstrap aws.reference.conf (advanced cluster)
```

Figure 8 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 10.0.1.224 ..... done
* Inspecting capabilities of 10.0.1.224 ..... done
* Normalizing 10.0.1.224 ..... done
* Installing ntp (1/2) .... done
* Installing curl (2/2) ..... done
* Mounting all instance disk drives ..... done
* Resizing instance root partition ..... done
* Rebooting 10.0.1.224 .... done
* Waiting for 10.0.1.224 to boot ..... done
* Installing repositories for Cloudera Manager ..... done
* Installing jdk (1/4) .... done
* Installing cloudera-manager-daemons (2/4) .... done
* Installing cloudera-manager-server (3/4) .... done
* Installing cloudera-manager-agent (4/4) ..... 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
* Starting Cloudera Management Services ..... done
* Inspecting capabilities of 10.0.1.224 ..... done
* Done ...
Cloudera Manager ready.
Creating cluster C5.Simple.AWS ...
* Starting .... done
* Requesting instances ..... done
* Preparing instances and deploying Cloudera Manager agents ..... done
* Creating CDH5 cluster using the new nodes .... done
* Downloading parcels: CDH-5.1.0-1.cdh5.1.0.p0.53 ... done
* Distributing parcels: CDH-5.1.0-1.cdh5.1.0.p0.53 ... done
* Switching parcel distribution rate limits back to defaults: 51200KB/s with 25 concurrent uploads ... done
* Activating parcels: CDH-5.1.0-1.cdh5.1.0.p0.53 ... done
* Configuring Hive Metastore database ... done
* Waiting on First Run command ... done
* Done ...
Cluster ready.
```

Figure 8: EDH Deployment Sequence

Cloudera Director also supports other command arguments, such as `terminate` and `status query`. For example:

```
./bin/cloudera-director status aws.simple.conf (simple cluster)
-OR-
./bin/cloudera-director status aws.reference.conf (advanced cluster)
```

```
[ec2-user@ip-10-0-2-111 cloudera-director-2.0.0]$ ./bin/cloudera-director status aws.simple.conf
Process logs can be found at /home/ec2-user/cloudera/cloudera-director-2.0.0/logs/application.log
Cloudera Director 2.0.0.M4 initializing ...

Cloudera Manager:
* Instance: 10.0.1.144 application=Cloudera Manager 5,owner=ec2-user
* Shell: ssh -i /home/ec2-user/kkr.pem ec2-user@10.0.1.144

Cluster Instances:
* Instance 1: 10.0.1.75 owner=ec2-user
* Shell 1: ssh -i /home/ec2-user/kkr.pem ec2-user@10.0.1.75

* Instance 2: 10.0.1.73 owner=ec2-user
* Shell 2: ssh -i /home/ec2-user/kkr.pem ec2-user@10.0.1.73

* Instance 3: 10.0.1.71 owner=ec2-user
* Shell 3: ssh -i /home/ec2-user/kkr.pem ec2-user@10.0.1.71

* Instance 4: 10.0.1.74 owner=ec2-user
* Shell 4: ssh -i /home/ec2-user/kkr.pem ec2-user@10.0.1.74

* Instance 5: 10.0.1.72 owner=ec2-user
* Shell 5: ssh -i /home/ec2-user/kkr.pem ec2-user@10.0.1.72

Command to map remote web console ports on the local machine:
* Gateway Shell: ssh -i /path/to/launchpad/host/keyName.pem -L 7180:10.0.1.144:7180 -L 7187:10.0.1.144:7187
ec2-user@ec2-54-169-110-21.ap-southeast-1.compute.amazonaws.com

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

[ec2-user@ip-10-0-2-111 cloudera-director-2.0.0]$
[ec2-user@ip-10-0-2-111 cloudera-director-2.0.0]$
```

Deploy Using Cloudera Director Server (Option 2)

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 (or, optionally, specify --port=port argument)
```

This command starts the server on port 7189 (default) of the cluster launcher instance.

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

```
./bin/cloudera-director bootstrap-remote aws.simple.conf --lp.remote.hostAndPort= 127.0.0.1:7189 (simple cluster)
```

-OR-

```
./bin/cloudera-director bootstrap-remote aws.reference.conf --lp.remote.hostAndPort= 127.0.0.1:7189 (advanced cluster)
```

Connect to 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, the following shows the node where Cloudera Manager application is running.

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
<input type="checkbox"/>	cloudera-director-i-bc3bcf97-c0acd7-3458-4881-...	i-043bcf2f	m3.2xlarge	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input type="checkbox"/>	cloudera-director-i-bc3bcf97-f6b22292-21e4-46bc-a...	i-073bcf2c	m3.2xlarge	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input type="checkbox"/>	cloudera-director-i-bc3bcf97-0a9909c0-d832-46d2-...	i-053bcf2e	m3.2xlarge	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input checked="" type="checkbox"/>	cloudera-director-i-bc3bcf97-1f465133-73b8-4896-b...	i-f224d0d9	m3.2xlarge	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input type="checkbox"/>	cloudera-director-i-bc3bcf97-19200dc5-77e0-4f5e-a...	i-033bcf28	m3.2xlarge	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input type="checkbox"/>	cloudera-director-i-bc3bcf97-3676077d-4802-446a-...	i-063bcf2d	m3.2xlarge	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input type="checkbox"/>	ClusterLauncher Instance (Public Subnet)	i-bc3bcf97	t2.small	ap-southeast-1a	● running	✔ 2/2 checks ...	None
<input type="checkbox"/>	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 9: Using instance Tags

In Figure 10, 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 using 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
```

