CLOUDERA

Educational Services

Cloudera Administrator Training for CDP PVC Base: Hands-On Exercises

Table of Contents

General Notes

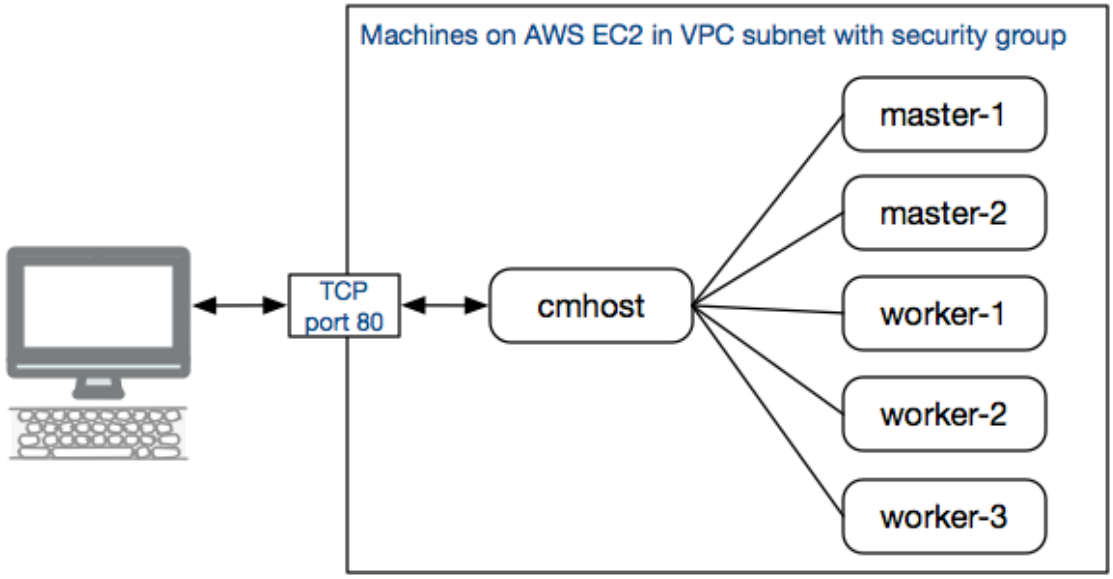
Important: Read this section before starting the exercises.

Using Your Exercise Environment

Exercise Environment Overview

In this course, you will install Cloudera Manager and CDP on six virtual machines (VMs) running in the cloud. These are referred to as the "cluster hosts."

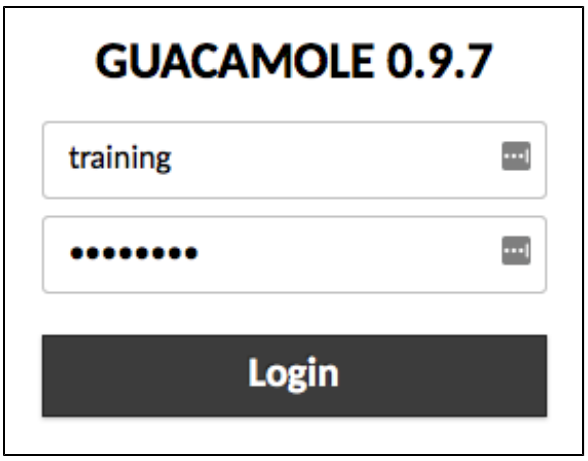
Your instructor will provide you with six IP addresses, one for each machine. In the first exercise, you will run a script to define which machine uses which IP address.



Accessing the Exercise Environment

One of the environment hosts is named `cmhost`. You will access your environment using the `cmhost` desktop. You interact with the desktop within your local web browser, which is available using the Apache Guacamole Remote Desktop Gateway.

- To use the desktop, enter the `cmhost` IP address in your browser. Enter your Guacamole client credentials: username `training`, password `Tr@inin9_20`.



- After a long idle period, the Guacamole client may time out with a message indicating a connection error. You will be prompted with a **Reconnect** button. If reconnecting using the button does not work, enter the `cmhost` IP address in your browser again.
- Note: If you stop and restart `cmhost`, the host will be assigned a new IP address. Use the *new address* to access the desktop.

Logging in to the Exercise Environment

All exercise steps are performed on the `cmhost` desktop, using the Firefox browser and terminal window(s).

Cluster host login credentials (all hosts):

- Username: `training`
- Password: `training`

Superuser access: You can use `sudo` without entering a password. The `training` user has unlimited, passwordless `sudo` privileges.

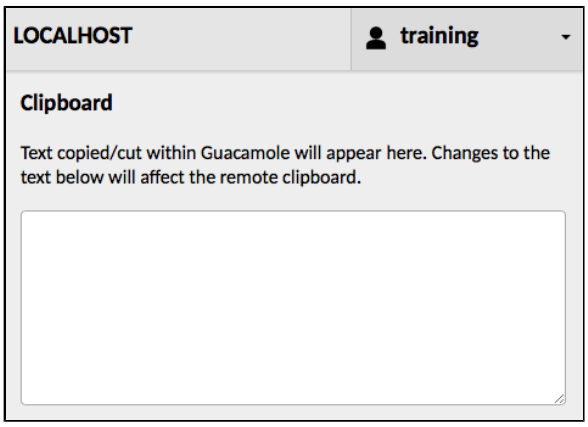
Logging in to other hosts: You can use `ssh` to connect from `cmhost` to the other cluster hosts without entering a password.

Note: Passwordless access is enabled for convenience in the course environment, but should not be enabled in a production environment.

Using the `cmhost` Remote Desktop

- Apache Guacamole provides the ability to copy text between the remote desktop and your local machine's clipboard.

1. Enter `Ctrl+Alt+Shift` to open the Guacamole menu. (Use `Ctrl+Cmd+Shift` on a Mac keyboard.)



---

2. Text on the remote desktop's clipboard will appear in the **Clipboard** area. You can copy the text to your local clipboard using `Ctrl+C`.

   You can paste text from your local clipboard into the **Clipboard** area using `Ctrl+V`. This will copy the text to the desktop clipboard.

---

3. Enter `Ctrl+Alt+Shift` or `Ctrl+Cmd+Shift` again to close Guacamole menu.

---

4. After closing the Guacamole you can paste the clipboard text into the remote desktop or your local desktop.

- Tips for using the cmhost desktop
  - Open a terminal window or start the Firefox browser from the **Applications > System Tools > MATE Terminal** desktop menu.
  - Change the size of the display using the `xrandr` command in a terminal on the desktop.
    - Show screen resolution options

```
$ xrandr
```

    - Example: change the screen resolution to 1680x1050

```
$ xrandr -s 1680x1050
```

- Change system settings using the **System > Preferences** desktop menu.
  - Configure or disable the screen saver in the **System > Preferences > Screensaver** section.
  - Configure or disable the lock screen in the **System > Lock Screen**.
- Launch the Terminal prompt in the **Applications, System Tools, MATE Terminal**.
- Get additional help using **System > Help**.

Points to Note While Doing the Exercises

Directories

The main directory for the exercises is `~/training_materials/admin`.

The `admin` directory contains these subdirectories:

- `data`—data files used in the exercises
- `java`—Java applications you will run during the exercises
- `scripts`—scripts you will run during the exercises
- `solutions`—text files containing the commands you will need to enter during the exercises
  - Use the solution files to make copying and pasting text easier.

Fewer Step-by-Step Instructions as You Work Through These Exercises

As the exercises progress, and you gain more familiarity with the tools and environment, we will provide fewer step-by-step instructions. As in the real world, we merely give you a requirement, and it is up to you to solve the problem. You should feel free to refer to the hints or solutions provided, ask your instructor for assistance, or consult with your fellow students.

Notational Convention

In some command-line steps in the exercises, you will see lines like this:

```
$ hdfs dfs -put shakespeare \
  /user/training/shakespeare
```

The backslash at the end of the first line signifies that the command is not complete and continues on the next line. You can enter the code exactly as shown (on multiple lines), or you can enter it on a single line and remove the backslashes.

Copying and Pasting Command Line Text from Exercise Manual

Sometimes you might need to copy commands from the exercises and paste them into your terminal window. There are two options to do this:

- Copy from the provided solution text files in the `training_materials/admin/solutions` directory. There is one text file for each exercise.
- Copy directly from this Exercise Manual PDF file using the Guacamole clipboard, as described in the section called "Using the **`cmhost`** Remote Desktop".

  However, note that copying multi-line commands from PDF to text can sometimes introduce problems. If you copy a line that ends a backslash (\), sometimes it will append the following line immediately after the backslash. If this happens, you need to manually remove the backslash.

Reset Script—Resetting Your Cluster

For most exercises, if you did not successfully complete the prerequisite exercise(s), **you can use the `reset_cluster.sh` script to set the state of your cluster as necessary to perform an exercise.**

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

Note the following points regarding the reset script:

- *Networking between hosts in your environment **must** be configured correctly before running the reset script. Be sure you have completed Hands-On Exercise: Configuring a Hadoop Cluster before running the reset script.*
- Some prerequisite exercises *cannot* be bypassed using a script. If you are unable to complete those exercises, speak to your instructor.
- The reset script is destructive: work you have already done on the cluster might be overwritten when the script runs.
- You can reset your environment to the original state, before Cloudera Manager is installed, by selecting **Full cluster reset**.
- Depending on which environment state you are resetting to, the script may take 30 minutes or more.

- The options you will be given include:
  - Full cluster reset - CM installed, no cluster
  - Cluster setup - CM and Basic cluster installed (thru ch 3)
  - High-availability setup - (thru ch 12)
  - Final setup - end of course (thru ch 15)
- After the cluster restarts, you might see some services with an orange dot indicating a concerning state due to swapping on `cmhost`. The warnings should resolve in about 15 minutes. You can continue with the exercises in the meantime.

Reset Data Script—Resetting Your Datasets

For most exercises, if you did not successfully complete the prerequisite exercise(s), **you can use the `reset_cluster.sh` script to set the state of your cluster as necessary to perform an exercise.**

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

Note the following points regarding the reset script:

- *Networking between hosts in your environment **must** be configured correctly before running the cluster reset script. If the cluster reset script needs to be ran, be sure you have completed Hands-On Exercise: Configuring a Hadoop Cluster before running the reset dataset script.*
- The options you will be given include:
  - Full tables and data clean up
  - Manage datasets on HDFS
  - Import data to HDFD with Sqoop
  - Create and manage external Hive tables

Hands-On Exercise: Configure the Exercise Network

In this exercise, you will configure your exercise environment's network in preparation for installing Cloudera Manager and deploying a Hadoop cluster.

IMPORTANT: Be sure to read through General Notes before starting this exercise.

Configure the Exercise Environment Network

1. Connect to the `cmhost` machine's remote desktop.

   - Your instructor will provide you with six IP addresses for the six hosts in your exercise environment. One of the addresses will be identified as `cmhost`.
   - Open a browser on your local machine (such as Google Chrome, Mozilla Firefox, Apple Safari, or Microsoft Internet Explorer 11 or more recent) and enter the `cmhost` IP address.
   - Enter the username `training` and password `Tr@inin9_20` and click **Login**.
   - If the remote desktop shows a blue screen saver
     - Press `Enter`.
     - At the password prompt, enter **training** as the password and click **Unlock**.

     Tip: You can change the screen saver and lock screen configuration using **Applications > System Tools > Settings**. See the section called "Using the `cmhost` Remote Desktop".

---

2. Open a terminal prompt on the `cmhost` remote desktop.

   - From the desktop **Applications** menu, choose **System Tools** > **Mate Terminal**.
   - You should see a terminal window with a `training@cmhost:~$` prompt.

---

3. In the terminal window, run a script to configure the `/etc/hosts` files on the virtual machines in your exercise environment so that they can all communicate.

```
training@cmhost:~$ ~/bin/config_ec2_hosts_files.sh
```

   - When prompted for the internal IP for `master-1`, enter the IP address for master-1 provided by the instructor.

- At each of the next four prompts, enter the IP address provided by the instructor for each of the machines that will host your cluster: master-2, worker-1, worker-2 and worker-3.

- The script will display the values you entered, asking you to confirm. Confirm the IP addresses are correct, then type `y`.

  Note: You can disregard any warnings.

The script should complete within about five minutes, after which it will display a message and return to the terminal prompt.

---

4. Verify that you can communicate with all the hosts in your cluster from `cmhost` by using the hostnames.

In the `cmhost` terminal, enter:

```
training@cmhost:~$ ping -c 3 cmhost
training@cmhost:~$ ping -c 3 master-1
training@cmhost:~$ ping -c 3 master-2
training@cmhost:~$ ping -c 3 worker-1
training@cmhost:~$ ping -c 3 worker-2
training@cmhost:~$ ping -c 3 worker-3
```

---

5. Confirm that the `training` user can use `ssh` to connect to the other virtual machines.

Note: Your environment is configured so that you can log in, submit commands, and copy files from `cmhost` to the other hosts in the environment without a password.

Passwordless SSH and `scp` **are not required to deploy a CDP cluster.** Passwordless connection is configured in the classroom environment as a convenience—do not do this in a production environment.

- In the same terminal window, connect to the `master-1` machine:

```
training@cmhost:~$ ssh master-1
```

- Confirm that the terminal window prompt has changed to `training@master-1:~$`, indicating that you are now connected to `master-1`.

Note: You might get a message starting with `Warning: Permanently added...`. You can disregard this message.

- Disconnect from the `master-1` machine.

```
training@master-1:~$ exit
```

- Confirm that the terminal window prompt has changed back to `training@cmhost:~$`.

- Use the same approach to connect to each of the other virtual machines. One at a time, connect to each of the other machines that will be part of the cluster, then exit each one after confirming that the prompt displays as show below.

```
training@cmhost$ ssh master-2
training@master-2:~$ exit
training@cmhost:~$ ssh worker-1
training@worker-1:~$ exit
training@cmhost:~$ ssh worker-2
training@worker-2:~$ exit
training@cmhost:~$ ssh worker-3
training@worker-3:~$ exit
```

Note: To simplify the instructions, the rest of the exercises will use a dollar sign (`$`) instead of the full prompt (such as `training@cmhost:~$`) to indicate the `cmhost` terminal prompt.

This is the end of the exercise.

Hands-On Exercise: Installing Cloudera Manager Server

For this installation, you will install Cloudera Manager Server on the `cmhost` machine. Before installing Cloudera Manager, you will configure an external database (MySQL) to be used by Cloudera Manager and some of the services, which you will install in the next exercise.

Configure the Internal Repository

In this section of the exercise, you will install the AWS CLI. Then you will install, configure and start the Apache Web server, and download the Cloudera Manager installation files. This will create the Cloudera Manager internal repository.

IMPORTANT: Complete all steps in this exercise on `worker-3`

1. Install the AWS CLI.

   In a terminal:

```
$ ssh worker-3
```

```
$ sudo yum install httpd
```

   Reply yes if prompted.

   Change the Cloudera Manager baseurl and gpgkey values.

```
$ sudo nano /etc/httpd/conf/httpd.conf
```

   Edit the port number on the line that reads **Listen 80** to use port 8060 instead of 80. Then save the file using Ctrl + O, and hit enter to accept the location. Exit with Ctrl + X.

   Start the Apache Web server, and then confirm the status as active:

```
$ sudo systemctl start httpd
```

```
$ sudo systemctl status httpd
```

Create the Cloudera Manager installation files. You will download the files, unzip them, set the file properties and restart the Apache web server.

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/cm7/
cm7.1.3-redhat7.tar.gz /var/www/html/cloudera-repos/cm7/
```

```
$ sudo tar xvfz /var/www/html/cloudera-repos/cm7/cm7.1.3-
redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-
components=1
```

```
$ sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cm7
```

```
$ sudo systemctl restart httpd.service
```

---

2. Test the repo URL. You will see the files you just uploaded.

   From **cmhost**, start a web browser and enter this URL:

```
$ http://worker-3:8060/cloudera-repos/cm7/
```

Verify Environment Configuration

IMPORTANT: Complete all steps in this exercise on `cmhost`

In this section of the exercise, you will verify some important settings prior to installing software. Although in these steps you will only verify settings on the `cmhost` machine, most of the settings below should be in place on all the machines that will be part of the cluster.

3. Verify the Oracle JDK is installed and that `JAVA_HOME` is defined and referenced in the system `PATH`.

   In a terminal:

```
$ java -version
```

The message returned in the terminal should show the java version is 1.8.0_232.

```
$ echo $JAVA_HOME
```

The result should show that `/usr/java/default` is the `JAVA_HOME` location.

```
$ env | grep PATH
```

The `PATH` value returned includes a reference to `/usr/java/default/bin`.

---

4. Verify Python is installed. It is a requirement for Hue, which you will install later in the course.

```
$ rpm -q python-2.7*
```

The command should return the package name of the installed version of python 2.7.

---

5. Verify MySQL Server is installed and running on `cmhost`.

```
$ systemctl status mysqld
```

The results of the command should show that the `mysqld.service` service is `“active” (running)`.

Note: Note that in a true production deployment you would also move the old InnoDB log files to a backup location and update the `/etc/my.cnf` MySQL configurations to conform with requirements as documented here.

---

6. Confirm no blocking is being done by Security-Enhanced Linux (SELinux)

```
$ sestatus
```

Note: Although it is not a requirement to set SELinux to disabled or permissive, it is important that SELinux not block during installation.

---

7. Verify IPv6 is disabled. This is a CDP requirement.

```
$ ip addr show
```

Notice that there is an IPv4 (`inet`) address for the `eth0` network interface, however there is no `inet6` address.

---

8. Check firewall settings.

```
$ systemctl list-unit-files | grep firewalld
```

The results of the command should show that the `firewalld.service` service is "`disabled`".

Note: Although it is not a requirement to disable the firewall, there are many ports that must not be blocked during and after installation.

---

9. Check Transparent Hugepage compaction is disabled.

