cloudera
Ask Bigger Questions

Cloudera Director User Guide

Important Notice

Cloudera, Inc.
1001 Page Mill Road Bldg 2
Palo Alto, CA 94304
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com

Release Information

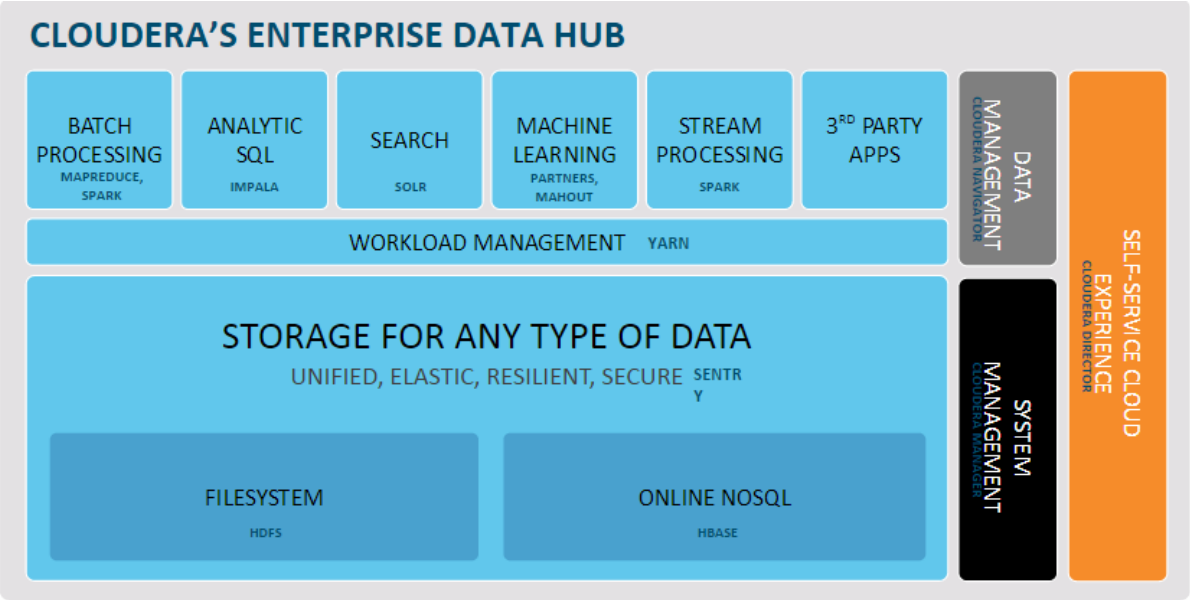
Version: 1.0.x
Date: October 29, 2014

Table of Contents

Introduction

Cloudera Director is the reliable, self-service experience for CDH and Cloudera Enterprise in the cloud. Cloudera Director brings choice and Hadoop expertise to the cloud, with the ability to leverage the robust Cloudera partner ecosystem and open, enterprise-grade platform.

Cloudera Director is designed to provide a single pane of glass administration experience for central IT to reduce costs and deliver agility, and for end-users to self-service provision and elastically scale clusters, all while ensuring auditability. Advanced users can interact with Cloudera Director programmatically via the REST API or through the CLI to maximize time-to-value for an enterprise data hub in cloud environments. Furthermore, the same high quality experience is available to admins, as well as end-users on multiple clouds, should the organization decide to leverage more than one cloud environment for data processing needs.



Cloudera Director is designed for both long running and ephemeral clusters. With long running clusters, you deploy one or more clusters that you can scale up to meet demand. With ephemeral clusters, you can launch a cluster, schedule any jobs, and shut the cluster down after the jobs complete.

Some of the reasons to run Cloudera in the cloud include:

- Faster procurement—it is usually faster to deploy servers in the cloud than go through a lengthy hardware acquisition process.
- Easier to scale—to meet increased cluster demand, it is easier to add new hosts in the cloud than in a bare metal environment.
- Infrastructure decisions—many organizations have already moved to a cloud architecture, while others are in the process of moving.

Cloudera Director supports two ways to quickly deploy clusters: through the client or the server.

Using the Cloudera Director client, you edit the cluster configuration file, launch an EC2 instance, and run Cloudera Director with a configuration file. The Cloudera Director client works well for proof of concept work and infrequent usage.

Using the Cloudera Director server, you can either send it a cluster configuration file or you can log into its UI and launch clusters directly. The server works well for launching and managing large numbers of clusters in a production environment.

Features

Cloudera Director provides a rich set of features for launching and managing clusters in cloud environments. Some of these features include:

Benefit	Capability	Features
Simplify cluster life cycle management	Simple UI to spin up, scale, and spin down clusters	▪ Self-Service spin up and tear down ▪ Dynamic scaling for spiky workloads ▪ Simple cloning of clusters ▪ Cloud blueprints for repeatable deployments
Eliminate lock-in	Flexible, open platform	▪ 100% open source Hadoop distribution ▪ Native support for hybrid deployments ▪ 3rd party software deployment in the same workflow ▪ Support for custom, workload-specific deployments
Accelerate time to value	Enterprise-ready security and administration	▪ Support for complex cluster topologies ▪ Minimum size cluster when capacity constrained ▪ Management tooling ▪ Compliance-ready security and governance ▪ Backup and disaster recovery with an optimized cloud storage connector
Reduce support costs	Monitoring and metering tools	▪ Multi-cluster health dashboard ▪ Instance tracking for account billing

▪ **Note:** Although you cannot deploy a high availability cluster through Cloudera Director, you can launch a cluster and upgrade it for high availability. For more information, see High Availability on page 32.