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

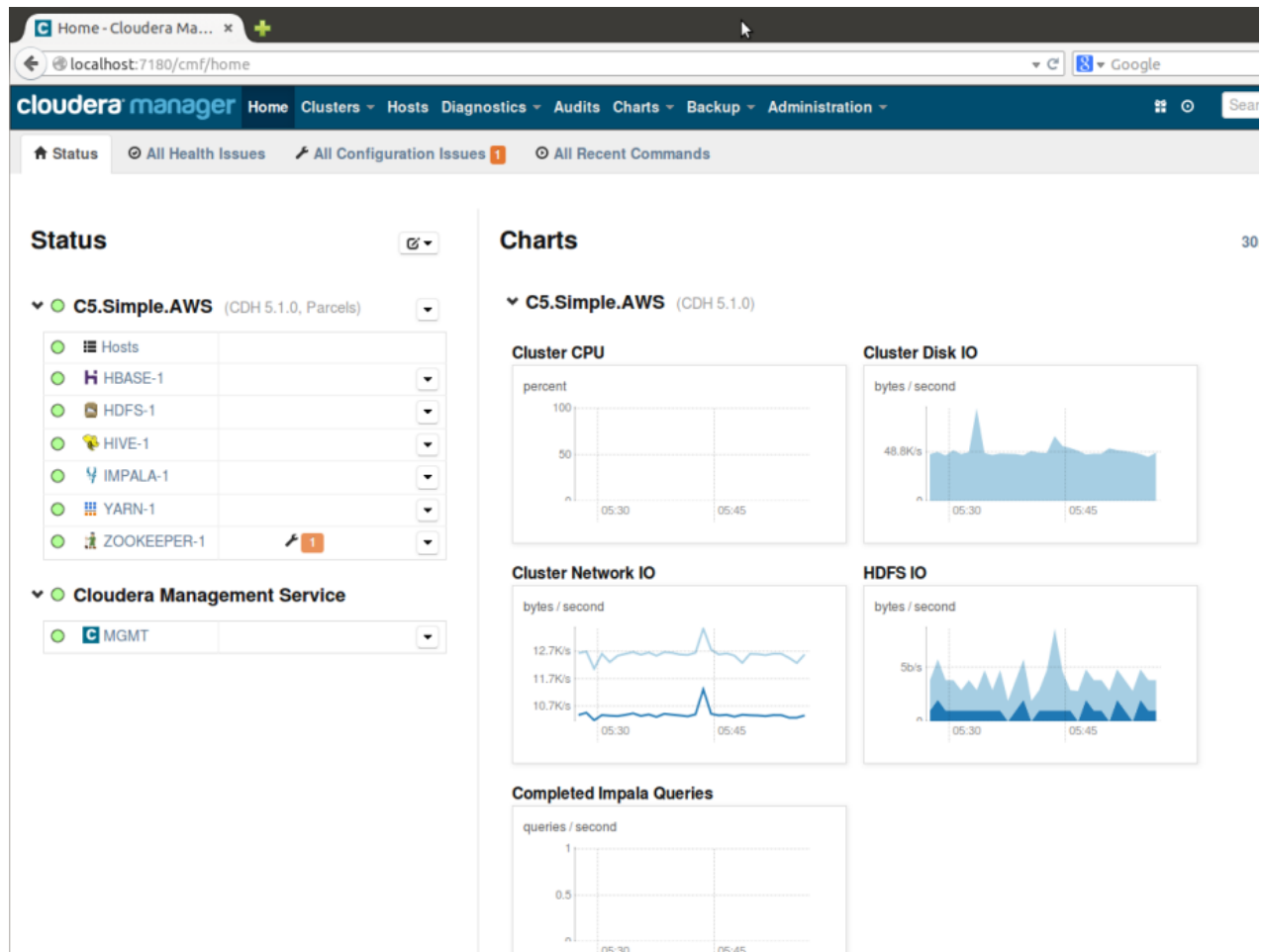


Figure 10: Connecting to Cloudera Manager

Connect to Cloudera Director

For ongoing management of the cluster or to launch additional clusters, you can use Cloudera Director’s web interface at <http://<ip address of Director server>:7189>. Log in with admin/admin.

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.

cloudera director

admin ?

All Environments Marketing Analytics Test bed

Add Environment

All Environments

Add Cluster

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

Figure 11: 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 GB (1680 GB)
c3.8xlarge	2 x 320 GB SSD (640 GB)
i2.2xlarge	2 x 800 GB SSD (1600 GB)
cc2.8xlarge	4 x 840 GB (3360 GB)
r3.8xlarge	2 X 320 GB (640 GB)
i2.4xlarge	4 x 800 GB SSD (3200 GB)
hs1.8xlarge	24 x 2048 GB (48 TB)
i2.8xlarge	8 x 800 GB SSD (6400 GB)

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 Information

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](#).

Appendix A: 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 from only instances deployed in the Amazon VPC
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
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

© 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.