```
$ cat /proc/meminfo | grep -i hugepages_total
```

A response of `HugePages_Total: 0` indicates the hugepage feature is turned off.

Note: The OS feature called transparent hugepages interacts poorly with Hadoop workloads and can seriously degrade performance. Cloudera recommends it be turned off.

---

10. Check the `vm.swappiness` Linux kernel setting.

```
$ cat /proc/sys/vm/swappiness
```

Note: Cloudera recommends that you set `vm.swappiness` to a value between 1 and 10, in order to reduce lengthy garbage collection pauses which can affect stability and performance.

---

11. Verify the service that ensures time consistency across machines is running.

```
$ ntpstat
```

The results of the command should show that the time is synchronized wth the ntp server.

---

12. Verify the MySQL JDBC Connector is installed. Sqoop (a part of CDP that you will install in this course) does not ship with a JDBC connector, but does require one.

```
$ ls -l /usr/share/java/mysql*
```

You should see that a symlink has been defined at `/usr/share/java/mysql-connector-java.jar` that points to a specific version of a MySQL connector JAR file that exists in the same directory.

Configure the External Cloudera Manager Database

As is typical in a production cluster, Cloudera Manager uses an external database system instead of the embedded PostgreSQL system.

13. Optional: Review the script you will run to create the required databases.

```
$ cat ~/training_materials/admin/scripts/mysql-setup.sql
```

This script creates databases and users for the Cloudera Manager database and databases required by other services on the cluster.

---

14. Run the script to create the databases.

```
$ mysql -u root -p < ~/training_materials/admin/scripts\
/mysql-setup.sql
```

When prompted for the password enter **training**

---

15. Confirm the databases were created.

```
$ mysql -u root -p -e "SHOW DATABASES"
```

When prompted for the password enter **training**

Confirm that the 13 databases created by the script include:`amon`, `cmserver`, `hue`, `metastore`, `oozie`, `streamsmsgmgr`, and `rman`. (The list will also include several previously created databases.)

---

16. Make your MySQL installation secure.

```
$ sudo /usr/bin/mysql_secure_installation
 [...]
 Enter current password for root (enter for none): training
 OK, successfully used password, moving on..
 [...]
 Change the root password? [Y/n] N
 Remove anonymous users? [Y/n] Y
 [...]
 Disallow root login remotely? [Y/n] Y
 [...]
 Remove test database and access to it [Y/n] Y
 [...]
 Reload privilege tables now? [Y/n] Y
 All done!
 [...]
 Thanks for using MySQL!
 Cleaning up . . .
```

---

17. Verify the Cloudera Manager local software repository.

Your instances contain a local yum repository of Cloudera Manager software to save download time in this course.

CentOS (and Red Hat) store software repository references in `/etc/yum.repos.d`.

- Issue the command below to edit the yum repository settings.

```
$ sudo nano /etc/yum.repos.d/cloudera-manager.repo
```

  - Update the file with the following settings:

    [cloudera-manager]

    baseurl = http://worker-3:8060/cloudera-repos/cm7/

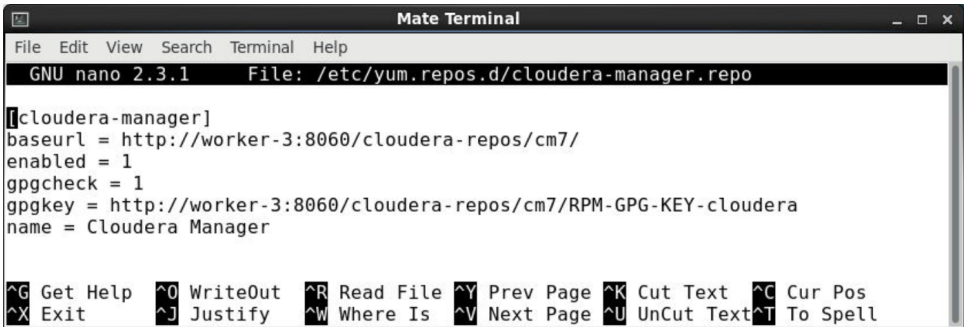
    enabled = 1

gpgcheck = 1

gpgkey = http://worker-3:8060/cloudera-repos/cm7/RPM-GPG-KEY-cloudera

name = Cloudera Manager

Notice the base URL in the output of the last command. You will type this URL into Cloudera Manager later when you install Cloudera Manager agents.



Install Cloudera Manager Server

18. Install Cloudera Manager Server.

```
$ sudo yum install -y cloudera-manager-daemons \
cloudera-manager-server
```

Note: The `-y` option provides an answer of `yes` in response to an expected confirmation prompt.

Note: This command will take some time due to the size of the new CDP RPM packages. The `yum install` command may fail at any point. If it does fail use the alternative command below:

```
$ cd ~/software/cm7/RPMS/x86_64
```

```
$ sudo yum localinstall -y cloudera-manager-server-7* \
cloudera-manager-daemons-7*
```

---

19. Run the script to prepare the Cloudera Manager database.

```
$ sudo /opt/cloudera/cm/schema/scm_prepare_database.sh \
mysql cmserver cmserveruser password
```

After running the command above you should see the message, "All done, your SCM database is configured correctly!"

---

20. Start the Cloudera Manager Server.

```
$ sudo systemctl start cloudera-scm-server
```

---

21. Confirm that Cloudera Manager Server process started successfully.

```
$ ps -ef | grep cloudera-scm-server
```

The results of the `ps` command above show that Cloudera Manager Server is using the JDBC MySQL connector to connect to MySQL. It also shows logging configuration and other details.

Note: You should see multiple entries for the `cloudera-scm-server` process. If there is only one entry for the `cloudera-scm-server` process that displays, it means that the service did not start correctly. This can usually be corrected by rebooting `cmhost`. Use `sudo reboot` and enter password `training` when prompted. After rebooting, you will need to reconnect your browser session to `cmhost`.

---

22. Review the Cloudera Manager Server log file to see what took place.

The path to the log file is `/var/log/cloudera-scm-server/cloudera-scm-server.log`. Note that you must use `sudo` to access Cloudera Manager logs because of restricted permissions on the Cloudera Manager log file directories.

```
$ sudo less \
  /var/log/cloudera-scm-server/cloudera-scm-\
  server.log
```

Tip: Bash tab-completion won't work for the filename in the above command, because the permissions on the directory do not allow the training user to view its contents until the `sudo` command actually runs.

Tip: Press the `Enter` key to scroll through the log file contents. Press the `Ctrl + c` to quit the `less` command file viewer.

This is the end of the exercise.

Hands-On Exercise: Cluster Installation

In this exercise, you will log in to the Cloudera Manager Admin Console to install, deploy and verify your cluster.

During this exercise, you will identify the hosts that Cloudera Manager will manage.

You will then be prompted to choose which CDP services you want to add in the cluster and to which machines you would like to add each role of each service.

At the end of this exercise, the Cloudera Manager and CDP services and roles will be deployed across your cluster as show below. (The services and roles added in this exercise are in blue).

	master-1	master-2	worker-1	worker-2	worker-3	cmhost
HDFS NameNode	✓					
HDFS Secondary NameNode		✓				
HttpFS	✓					
HDFS Balancer	✓					
HDFS DataNode			✓	✓		✓
Hive Metastore	✓					
HiveServer 2 on Tez	✓					
Oozie Server		✓				
Spark History Server		✓				
Spark Gateway	✓	✓				
YARN Resource Manager	✓					
YARN JobHistory Server	✓					
YARN NodeManager		✓	✓	✓		✓
Zookeeper Server	✓	✓	✓			
Cloudera Manager Server						✓
Cloudera Manager Server Database						✓
Cloudera Management Services						✓
Cloudera Manager Agent	✓	✓	✓	✓		✓

In subsequent exercises, you will add more services to your cluster.

After completing the installation steps, you will review a Cloudera Manager Agent log file and review processes running on a machine in the cluster.

All steps in this exercise that use a terminal window should be run on `cmhost`.

Download CDP version 7.1.2 Parcel

In this section of the exercise, you will download the files needed for the cluster installation of CDP 7.1.2.

1. In a terminal on worker-3:

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/7.1.2/
CDH-7.1.2-1.cdh7.1.2.p0.4253134-el7.parcel /var/www/html/
cloudera-repos/7.1.2/
```

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/7.1.2/
CDH-7.1.2-1.cdh7.1.2.p0.4253134-el7.parcel.sha /var/www/
html/cloudera-repos/7.1.2/
```

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/7.1.2/
manifest.json /var/www/html/cloudera-repos/7.1.2/
```

   This places the repository files where they can be reached.

---

2. Test the repo URL.

   From **cmhost**, start a web browser and enter this URL: You will see the files you just uploaded.

```
$ http://worker-3:8060/cloudera-repos/7.1.2/
```

Log in to Cloudera Manager Admin UI

3. If Firefox is not yet running, launch it from the `cmhost` desktop's **Applications, Internet, Firefox**.

   Note: Firefox may take a minute to launch the first time it is started.

---

4. In Firefox, load the Cloudera Manager Admin Console by entering `http://cmhost:7180`.

   Note: If an "Unable to Connect" message appears, the Cloudera Manager server has not yet fully started. Wait several moments, and then attempt to connect again.

---

5. Log in with the username `admin` and password `admin`.

   The **Welcome to Cloudera Manager** page.

---

6. On the "Welcome to Cloudera Manager 7.1.3" page, select **Try Cloudera Data Platform for 60 days**. Add a checkmark in the box next to **Yes, I accept the Cloudera Standard License Terms and Conditions** and click **Continue**.

---

7. The **Add Cluster - Installation Welcome** page. Click **Continue**.

---

8. The **Add Cluster - Installation Cluster Basics** page will appear. Enter the cluster name or accept the default Cluster 1. Click **Continue**. A Regular Cluster will be created.

Install Cloudera Manager agents and create cluster

9. The **Specify Hosts** for your CDH cluster installation page appears.

   Type in the names of five of the six hosts in the environment, separated by spaces:

   ```
   cmhost master-1 master-2 worker-1 worker-2
   ```

   Note: Do **not** include `worker-3` at this time. That host will be used in a different exercise.

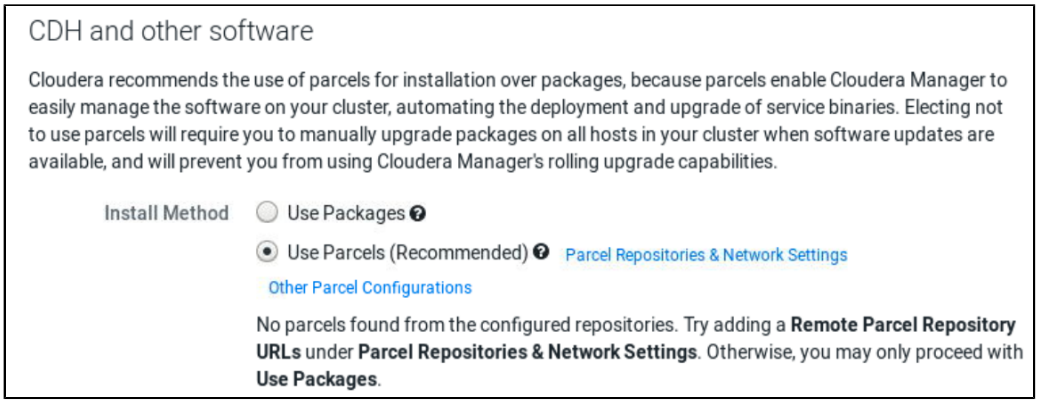
   Click **Search**. All five hosts should be found. Make sure that all five are selected, then click **Continue**.

Expanded Query	Hostname (FQDN)	IP Address	Currently Managed	Result
cmhost	cmhost.example.com	10.0.1.19	No	Host was successfully scanned.
master-1	master-1.example.com	10.0.5.114	No	Host was successfully scanned.
master-2	master-2.example.com	10.0.2.70	No	Host was successfully scanned.
worker-1	worker-1.example.com	10.0.8.213	No	Host was successfully scanned.
worker-2	worker-2.example.com	10.0.5.207	No	Host was successfully scanned.

   The **Select Repository** page appears. In this screen, you will identify the location of the CDP parcel and the Cloudera Manager Agent installer.

---

10. In the **Install Method** section, under **CDH and other software** area of the page, ensure the **Use parcels** option is selected.



---

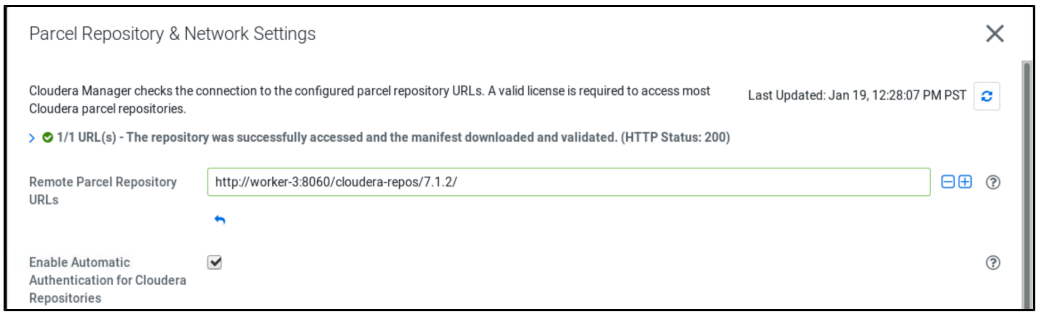
11. The **Parcel Repository & Network Settings** pop-up will appear. -->

In the **Parcel Repository and Network Settings** popup, click on each of the minus sign (**-**) icons to remove *ALL* the current repository references.

Click on the **plus sign** and add the following entries:

http://worker-3:8060/cloudera-repos/7.1.2/

Click **Save & Verify Configuration**. Once you have verified all errors have been removed click on **Close** to return to the **Select Repository** page.
-->



---

12. Click the **Other Parcel Configurations** link

In the **Other Parcel Configurations** popup we will make a change toone of the settings. Notice that there are severalproperties that can be set during this installation. Change the **Parcel Update Frequency** to 5 minute(s)

Click **Save Changes** to return to the **Select Repository** page.

---

13. Ensure that Version `Cloudera Runtime 7.1.2-1.cdh7.1.2.p0.4253134` is selected. Note that there will be several versions listed. Please ensure you have the correct one selected. Although we installed Cloudera manager 7.1.3, we will

be installing 7.1.2 to the cluster. W We will upgrade the cluster to 7.1.3 later in the course.

---

14. Click **Continue** to save the installation repository settings.

---

15. The **Select JDK** page appears.

A supported version of the Oracle JDK is already installed in your exercise environment. Therefore, select the option to **Manually manage JDK**.

Click **Continue**.

---

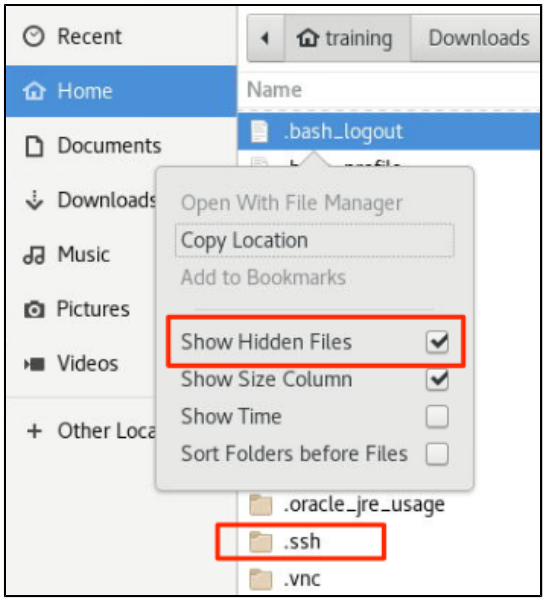
16. The **Enter Login Credentials** page appears.

a. Ensure that **Login To All Hosts As** is changed to **Another user** and enter **training**.

b. For **Authentication Method**, choose **All hosts accept same private key**.

c. Click the **Choose File** button to select a private key file.

Select **Home** in the location selector panel on the left. Then right-click in the **Name** area and select **Show Hidden Files**.



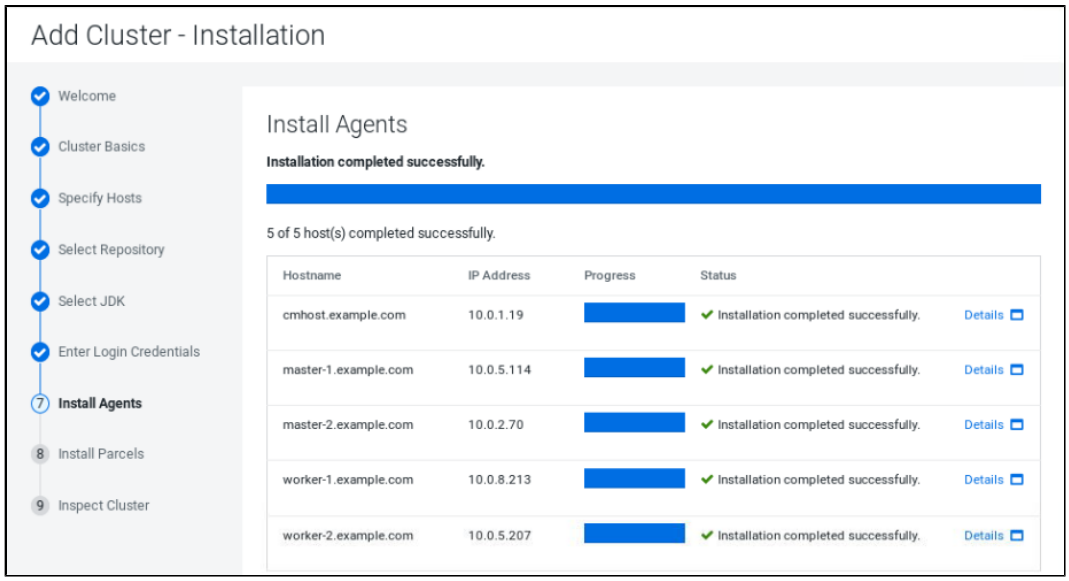
Double-click to open the `.ssh` directory, select the `id_rsa` file, and click **Open**.

d. Leave the passphrase fields blank fields and click **Continue**.