Requirements

Cloudera Director supports the following:

- Deployment of instances running RHEL 6.4 AMIs, CentOS 6.4, or CentOS 6.5
- Use of RHEL5, RHEL6, CentOS 5, CentOS 6, Ubuntu Precise, Ubuntu Trusty, Debian Wheezy or SLES11 for the Cloudera Director server and client.
- Cloudera Manager 5
- CDH 4 and 5

Make sure you have the following:

- An AWS account
- An SSH key pair (either local or generated by AWS)
- The Cloudera Director software

> **Note:** Cloudera Director uses the H2 Database Engine to store environment and cluster data, which it stores in the `state.h2.db` file. To avoid losing this data, make sure to back up this file.

Setting up an Environment

Whether you are using the Cloudera Director client or server, you must first set up the environment.

Setting Up VPC

Within AWS, Cloudera Director requires Amazon Virtual Private Cloud (VPC).

> **Note:** The AWS VPC must be set up for forward and reverse hostname resolution.

To manually set up a VPC, follow these steps:

1. Log in to web console at https://aws.amazon.com/console.
2. In the upper right of the AWS Console, select a region.
3. Select **VPC** from the **Services** navigation list box.
4. Click **Start VPC Wizard**. The Select a VPC Configuration page appears.
5. Specify IP address settings, a VPC name, and any other preferences. The easiest way to get started is to select **VPC with a Single Public Subnet**. For more information, see the VPC Documentation.
6. Click **Create VPC**. AWS creates a VPC.
7. In the left pane, click **Subnets**. A list of currently configured subnets appears.
8. Click **Create Subnet**. The Create Subnet dialog box appears.
9. Configure a subnet of the VPC you created and click **Yes, Create**.
10. In the left pane, click **Security Groups**. A list of currently configured security groups appears.
11. Click **Create Security Group**. The Create Security Group dialog box appears.
12. Enter a name and description. Make sure to select the VPC you created from the VPC list box.
13. Click **Yes, Create**.
14. Select the newly created security group and add the following rules:
    a. Add the `All traffic, all protocols, all ports, <id of this security group>` and `SSH(22), TCP(6), 22, 0.0.0.0/0` inbound rules (you can secure this further later).
    b. Add the `All traffic, all protocols, all ports, 0.0.0.0/0` outbound rule.

Creating a Key Pair

To interact with the cluster launcher and other instances, you must create an SSH key pair.

> **Note:** For information on importing an existing key pair, see the EC2 Documentation.

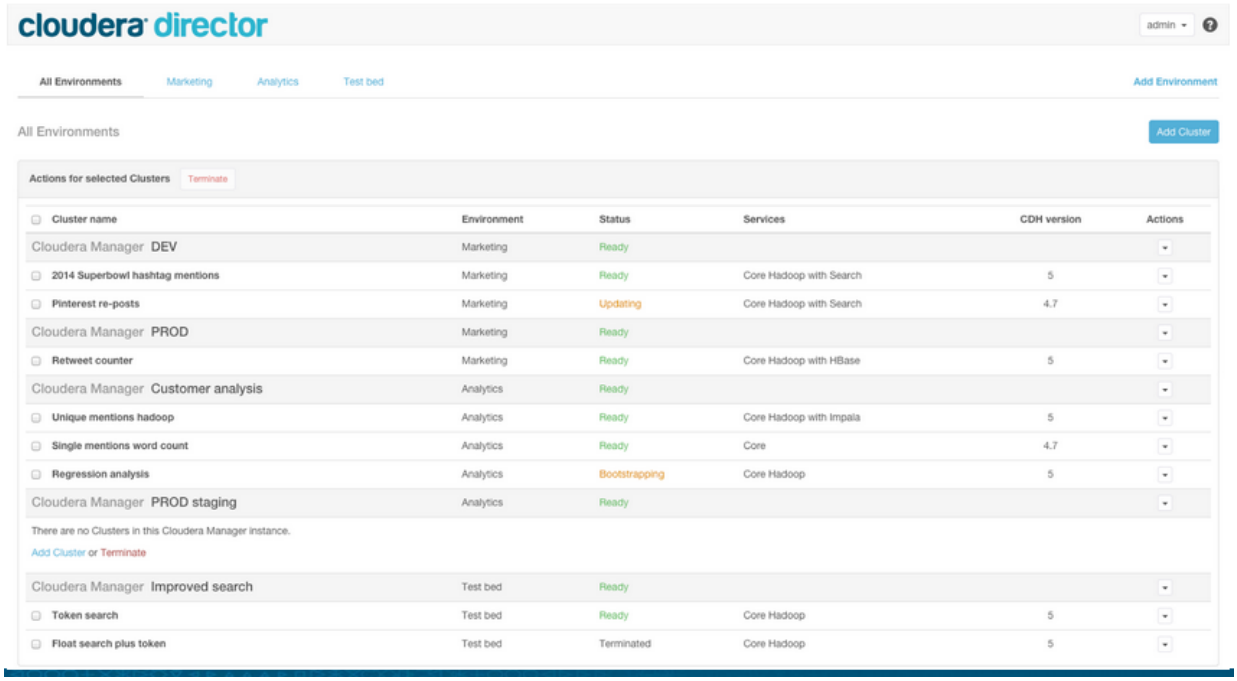
If you do not have a key pair, follow these steps:

1. Select **EC2** from the **Services** navigation list box.
2. In the left pane, click **Key Pairs**. A list of currently configured key pairs appears.
3. Click **Create Key Pair**. The Create Key Pair dialog box appears.
4. Enter a name for the key pair and click **Create**.
5. Note the key pair name. Move the automatically downloaded keyfile (with .pem extension) to a secure location and note the location.

Cloudera Director Server: UI

To deploy multiple clusters through the UI, you launch a Cloudera Director server, log into the server, and configure settings to launch one or more clusters.

The following shows the Cloudera Director server UI:



Setting Up the Cloudera Director Server

This section describes how to set up the Cloudera Director Server.

To set up the server, follow these steps:

1. Review the requirements.
2. Set up VPC.
3. Make sure you are logged in to web console at http://aws.amazon.com/console/.
4. Select **EC2** from the **Services** navigation list box.
5. Click **Launch Instance**.
6. Get the AMI ID for the instance. Cloudera recommends Red Hat Linux 6.4 (ami-b8a63b88 in US-West-2) with a c3.xlarge instance type. If the AMI does not show up in the list, go to https://aws.amazon.com/partners/redhat/ and select a 64-bit version of Red Hat Linux 6.4 for the region in which you are launching the cluster.
7. Click **Community AMIs** in the left pane.
8. Enter the **AMI ID**. The AMI appears in the list.
9. Click **Select**. The Choose an Instance Type page appears.
10. Click **Next: Configure Instance Details**.
    a. Make sure to select the VPC and subnet that you created or noted earlier.
    b. The cluster launcher needs Internet access; from the Auto-assign Public IP list box, select **Enable**.
11. Click **Next: Add Storage**.

12. Click **Next: Tag Instance** and create any tags to quickly find the instance.
13. Click **Next: Configure Security Group**. Then, select the **Select an existing security group** radio button and select the security group you noted or created earlier.
14. Click **Review and Launch**.
15. Click **Launch**. When prompted, make sure to launch the instance with the key pair that you created. If you selected SSD storage, you might be prompted to choose the storage type. The instance does not require SSD storage.
16. Click **Launch Instances**.
17. After the instance is created, note its public and private IP addresses.
18. Install the latest version of the Cloudera Director server from the Cloudera Director Download Page.

Configuring an Environment and Deploying a Cluster

The environment defines common settings used with a cloud infrastructure provider, such as AWS. While creating an environment, you are also prompted to deploy its first cluster.

To add an environment:

1. Open Cloudera Director through a web browser using the public IP address you noted in Setting Up the Cloudera Director Server on page 9. For example, http://100.100.100.100:7189.

   You are prompted to log in.

2. Enter the username and password (default: admin/admin).
3. Click **Add Environment**.

   The Add Environment page opens.

4. Enter a name in the **Environment Name** field.
5. Select a region from the **Region** field.
6. Enter your keys in the **Access key ID** and **Secret access key** fields.
7. To make the keys available to the cluster, select the **Make access keys available to Hadoop** check box.
8. Enter the name of the EC2 key pair in the **EC2 Public key name** field.
9. Enter the name of the SSH user in the **SSH username** field. For example, ec2-user.
10. Copy the SSH private key into the **SSH private key** field.
11. Enter the SSH passphrase in the **SSH passphrase** field. If the SSH key is not encrypted, leave this blank.
12. To provide the instances with public IP addresses, select the **Associate public IP addresses with instances** check box.
13. Click **Continue**. The Add Cloudera Manager page appears.
14. Enter a name for the Cloudera Manager in the **Cloudera Manager name** field.
15. If you want to enable a trial of Cloudera Enterprise, select the **Enable Cloudera Enterprise trial** check box.
16. Select whether to create a new template or use an existing one from the **Instance Template** list box.
17. If you selected **Create New Instance Template** configure the following options:

    - **Instance Template name** - enter a name for the template.
    - **Type** - select the instance type.
    - **Amazon Machine Image (AMI)** - enter the AMI ID to use for Cloudera Manager.
    - **Tags** - specify one or more tags to associate with the instance.
    - **Root volume size** - select the size of the root volume.
    - **VPC subnet ID** - enter the ID of the VPC subnet in which the instance will be located.
    - **Security group IDs** - enter one or more security group IDs with which the instance will be associated.
    - **Bootstrap script** (optional) - enter a Linux bootstrap script. After the instance boots, this script automatically runs. This script can contain anything you need for your environment including libraries, monitoring tools, security configurations, and so on.

18. Select whether to override the default Cloudera Manager repository.
19. Click **Continue**. You are prompted for confirmation.
20. Click **OK**.

    The **Add Cluster** page appears.

21. Enter a name for the cluster in the **Cluster name** field.
22. Select the version of CDH to deploy in the **Version** field.
23. Select the type of cluster to deploy from **Services**.
24. Select the numbers of masters, workers, and gateways to deploy. Then, select an instance template for each or create one or more new templates.

    > **Note:** You must deploy at least one master and one worker.

25. When you are finished, click **Continue**. You are prompted for confirmation.
26. Click **OK**.

    Cloudera Director begins deploying the cluster.

Deploying Clusters in an Existing Environment

If you already configured an environment, you can easily deploy new clusters.

To deploy a cluster:

1. Log in to Cloudera Director. For example, http://example.com:7189.
2. Click **Add Cluster**.

   The Add Cluster page appears.

3. Select an environment from the **Environment** list box. If you have not created an environment, see Configuring an Environment and Deploying a Cluster on page 10.
4. Select a Cloudera Manager from the **Cloudera Manager** list box.
5. To clone an existing cluster, select **Clone from existing** and select a cluster. To specify cluster settings, select **Create from scratch**.
6. Enter a name for the cluster in the **Cluster name** field.
7. Select the version of CDH to deploy in the **Version** field.
8. Select the type of cluster to deploy from **Services**.
9. Select the numbers of masters, workers, and gateways to deploy. Then, select an instance template for each or create one or more new templates.
10. When you are finished, click **Continue**.

    You are prompted for confirmation.

11. Click **OK**.

    Cloudera Director begins deploying the cluster.

Opening Cloudera Manager

After deploying a cluster, you can manage it using Cloudera Manager.

To manage a cluster through Cloudera Manager:

1. Log in to Cloudera Director. For example, http://example.com:7189.

   Cloudera Director opens with a list of clusters.

2. Locate the cluster to manage and click its Cloudera Manager. If the Cloudera Manager is not ready, the link does not appear.

   The Cloudera Manager Login page appears.

3. Enter your credentials and click **Login**.

   Cloudera Manager opens.

Growing a Cluster

Cloudera Director can grow the size of a cluster. If you are adding a large number of DataNodes, you should rebalance the cluster through Cloudera Manager.

To grow a cluster:

1. Log in to Cloudera Director. For example, http://example.com:7189.

   Cloudera Director opens with a list of clusters.