---

17. The **Install Agents** page appears.

Cloudera Manager installs the Cloudera Manager Agent on each machine. It may take up to five minutes.



After the Cloudera Manager Agent is successfully installed on all five machines, click **Continue** if needed.

---

18. In the next step, the parcel is downloaded, distributed, unpacked, and activated on all hosts in the cluster.

When the parcel is activated on all five hosts, click **Continue**.

---

19. The **Inspect cluster** page appears.

Select the **Inspect Network Performance** button. After a few moments, the results will appear. All validations should succeed.

Then click on the **Inspect Hosts** button. After a few moments, the results will appear. All validations should succeed.

If you wish you can review the reports from each inspection.

Click **Continue**.

Tip: The Host Inspector can be run on existing cluster hosts at any time from the Cloudera Manager admin console.

---

20. You have now completed installing Cloudera Manager and creating a cluster.

The **Add Cluster - Configuration** wizard will walk you through installing and deploying services on the cluster starts automatically.

Set up Cluster Services

21. The first page of the Add Cluster - Configuration wizard is the **Select Services** page.

Notice the note at the bottom of the screen stating "This wizard will also install the **Cloudera Management Service**."

Click **Custom Services**, which will display a table appears with a list of CDP service types.

---

22. Select the following services:

- HDFS
- Hive
- Hive on Tez
- Oozie
- Spark
- Tez
- YARN
- ZooKeeper

Double check that you have selected the correct services before continuing, then click **Continue**.

---

23. On the next page, specify which hosts in the cluster serve which roles.

Assign roles to hosts as shown in the table below.

Note: The instructions below have you assign the HDFS DataNode role to `cmhost`. This is contrary to standard recommendations, even on a small cluster. Follow the instructions as shown, for now. You can modify this setup later.

Tip: To assign a role to a particular host, click on a field with one or more hostnames in it. For example, the field under **SecondaryNameNode** might initially have the value `worker-1`. Change this to a different host by clicking on the field, which will open a window where you can choose a new host. Note that the interface will include '.example.com' in all host names, as in 'master-1.example.com'.

	Role	Node(s)
HDFS		
	NameNode	`master-1`
	SecondaryNameNode	`master-2`
	Balancer	`master-1`
	HttpFS	`master-1`
	NFS Gateway	Do not specify any hosts
	DataNode	Custom: `cmhost`, `worker-1`, `worker-2`
Hive		
	Gateway	Do not specify any hosts
	Hive Metastore Server	`master-1`
	WebHCat Server	Do not specify any hosts
	HiveServer2	Do not specify any hosts
Hive on Tez		
	Gateway	`master-1, master-2`
	HiveServer2	`master-2`
Cloudera Management Service		
	Service Monitor	`cmhost`
	Activity Monitor	`cmhost`
	Host Monitor	`cmhost`
	Reports Manager	`cmhost`
	Event Server	`cmhost`
	Alert Publisher	`cmhost`
	Telemetry Publisher	Do not specify any hosts
Oozie		
	Oozie Server	`master-2`
Spark		
	HistoryServer	`master-2`
	Gateway	`master-1, master-2, cmhost`
Tez		
	Gateway	`cmhost`, `master-2`

	Role	Node(s)
YARN		
	ResourceManager	`master-1`
	JobHistoryServer	`master-1`
	NodeManager	Same as DataNode
ZooKeeper		
	Server	`cmhost`, `master-1`, `master-2`

When you have finished assigning roles, carefully verify that your role assignments are correct. When you are certain that the settings are correct, click **Continue**.

---

24. The **Setup Database** page appears. This allows you to specify the database connection details for each service's database in the Cloudera Manager database system. The databases and access credentials were created when you ran `scm_prepare_database.sh` during the Cloudera Manager installation process.

Note: The Database Hostname should be set to `cmhost` or `cmhost.example.com` for each service. Either one will work.

Fill in the details as shown here.

Service	Database Hostname	Database Type	Database Name	Username	Password
Hive	`cmhost`	`MySQL`	`metastore`	`hiveuser`	`password`
Activity Monitor	`cmhost`	`MySQL`	`amon`	`amonuser`	`password`
Reports Manager	`cmhost`	`MySQL`	`rman`	`rmanuser`	`password`
Oozie Server	`cmhost`	`MySQL`	`oozie`	`oozieuser`	`password`

Click **Test Connection** to verify that Cloudera Manager can connect to the MySQL databases you created in an earlier exercise in this course.

After you have verified that all connections are successful, click **Continue**.

---

25. The **Review Changes** page appears. Leave the default values for all settings. Click **Continue**.

---

26. The **First Run Command** page appears. Progress messages appear while cluster services are created and started.

When all the cluster services have started, click **Continue**.

---

27. The **Summary** page appears.

The page indicates that services have been added and are now configured and running on your cluster. Click **Finish**.

The Cloudera Manager home page will appear.

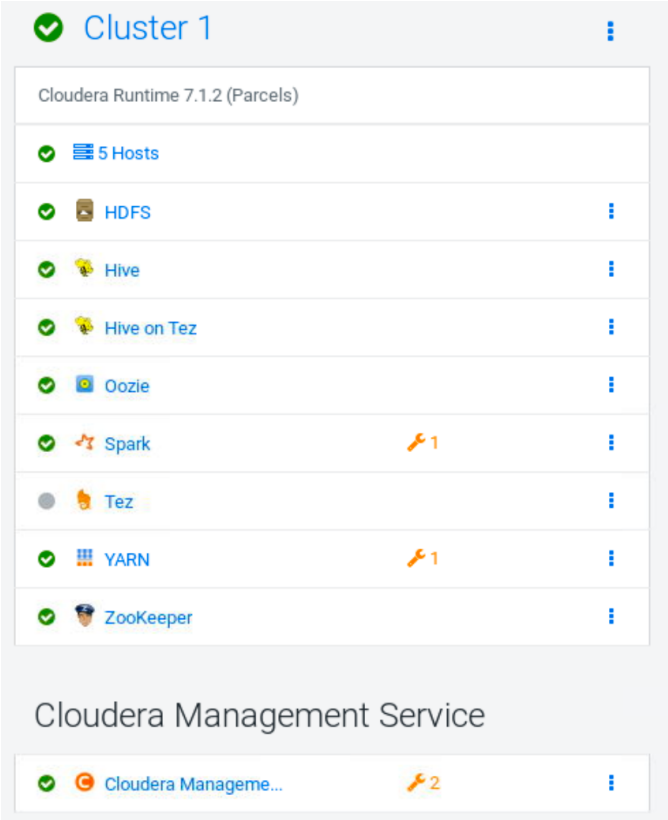
---

28. On the home page, confirm that all services are running correctly—that is, they should have a green checkmark indicator. If any of the services are not running, open the drop-down menu to the right of the service name and select **Restart**.

Configuration Warnings

The configuration warnings—as seen in the screenshot below—are expected, and indicate that, although Cloudera Management Services and the Service Monitor are in good health, but they do not have the recommended amount of memory available. There is also an indicator that YARN needs the YARN Queue Manager which we will install later. There is also an erroneous message indicating that we need a Spark Gateway on master-1 and master-2, which already exist.

In a production deployment, you would need to ensure these warnings were addressed. However, in the exercise environment, you can safely ignore them at this time.



Cluster installation is now complete.

This is the end of the exercise.

Hands-On Exercise: Configuring a Hadoop Cluster

In this exercise, you will modify a service configuration, activate additional parcels, and add additional services to your cluster. You will then create a Cloudera Manager host template and apply it to an existing host in the cluster.

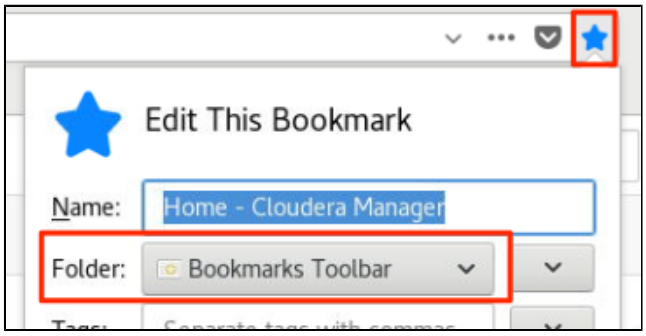
Modify a service configuration

In this step, you will practice changing configuration settings by changing the HDFS replication factor. (The replication factor determines how many copies of each file in HDFS are kept on the cluster.)

1. Locate and change the HDFS replication factor setting.

   a. In the remote desktop browser, go to the Cloudera Manager admin console home page (`cmhost:7180`).

      Tip: You might wish to bookmark the Cloudera Manager home page, because you will visit it frequently throughout the exercises.



   b. Go the HDFS service page by clicking the **HDFS** service in the **Cluster1** list of services.

   c. Select the **Configuration** tab.

   d. From the **Category** set of filters on the left, select the **Replication** filter.

   e. Change the replication Factor to **2**.

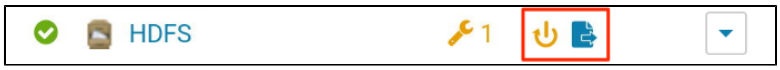
   f. Click **Save Changes**.

---

2. Restart stale services and redeploy client configurations:

a. Return to the CM home page by clicking the Cloudera Manager logo in the upper left corner.

b. Note that the HDFS service (and several services that depend on HDFS) show two new status icons. One is the "Stale Configuration: Restart needed" icon. The other is the "Stale Configuration: Client configuration redeployment needed" icon.



Click on either one of the two icons. This opens the **Stale Configurations** page.

c. Review the changes that Cloudera Manager will push out to the cluster hosts and then click **Restart Stale Services**.

d. In the **Review Changes** screen, keep **Re-deploy client configuration** checked and click **Restart Now**.

e. The **Restart Awaiting Staleness Computation Command** page is displayed. Wait for the commands to complete, which should take less than five minutes. Then click **Finish**.

> **Resolving Stale Configurations**
> In future exercises, you might be instructed to change configuration settings without being explicitly reminded to redeploy the changes. Whenever you see stale configuration icons, you can resolve them as you did above.

Set an unexposed property using a configuration snippet

Most configuration properties can be set directly in Cloudera Manager. However, some less common properties are not exposed, and must be set using a "safety valve" snippet, which modifies a configuration file directly.

One such property is `dfs.datanode.scan.period.hours`, which determines how often HDFS scans for corrupt files.

3. In Cloudera Manager, return to the HDFS **Configuration** tab.

---

4. Find the **HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** property.

Tip: You can find this either by selecting **Category > Advanced** in the **Filters** panel on the left, or by entering `hdfs-site` in the search box.

---

5. Notice that the advanced configuration snippet currently contains no settings. Click the plus sign icon next to the configuration name to add a setting.

---

6. In the **Name** field, enter `dfs.datanode.scan.period.hours`.

---

7. In the **Value** field, enter `240`. This overrides the default scan frequency (every three weeks) to be every 10 days.

---

8. *Optional*: Enter a description of the change, such as `scan every 10 days`.

---

9. Click **View as XML**. This displays the actual command that will be added to the `hdfs-site.xml`.

---

10. As you did in the previous section, save your changes, then redeploy the configuration files and restart the affected services.

Install CFM parcels

11. Open a **terminal** window on worker-3 and download the parcels for CFM.

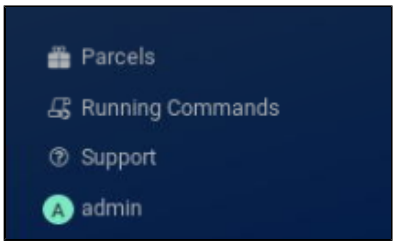
```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/CFM/
CFM-2.0.1.0-71-el7.parcel /var/www/html/cloudera-repos/CFM/
```

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/CFM/
CFM-2.0.1.0-71-el7.parcel.sha /var/www/html/cloudera-repos/
CFM/
```

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/CFM/
manifest.json /var/www/html/cloudera-repos/CFM/
```

Distribute and activate parcels

12. Go to the **Parcels** page in Cloudera Manager by clicking on the parcel icon on the bottom left.



---

13. Select the **Parcel Repositories and Network Settings** button and add the following URL: http://worker-3:8060/cloudera-repos/CFM/. Save and Close that window.

---

14. Under **Location** in the left filter panel, ensure that **Cluster 1** (the name of your cluster) is highlighted.

---

15. Select the parcel named **CFM** and click on **Download**.

---

16. When CM has downloaded the parcels, click **Distribute**.

---

17. When CM has distributed the parcels, click **Activate**. If you are prompted for whether you are sure want to activate the parcel, click **OK**. Once the parcel is activated, the button will display Deativate. We will leave it in that stage.

---

18. If any of the cluster services show stale configuration icons, resolve the stale configuration as you did earlier.

Add Nifi to the cluster

19. From the Cloudera Manager home page, choose **Add Service** from your cluster drop-down menu.

---

20. Choose **NiFi** and click **Continue**.

---

21. On the next screen, assign the **NiFi Node** role to **master-2**. Click **Continue**.

---

22. Continue through the rest of the wizard, keeping all default settings, until NiFi is installed. Click **Finish**.

---

23. If any of the cluster services show stale configuration icons, resolve the stale configuration as you did earlier in this exercise.

---

24. NiFi will appear as it is running, and two health issues will appear on the cluster status page associated with NiFi.

    This problem is a known issue with this version of Nifi running on CDP 7.1.2. Upon research you would find a workaround as explained in the next steps.

---

25. Locate the advanced configuration snippet again for `staging/state-management.xml` and add two entries:

    a. Click **NiFi** from the services list, and then click the **Configuration** tab.

    b. In the search box, enter **Node Advanced Configuration Snippet (Safety Valve) for staging/state-management.xml**.

    c. Notice that the advanced configuration snippet currently contains no settings. Click on the plus sign to add a new setting.

    d. Add a new setting with the following values:

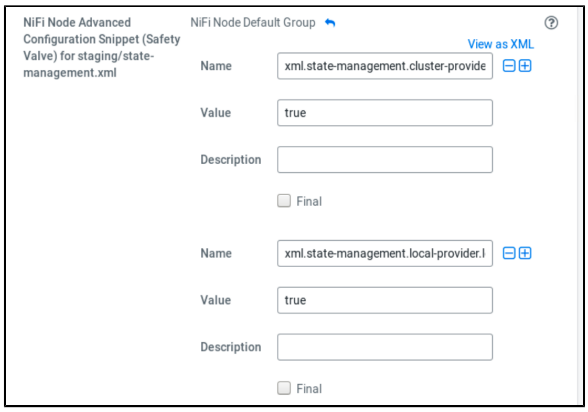
    Name: `xml.state-management.cluster-provider.zk-provider.enabled`

    Value: `true`

    e. Add another new setting with the following values:

    Name: `xml.state-management.local-provider.local-provider.enabled`

    Value: `true`



f. Save your changes.

g. Click on the "stale configuration" icon. (It may take a moment for it to appear.)

h. Click **Restart Stale Services**, **Restart Now** and then **Finish**.

Add Sqoop 1 to the cluster

Sqoop is a client install only. We will add the service, then add a Sqoop gateway in the next exercise when we create a gateway template.

26. From the Cloudera Manager home page, choose **Add Service** from your cluster drop-down menu.

---

27. Choose **Sqoop** and click **Continue**.

---

28. Do not deploy the **Gateway** role to any host at this time.

---

29. Continue with default settings through the rest of the wizard.

Create a host template for gateway hosts

The `cmhost` machine is currently a utility node. In this step, you will configure it with gateway roles so that it also plays the role of a gateway (or edge) node. This setup is consistent with Cloudera's recommendation for a cluster with fewer than 20 hosts.

30. Go to **Hosts** > **Host Templates**.

---

31. Click **Create**.

---

32. Enter template name `Gateway`.

---

33. Expand the HDFS and check the **Gateway**. Leave the gateway group set to **Gateway Default Group**.

Continue this process to add the Gateway roles for each of the following services as well:

- HDFS
- Hive
- Hive-on-Tez
- Sqoop
- YARN

---

34. Click **Create** to save the template.

---

35. Verify the settings of the **Gateway** template: five **Gateway Default Group** roles listed in the **Groups** column.

Apply the Gateway host template to `cmhost`

36. Select **Hosts** > **All Hosts**.

---

37. Check the box next to `cmhost`.

---

38. From the **Actions for Selected** menu, choose **Apply Host Template**.

---

39. Choose the **Gateway** host template you just created.

---

40. Place a checkmark next to **Deploy client configurations...**

---

41. Click **Confirm**.

---

42. The gateway role instances identified by the template you created will be deployed to `cmhost`. This will take a few minutes to complete.

When it completes, click **Close**.

---

43. Return to the Cloudera Manager home page.

---

44. If any of the cluster services show stale configuration icons, resolve the stale configuration.

---

45. Return to the **Hosts** > **All Hosts** page. Then click on blue arrow icon in the **Roles** column for `cmhost`.

Notice that `cmhost` now hosts the gateway roles that are part of the **Gateway** template, in addition to the other roles assigned to it earlier.

Tip: The gateway roles do not have running indicators (green checkmark icons) or stopped indicators because the gateway roles do not include daemons. They are client libraries that allow users to interact with the services to which they connect.

This is the end of the exercise.

Hands-On Exercise: Working with HDFS

In this hands-on exercise you will explore several ways to access and manage HDFS. You will also create a host template and use it to add a new worker host to your cluster.

IMPORTANT: This exercise depends on Hands-On Exercise: Configuring a Hadoop Cluster. Be sure to complete that exercise before continuing.

All steps in this exercise that use a terminal window should be run on `cmhost`.

Create a host template for worker hosts

In the next section, you will add a new worker host to the cluster. As you have probably noticed, we have reached critical data size on cmhost. We will move the DataNode off of cmhost. Before you do that, you will create a `worker` host template to use for the new host that includes the HDFS DataNode role.

1. In Cloudera Manager, select **Hosts** > **Host Templates** and click **Create**.

   a. Enter the template name **Worker**

   b. Select the following role groups to include in role groups in the template:

   - **HDFS DataNode** in **DataNode Default Group**
   - **YARN NodeManager** in the **NodeManager Default Group**

   c. Click **Create**.

   Verify that there are two different "... Default Group" roles in the `Worker` template.

Add a worker host to the cluster

The `worker-3` host was provisioned earlier, but was not added to the cluster. In this step, you will add it to the cluster using the `Worker` host template you created above.

2. Go to **Hosts** > **All Hosts**.

---

3. Click the **Add Hosts** button.

---

4. Select **Add hosts to cluster** option, and then click **Continue** to start the Add Hosts wizard.

---

5. In the **Specify Hosts** step, **Search** for the `worker-3` host. Confirm it is selected, then click **Continue.**

---

6. In the **Select Repository** screen, choose the **Custom Repository** option and click **Continue**.

---

7. Make sure that the **Manually manage JDK** option is selected, then click **Continue**.

---

8. Configure the new host's login credentials.

   a. Make sure that **Login To All Hosts As** is changed to **Another user** and enter **training**.

   b. Choose authentication method **All hosts accept the same private key**, then:

      i. Browse to `/home/training/.ssh` if needed.

      Tip: You may need to right-click and choose **Show hidden files** to see the `.ssh` directory.

      ii. Select the `id_rsa` file and click **Open**.

      iii. Click **Continue.**

---

9. The wizard will proceed to install the Cloudera Manager Agent. This may take a few minutes to complete.

   When it completes, click **Continue**.

---

10. The **Select Host Template** screen appears showing the available templates.

---

11. Choose the **Worker** template you created above and click **Continue**.

---

12. The role instances identified by the template you created will be applied to the new host.

    This may take a few minutes to complete.

---

13. After all parcels are activated on the new host, click **Continue** if needed. Typically it will move to the next page automatically.

---

14. On **Inspect hosts** screen, confirm that all validations succeeded. Click **Continue**.

---

15. After all the steps that start roles and deploy client configurations complete, click **Continue**.

16. Click **Finish**.

17. From the Status page of the Cloudera Manager Home page, click on any one of the **Stale Services** icons, and **Restart Stale Services** (be sure to redeploy client configurations), then click **Restart Now**. The cluster restart may take up to five minutes to complete.

18. After cicking **Finish**, go to the HDFS **Status** tab.

    Note that the configured capacity has increased to about 444 GB because you just added a fourth DataNode to your cluster.

Decommission and delete the `cmhost` DateNode and NodeManager roles

In the exercise environment, `cmhost` is currently serving as a YARN NodeManager and HDFS DataNode. Having a utility node serve these roles is not a good practice. Adding `worker-3` as a new DataNode and NodeManager added more capacity to the system, so you can now remove those roles from `cmhost`.

19. Go to the HDFS **Instances** tab.

20. Check the box next to **DataNode** that is deployed to `cmhost`.

21. Select **Actions for Selected > Decommission**.

22. In the confirmation dialog, click **Decommission** and wait for the action to complete, then click **Close**.

23. Verify on the HDFS **Instances** tab that the status of the DataNode role on `cmhost` is now **Decommissioned**.

24. Check the box next to **DataNode** that is deployed to `cmhost` again.

25. Select **Actions for Selected > Delete**, then confirm that you want to delete the role on the host.

26. Go to the **YARN** service page, then follow the steps above to decommission and delete the NodeManager role from `cmhost` as well.

---

27. Go to the Cloudera Manager home page and restart the stale services. Be sure to re-deploy stale configurations.

Confirm HDFS Processes and Roles

28. Confirm that all HDFS processes are running.

a. From the Cloudera Manager, click **HDFS** to go to the HDFS service page, then go to the **Instances** tab.

b. Notice that that the HDFS daemons (`NameNode`, `SecondaryNameNode`, and multiple `DataNodes`) are running on various hosts in the cluster. There is also a `Gateway` role deployed to the `cmhost` machine.

---

29. Use the search box in the HDFS **Configuration** tab, and search for **block size**.

Note that the HDFS Block Size setting defaults to 128MB.

View the HDFS File Browser in Cloudera Manager

30. On the HDFS service page, select the **File Browser** tab.

---

31. Click into the `user` directory.

Notice the user directories exist for the OS users that run the CDP services that exist in the cluster, such as `hive` and `hue`. However, there is no user space defined yet in HDFS for the `training` user. In the next section you will create the `training` user directory.

View HDFS User Folders

32. In the cmhost terminal, enter

```
$ hdfs dfs -ls /user/
```

Notice that there is no `/user/training` directory in HDFS.

---

33. Using the `hdfs` user, create a home directory for the training user in HDFS. Then set the correct permissions.

In the `cmhost` terminal, enter:

```
$ sudo -u hdfs hdfs dfs -mkdir /user/training
$ sudo -u hdfs hdfs dfs -chown training /user/training
```

---

34. Check to ensiure the folder has now been created:

```
$ hdfs dfs -ls /user/
```

Manage the HDFS Superuser Group

35. Try creating a new directory in HDFS:

```
$ hdfs dfs -mkdir /testdir
```

Note: The command fails with a "Permission denied" error, because the `training` user does not have permission to write to the root directory of HDFS.

---

36. In Cloudera Manager, go to the HDFS configuration tab.

---

37. Search for **dfs.permissions**. Note that permission-checking is enabled.

---

38. Note that `dfs.permissions.superusergroup` defaults to a group named `supergroup`. Users in `supergroup` are HDFS "superusers"—that is, they have full permissions to perform any operation in HDFS.

The `supergroup` OS user group does not exist yet on the NameNode host. Create the group, and add the `training` OS user to the group on the NameNode host (`master-1`):

```
$ ssh training@master-1 sudo groupadd supergroup
$ ssh master-1 sudo usermod -aG supergroup training
```

Note: If these commands fail with `command not found`, use these commands instead:

```
$ ssh training@master-1 sudo /usr/sbin/groupadd supergroup
$ ssh master-1 sudo /usr/sbin/usermod -aG supergroup training
```

> For simplicity, these exercises add the `supergroup` group only to the NameNode host. In a production environment, the cluster would likely be configured to get user and group information from an external source, such as LDAP or Active Directory. You would add the group in one place and it would propagate throughout the cluster in a consistent way.

---

39. Confirm the group was created.

```
$ ssh master-1 groups training
```

The command should show:

```
training : training wheel supergroup
```

---

40. Run the `hdfs dfsadmin` command to refresh the HDFS user group mapping cache.

```
$ hdfs dfsadmin -refreshUserToGroupsMappings
```

Add directories and files using the HDFS CLI

41. Re-run the command that failed due to a permission error earlier.

```
$ hdfs dfs -mkdir /testdir
```

---

42. Verify the directory was created.

```
$ hdfs dfs -ls /
```

---

43. Create three test files on the local `cmhost` file system and copy them into HDFS.

```
$ cd /tmp
$ touch testfile1 testfile2 testfile3
$ hdfs dfs -put testfile* /testdir/
```

---

44. Verify the files were placed in HDFS.

```
$ hdfs dfs -ls /testdir
```

---

45. Delete one of the files from HDFS.

```
$ hdfs dfs -rm /testdir/testfile1
```

Note the info message saying that the file was moved to the trash.

---

46. Delete another one of the files, this time skipping the trash.

```
$ hdfs dfs -rm -skipTrash /testdir/testfile2
```

---

47. Try to delete the directory.

```
$ hdfs dfs -rmdir /testdir
```

The command did not work, because there is still a file in the directory.

---

48. Delete the directory recursively, which deletes the directory and all its contents.

```
$ hdfs dfs -rm -r /testdir
```

---

49. Empty the `training` user's trash.

```
$ hdfs dfs -expunge
```

---

50. Run this command to restore the training user's permissions to the original state by removing the `training` user from the `supergroup` OS group on the NameNode:

```
$ ssh master-1 sudo gpasswd -d training supergroup
```

---

51. Force an HDFS user group mapping cache refresh.

```
$ hdfs dfsadmin -refreshUserToGroupsMappings
```

---

52. Test that the `training` user can no longer write to the HDFS root directory.

```
$ hdfs dfs -mkdir /weblogs
```

The command should fail with a permission denied error.

---

53. The `hdfs` user owns the NameNode process and has superuser permissions to HDFS, even without being in the group specified by the `dfs.permissions.supergroup` setting.

Run the `mkdir` command as the `hdfs` user.

```
$ sudo -u hdfs hdfs dfs -mkdir /weblogs
```

Tip: Remember that, in the exercise environment, the `training` OS user has full passwordless `sudo` permissions, so you can run any command as any user. This would not be typical for an end user in a production cluster.

---

54. Verify ownership of the new directory:

```
$ hdfs dfs -ls /
```

---

55. Change the owner of the `weblogs` directory to be the `training` user, then verify the new ownership.

```
$ sudo -u hdfs hdfs dfs -chown training /weblogs
$ hdfs dfs -ls /
```

Add web server log data to HDFS

In this section you will upload to HDFS a set of files containing log output from a web server.

56. Find the size of the data on the Linux filesystem.

```
$ du -sm ~/training_materials/admin/data/weblogs
```

The command shows the size in MB.

Find the number of files in the `weblogs` directory.

```
$ ls ~/training_materials/admin/data/weblogs | wc -l
```

---

57. Add the files to the new directory in HDFS.

```
$ hdfs dfs -put ~/training_materials/admin/data/weblogs/* \
  /weblogs/
```

The command will take a moment to copy the files from the `cmhost` machine's local filesystem to the cluster HDFS storage (the machines where the DataNode roles are running).

---

58. View a summary of how the weblogs are being stored.

```
$ hdfs fsck /weblogs -blocks
```

Tip: The "Total size" statistic shown in the results of the command is displayed in bytes.

---

59. Use the `fsck` utility to discover more detail about the files, including which DataNodes host copies (replicas) of the blocks that make up each file in the weblogs dataset.

```
$ hdfs fsck /weblogs -blocks -files -racks
```

Load the Ngrams dataset into HDFS

The Google Ngrams Dataset

In this section, you will load data from the Google Books Ngrams dataset, which is licensed under a Creative Commons Attribution 3.0 Unported License https://creativecommons.org/licenses/by/3.0/. The American English 1grams dataset is the one you will use.

It contains all the words extracted from American English publications from the Google Books corpus.

See https://aws.amazon.com/datasets/google-books-ngrams/ and http://storage.googleapis.com/books/ngrams/books/datasetsv2.html for more details.

60. In Cloudera Manager, go to the HDFS **Status** tab.

---

61. In the **HDFS Summary** panel, note the configured capacity.

It should show a total of 333 GB of storage capacity.

---

62. Use the Linux `du` command to view the size of the ngrams dataset. For this exercise, you will be using a subset of the dataset: the files ending with `a` through `e`.

```
$ du -h /ngrams/unzipped/*[a-e]
```

The approximate size of the sizes of the files ending with `a` through `e` is 7GB.

---

63. Copy dataset into HDFS.

```
$ hdfs dfs -mkdir /tmp/ngrams
$ hdfs dfs -put /ngrams/unzipped/*[a-e] /tmp/ngrams/
```

---

64. Open the NameNode web UI at http://master-1:9870 and click on the **Datanodes** page. Note the number of blocks stored on each worker host.

Refresh the page a few times to see that the number of blocks on each DataNode is increasing.

---

65. Return to the Cloudera Manager HDFS **Status** tab.

Note the **Total Bytes Written Across DataNodes** and **HDFS Capacity** charts. They should reflect the process of uploading the ngram data.

---

66. When the `put` action has completed, refresh the HDFS **Status** tab.

Configured Capacity should now show an additional 14 GB used of the total 333GB of space available. The new value reflects the prior level plus the 7GB you just uploaded, stored with 2x replication = 14GB space used to store the ngrams subset data.

Explore replication and block information

67. Return to the NameNode web UI and choose **Utilities** > **Browse the file system**. Open the `/tmp/ngrams` directory and note the information available for each block of data, including the file size, replication factor, and block size. Notice that the file sizes shown in the **Size** column are larger than the block sizes.

---

68. Click on the first file in the list: `googlebooks-eng-all-1gram-20120701-a`. This opens the **File information** pop-up window.

Notice that the size of the first block of the file (**Block 0**) is 134,217,728 bytes (128 MB). This is because the HDFS block size on your cluster is 128 MB.

---

69. Click on the **Block information** drop-down menu and switch the view to last block in the file (**Block 13**).

Notice that the last block is smaller than the first: about 57 MB.

In the **Availability** section, take note of which DataNodes hold a copy of each block. In a later step, you will locate the file on disk, so you need to know which DataNodes have a copy.

---

70. Select the value of the `Block ID` field for **Block 13** and copy it to the clipboard. You will need this value for the next step in this exercise.

---

71. Locate the HDFS block on the Linux file systems on one of the NameNodes in the cluster that has a copy of the file.

a. In the Cloudera Manager HDFS **Configuration** tab, search for "data directory." Note that the DataNode Data Directory is `/dfs/dn`.

b. In the terminal window, use the Linux `find` command to locate a copy of the block on disk.

Note: In the following command, replace `datanode` with the hostname of one of the two DataNodes that has a copy of the block. Also replace **block id** with the actual Block ID you copied from the NameNode Web UI. Be sure to keep the `'` and `*` characters as shown.

```
$ ssh training@datanode sudo find \
  /dfs/dn -name '*block-id*' -ls
```

c. Verify that two files with the block ID you copied appear in the `find` command output—one file with extension `.meta`, and another file without any extension.

The `.meta` file is a metadata file that contains checksums for sections of the block.

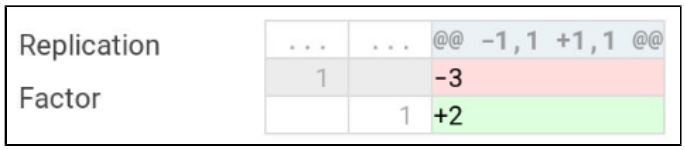
Increase the HDFS replication factor for new files

72. In the HDFS **Configuration** tab, click on **History and Rollback** on the right side of the page.

---

73. Locate the **Modified Replication Factor** revision listed towards to bottom

There may be more than one revision with that label. Use the **Details** link to find the one in which you previously changed the replication factor from the default value (3) to 2, as shown below.



74. After confirming in the **Revision Details** that you have selected the correct revision, click **Revert Configuration Changes**.

75. Redeploy the configurations and restart the stale services.

76. In a `cmhost` terminal, confirm the replication factor of new files.

```
$ echo "test file contents" > testfile
$ hdfs dfs -put testfile /tmp/
$ hdfs dfs -ls /tmp/testfile
```

The result of the last command should show that the replication factor of the new file is 3:

```
-rw-r--r--   3 training supergroup
```

Increase the replication for selected existing files

Changing the HDFS replication factor configuration does not automatically change the replication of existing files. In this section, you will manually change the replication of the files in the `weblogs` HDFS directory.

77. Verify that the current web log files are replicated twice.

```
$ hdfs dfs -ls /weblogs
```

78. Increase the replication factor of the files in the `weblogs` directory to 3.

```
$ hdfs dfs -setrep -R 3 /weblogs
```

79. Rerun the command to list the files to confirm that the replication factor has been correctly set to 3.

This is the end of the exercise.

Hands-On Exercise: Storing Data in Amazon S3

In this exercise, you will copy data from AWS S3 to HDFS.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

View Your Environment's AWS Credentials

1. When your exercise environment was created, you were assigned two AWS keys that will give you access to the course S3 bucket. You will find these values in the README.txt file in ~/training_materials/admin/ .

Copy Data from S3 to HDFS Using the `hadoop distcp` Command

2. Copy `shakespeare.txt` file from S3 into HDFS using the `hadoop distcp` utility. Enter the command below, which uses the AWS credentials provided in the README.txt file. Enter the correct value in place of $AWS_ACCESS_KEY and $AWS_SECRET_KEY.

```
$ hadoop distcp \
  -Dfs.s3a.access.key=$AWS_ACCESS_KEY \
  -Dfs.s3a.secret.key=$AWS_SECRET_KEY \
  s3a://admin-public/shakespeare.txt /tmp/
```

---

3. Confirm that the data is now in HDFS:

```
$ hdfs dfs -tail /tmp/shakespeare.txt
```

This is the end of the exercise.

Hands-On Exercise: Importing Data Using Sqoop

In this exercise, you will import data from a relational database using Sqoop.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. If you did not complete that exercise, run the `reset_cluster.sh` script. The script will reset your cluster, then prompt for which state the cluster should be set to. Choose the **Cluster setup** option to prepare for this exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Review the Database Tables

The MySQL database `movielens` contains a subset of data from the MovieLens project from University of Minnesota. (See note at the end of this exercise.) The database consists of several related tables, but you will import only two tables: `movie`, which contains about 3,900 movies, and `movierating`, which has about one million ratings of those movies.

Review the database tables to be loaded into HDFS.