2. If the cluster is in the **Ready** state, select the list box to the right of the cluster to grow and select **Grow Cluster**.

   The Grow Cluster page appears, displaying the number of masters, workers, and gateways.

3. Increase the number of workers to the desired size. If you increase the number of workers by 30% or more, we recommend rebalancing the cluster through Cloudera Manager.

Terminating a Cluster

You can terminate a cluster at any time. To terminate a cluster:

1. Log in to Cloudera Director. For example, http://example.com:7189.

   Cloudera Director opens with a list of clusters.

2. Select the check box for each cluster to terminate and click **Terminate**.

   The Terminate All Selected dialog box appears.

3. Click **Terminate**.

   Cloudera Director begins terminating the cluster(s).

Starting and Stopping the Cloudera Director Server

Although you can stop and start Cloudera Director at any time, you should terminate any running clusters first.

To start or stop the server, enter the following:

```
$ sudo service cloudera-director-server [start | stop]
```

Viewing the Status Page

You can view the status page at any time.

To view the status page, open the following with a web browser:

```
http://host:7189
```

CLI Deployment

Deployment through the Cloudera Director client involves launching an instance called the cluster launcher, copying the software to the cluster launcher, editing a configuration file, and running Cloudera Director from the command line.

Starting an Instance

Cloudera Director needs a dedicated instance in the same subnet that can access the new instances on the private network.

To start the instance:

> **Note:** For general information about copying files to and from an instance, see the EC2 Documentation.

1. Make sure you are logged in to web console at http://aws.amazon.com/console/.
2. Select **EC2** from the **Services** navigation list box.
3. Click **Launch Instance**.
4. Get the AMI ID for the instance. Cloudera recommends Red Hat Linux 6.4 (ami-b8a63b88 in US-West-2) with a c3.xlarge instance type. If the AMI does not show up in the list, go to https://aws.amazon.com/partners/redhat/ and select a 64-bit version of Red Hat Linux 6.4 for the region in which you are launching the cluster.
5. Click **Community AMIs** in the left pane.
6. Enter the **AMI ID**. The AMI appears in the list.
7. Click **Select**. The Choose an Instance Type page appears.
8. Click **Next: Configure Instance Details**.
   a. Make sure to select the VPC and subnet that you created or noted earlier.
   b. The cluster launcher needs Internet access; from the Auto-assign Public IP list box, select **Enable**.
9. Click **Next: Add Storage**.
10. Click **Next: Tag Instance** and create any tags to quickly find the instance.
11. Click **Next: Configure Security Group**. Then, select the **Select an existing security group** radio button and select the security group you noted or created earlier.
12. Click **Review and Launch**.
13. Click **Launch**. When prompted, make sure to launch the instance with the key pair that you created. If you selected SSD storage, you might be prompted to choose the storage type. The instance does not require SSD storage.
14. Click **Launch Instances**.
15. After the instance is created, note its public and private IP addresses.

Installing Cloudera Director

Installation is simple; you only need to install the software and copy your private key to the instance.

To install Cloudera Director:

1. Install the latest version of Cloudera Director from the Cloudera Director Download Page.

2. SSH to the instance and copy the key file.

```
[ec2-user@ip-10-1-1-18 ~]$ ssh -i [keyfile].pem ec2-user@100.100.100.100
[ec2-user@ip-10-1-1-18 ~]$ cp [keyfile].pem /home/ec2-user/.ssh/id_rsa
[ec2-user@ip-10-1-1-18 ~]$ chmod 400 /home/ec2-users/.ssh/id_rsa
```

Choosing an AMI

AMIs specify the operating system, architecture (32-bit vs 64-bit), AWS Region, and virtualization type (Paravirtualization or HVM).

Cloudera Director, CDH, and Cloudera Manager only support 64-bit Linux. Cloudera Director has only been tested with Red Hat Enterprise Linux 6.4.

The virtualization type depends on the instance type that you want to use. After selecting an instance type based on the expected storage and computational load, check the supported virtualization types. Then, identify the correct the AMI based on architecture, AWS Region, and virtualization type.

Modifying the Configuration File

The configuration file contains information Cloudera Director needs to operate and settings that define your cluster.

To modify the configuration file:

1. Copy the `aws.simple.conf` file to `aws.conf`. For advanced cluster configuration, use `aws.reference.conf`.

   > **Note:** The configuration file must use the `.conf` file extension.

2. Open `aws.conf` with a text editor.
3. Configure the basic settings:
   - **name** - change to something that makes the cluster easy to identify.
   - **id** - leave this set to aws.
   - **accessKeyId** - AWS Access Key. Make sure the value is enclosed in double quotes.
   - **secretAccessKey** - AWS Secret Key. Make sure the value is enclosed in double quotes.
   - **region** - specify the region (for example, us-west-2).
   - **keyName** - specify the name of the key pair used to start the cluster launcher. Key pairs are region-specific. If you create a key pair in US-West-2, it will not be available in US-West-1.
   - **subnetId** - ID of the subnet that you noted earlier.
   - **securityGroupsIds** - ID of the security group that you noted earlier. Use the ID of the group, not the name (for example, sg-b139d3d3, not default).
   - **instanceNamePrefix** - enter the prefix to prepend to each instance's name.
   - **image** - specifies the AMI to use. Cloudera recommends Red Hat Enterprise Linux 6.4 (64bit). To find the correct AMI for the selected region, visit the Red Hat AWS Partner page.

     > **Note:** If you use your own AMI, make sure to disable any software that prevents the instance from rebooting during the deployment of the cluster.

4. Configure the following cluster settings:
   a. You can only use Cloudera Manager 5. No changes are needed for repository and repository key URLs and you must set the parcel repositories to match the CDH and Impala versions you plan to install.
   b. Specify services to start on the cluster. For a complete list of allowed values, see the Cloudera Manager API Service Types.