1. Connect to MySQL in a `cmhost` terminal window.

   ```
   $ mysql -p movielens
   ```

   When prompted for the password, type in `training`.

---

2. Review the structure and contents of the `movie` table.

   ```
   mysql> DESCRIBE movie;
   …
   mysql> SELECT * FROM movie LIMIT 5;
   ```

---

3. Review the structure and contents of the `movierating` table.

   ```
   mysql> DESCRIBE movierating;
   …
   mysql> SELECT * FROM movierating LIMIT 5;
   ```

---

4. Exit MySQL.

```
mysql> quit;
```

Import RDBMS Data Using Sqoop

You invoke Sqoop on the command line to perform several commands. To connect to a database, provide a connect string to identify the server, database, and username and password.

5. Show the commands available in Sqoop.

```
$ sqoop help
```

You can safely ignore the warning that Accumulo does not exist, because this course does not use Accumulo.

6. List the databases (schemas) in your database server.

```
$ sqoop list-databases \
  --connect jdbc:mysql://cmhost \
  --username training --password training
```

Note: Instead of entering `--password training` on the command line, you can enter `-P` so Sqoop will prompt you for the password.

7. List the tables in the `movielens` database.

```
$ sqoop list-tables \
  --connect jdbc:mysql://cmhost/movielens \
  --username training --password training
```

8. Create a target directory in HDFS to import database data into.

```
$ hdfs dfs -mkdir /tmp/moviedata
```

9. Import the `movie` table into the HDFS directory you created above.

```
$ sqoop import \
  --connect jdbc:mysql://cmhost/movielens \
  --username training --password training \
  --table movie --fields-terminated-by '\t' \
  --target-dir /tmp/moviedata/movie
```

You can disregard warnings about missing packages.

> The `--fields-terminated-by '\t'` option separates the fields in the HDFS file with the tab character, which is sometimes useful if users will be working with Hive or Spark.

---

10. Verify that the command worked.

```
$ hdfs dfs -ls /tmp/moviedata/movie
$ hdfs dfs -tail /tmp/moviedata/movie/part-m-00000
```

---

11. Import the `movierating` table into HDFS. This time, save the imported data in Parquet format instead of tab-delimited text files.

```
$ sqoop import \
  --connect jdbc:mysql://cmhost/movielens \
  --username training --password training \
  --table movierating --as-parquetfile \
  --target-dir /tmp/moviedata/movierating
```

---

12. List the contents of the `/tmp/moviedata/movierating` directory to confirm the data was exported as Parquet files.

---

13. Use the `parquet-tools head` command to confirm that the data in the directory was stored correctly. This will display the column values of the first few records

in the imported dataset. Note that the file will need to be local. You will connect to master-1 and copy the file to the local file system.

```
$ hdfs dfs -get /tmp/moviedata/movierating/\
part-m-00000.parquet /tmp
```

```
$ parquet-tools head /tmp/part-m-00000.parquet
```

Note:

This exercise uses the `MovieLens` data set, or subsets thereof. This data is freely available for academic purposes, and is used and distributed by Cloudera with the express permission of the UMN GroupLens ResearchGroup. If you would like to use this data for your own research purposes, you are free to do so, as long as you cite the GroupLens Research Group in any resulting publications. If you would like to use this data for commercial purposes, you must obtain explicit permission. You can find the full dataset, as well as detailed license terms, at http://www.grouplens.org/node/73.

This is the end of the exercise.

Hands-On Exercise: NiFi Verification

In this exercise, you will verify the installation of NiFi by creating a simple dataflow. You will generate a FlowFile and log FlowFile attributes. The exercise is not intended to describe the details of how the dataflow works at this point.

It is recommended that the instructor walk students through this short exercise.

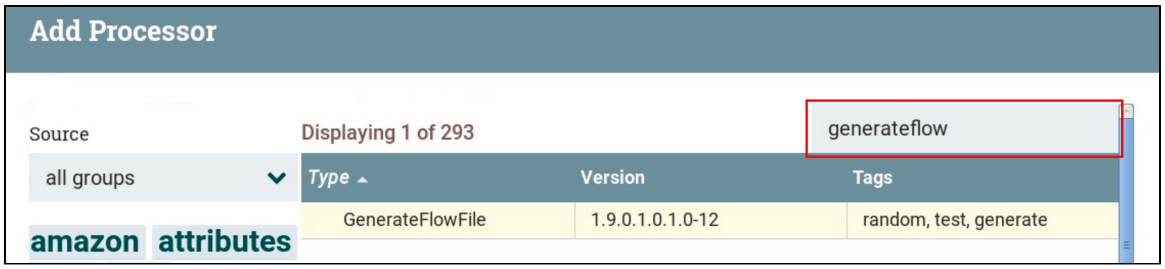
Create a Dataflow to Generate a FlowFile and log FlowFile Attributes

1. Configure a `GenerateFlowFile` processor.

   a. If you have not already done so, start the Firefox browser on your remote desktop.

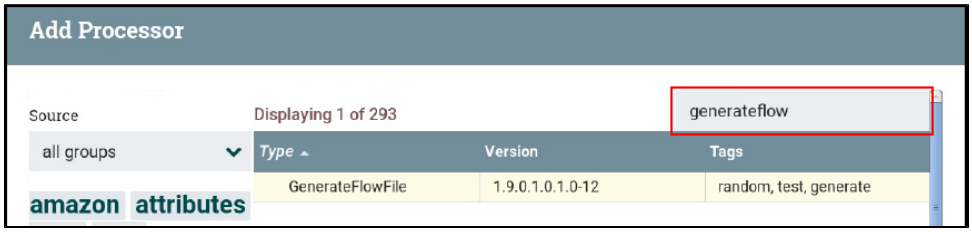
   b. Enter the URL for NiFi: `http://master-2:8080/nifi`.

   c. In the NiFi UI, drag a processor onto the canvas and filter with `GenerateFlowFile`.



   d. Click **ADD**.

   e. Right-click on the `GenerateFlowFile` processor and select **Configure**. (You can also double-click on the processor).



   f. On the **SETTINGS** tab, name the processor `First GenerateFlowFile`.

g. Click **APPLY**.

---

2. Configure a `LogAttribute` processor.

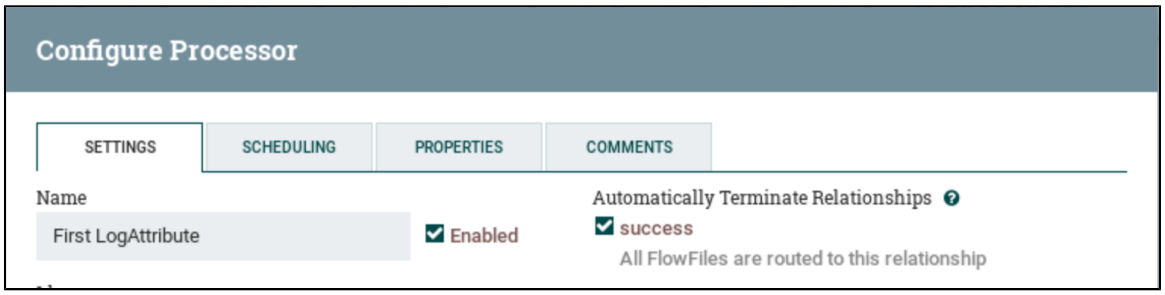
   a. Drag a processor to the canvas and filter with `logattribute` to add it.

   b. Right-click and select **Configure** or double-click on the processor.

   c. On the **SETTINGS** tab, name the processor `First LogAttribute`.

   d. Check the box next to **success** for **Automatically Terminate Relationships**.

   Your **SETTINGS** tab should look like this:

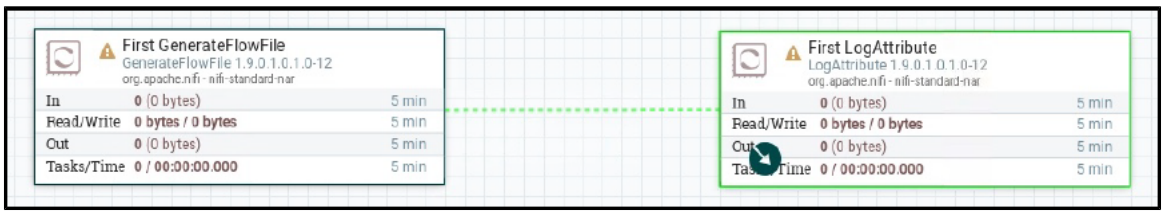


   e. Click **APPLY**.

---

3. Connect the `GenerateFlowFile` and `LogAttribute` processors.

   a. Hover over the `GenerateFlowFile` processor with your mouse and a connection symbol will appear ( ). Click on this symbol and drag it on top of the `LogAttribute` processor. A dashed line will appear and turn green when the connection is properly positioned.



   b. You will see a **Create Connection** screen. Ensure that **success** is checked on the **DETAILS** tab.

c. Click **ADD**.

Your dataflow should look like this:

---

4. Start the dataflow.

a. While holding the shift key, select the processors in the dataflow. (You can also hold down the shift key and drag an outline around all components).

b. On the **Operate** palette, click the "Start" button (shown as arrow icon) to start the dataflow

After a few seconds, you should see statistics on the surface of the processors, indicating that data is moving through your dataflow.

c. After observing the dataflow operate and noticing some statistics on the processors, stop the dataflow. Ensure that all dataflow processors are still selected, then click the "Stop" button (shown as a square stop icon) on the **Operate** palette.

This is the end of the exercise.

Hands-On Exercise: Working with Kafka

In this exercise, you will install Kafka and deploy it to your exercise cluster. Then you will use Kafka's command line tool to create a Kafka topic and use the command line producer and consumer clients to publish and read messages.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Add Kafka to the cluster

1. Return to the cluster Status page and add the Kafka service to your cluster.

   a. From the drop-down menu to the right of your cluster name, choose **Add Service**.

   b. Select **Kafka** and click **Continue**.

   c. Select **Zookeeper** and **HDFS** as dependencies, and click **Continue**.

   d. Add the **Kafka Broker** role to the `master-1`, `master-2`, and `cmhost` hosts.

   e. Do not deploy the **Gateway** or **Kafka MirrorMaker** or **Kafka Connect** roles to any host.

   f. Click **Continue**.

   g. On the **Review Changes** page, leave all configuration settings as their default values. Click **Continue**.

   h. Select **Finish**.

   i. Return to the Cloudera Manager home page. If prompted, confirm that you want to leave the current page.

---

2. Adjust the **Kafka Broker's Java heap size** and logging level.

   a. Go to the Kafka **Configuration** tab and make the following two configuration changes:

      - **Java Heap Size** = 512

- **Gateway Logging Threshold** = **WARN**

b. Save the changes.

---

3. Open Kafka's **Actions** menu and select **Restart**. When prompted, confirm that you want to restart the service.

When the service has started, click **Finish**. Notice that this method did not deploy the configuration changes. Select the icon to deploy all configuration changes.

Creating a Kafka Topic

4. In a `cmhost` terminal window, create a Kafka topic named `weblogs` that will contain messages representing lines in Loudacre's web server logs.

```
$ kafka-topics --create \
 --bootstrap-server
 master-1:9092,master-2:9092,cmhost:9092\
 --replication-factor 3 \
 --partitions 2 \
 --topic weblogs
```

---

5. Display all Kafka topics to confirm that the new topic you just created is listed:

```
$ kafka-topics --list \
 --bootstrap-server master-1:9092,master-2:9092,cmhost:9092
```

---

6. Review the details of the `weblogs` topic.

```
$ kafka-topics --describe \
 --bootstrap-server master-1:9092,master-2:9092,cmhost:9092
```

Producing and Consuming Messages

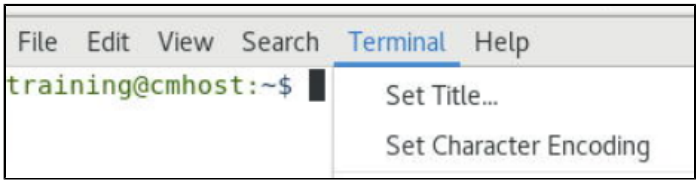
You will now use Kafka command line utilities to start producers and consumers for the topic created earlier.

7. Start a Kafka producer for the `weblogs` topic:

```
$ kafka-console-producer \
 --broker-list master-1:9092,master-2:9092,cmhost:9092 \
 --topic weblogs
```

You will see a few `SLF4J` messages, at which point the producer is ready to accept messages on the command line.

Tip: This exercise involves using multiple terminal windows. To avoid confusion, set a different title for each one by selecting **Set Title...** on the **Terminal** menu:



Set the title for this window to "Kafka Producer."

---

8. Publish a test message to the `weblogs` topic by typing the message text and then pressing `Enter`. For example:

```
test weblog entry 1
```

---

9. Open a new terminal window and adjust it to fit on the window beneath the producer window. Set the title for this window to "Kafka Consumer."

---

10. In the new terminal window, start a Kafka consumer that will read from the beginning of the `weblogs` topic:

```
$ kafka-console-consumer \
 --bootstrap-server master-1:9092\
 --topic weblogs \
 --from-beginning
```

After a few `SLF4J` messages, you should see the status message you sent above (with the producer) displayed on the consumer's console, such as:

```
test weblog entry 1
```

---

11. Press `Ctrl+C` to stop the weblogs consumer, and restart it, but this time omit the `--from-beginning` option to this command. You should see that no messages are displayed.

---

12. Switch back to the producer window and type another test message into the terminal, followed by the `Enter` key:

```
test weblog entry 2
```

---

13. Return to the consumer window and verify that it now displays the alert message you published from the producer in the previous step.

---

14. When you are done testing, stop both the producer and consumer processes with `Ctrl+C`.

This is the end of the exercise.

Hands-On Exercise: Install Impala and Hue

In this exercise, you will install the Impala and Hue services.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Add Impala and Hue to the cluster

1. Go to the Cloudera Manager home page.

---

2. Add the Impala service to your cluster.
   a. From the drop-down menu to the right of your cluster name, choose **Add Service**.
   b. Select **Impala** and click **Continue**.
   c. Add the **Impala StateStore** role to `master-2`.
   d. Add the **Impala Catalog Server** role to `master-2`.
   e. Add the **Impala Daemon** role to Custom: `worker-1, worker-2, worker-3`.
   f. Click **Continue**.
   g. On the **Review Changes** page, leave all configuration settings as their default values. Click **Continue**.
   h. Select **Finish**.
   i. Return to the Impala **Configuration** page.
   j. Search for the `MEM_LIMIT` setting. Change the limit to `1 Gib` for the Impala Daemon Default Group.
   k. **Save the Changes** and update the stale configuration.

      MEM_LIMIT: setting that the Impala Daemon uses up to this amount of memory for query processing, cached data, network buffers, background operations,

etc. If exceeded, queries running on the Impala Daemon will be killed until the Impala Daemon is under the memory limit.

---

3. Add the Hue service to your cluster.

   a. From the drop-down menu to the right of your cluster name, choose **Add Service**.

   b. Select **Hue** from the list of services and select **Continue**.

   c. From the Select Dependencies page, add the Optional Dependencies of Hive on Tez, Impala, Oozie, Spark, Tez.

   d. Click **Continue**.

   e. Add the **Hue Server** role to `cmhost`.

   f. Add the **Load Balancer** role to `cmhost`.

   g. Click **Continue**.

   h. One the Setup Database page, enter **cmhost** for the Database hostname, **hue** as the Database Name, **hueuser** as the Username and **password** as the Password.

   i. Click **Test Connection**.

   j. After connection is confirmed to be successful, click **Continue**.

   k. On the **Review Changes** page, leave all configuration settings as their default values. Click **Continue**.

   l. Once the First Run is complete, click **Continue**.

   m. Select **Finish**.

   n. Return to the Cloudera Manager home page. If prompted, confirm that you want to leave the current page. If there are any indications of stale configurations, ensure those are all updated.

This is the end of the exercise.

Hands-On Exercise: Using Hue, Hive and Impala

In this exercise, you will configure a Hue environment that provides business analysts with the following capabilities:

- Submitting Hive, and Impala queries via command line
- Create table in Hive metastore
- Use Hive and Impala in Hue

Users will be able to access their environments by using a Web browser, eliminating the need for administrators to install Hadoop client environments on the analysts' systems.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

Set Up Hue Permissions for Business Analysts

Analysts should be able to submit Hive and Impala queries, browse HDFS, manage table definitions in the Hive Metastore, and browse jobs.

1. Log in to Hue (`http://cmhost.example.com:8889`) as the user `training` with password `training`.

---

2. Click on your username in the lower left corner and select **Manage Users**.

---

3. Select the **Groups** tab.

---

4. Click **Add Group** and name the new group `analysts`.

---

5. In the **permissions** area, scroll or use the search field to enable the following permissions to access and launch applications:
   - `about.access`
   - `beeswax.access`
   - `filebrowser.access (all)`
   - `help.access`
   - `hive.access`
   - `impala.access`

- `jobbrowser.access`
- `metastore.write`
- `metastore.access`

---

6. Click **Add group**.

---

7. Select the **Users** tab.

---

8. Click **Add user**.

---

9. In the **Step 1: Credentials** step, create a user named `ada` with the password `ada`. Check **Create home directory**, then click **Next**.

---

10. In the **Step 2: Profile and Groups** step, enter first and last name **Ada Lovelace**. Make `ada` a member of the `analysts` group and remove her from the `default` group.

---

11. Click **Add User** to save Ada's user information and credentials.

---

12. Click your username (`training`) in the lower left corner and select **Sign out**.

---

13. Log back in to Hue as user `ada` with password `ada`.

    Because this is the first time this user has logged in to Hue, you will be offered an introduction to Hue 4. Click the **X** to close the pop-up window.

---

14. Verify that Ada can access only the Hue functionality configured for the `analysts` group.

    - Click the **</>** (Editor) menu and confirm that only the Impala and Hive query editors are available. Other options such as **Scheduler** and **MapReduce** query options should not be visible.

    - Open Ada's user menu and confirm that **Manage Users** and **Hue Administration** options are not available.

    - On the left Navigation bar, confirm that **Documents**, **Files**, **Tables**, and **Jobs** browser options are available, but **S3** is not.

- Go to the **Jobs** browser, and confirm that **Jobs** and **Impala** tabs are present, but the Oozie-related tabs such as **Workflows** are not.

---

15. Sign out of Hue, and log back in as the `training` user for remaining exercises.

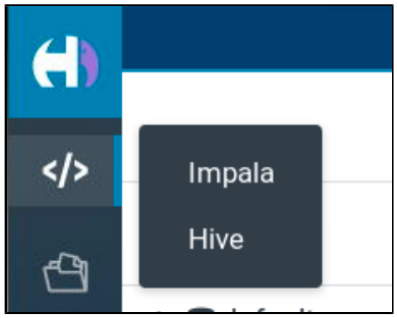
Create an external table in Hive for data stored in HDFS

16. Enter the following commands in a terminal window to set access to data.

```
$ sudo -u hdfs hdfs dfs -ls /warehouse/tablespace/external/
hive
$ sudo -u hdfs hdfs dfs -chown hive /tmp/ngrams
```

---

17. From the Navigation bar in Hue, select **Editor** > **Hive** to open the **Hive Query Editor**.



---

18. Enter the following SQL command in the query editor to create an internal table.

```
CREATE EXTERNAL TABLE ngrams (
    gram STRING,
    year INT,
    occurrences BIGINT,
    pages BIGINT,
    books BIGINT
  )
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY '\t';
```

---

19. Click the execute icon (triangle icon) to execute the SQL command to create the table in the metastore.

---

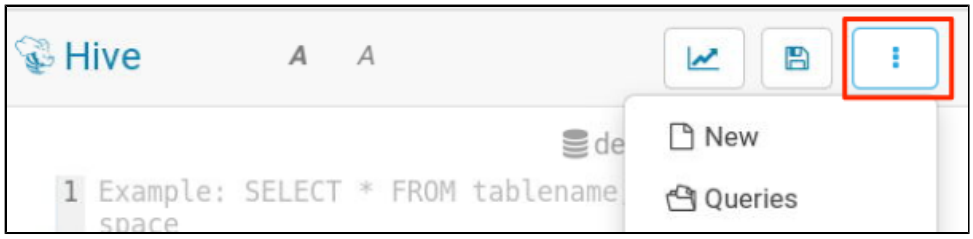
20. Click on the **default** database in the left side panel and confirm that the `ngrams` table was created.

---

21. Click on the table name in the left panel. The columns you just defined (`gram`, `year`, `occurrences`, `pages`, and `books`) should appear.

---

22. Select the **New** menu item to clear the previous SQL command.



---

23. Drag the `ngrams` table name into the query editor and release it. In the pop up menu that appears, choose the `SELECT * FROM default.ngrams...` option.

---

24. Run the `SELECT * FROM default.ngrams LIMIT 100;` query by clicking on the play icon (triangle icon).

No results are returned because there is no data in the table yet.

---

25. Clear the existing query again from the editor again, then run the following command to load the data into the table you created.

```
LOAD DATA INPATH '/tmp/ngrams' INTO TABLE ngrams;
```

This moves the `/tmp/ngrams` directory that was previously copied into the HDFS into the default Hive data warehouse location: `/warehouse/tablespace/external/hive/ngrams`.

---

26. In the **Query History** tab below the query editor, click on the `SELECT * FROM default.ngrams LIMIT 100` query that you ran earlier and run it again.

---

27. This time, 100 results should display in the **Results** tab.

---

28. Return to the Cloudera Manager HDFS file browser. Confirm that the data is no longer in `/tmp/ngrams` and has been moved to `/warehouse/tablespace/external/hive/ngrams`.

Note: The data was moved automatically by the `LOAD` command.

---

29. In the Hive Query Editor in Hue, run a more interesting query:

```
SELECT * FROM ngrams WHERE gram='computer';
```

This command will take a few minutes to run, because it must search through a large amount of data. You can continue with the exercises below while waiting for the command to finish.

---

30. While the query job is running, go to Cloudera Manager in a separate tab. Go to the YARN **Applications** tab.

a. Observe that the Hive query runs as a Tez job.

b. Click on the application **ID** link to go to the job details in the YARN ResourceManager UI.

---

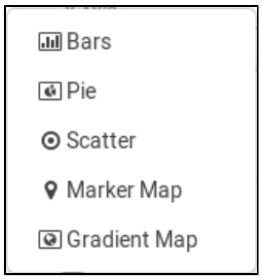
31. Return to the Cloudera Manager YARN **Applications** tab. This page shows a variety of metrics about the job. After the query has finished, reload the page and take note of how long the job took; you will notice that the resources remained available after the query finished.

---

32. View a graph of the results in the **Results** tab in the Hue Query Editor.

a. Scroll to the bottom of the result set to see the last year that returned data. (You must scroll to the end result row in order to ensure the values are cached to graph them in the next step.)

b. Click on the down arrow next to the bar chart icon in the **Results** tab (just to the left of the rows of data returned by the query) and choose **Bars**.



c. Set the details:

- X-AXIS: `ngrams.year`
- Y-AXIS: `ngrams.occurrences`

d. Observe the relationship between occurrences of the term *computer* and the year of publication.

Use the Beeline CLI for Hive

Create another table in the metastore, this time based on *compressed* ngrams data. For this section, you will use the Beeline CLI, which is an alternative to using Hue to access Hive.

33. On `cmhost`, run the following commands to load zipped (compressed) versions of the ngram dataset files into HDFS.

```
$ hdfs dfs -mkdir /tmp/ngrams_gz
$ hdfs dfs -put /ngrams/gz/*[a-e].gz /tmp/ngrams_gz/
```

---

34. In a `cmhost` terminal session, run the following command. This will start the Beeline shell and connect to the Hiveserver 2 service running on `master-2`.

```
$ beeline -u jdbc:hive2://master-2:10000 -n training
```

---

35. In Beeline, enter the following SQL commands.

```
> SHOW TABLES;
> CREATE EXTERNAL TABLE ngrams_zipped (
    gram STRING,
    year INT,
    occurrences BIGINT,
    pages BIGINT,
    books BIGINT
  )
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY '\t'
 LOCATION '/tmp/ngrams_gz';
> SELECT * FROM ngrams_zipped WHERE gram='data' LIMIT 5;
```

---

36. Exit the Beeline shell.

```
> !quit
```

Use the Impala Query Editor in Hue

Note: Even though the `ngrams_zipped` table was created in Hive, you can query the table using Impala. Both services access the same metastore. However, Impala caches data from the metastore, so you must invalidate the Impala cache after making any changes in Hive.

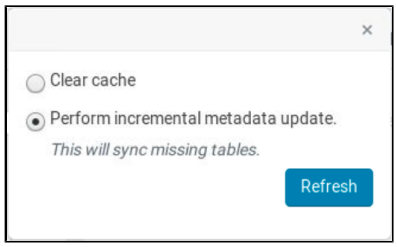
37. In Hue, select **Editor** > **Impala**. The Impala Query Editor looks very similar to the Hive Query Editor, but the queries you enter are executed by Impala instead of a Hive job on YARN.

---

38. In the "Assist" panel on the left, click the back arrow until the panel shows data sources **Impala** and **Hive**. Select **Impala**.

---

39. In the panel on the left, click the refresh icon next to the table list, then select **Perform incremental metadata update** and click **Refresh**.



40. Run a simple query

```
SELECT * FROM default.ngrams_zipped LIMIT 100;
```

Use the Impala Shell

41. Start the Impala shell in a terminal on `cmhost`.

```
$ impala-shell
```

42. Connect to an Impala daemon on any of the hosts where the daemon is running, such as `worker-1`.

```
> connect worker-1;
```

43. Execute the following commands to list available databases, connect to the `default` database, view the tables in the database, and execute a query.

```
> SHOW DATABASES;
> USE default;
> SHOW TABLES;
> SELECT * FROM ngrams WHERE gram='computer';
```

44. While the query is running, review the query details.

a. Go to the Impala service page in Cloudera Manager, then select the **Queries** tab.

b. Locate the drop-down menu on the right side of the **Results** row where the query is executing and choose **Query Details**.

The **Query Details** page shows the query plan and query information.

c. Visit the web UI of the Impala Daemon that you connected to to the query: `http://worker-1:25000/queries`. On the **Queries** page, locate the query you are running and click on the **Details** link in that row.

d. The details page show a graphical query plan. You can see the text version of the same plan by selecting the **Text plan** tab. This is the same plan you saw in Cloudera Manager. Query plan details can be helpful when optimizing complex queries.

e. When you are done exploring details about how the query is running (or *ran*, if it finished), return to the Impala shell. When the query is finished, note how long it took to complete. Compare the time it took to run the query using Impala and how long it took using Hive. (The time to fetch the requested rows is displayed immediately below the query output.)

f. Exit the Impala shell.

```
> exit;
```

This is the end of the exercise.

Hands-On Exercise: Running YARN Applications

In this exercise, you will run two YARN applications on your cluster. The application is a MapReduce job.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Submit a MapReduce application to your cluster

Run the WordCount application to count the number of occurrences of each unique word in a file containing the collected works of Shakespeare. WordCount a classic MapReduce program that demonstrates the map-reduce paradigm. It is included in the Hadoop installation.

1. In a `cmhost` terminal session, unzip and copy the data file to HDFS.

```
$ gunzip ~/training_materials/admin/data/shakespeare.txt.gz
$ hdfs dfs -put \
  ~/training_materials/admin/data/shakespeare.txt
```

Note: This command uploads the file to the default HDFS directory which is the user's home directory—`/user/training`.

---

2. WordCount is a Java application, packaged in a Java Archive (JAR) file. Use the `hadoop jar` command to submit it to the cluster. Like many MapReduce programs, WordCount accepts two arguments: the HDFS directory path containing the input and the HDFS directory into which output should be saved.

```
$ hadoop jar \
  /opt/cloudera/parcels/CDH/lib/hadoop-\
mapreduce/hadoop-mapreduce-examples.jar \
  wordcount shakespeare.txt counts/
```

---

3. After the program has completed, list the contents of the output directory in HDFS: `counts`.

```
$ hdfs dfs -ls counts
```

The output directory will include a `_SUCCESS` flag file and one file created by each of the application's reducer tasks.

---

4. View the contents of one of the output files.

```
$ hdfs dfs -cat counts/part-r-00000
```

Review the MapReduce application details and logs

In this section, you will start by looking at details in the YARN **Applications** tab and the MapReduce HistoryServer Web UI.

As you go through the steps below, try to reconstruct what occurred when you ran the MapReduce job above by creating a chart like the one below.

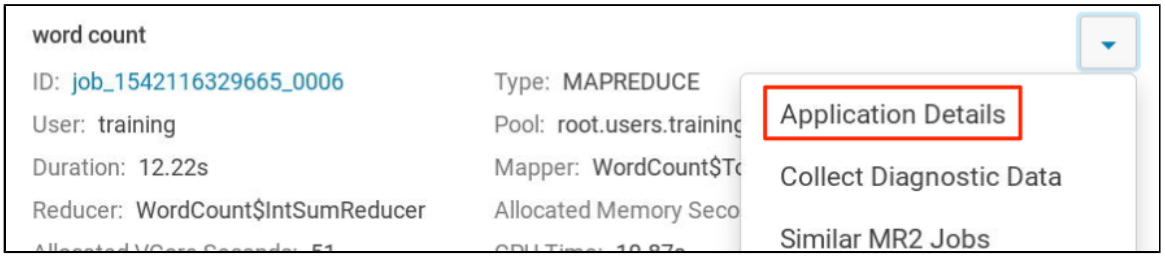
	Node(s)
Application Master	
Map Task(s)	
Reduce Task(s)	

5. From the Cloudera Manager home page, go to the **YARN** service page, the go to the **Applications** tab.

---

6. The most recently run application is displayed first in the list of applications in the **Results**. The application name should be **word count** and the type should be **MAPREDUCE**.

   From the drop-down menu to the right of the "word count" application, choose **Application Details**.



   This opens the the HistoryServer Web UI with details about the job.

---

7. In the **ApplicationMaster** area, note the cluster node on which the ApplicationMaster ran. Follow the **logs** link to view the ApplicationMaster log.

   Click the browser's back button when done viewing the log.

   Also note the number of map and reduce tasks that ran to complete the word count job. The number of reducers run by the job corresponds to the number of `part-r-#####` files you saw when you listed the files in the output directory earlier.

---

8. Identify the node where the mapper task ran and view the task log.

   a. From the HistoryServer Web UI, select the **Job** name, choose **Map** task type.

   b. From the **Map Tasks** table, click on the link in the **Name** column for the task.

      In the **Attempts** table, note that the **Node** column shows which node the map task attempt ran on.

   c. Follow the **logs** link and review the contents of the mapper task log. When done, click the browser back button to return to the previous page.

---

9. Identify the nodes where the reduce tasks ran and view the logs.

   a. From the HistoryServer Web **Job** menu, choose **Reduce** task type.

b. In the **Reduce Tasks** table, click on the link in the **Name** column for one of the tasks.

In the **Attempts table**, note that the "Node" column shows which node this reducer task ran on.

c. Click the **logs** link and review the contents of the log. Observe the amount of output in the reducer task log. When done, click the browser back button to return to the previous page.

---

10. Note the default log level for reducer tasks for the `word count` job.

a. Expand the **Job** menu and choose **Configuration**. The **Configuration** tab will display the first twenty entries from the job configuration settings that were in effect when the `word count` job ran.

b. In the Search field, enter `log.level`. Note that both `mapreduce.reduce.log.level` and `mapreduce.map.log.level` are set to `INFO`.

Note: `INFO` is default value for the "JobHistory Server Logging Threshold", which can be set in the Cloudera Manager YARN **Configuration** tab for your cluster.

Set the reducer task log level

MapReduce applications that use the Hadoop ToolRunner API allow you to pass several types of arguments to Hadoop, such as runtime configuration parameters, including log level settings.

11. You must delete the `counts` directory before running the WordCount program a second time. MapReduce applications will not run if you specify an output path that already exists. Remove the `counts` directory from HDFS.

```
$ hdfs dfs -rm -r counts
```

---

12. Rerun the WordCount program, this time passing it an argument to set the log level for reduce tasks.

```
$ hadoop jar \
  /opt/cloudera/parcels/CDH/lib/hadoop-\
mapreduce/hadoop-mapreduce-examples.jar wordcount \
  -D mapreduce.reduce.log.level=DEBUG \
  shakespeare.txt counts
```

The `-D` option allows you override a default property setting by specifying the property and the value you want to assign.

---

13. After the job completes, view one of the reducer logs.

From the YARN **Applications** tab for your cluster, locate the entry for the application that you just ran.

Click on the ID link, then go the the list of reduce tasks: **Job** > **Configuration**. Verify that:

- The value of the `mapreduce.reduce.log.level` configuration attribute is `DEBUG`.
- The reducer task's logs for this job contain `DEBUG` log records and the logs are larger than the number of records written to the reducer task's logs during the previous WordCount job execution.

This is the end of the exercise.

Hands-On Exercise: Running Spark Applications

In this exercise, you will run the interactive Spark shell, as well as a standalone Spark application. You will also use the Spark Application UI and history server to explore how Spark applications run on YARN.

IMPORTANT: This exercise depends on Hands-On Exercise: Install Impala and Hue. Be sure to complete that exercise before continuing.

Run and review a Spark application

1. In a `cmhost` terminal session, run the `CountNgramsByYear.py` Python Spark application, which counts the number of records in the `ngrams` Hive table by year. This application takes one argument: the directory to save the results to. Deploy the application in `cluster` mode.

```
$ spark-submit --deploy-mode cluster \
  ~/training_materials/admin/scripts/CountNgramsByYear.py \
  yearcounts/
```

   The application will run for about a minute while displaying log messages to the console.

---

2. In your browser, visit the ResourceManager UI—you can either go to the URL directly (`http://master-1:8088`), or start at the Cloudera Manager YARN **Applications** tab. Locate the most recently run application on the **Application** tab. Note the application name—**CountNgramsByYear.py**. Spark application names default to the file or class containing the application. Also note the application type, which should be **Spark**.

---

3. Click on the application name and review the application in the Spark History Server UI: Click on the **History** link on the right side.

---

4. The starting page is **Jobs**. Note that the application ran two jobs.

   Click on the description in the **Description** column for the job at the bottom of the list to see the job's details. This page shows that the job consisted of two stages.

---

5. Click on the description of one of the stages to view details about the tasks that made up the stage.

Review the **Tasks** section. This lists all the tasks in the stage, including the worker host the task ran on, metrics about the the time, the amount of data the task read and wrote to HDFS, and how long the task took.

Run the Spark shell

6. In the terminal session, start the Python Spark shell.

```
$ pyspark
```

Note: You cannot deploy the shell application in cluster mode. The driver must run on your local host in order to allow you to interact with the application.

---

7. In the Spark shell, enter a command to start a job:

```
>>> spark.read.table("ngrams").count()
```

This command returns the number of row in the `ngrams` Hive table.

---

8. In your browser, visit the ResourceManager UI. Note that the **PySparkShell** application is currently running. It will stay in the **RUNNING** state until you exit the shell.

---

9. Because the application is currently running, you can visit the Spark Application UI instead of the history server UI. Click on the **ApplicationMaster** link in the **Tracking UI** column on the Applications tab of the Resource Manager UI.

   The job started by the shell command you entered above should be displayed in the Spark Application UI.