> **Note:** Include Flume in the list of services only when customizing role assignments. See the configuration file aws.reference.conf included in the Cloudera Director download for examples on how to configure customized role assignments. If Flume is required, it should be excluded from the list of services in the configuration file and added as a service using Cloudera Manager UI or API after the cluster is deployed. When adding Flume as a service, you must assign Flume agents (which Cloudera Manager does not do automatically).

c. Specify the number of nodes in the cluster.

5. Save the file and exit.

Running Cloudera Director

After you modify the configuration file, you are ready to run Cloudera Director.

> **Note:** If you are restarting Cloudera Director, it will prompt you to resume from where it stopped or start over. If you made changes to the configuration file between deployments or if you need to start the run from scratch, you should start over.

1. From the cluster launcher, enter the following:

```
[ec2-user@ip-10-1-1-18 cloudera-director-1.1.0]$ cloudera-director bootstrap
aws.conf
```

Cloudera Director displays output similar to the following:

```
Installing Cloudera Manager ...
* Starting ... done
* Requesting an instance for Cloudera Manager ................... done
* Inspecting capabilities of 10.1.1.194 ............... done
* Normalizing 10.1.1.194 ................... done
* Installing python (1/4) .... done
* Installing ntp (2/4) .... done
* Installing curl (3/4) .... done
* Installing wget (4/4) ................ done
* Installing repositories for Cloudera Manager ............... done
* Installing jdk (1/5) ..... done
* Installing cloudera-manager-daemons (2/5) ..... done
* Installing cloudera-manager-server (3/5) ..... done
* Installing cloudera-manager-server-db-2 (4/5) ..... done
* Installing cloudera-manager-agent (5/5) .... 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 10.1.1.194 ......... done
* Done ...
Cloudera Manager ready.
Creating cluster C5-Sandbox-AWS ...
* Starting ... done
* Requesting 3 instance(s) .......... done
* Inspecting capabilities of new instance(s) ....... done
* Running basic normalization scripts ......... done
* Registering instance(s) with Cloudera Manager .... done
* Waiting for Cloudera Manager to deploy agents on instances ... done
* Creating CDH4 cluster using the new nodes ...... done
* Downloading CDH-4.6.0-1.cdh4.6.0.p0.26 parcel ..... done
* Distributing CDH-4.6.0-1.cdh4.6.0.p0.26 parcel ... done
* Activating CDH-4.6.0-1.cdh4.6.0.p0.26 parcel ...... done
* Done ...
Cluster ready.
```

> **Note:** If you have a large root disk partition or if you are using a hardware virtual machine (HVM) AMI, the instances can take a long time to reboot. Expect to wait 20-25 minutes for Cloudera Manager to become available.

2. To monitor Cloudera Director, log in to the cluster launcher and view the application log:

```
 $ ssh ec2-user@54.186.148.151
Last login: Tue Mar 18 20:33:38 2014 from 65.50.196.130
[ec2-user@ip-10-1-1-18]$ tail -f ~/.cloudera-director/logs/application.log
[...]
```

> **Note:** If you have deployment issues and need help troubleshooting, be careful distributing the state.h2.db or application.log files. They contain sensitive information such as your AWS keys and SSH keys.

Connecting to Cloudera Manager

After the cluster is ready, log in to Cloudera Manager and access the cluster.

To access Cloudera Manager:

1. Use the status command to get the host IP address of Cloudera Manager:

```
$ cloudera-director status aws.conf
```

Cloudera Director displays output similar to the following:

```
Cloudera Launchpad 1.0.0 initializing ...

Cloudera Manager:
* Instance: 10.0.0.110 Owner=wintermute,Group=manager
* Shell: ssh -i /root/.ssh/launchpad root@10.0.0.110

Cluster Instances:
* Instance 1: 10.0.0.39 Owner=wintermute,Group=master
* Shell 1: ssh -i /root/.ssh/launchpad root@10.0.0.39

* Instance 2: 10.0.0.148 Owner=wintermute,Group=slave
* Shell 2: ssh -i /root/.ssh/launchpad root@10.0.0.148

* Instance 3: 10.0.0.150 Owner=wintermute,Group=slave
* Shell 3: ssh -i /root/.ssh/launchpad root@10.0.0.150

* Instance 4: 10.0.0.147 Owner=wintermute,Group=slave
* Shell 4: ssh -i /root/.ssh/launchpad root@10.0.0.147

* Instance 5: 10.0.0.149 Owner=wintermute,Group=slave
* Shell 5: ssh -i /root/.ssh/launchpad root@10.0.0.149

* Instance 6: 10.0.0.151 Owner=wintermute,Group=slave
* Shell 6: ssh -i /root/.ssh/launchpad root@10.0.0.151

* Instance 7: 10.0.0.254 Owner=wintermute,Group=gateway
* Shell 7: ssh -i /root/.ssh/launchpad root@10.0.0.254

* Instance 8: 10.0.0.32 Owner=wintermute,Group=master
* Shell 8: ssh -i /root/.ssh/launchpad root@10.0.0.32

* Instance 9: 10.0.0.22 Owner=wintermute,Group=master
* Shell 9: ssh -i /root/.ssh/launchpad root@10.0.0.22

Launchpad Gateway:
```

```
* Gateway Shell: ssh -i /path/to/launchpad/host/keyName.pem -L 7180:10.0.0.110:7180
 -L 7187:10.0.0.110:7187 root@ec2-54-77-57-3.eu-west-1.compute.amazonaws.com

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

In this example, it is 10.0.0.110.

2. Change to the directory where your `keyfile.pem` file is located. Then, route the connection over SSH:

```
$ ssh -L 7180:cm-host-private-ip:7180 ec2-user@cm-host-public-ip
# go to http://localhost:7180 in your browser and login with admin/admin
```

   - **Note:** If you get a permission error, add the `.pem` file from the command line:

```
$ ssh -i <your-cert.pem> -L 7180:cm-host-private-ip:7180
ec2-user@cm-host-public-ip
```

3. Open a web browser and enter http://localhost:7180 to connect to Cloudera Manager. Use admin as the username and password.
4. Complete the setup by adding any additional services to the cluster. The CDH 5 parcel was already distributed by Cloudera Director.

Cloudera Director Server: Command Line

In addition to deploying clusters through the Cloudera Director server UI, you can use the Cloudera Director client to send configuration files that the server uses to deploy clusters.

This section describes how to submit a cluster configuration file to the Cloudera Director server. For information on how to deploy the server, see Cloudera Director Server: UI on page 9.

Submitting a Cluster Configuration File

When you submit a cluster configuration from a Cloudera Director client to the Cloudera Director Server, all communications are transmitted in the clear (including the AWS credentials).

If the client and server communicate over the Internet, make sure the communication occurs through a VPN.

To submit a cluster configuration file to the Cloudera Director Server, follow these steps:

1. Create a configuration file. See Modifying the Configuration File on page 15.
2. Install the latest version of the Cloudera Director client from the Cloudera Director Download Page.
3. Unzip the Cloudera Director client.
4. Change to the client directory and enter the following:

```
cloudera-director bootstrap-remote myconfig.conf --lp.remote.hostAndPort=
          host:port
```

   where *myconfig.conf* is the name of your configuration file, *host* is the name or IP address of the host, and *port* is its port.

   The Cloudera Director client provides deployment status.

Cluster Configuration

This section describes how to configure the cluster you deploy through Cloudera Director.

File Location

To create a configuration file, download and unpack Cloudera Director, open `aws.simple.conf` or `aws.reference.conf`, and save it as `aws.conf`.

Environment Settings

This section describes basic settings you must configure before deploying a cluster.

Setting	Type	Required	Description
name	string	yes	Specifies the name of the cluster in Cloudera Manager. Example: C5-Reference-AWS Default: none
provider	container	yes	Container for the cloud infrastructure provider.
id	string	yes	The ID of the cloud infrastructure provider; leave this set to aws. Example: aws Valid Values: aws Default: aws
accessKeyId	string	yes	The access key used to make AWS requests. Make sure the value is enclosed in double quotes. Example: RQU1JC3XKTTYYJTXDR Valid Values: Valid AWS access key. Default: none
secretAccessKey	string	yes	The secret access key used to make AWS requests. Make sure the value is enclosed in double quotes. Example: vWdg/y4AVuxdzZo00613exTSdbV558 Valid Values: Valid AWS secret key.

Setting	Type	Required	Description
			Default: none
publishAccessKeys	boolean	no	Specifies whether Cloudera Director automatically publishes your credentials as cluster configurations for Amazon S3 access. Example: true Valid Values: true | false Default: false
region	string	yes	The region in which to launch the cluster. Example: us-west-2 Valid Values: See Availability Zones . Default: none
regionEndpoint	string	no	Specifies the region endpoint for clusters launched in the .gov region. If you are not launching in the .gov region, leave this commented out. Example: ec2.us-gov-west-1.amazonaws.com Valid Values: any valid .gov region. Default: none
keyName	string	yes	The name of the key pair used to start the cluster launcher. Example: my-cloudera-keypair Valid Values: any valid key pair associated with the region. Default: none
subnetId	string	yes	ID of the subnet that you noted earlier. Example: subnet-5b818f1d Valid Values: any valid subnet ID in the region. Default: none

Setting	Type	Required	Description
securityGroupsIds	string	yes	ID of the security group that you noted earlier. Use the ID of the group, not the name (for example, sg-b139d3d3, not default). To specify more than one security group, separate them with commas and enclose the string with quotes. Example: sg-b139d3d3 Valid Values: any valid security group ID in the region. Default: none
instanceNamePrefix	string	yes	The prefix used to launch instances. This prefix is part of the instance name which you can use to find instances started by Cloudera Director in the AWS Console. Example: skynet-cluster-1 Valid Values: any string Default: none
rootVolumeSizeGB	integer	yes	Sets the size of the root volume for the cluster launcher. Example: 100 Default: 50
associatePublicIpAddresses	boolean	no	Specifies whether nodes will have public IP addresses. To optimize Amazon S3 data transfer performance, set this to true. Example: true Valid Values: true | false Default: false
image	string	yes	Specifies the AMI to use. Cloudera recommends Red Hat Enterprise Linux 6.4 (64bit). To find the correct AMI for the selected region, visit the Red Hat AWS Partner Page.

Setting	Type	Required	Description
			Example: ami-22558833 Valid Values: Any valid AMI running Enterprise Enterprise Linux 6.4 (64bit) Default: none Note: For more information about AMI selection, see Choosing an AMI on page 15.
ssh	container	yes	Container for SSH settings.
username	string	yes	Specifies the username for SSH access to the instances. Example: ec2-user Default: none
privateKey	string	yes	Specifies the location of the SSH private key. Example: ${?HOME}/.ssh/director_id_rsa Default: none

Instance Settings

This section describes settings that define instances. Once defined, you can launch these instance types for Cloudera Manager and nodes in the cluster.

Setting	Type	Required	Description
instances	container	yes	The container that specifies instance settings.
instance_type	container	yes	A container that specifies settings for a type of instance to launch. You can specify any string value. For example, you can create an instance type called "cm" that uses an m1.large instance and another instance type called "node" that uses an m1.xlarge instance.
type	string	yes	The type of instances to launch. Example: m3.2xlarge Valid Values: Any valid instance name. For a list

Setting	Type	Required	Description
			of valid instance types, go to Instance Types. Default: none
bootstrapScript	string	yes	Linux shell script that executes whenever a cluster instance reboots. After the instance boots, this script automatically runs. This script can contain anything you need for your environment including libraries, monitoring tools, security configurations, and so on. Example: """#!/bin/sh # This is an embedded bootstrap script that runs # as root and can be used to customize # the instances immediately after boot and before # any other Cloudera Director action # If the exit code is not zero Cloudera Director will # automatically retry echo 'Hello World!' exit 0 """ Valid Values: any valid script Default: none