---

10. Return to the terminal window and exit the shell.

```
>>> exit()
```

Review the Spark application logs

Access the Spark application logs from the command line.

11. Find the application ID using the `yarn application` command.

```
$ yarn application -list -appStates FINISHED
```

Locate the most recent YARN application in the list by looking at the application IDs for Spark (not Tez) jopbs. The most recent application run will have the highest application ID. Confirm that the application name corresponds to the last application you ran: `PySparkShell`. Copy the application ID to the clipboard.

---

12. Display the logs for the application using the `yarn logs` command. Substitute the application ID you copied above for *`appId`*.

```
$ yarn logs -applicationId appId | less
```

Use the space bar to scroll through the logs. Notice that the aggregated logs contain all the logs for the individual containers that ran the Spark executors, such as:

```
Container: container_1544099788748_0014_01_000001 on
 worker-1.example.com_8041
==========================================================
```

Enter `q` when you are done reviewing the logs.

Note: These Spark application logs are stored in HDFS in the `/user/spark/applicationHistory` directory.

This is the end of the exercise.

Hands-On Exercise: Using The Capacity Scheduler

In this exercise, you will insstall the YARN Queue Manager, submit jobs to the cluster and observe the behavior of the Capacity Scheduler.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Install the YARN Queue Manager

1. From the drop-down menu to the right of your cluster name, choose **Add Service**.

2. From the Add Service to Cluster 1 pages, Select **YARN Queue Manager**

3. Click **Continue**.

4. Add the **YARN Queue Manager Webapp** role to `master-1`.

5. Click **Continue**.

6. On the **Review Changes** page, leave all configuration settings as their default values. Click **Continue**.

7. *If you receive an error during installation*, follow the instructions below. Otherwise go to step 9 and continue from there.

8. Open a terminal window connected to master-1, and enter the following commands:

   a. sudo rm -rf /var/lib/hadoop-yarn

   b. sudo mkdir /var/lib/hadoop-yarn

   c. sudo chmod 755 /var/lib/hadoop-yarn

   d. Restart the YARN QueueManager Store role from the YARN **Instances** page. You will need to exit the installation wizard by selecting the **Cloudera Manger** logo and clicking on **Leave page** and skip the next two steps.

9. Click **Continue** on the First Run page.

10. Select **Finish**.

11. Enable the YARN Queue Manager by going to the **Configuration** page for YARN.

12. Search for queue and enable the **Queue Manager Service** by selecting the check box.

13. Click **Save Changes** and return to the Cluster Status page.

14. Update the stale configuration.

Adding Queues

15. Select the **Clusters** menu from the navigation bar and click on **YARN Queue Manager UI**.

16. View the only existing LEVEL 1 pool - the root pool.

17. Select the three dots for the LEVEL 1 **root** pool, and select **More Information**.

    The root pool information will open in a new tab.

18. Select the three dots for the LEVEL 2 **Default** pool, and select **More Information**.

    The **default** pool information will open in a new tab.

19. Return to Cloudera Manager And the YARN Queue Manager page. Select the three dots and select **Add Child Queue**.

20. Enter the queue name of **Analysts**. Then select **Save** and **OK**.

21. Select the three dots by the **Analysts** queue. Select View/Edit Queue Properties. Change the Maximum Applications to **4**. Then select **Save** and **OK**.

Run jobs on YARN with new pool

22. Select the three dots for the **Analyst** pool, and select **More Information**. A new tab will open. Close any other remaining tabs no longer being used.

---

23. To make it easier to start MapReduce jobs during this exercise, a script has been provided—`pools.sh`. The script starts a job in the pool you specify. Each job is long-running and is configured to ensure that there will be contention for system resources when multiple jobs are running.

Review the script to see how it works.

```
$ cat ~/training_materials/admin/scripts/pools.sh
```

---

24. Start a Hadoop job in the new pool created - Analysts.

```
$ cd ~/training_materials/admin/scripts/
```

```
$ ./pools.sh Analysts
```

---

25. Return to the **Applications** tab in the Analyst GUI to verify that the sleep job has started.

---

26. Click on the application name link for the application running in the Analysts queue.

---

27. Select the **Resource Usage** sub-tab to view the resources. If the application have finished, run it again to view the resources being used.

---

28. Explore the User interface for the Analysts queue, restarting the application as needed.

Configure Impala Admission Control

29. Select **Impala Admission Control Configuration** from the **Clusters** menu.

---

30. Click on the **Create Resource Pool** button.

---

31. Enter the name of `pool1`. Set the **Max Memory to 3 GiB** per cluster, then Minimumum query memory limit to 1 GiB, and set the **Max Running Queries** to **4**.

---

32. Click on the **Create** button.

---

33. Click on the **Refresh Dynamic Resource Pools** button.

Run an Impala query and view the process details

34. Open **Hue** and the **Impala** editor. By default the query will run against the root:default pool. Run the following query:

```
SELECT * FROM ngrams WHERE gram='computer';
```

---

35. In **Hue**, view the **Job Brower** and select the **Impala** tab. You will see the details on how the query including the amount of time it took.

---

36. Next we will set the pool to pool1 and run it again. Enter the follwoing command in the Impala editor.

```
SET request_pool='pool1'
```

```
SELECT * FROM ngrams WHERE gram='computer';
```

---

37. In **Hue**, view the **Job Brower** and select the **Impala** tab. You will see the details on how the new query including the amount of time it took. Compare that to the previous running on the root.default pool. Feel free to re-run the commands as needed.

This is the end of the exercise.

Hands-On Exercise: Configuring HDFS for High Availability

In this exercise, you will configure HDFS high availability, eliminating the NameNode as a single point of failure for your Hadoop cluster.

At the end of this exercise, the HDFS roles will be deployed across your cluster as depicted here. You will be removing all other roles as we will not be ising them any longer.

	master-1	master-2	worker-1	worker-2	worker-3	cmhost
HDFS NameNode	✓					
HDFS Secondary NameNode		✓				
HttpFS	✓					
HDFS Balancer	✓					
HDFS DataNode			✓	✓	✓	
YARN Resource Manager	✓					
YARN JobHistory Server	✓					
YARN NodeManager			✓	✓	✓	
Zookeeper Server	✓	✓	✓			
Sqoop	✓					
Cloudera Manager Server						✓
Cloudera Manager Server Database						✓
Cloudera Management Services						✓
Cloudera Manager Agent	✓	✓	✓	✓	✓	✓

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

Bring down uneeded Services

Since you will no longer use Hue, Oozie, Impala, Nifi, or Hive for the remaining execises, you can stop their services to improve your cluster's performance

Verify that the only services that are still up and running on your cluster are the Hosts, HDFS, Spark, YARN, ZooKeeper, and Cloudera Manager Service services. All these services should have good health.

1. In Cloudera Manager, navigate to the **Home** page.

In the row for the **Hue** service, chose **Stop** from the drop-down menu.

Click **Stop** in the confirmation window.

Click **Close** after messages in the Stop Command page indicate that the Hue service has stopped.

The Home page reappears. The status of the **Hue** service should have changed to Stopped.

---

2. Using steps like the ones you followed to stop the Hue service, stop the Oozie service.

---

3. Stop the Impala service.

---

4. Stop the Hive service.

---

5. Stop the Hive on Tez service.

---

6. Stop the NiFi service.

---

7. Stop the Kafka service.

Enable HDFS High Availability

8. In Cloudera Manager, go to the HDFS **Instances** tab.

   Take note of the HDFS service role instances that make up the HDFS service in the current non-high availability configuration.

   - Balancer
   - DataNotes
   - NameNode
   - SecondaryNameNode
   - HttpFS

   These roles will change when you enable high availability below.

---

9. From the **Actions** menu on the HDFS service page, choose **Enable High Availability**. This starts a wizard to walk you through the enable process.

---

10. Set **Nameservice Name** to `my-name-service`, then click **Continue**.

---

11. In the next step, assign roles as follows:

    - NameNode Hosts: `master-1` and `master-2`
    - JournalNode Hosts: `master-1`, `worker-2`, and `worker-3`

    Click **OK**.

    Click **Continue**.

---

12. In the **Review Changes** step, note that the JournalNode edits directory values indicate **Inherited value is empty**. Click the value field and specify the directory `/dfs/jn` for all three JournalNode hosts.

Parameter	Group	Value	Description
Service HDFS			
NameNode Data Directories* dfs.namenode.name.dir	master-1	/dfs/nn Inherited from: NameNode Default Group	Determines where on the local file system the NameNode should store the name table (fsimage). For redundancy, enter a comma-delimited list of directories to replicate the name table in all of the directories. Typical values are /data/N/dfs/nn where N=1..3.
	master-2	/dfs/nn Inherited from: NameNode Default Group	
JournalNode Edits Directory* dfs.journalnode.edits.dir	master-1	Inherited value is empty. Click to edit.	Directory on the local file system where NameNode edits are written.
	worker-2	Inherited value is empty. Click to edit.	
	worker-3	Inherited value is empty. Click to edit.	

Click **Continue**.

---

13. In the next step, Cloudera Manager will execute the commands to enable high availability. This might take a few minutes.

    Note: You will probably get an warning message when the wizard attempts to format the name directories. This is expected because data already exists in the current NameNode directory. You can disregard the error message.

    Click **Continue** after the commands are complete.

---

14. On the final step of the wizard, an informational message displays informing you of post-setup steps regarding the Hive Metastore. You will perform these steps below.

    Click **Finish** to complete the wizard.

---

15. Return to the HDFS **Instances** tab and note the changes to the role instances that make up the HDFS service:

    - Two NameNodes, one active and one standby, instead of one NameNode and one SecondaryNameNode
    - A failover controller on each NameNode host
    - Three JournalNodes

Test Automatic NameNode Failover

16. Open the HDFS **Instances** tab and note that one of the NameNodes is active and the other NameNode is standby.

---

17. Select the check box next to the active NameNode (`master-1`).

---

18. Choose **Actions for Selected (1)** > **Restart**, then confirm the action when prompted.

    When the restart operation is complete, click **Close**.

---

19. Verify in the role instances list that the NameNode that was originally active (`master-1`) is now the standby, and the other NameNode (`master-2`) is now active. (It might take a few seconds for the page to display the change.)

---

20. On the **Diagnostics** menu, select **Events**. Note the several recent event entries related to restarting the NameNode.

---

21. Return to the HDFS **Instances** tab, then restart the active NameNode (`master-2`).

    After the restart has completed, verify that the states of the NameNodes have reversed again.

This is the end of the exercise.

Hands-On Exercise: Creating and Using a Snapshot

In this exercise, you will enable HDFS snapshots on a directory and the practice restoring data from a snapshot.

1. Open a terminal:

   List all the snapshottable directories in the HDFS

   ```
   $ hdfs lsSnapshottableDir
   ```

   Since we have yet to enable snapshots in the cluster the command returns nothing.

---

2. Enable snapshots on a directory in HDFS.

   In Cloudera Manager, go to the HDFS page for your cluster and click **File Browser**.

   Browse to /user/training, then click **Enable Snapshots** either by using the button in the lower right, or by choosing it from the drop down menu on the right side.

   In the "Enable Snapshots" window, keep the Snapshottable Path set to `/user/training` and read the note about how child paths can no loner be snapshotted. It is also important to note that you cannot delete a directory that currently contains a snapshot.

   Click **Enable Snapshots**.

   Once the command completes, notice in the message displayed on the "Program:" line that snapshots can also be enabled from the command line using the hdfs dfsadmin tool.

   Click **Close**. Notice that there is now a "Take Snapshot" button.

---

3. List the snapshottable directories again.

   In the terminal window:

   ```
   $ hdfs lsSnapshottableDir
   ```

   Now that we have enabled snapshots on the /user/training directory it shows up in the output of the command.

---

4. Put some data into the `training` user home directory.

   Open a terminal:

Put some data into /user/training.

```
$ cd ~/training_materials/admin/data/
```

```
$ hdfs dfs -put * /user/training
```

Confirm there is now several files and folders inside the HDFS

```
$ hdfs dfs -ls
```

---

5. Take a snapshot.

   Back in the Cloudera Manager File Browser at `/user/training`, Click **Take Snapshot**. Give it the name snap1 and click OK.

   After the snapshot completes click **Close**.

   The snapshot section should now show your "snap1" listing.

---

6. Delete data from /user/training then restore data from the snapshot.

   Now let's see what happens if we delete some data.

   From the terminal:

```
$ hdfs dfs -rm -r weblogs
```

   This command should show that the weblogs directory is now gone.

```
$ hdfs dfs -ls /user/training
```

   However weblogs data is still available, which you can see by running the commands here:

```
$ hdfs dfs -ls /user/training/.snapshot/snap1/weblogs
```

```
$ hdfs dfs -tail .snapshot/snap1/weblogs/2014-03-06.log
```

---

7. Take another snapshot.

Back in the Cloudera Manager File Browser at `/user/training` since “Take Snapshot” no longer exists and has been replaced with a list of current snapshots to take another snapshot go to the drop down menu above next to "/user/training/" and choose **Take Snapshot**. Give it the name `snap2` and click **OK**.

After the snapshot completes click **Close**.

The snapshot section should now show your “snap2” in the listing.

---

8. Let’s make a couple more changes and then take another snapshot.

From the terminal:

```
$ hdfs dfs -mv moviedata movies
```

---

9. Take another snapshot.

Back in the Cloudera Manager File Browser at `/user/training` go to the drop down menu above next to "/user/training/" and choose **Take Snapshot**. Give it the name `snap3` and click **OK**.

After the snapshot completes click **Close**.

The snapshot section should now show your `snap3` in the listing.

---

10. Compare the snapshots.

We can use the `snapshotDiff` command to compare snapshots and see what the changes are

From the terminal:

```
$ hdfs snapshotDiff /user/training snap1 snap2
```

The output of the snapshot diff displays the files and/or directories that have been altered.

Run the command again to compare snap2 to and snap3

The output uses 4 icons to show the changes between the snapshots.

The "+" indicates a new file/directory.

The "-" indicates a deleted file/directory.

The "M" indicates a modified file/directory.

The "R" indicates a renamed file/directory.

Create a Snapshot Policy

We can also create snapshot policies inside Cloudera Manager to automatically manage the creation and deletion of HDFS snapshots

Service HDFS (Cluster 1)

Name: Training Home

Description: Home snapshots

Paths: /user/training

Schedule: Choose **Weekly**

Note: Review the other configuration options to alert on failure, success, and to choose the number of snapshots to keep but leave them at their defaults.

Note the information in the table and the details on the left side, particularly the snapshot prefix, which is the snapshot name.

11. From the navigation click on the **Replication** and choose **Snapshot Policies**

---

12. Click on the **Create Snapshot Policy** button.

---

13. In the Create Snapshot Policy pop-up give the policy the following values

---

14. For the "Take snapshot every week at" value, look at the system time in the upper right and choose a time that is just a few minutes from now, making sure to pick the right day.

---

15. Click **Save Policy**

---

16. After the time configured has been reached we can review the actions of the snapshot policy. Refresh the page in order to check if the policy fired, you should now see a value under the "Last Run" column

---

17. Click the **Actions** button and choose **Show History**.

---

18. Navigate back to the HDFS File Browser and to the `/user/training/` folder. You can see that the automatically generated snapshot is now in the snapshot list in the lower right corner.

---

19. The name is too long to display in the normal window, so click the **Show All** link at the top of the snapshot list. Copy the name of the snapshot.

---

20. Compare the auto generated snapshot to the one you first created.

From the terminal:

```
$ hdfs snapshotDiff /user/training snap1 auto-generated-snapshot
```

Note: The snapshots created by Cloudera Manager's Snapshot Policies are no different than those created manually, whether they are from Policies, the File Browser or the command line they all behave the same. Snapshot Policies just give us a way of maintaining snapshots automatically.

Restore from a snapshot

21. You can restore single files/folders or the entire contents of a snapshot by using the `hdfs dfs -cp` command or you can restore the entire snapshot from the Cloudera Manager File Browser.

Note: Despite Cloudera Manager using the words "Restore From Snapshot" there is no actual restore function in the HDFS API. Snapshots are read-only folders and to restore from them we simply copy the data from them.

Before restoring our snapshot let's review the data.

Navigate to the active NameNode's web UI to take note how the blocks are used by snapshots and what happens during a restore.

---

22. You can use the "Web UI" menu on the HDFS Service page. Make sure you pick the (active) NameNode. Leave the Cloudera Manager tab/window open, we will use it again shortly.

---

23. Go to the File Browser by clicking the **Utilities** tab in the upper right and choosing **Browse the file system**.

---

24. Navigate to the `/user/training folder`

---

25. Click the link for the `training.sql` file and the file information dialog will pop up. Take note of the Block ID value here and write it down or copy and paste it to a notepad, we will refer to this shortly. **Close** the pop-up.

---

26. In the file pathway below Browse Directory set the path to `/user/training/.snapshot/snap1`. Again click the link for the `training.sql` file and note that the Block ID value is the same as it was in the parent folder as HDFS snapshots do not copy blocks but simply refer to the orignal. Leave this tab/window open.

---

27. From the File Browser in Cloudera Manager click the drop-down menu next to `/user/training` and select **Restore Directory From Snapshot**, select `snap1` in the Snapshot field and leave the default Restore Method as **Use HDFS copy command**. The contents of the snapshot will now be copied back into the parent directory returning it to the state it was in when the snapshot was taken. Click **Close** after the process completes.

Note: The `hdfs dfs -cp` command was used to restore the folder which means data was actually copied and new blocks were generated as part of this process. We now have 2 copies of all the data in `snap1`.

---

28. Navigate back to the NameNode web UI and to the `/user/training` folder and once again click the `training.sql` file. Note that the Block ID has changed confirming that new blocks were generated as part of the restore process.

This is the end of the exercise.

Hands-On Exercise: Upgrade the Cluster

In this exercise, you will upgrade your cluster from CDP 7.1.2 to CDP 7.1.3.

All steps in this exercise that use a terminal window should be run on `cmhost` unless otherwise stated.

Back up Cloudera Manager and CDP

1. View the Cloudera interactive Upgrade Guide: https://docs.cloudera.com/cdp-private-cloud/latest/upgrade-cdh/topics/ug_cdh_upgrade_backup.html#ug_cdh_upgrade_backup.

   Select values in the **My Environment** section to see instructions specifically for your environment:

   - **Current Cloudera Manager Version**: Cloudera Manager 7.1.3
   - **Install Method**: Parcels
   - **Operating System**: RedHat/CentOS 7
   - **HDFS High Availability**: Enabled
   - **Using Cloudera Navigator**: No
   - **Current CDP Version**: 7.1.2
   - **New CDP Version**: 7.1.3

---

2. Review the steps on how to prepare your cluster for upgrade, including how to back up the CDP and Cloudera Manager system data.

   Note: This process does not back up data in HDFS. It backs up only metadata and other data necessary to run the cluster. HDFS automatically preserves the previous working directories during an upgrade and provides the ability to rollback.

---

3. In order to save time, we will automate the backup process, please review and then run the script listed below on `cmhost`.

```
$ ~/training_materials/admin/scripts/pre-upgrade-backup.sh
```

---

4. Notice the steps to do the backup process manually include (on each NameNode):

   - Create a backup directory for CDH components
   - Back up CDP databases in MySQL
   - Back up Zookeeper data directory on all Zookeeper hosts
   - Back up JournalNode data on JournalNode hosts
   - Create a rollback directory on NameNode hosts
   - Create a rollback directory for the runtime data for all DataNodes
   - Back up Hue Server registry file

Download CDP version 7.1.3 Parcel

In this section of the exercise, you will download the files needed for the cluster ugrade to CDP 7.1.3.

1. In a terminal:

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/7.1.3/
manifest.json /var/www/html/cloudera-repos/7.1.3/
```

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/7.1.3/
CDH-7.1.3-1.cdh7.1.3.p0.4992530-el7.parcel /var/www/html/
cloudera-repos/7.1.3/
```

```
$ sudo aws s3 cp s3://admin-public/cloudera-parcels/7.1.3/
CDH-7.1.3-1.cdh7.1.3.p0.4992530-el7.parcel.sha /var/www/
html/cloudera-repos/7.1.3/
```

   This places the repository files where they can be reached.

---

2. Test the repo URL.

From **cmhost**, start a web browser and enter this URL:

```
$ http://worker-3:8060/cloudera-repos/7.1.3/
```

Upgrade the CDP Components

3. Enter maintenance mode:

   On the Cloudera Manager home **Status** tab, open the drop-down menu next to the cluster name (Cluster 1) and select **Enter Maintenance Mode**. Confirm when prompted.

   Note the gear icon next to cluster menu, indicating that the cluster is in maintenance mode.

---

4. On the Cluster 1 drop-down menu, select **Upgrade Cluster** to start the Upgrade Cluster wizard.

---

5. Select CDH 7.1.3-1 in the **Upgrade to version** drop-down menu.

   You will see a set of pre-upgrade checks. Some will show a green checkmark. The ones with a yellow mark indicate that a step needs to be taken before continuing the upgrade process.

   a. Scroll down to the inspection checks.

   b. Under the **Download and Distribute Parcel section**, select the **Download and Distribute Parcel** button. You will see that the parcel has already been downloaded. We did this with the preupgrade script ran. Ensure all the checks have been completed. You will need to complete a couple more steps, including indicating that you have performed the database backup.

---

6. Click **Continue** when the distribution process is completed.

---

7. In the next step select **Full Cluster Restart** and click **Continue**.

   Note: This process will take quite a while (15 minutes+) and is completely automated, this is a good time for a break

---

8. In the next step, wait while the wizard stops all services, activates CDP 7.1.3 parcel on all hosts, then restarts the services.

---

9. Return to the CM home page, then click the gear icon next to the cluster name to exit the cluster from maintenance mode.

This is the end of the exercise.

Hands-On Exercise: Breaking the Cluster

In this exercise, you will see what happens when an HDFS DataNode cluster goes down.

IMPORTANT: This exercise depends on Hands-On Exercise: Configuring HDFS for High Availability. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

1. Upload a large example file to HDFS for demonstration purposes. Use the `-l` option to save the file "lazily", which saves only a single version of the file, without replicating blocks.

```
$ hdfs dfs -put -l \
  ~/training_materials/admin/data/bigfile.txt.gz /tmp/
```

---

2. Manually set the replication factor for the file to two. HDFS will copy the file's blocks to two different DataNodes.

```
$ hdfs dfs -setrep 2 /tmp/bigfile.txt.gz
```

---

3. Identify the active NameNode by looking in the Cloudera Manager HDFS **Instances** tab—either `master-1` or `master-2`.

---

4. Choose a block in the new example file to use for testing and verification for this exercise.

   a. Visit the NameNode Web UI running on the active NameNode—either `http://master-1:9870` or `http://master-2:9870`. (Open a new tab so that you can easily switch back and forth between viewing the NameNode UI and Cloudera Manager.)

      Note: If you try to use the standby NameNode, you will get an error: "Operation category READ is not supported in state standby."

   b. Select **Utilities** > **Browse the File System**.

   c. Browse to the `/tmp` folder and click on the `bigfile.txt.gz` file. This will open the **File Information** popup window.

d. The popup window shows the information for `Block 0` by default. Take note of the block ID. You will need the ID later in the exercise.

e. In the **Availability** area, confirm that the block is replicated on two servers.

Choose one of the two servers to use for the rest of the exercise. In a later step, you will shut down that server.

---

5. Intentionally cause a node failure and observe what happens.

From the Cloudera Manager HDFS **Instances** tab, stop the DataNode running on the worker node you selected above.

---

6. Return to the NameNode UI, then click on **Datanodes**. Refresh the browser tab several times and notice that the "Last contact" time for the DataNode you stopped stays the same, while the times for the other DataNodes continue to update with newer timestamps.

---

7. Do an HDFS file system consistency check to see that the NameNode currently thinks there are no problems.

```
$ sudo -u hdfs hdfs fsck /
```

---

8. Wait for at least ten minutes, thirty seconds to pass before starting the next exercise.

To set a timer in any terminal session:

```
$ sleep 630 && echo "10 minutes have passed."
```

This is the end of the exercise.

Hands-On Exercise: Confirm Cluster Healing and Configuring Email Alerts

In this exercise, you will confirm that the cluster has healed from the previsou exercise and you will configure Cloudera Manager to use an email server to send alerts.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Confirm the healing of the cluster

1. In the NameNode UI, go to the **Datanodes** tab and confirm that you now have one dead host.

---

2. In the NameNode UI file browser, browse to the file you uploaded in the last exercises—`bigfile.txt.gz`. Click on the file name and review the information for `Block 0`. Notice that Hadoop has automatically re-replicated the data to another DataNode to maintain 2x replication.

---

3. In Cloudera Manager, go to the HDFS **Charts Library** tab, then click on the **Blocks and Files** chart filter.

   Scroll down and view the charts which show data about what occurred when you stopped a DataNode in the last exercise. For example, view the **Under-replicated Blocks**, **Pending Replication Blocks**, and **Scheduled Replication Blocks** charts.

   Note the spike in activity that occurred after the DataNode went down.

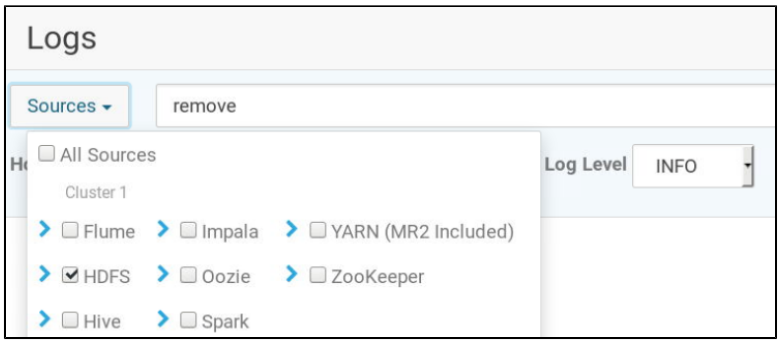
---

4. View the audit and log trails in Cloudera Manager.

   a. In Cloudera Manager, click on **Audits**.

   b. Note the timestamp for when the HDFS service was stopped.

   c. Choose **Diagnostics** > **Logs**, select source **HDFS** only, set the minimum log level to **INFO**, and enter the search term `remove`.



d. Click **Search Logs**.

e. Scroll down to the log entries that occurred over the 10 minutes after the DataNode was stopped. Notice the log messages related blocks being replicated. You can also find these log entries by searching for `blockstatechange`.

---

5. Run the `hdfs fsck` command again to observe that the filesystem is still healthy.

```
$ sudo -u hdfs hdfs fsck /
```

---

6. Run the `hdfs dfsadmin -report` command to see that one dead DataNode is now reported.

```
$ sudo -u hdfs hdfs dfsadmin -report
```

---

7. Return to the Cloudera Manager HDFS **Instances** tab and restart the DataNode you stopped earlier, bringing your cluster back to full function.

---

8. Re-run the `hdfs fsck` command and observe the temporary under-replication of blocks.

The under-replication situation will resolve itself (if it has not already) now that the previously unavailable DataNode is once again running.

If the command above did not show any under-replicated blocks, go to **Diagnostics** > **Logs** in Cloudera Manager and search the **HDFS** source for `replica`. You should find evidence of the temporary under-replication in the log entries corresponding to the time range just after the DataNode was started again.

Enable Email Alerts

Configure Cloudera Manager to send email alerts using the email server on `cmhost`.

9. In Cloudera Manager, choose **Clusters > Cloudera Management Service**.

---

10. Go to the **Configuration** tab, and then choose the **Scope > Alert Publisher** filter.

---

11. Confirm that the **Alerts: Enable Email Alerts** property is *checked*.

---

12. Set the following property values:

- Alerts: Mail Server Username: `training`
- Alerts: Mail Server Password: `training`
- Alerts: Mail Message Recipients: `training@localhost`
- Alerts: Mail Message Format: `text`

---

13. Save the changes.

---

14. Restart Cloudera Management Service.

---

15. Send a test alert from Cloudera Manager.

In Cloudera Manager, go to **Administration** > **Alerts**, **Mail Server**. You should see that the recipient(s) of alerts is now set to `training@localhost`.

Click on the **Send Test Alert** button at the top of the page and confirm by clicking **Close**.

---

16. Confirm emails are being received from Cloudera Manager on `cmhost`.

```
$ mail
```

The "Test Alert" email should show as new (N).

At the & prompt, type in the number that appears to the right of the `N` and hit the `Enter` key to read the email.

After you are done reading the email, enter `q` and to exit the mail client.

Test Alerts

17. On the **Administration, Alerts** page in Cloudera Manager, select the **Health Alerts** tab. Scroll down through the list of enabled and disabled alerts. In the **hdfs** category, note that **JournalNode Default Group** health is enabled.

---

18. Go to the HDFS **Instances** tab and stop the `worker-3` JournalNode instance.

---

19. Go to the HDFS **Status** tab to confirm that the NameNode shows a red status marker. If you do not see it, wait a minute and reload the page. It can take up to three minutes for the status to register.

---

20. View the `training` user's email in the `cmhost` terminal window.

```
$ mail
```

You should see a message with a subject line similar to `[Cloudera Alert] 2 Alerts since 10:29:21 AM`. This means that Cloudera Manager has sent multiple alerts combined into a single email message. View the message and confirm that the mail includes an alert for `NAME_NODE_JOURNAL_NODE_SYNC_STATUS`.

---

21. Return to the HDFS **Instances** tab and restart the role you shut down above.

This is the end of the exercise.

Hands-On Exercise: Troubleshooting a Cluster

In this exercise, you will practice troubleshooting techniques to find and fix cluster issues.

IMPORTANT: This exercise depends on Hands-On Exercise: Working with HDFS. Be sure to complete that exercise before continuing.

```
$ ~/training_materials/admin/scripts/reset_cluster.sh
```

All steps in this exercise that use a terminal window should be run on `cmhost`.

Troubleshooting Challenge 1: HFDS Troubles

1. Run a script that will cause HDFS to enter a critical state. *Do not view the script—that would ruin the challenge!*

   The script may take a few minutes to complete.

   ```
   $ ~/training_materials/admin/scripts/break-hdfs.sh
   ```

---

2. After the script completes, go to the Cloudera Manager home page. You should see a "bad health" indicator for HDFS. (The error indicator make take a few moments to appear. If you do not see it right away, wait a little while and reload the page.)

   Try to find the source of the problem and how to fix it. As you go through the process, see if you can answer these questions.

   - Which role or roles are unhealthy?
   - Which role instance or instances are unhealthy?
   - Which hosts are the unhealthy instances running on?
   - Have any related alerts occurred?
   - Are there error or warning messages in the logs that relate to the problem?
   - Are there any charts that show changes related to the problem?

---

3. *If you need further hints, turn to the next page.*

Here are some steps you could take to find and solve the problem. Note that this is only one approach. There are usually many ways to track down an issue.

1. Go to the HDFS service **Status** tab.

---

2. Under **Health Tests**, note that the **DataNode** role is unhealthy.

   Note: You could also check if any email alerts were sent.

---

3. Under **Status Summary** note that two DataNode instances are healthy and one is unhealthy.

---

4. Select **Hosts > All Hosts** to see which host is unhealthy (`worker-3`).

   Note that the **Disk Usage** column for `worker-3` indicates that disk storage on the host is nearly at capacity.

---

5. Click on the name of the host running the unhealthy role instance to open the host's **Status** tab.

---

6. In the **File Systems** area, note that the disk storage on the `/dev/xvda1` disk is critically full.

---

7. Open a terminal session on `worker-3` by using the `ssh` command on `cmhost`.

```
$ ssh worker-3
```

---

8. Find the mount point for the disk that is critically full using the `df` (disk free) command.

```
training@worker-3: df /dev/xvda1
```

   The output shows that the disk is mounted at the file system root (`/`).

---

9. Use the du (disk usage) command to see what directories and files are using the most space, starting at the root directory. (The `sort -nr` command sorts the output in reverse numerical order, so that the biggest directories are first on the list.)

```
training@worker-3: sudo du / | sort -nr | less
```

Enter `q` to exit the `less` output viewer.

---

10. You should see that the largest subdirectory is `/tmp/ngrams_gz`. That directory is taking up about 23GB of space. You have found the source of the problem.

In the real world, you would need to decide how to handle the excess data. Is it important that the data be on this node? Could the data be compressed to save space? Should you move the data to another partition, or add another disk?

For this exercise, the data is unnecessary, so you can just deleted it.

```
training@worker-3: rm -rf /tmp/ngrams_gz
```

---

11. Rerun the `df` command to confirm that there is now disk space available.

---

12. Exit the `worker-3` terminal session.

```
training@worker-3: exit
```

---

13. Go to Cloudera Manager to confirm that the HDFS service has returned to good health. (It may take a little while for CM to register the change. Wait a minute or two and reload the page.)

Troubleshooting Challenge 2: Heap O' Trouble

A cluster user is trying to run a MapReduce job and is getting an out of memory Java exception, despite the fact that there is plenty of available memory on the cluster.

Recreating the Problem

1. Upload the test files.

```
$ hdfs dfs -mkdir weblog
$ gunzip -c \
  ~/training_materials/admin/data/access_log.gz \
  | hdfs dfs -put - weblog/access_log
```

Note: You may get a message indicating that these files have already been uploaded to HDFS. You can disregard the messages and continue the exercises below.

---

2. Run the "Heap of Trouble" program.

```
$ hadoop jar ~/training_materials/admin/java/EvilJobs.jar \
  HeapOfTrouble \
  weblog/access_log heapOfTrouble
```

Tip: If you try to rerun the job, you will need to delete the `heapOfTrouble` HDFS file or choose a different name. Otherwise you will get an error.

Troubleshooting the Problem

The primary goal of this troubleshooting exercise is to start to become more comfortable analyzing problem scenarios by using Hadoop's log files and Web UIs. **Although you might be able to determine the source of the problem and fix it, doing so successfully is not the primary goal here.**

Take as many actions as you can think of to troubleshoot this problem. Please write down the actions that you take while performing this challenge so that you can share them with other members of the class when you discuss this exercise later.

Fix the problem if you can.

Do not turn to the next page unless you are ready for some hints.

Some Questions to Ask While Troubleshooting a Problem

This list of questions provides some steps that you could follow while troubleshooting a Hadoop problem. Not all the steps necessarily apply to all Hadoop issues, but this list is a good place to start.

- What is there that is different in the environment that was not there before the problem started occurring?
- Is there a pattern to the failure? Is it repeatable?
- If a specific job seems to be the cause of the problem, locate the task logs for the job, including the ApplicationMaster logs, and review them. Does anything stand out?
- Are there any unexpected messages in the NameNode, ResourceManager, and NodeManager logs?
- How is the health of your cluster?
  - Is there adequate disk space?
  - More specifically, does the `/var/log` directory have adequate disk space?
  - Might this be a swapping issue?
  - Is network utilization extraordinarily high?
  - Is CPU utilization extraordinarily high?
- Can you correlate this event with any of the issues?
- Does searching the web for the error provide any useful hints?

Fixing the Problem

If you have time and are able to, fix the problem so that the application runs successfully.

Post-Exercise Discussion

After some time has passed, your instructor will ask you to stop troubleshooting and will lead the class in a discussion of troubleshooting techniques.

This is the end of the exercise.