tags	container	yes	Container for any tags to apply to the instances. These tags can be used to find your instances in the AWS Console or on your AWS invoice.
tag	string	yes	Specifies the name and value of the tag. Example: department: "Data Science"

Setting	Type	Required	Description
			Valid Values: Any valid name/value pair. Default: none

Cloudera Manager Settings

This section describes settings for the Cloudera Manager instance.

Setting	Type	Required	Description
cloudera-manager	container	yes	The container for Cloudera Manager settings.
instance	string	yes	Specifies the instance type to use that you defined in Instance Settings. Example: ${instances.cm} Valid Values: any instance type that you defined earlier. Default: none
tags	container	yes	Container for any tags to apply to the Cloudera Manager instance.
tag	string	yes	Specifies the name and value of the tag. Example: application: "Cloudera Manager 5" Valid Values: Any valid name/value pair. Default: none
customBannerText	string	no	Specifies custom banner text to display in Cloudera Manager. Example: "Managed by Cloudera Director" Valid Values: any valid string Default: none
enableEnterpriseTrial	boolean	no	When set to true, automatically enables a 60-day Cloudera Enterprise trial. Example: true Valid Values: true | false

Setting	Type	Required	Description
			Default: false

Database Settings

This section describes settings for configuring external databases. This section is optional. If no settings are specified, Cloudera Director uses the embedded PostgreSQL database.

Setting	Type	Required	Description
databases	container	no	The container for databases.
CLOUDERA_MANAGER	container	no	The container for the database used by Cloudera Manager.
ACTIVITYMONITOR	container	no	The container for the database used by the activity monitor.
REPORTSMANAGER	container	no	The container for the database used by the reports manager.
NAVIGATOR	container	no	The container for the database used by Navigator.
type	string	no	The type of database. Example: postgresql Valid Values: postgresql | mysql. Default: none Note: Cloudera currently provides PostgreSQL drivers. Drivers for other databases must be added with the bootstrap script.
host	string	no	The database host. Example: db.example.com Default: none
port	string	no	The database port. Example: 123 Default: none
user	string	no	A database user. Example: dbuser Default: none

Setting	Type	Required	Description
password	string	no	The password of the database user. Example: Pa$$word Default: none
name	string	no	The name of database. Example: cmdb Default: none

Cluster Settings

This section describes products and services to launch on instances in the cluster.

Setting	Type	Required	Description
cluster	container	yes	The container for the cluster.
products	container	yes	The container for products to launch.
CDH	string	no	The version of CDH to launch. Example: 5 Valid Values: 4 | 5 Default: 4
IMPALA	string	yes	The version of Impala to launch. Example: 1.2 Default: none
services	array	yes	An array of services to launch. Options include: Example: [HDFS, YARN, ZOOKEEPER, HBASE, HIVE, HUE, OOZIE] Valid Values: HBASE, HDFS, HIVE, HUE, IMPALA, KS_INDEXER, MAPREDUCE, OOZIE, SOLR, SPARK, SQOOP, YARN, and ZOOKEEPER. Default: none
HIVE	container	no	The container for the database used by Hive. All Hive database settings are commented out by default.

Setting	Type	Required	Description
type	string	no	The type of Hive database. Example: postgresql Valid Values: postgresql | mysql. Default: none
host	string	no	The Hive database host. Example: db.example.com Default: none
port	string	no	The Hive database port. Example: 123 Default: none
user	string	no	A database user for Hive. Example: dbuser Default: none
password	string	no	The password of the database user. Example: Pa$$word Default: none
name	string	no	The name of Hive database. Example: cmdb Default: none
masters	container	yes	The container for service masters.
count	integer	yes	The number of instances to launch.
instance	string	yes	Specifies the instance type to use that you defined in Instance Settings. Example: ${instances.nodes} Valid Values: any instance type that you defined earlier. Default: none
tags	container	yes	Container for any tags to apply to the instances.

Setting	Type	Required	Description
tag	string	yes	Specifies the name and value of the tag. Example: group: master Valid Values: Any valid name/value pair. Default: none
roles	container	yes	Container for roles.
role	string	yes	Specifies the roles to apply to the masters. Example: HDFS: ${roles.HDFS_MASTERS} YARN: ${roles.YARN_MASTERS} ZOOKEEPER: ${roles.ZOOKEEPER_MASTERS} HBASE: ${roles.HBASE_MASTERS} HIVE: ${roles.HIVE_MASTERS} HUE: ${roles.HUE_MASTERS} OOZIE: ${roles.OOZIE_MASTERS} Default: none
workers	container	yes	Container for workers to launch.
count	integer	yes	The number of instances to launch.
instance	string	yes	Specifies the instance type that you defined in Instance Settings. Example: ${instances.nodes} Valid Values: any instance type that you defined earlier. Default: none
tags	container	yes	Container for any tags to apply to the instances.

Setting	Type	Required	Description
tag	string	yes	Specifies the name and value of the tag. Example: group: master Valid Values: Any valid name/value pair. Default: none
roles	container	yes	Container for roles.
role	string	yes	Specifies the roles to apply to the masters. Example: HDFS: ${roles.HDFS_MASTERS} YARN: ${roles.YARN_MASTERS} HBASE: ${roles.HBASE_MASTERS} Default: none
placementGroup	string	yes	Specifies the placement group in which to launch the instance. For more information, see Placement Groups.
gateways	container	yes	Container for gateways to launch. Note: Although this container is called gateways, containers at this level can use any name to launch a set of instances with shared instance settings and roles.
count	integer	yes	The number of instances to launch.
instance	string	yes	Specifies the instance type that you defined in Instance Settings. Example: ${instances.nodes} Valid Values: any instance type that you defined earlier. Default: none

Setting	Type	Required	Description
tags	container	yes	Container for any tags to apply to the instances.
tag	string	yes	Specifies the name and value of the tag. Example: group: master Valid Values: Any valid name/value pair. Default: none
roles	container	yes	Container for roles.
role	string	yes	Specifies the roles to apply to the masters. Example: HIVE: ${roles.HIVE_MASTERS} Default: none

- **Note:** Although you can deploy Flume through Cloudera Director, you must start it manually using Cloudera Manager.

Cluster Updates

This section describes how to make changes to the cluster through Cloudera Director.

Instance Updates

After launching a cluster, you can add instances.

To add instances to the cluster, follow these steps:

1. Open the `aws.conf` file that you used to launch the cluster.
2. Change the value for type of instance you want to increase. For example, the following increases the number of workers to 15:

```
workers {
      count: 15
      minCount: 5

      instance: ${instances.hs18} {
        tags {
          group: worker
        }
      }
```

3. Enter the following command:

```
cloudera-director update aws.conf
```

   Cloudera Director increases the number of worker instances.
4. Assign roles to the new master instances through Cloudera Manager. Cloudera Director does not automatically assign roles.

High Availability

After launching a cluster, you can upgrade it for high availability.

To upgrade the cluster, follow these steps:

1. Open the `aws.conf` file that you used to launch the cluster.
2. Change the count for the master instances. For example, the following adds two master instances to the one existing instance:

```
masters {
      count: 3

      instance: ${instances.cc28} {
        tags {
          group: master
        }
      }

      roles {
        HDFS: ${roles.HDFS_MASTERS}
        YARN: ${roles.YARN_MASTERS}
        ZOOKEEPER: ${roles.ZOOKEEPER_MASTERS}
        HBASE: ${roles.HBASE_MASTERS}
        HIVE: ${roles.HIVE_MASTERS}
        HUE: ${roles.HUE_MASTERS}
        OOZIE: ${roles.OOZIE_MASTERS}
```

```
      }
    }
```

3. Enter the following command:

```
cloudera-director update aws.conf
```

   Cloudera Director adds two master instances.
4. Assign roles to the new master instances through Cloudera Manager. Cloudera Director does not automatically assign roles.
5. Refer to the topics to enable high availability through Cloudera Manager.

Troubleshooting

This section describes common configuration and setup errors.

Configuration Issues

Depending on the size of the cluster, it can take up to 45 minutes or longer to deploy. If issues occur during deployment, check the following:

1. Incorrectly configured credentials
2. Overly specific region (us-west-2a instead of us-west-2)
3. Specifying default as the security group instead of its security group ID
4. Specifying a security group that is not part of the VPC

To troubleshoot configuration issues, view the `application.log` file.

After you correct the issue, delete the state.h2.db file and run Cloudera Director again.

DNS Issues

The AWS VPC must be set up for forward and reverse hostname resolution.

During deployment, the instance will ssh into the Cloudera Manager. Each time the cluster launcher attempts to ssh, it adds an application log entry similar to the following:

```
[2014-05-27 14:34:23] INFO  - c.c.l.b.UnboundedWaitForServerOnPort: Attempting connection
 to /10.0.4.42:22
```

DHCP Issues

Depending on the size of the disk, it can take a while for the Cloudera Manager to become available. If it takes too long, do the following:

1. Log in to web console at https://aws.amazon.com/console.
2. Select **VPC** from the **Services** navigation list box.
3. In the left pane, click **Your VPCs**. A list of currently configured **VPCs** appears.
4. Select the **VPC** you are using and note the **DHCP options set ID**.
5. In the left pane, click **DHCP Option Sets**. A list of currently configured DHCP Option Sets appears.
6. Select the option set used by the VPC.
7. Check for an entry similar to the following and make sure domain-name is specified:

   ```
   domain-name = ec2.internal
   domain-name-servers = AmazonProvidedDNS
   ```

8. If it is not configured correctly, create a new DHCP option set for the specified region and assign it to the VPC. For information on how to specify the correct domain name, see the AWS Documentation.

Known Issues

This section describes known issues in Cloudera Director.

Terminating clusters that are 'Bootstrapping' must be terminated twice for the instances to be terminated.

Description: Terminating a cluster that is 'Bootstrapping' stops on-going processes, but keeps the cluster in 'Bootstrapping' phase.

Severity: Low

Workaround: To transition the cluster to the 'Terminated' phase, terminate the cluster again.

Cloudera Director Glossary

availability zone

A distinct location in the region that is insulated from failures in other availability zones. For a list of regions and availability zones, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html.

Cloudera Director

An application for deploying and managing CDH clusters using configuration template files.

Cloudera Manager

An end-to-end management application for CDH clusters. Cloudera Manager enables administrators to easily and effectively provision, monitor, and manage Hadoop clusters and CDH installations.

cluster

A set of computers that contains an HDFS file system and other CDH components.

cluster launcher

An instance that launches a cluster using Cloudera Director and the configuration file.

configuration file

A template file used by Cloudera Director that you modify to launch a CDH cluster.

deployment

See cluster. Additionally, deployment refers to the process of launching a cluster.

environment

The region, account credentials, and other information used to deploy clusters in a cloud infrastructure provider.

ephemeral cluster

A short lived cluster that launches, processes a set of data, and terminates. Ephemeral clusters are ideal for periodic jobs.

instance

One virtual server running in a cloud environment, such as AWS.

instance group

A specification that includes general instance settings (such as the instance type and role settings), which you can use to launch instances without specifying settings for each individual instance.

instance type

A specification that defines the memory, CPU, storage capacity, and hourly cost for an instance.

keys

The combination of your AWS access key ID and secret access key used to sign AWS requests.

long-lived cluster

A cluster that remains running and available.

provider

A company that offers a cloud infrastructure which includes computing, storage, and platform services. Providers include AWS, Rackspace, and HP Public Cloud.

region

A distinct geographical AWS data center location. Each region contains at least two availability zones. For a list of regions and availability zones, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html.

tags

Metadata (name/value pairs) that you can define and assign to instances. Tags make is easier to find instances using environment management tools. For example, AWS provides the AWS Management Console.

template

A template file that contains settings that you use to launch clusters.