Cloudera QuickStart

Cloudera, Inc.
1001 Page Mill Road, Bldg 3
Palo Alto, CA 94304
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com

Release Information

Version: 5.5.x
Date: May 27, 2016

Table of Contents

Cloudera QuickStart VM

Cloudera QuickStart virtual machines (VMs) include everything you need to try CDH, Cloudera Manager, Cloudera Impala, and Cloudera Search.

The VM uses a package-based install. This allows you to work with or without Cloudera Manager. Parcels do not work with the VM unless you first migrate your CDH installation to use parcels. On your production systems, Cloudera recommends that you use parcels.

Prerequisites

- These 64-bit VMs require a 64-bit host OS and a virtualization product that can support a 64-bit guest OS.
- To use a VMware VM, you must use a player compatible with WorkStation 8.x or higher:

  – Player 4.x or higher
  – Fusion 4.x or higher

  Older versions of WorkStation can be used to create a new VM using the same virtual disk (VMDK file), but some features in VMware Tools are not available.

- The amount of RAM required varies by the run-time option you choose:

CDH and Cloudera Manager Version	RAM Required by VM
CDH 5 (default)	4+ GiB*
Cloudera Express	8+ GiB*
Cloudera Enterprise (trial)	10+ GiB*

*Minimum recommended memory. If you are running workloads larger than the examples provided, consider allocating additional memory.

Downloading a Cloudera QuickStart VM

Cloudera QuickStart VMs are available as Zip archives in VMware, KVM, and VirtualBox formats. Cloudera recommends that you use 7-Zip to extract these files, when possible. (7-Zip performs well with large files.) To download the latest VM in the required format, see Cloudera QuickStart VM Download.

Installed Products

For information on using the products installed with the VM, see QuickStart VM Software Versions and Documentation on page 5.

QuickStart VM Software Versions and Documentation

VM Version	Documentation
CDH 5 and Cloudera Manager 5	- To learn more about CDH 5 and Cloudera Manager 5, see the Cloudera 5 documentation. - For the latest important information about new features, incompatible changes, and known issues, see the Release Guide. - For information on the versions of the components in the latest release, and links to each project's changes files and release notes, see the packaging section of Version and Download Information. - Cloudera Manager is installed in the VM but is turned off by default. If you would like to use Cloudera Manager, open the **Launch Cloudera Manager** icon on the desktop. It is strongly

VM Version	Documentation
	recommended that before you do so, you configure the VM with a minimum of 8 GiB of RAM and 2 virtual CPU cores (by default it will use 4 GiB of RAM and 1 virtual CPU core). Cloudera Manager and all of the CDH services might not launch properly with less RAM. After launching Cloudera Manager, all of the services in CDH are started, although it might take several minutes for Cloudera Manager to start all of the services. To conserve resources and improve performance, it is recommended that you stop services you do not plan to use. Changes made to configuration files before launching Cloudera Manager are not preserved.
	You can start or reconfigure any installed services using the web interface that is automatically displayed when the VM starts.
	Warning: If Cloudera Manager is running, do not use init scripts from the command line to start, stop, or configure CDH components. Doing so will leave the cluster in an undefined state that could result in data loss. Use only Cloudera Manager to start, stop, or configure CDH components when running Cloudera Manager.

QuickStart VM Administrative Information

In most cases, the QuickStart VM requires no administration beyond managing the installed products and services. In the event that additional administration is required or that problems occur, this page provides information on accounts and possible explanations and solutions to some common problems.

Accounts

Once you launch the VM, you are automatically logged in as the `cloudera` user. The account details are:

- username: `cloudera`
- password: `cloudera`

The `cloudera` account has `sudo` privileges in the VM. The `root` account password is `cloudera`.

The root MySQL password (and the password for other MySQL user accounts) is also `cloudera`.

Hue and Cloudera Manager use the same credentials.

QuickStart VMware Image

To launch the VMware image, you will either need VMware Player for Windows and Linux, or VMware Fusion for Mac. Note that VMware Fusion only works on Intel architectures, so older Macs with PowerPC processors cannot run the QuickStart VM.

QuickStart VirtualBox Image

Some users have reported problems running CentOS 6.4 in VirtualBox. If a kernel panic occurs while the VirtualBox VM is booting, you can try working around this problem by opening the **Settings** > **System** > **Motherboard** tab, and selecting **ICH9** instead of **PIIX3** for the chip set. If you have not already done so, you must also enable **I/O APIC** on the same tab.

QuickStart KVM Image

The KVM image provides a raw disk image that can be used by many hypervisors. Configure machines that use this image with sufficient RAM. See <u>Cloudera QuickStart VM</u> on page 5 for the VM size requirements.

Cloudera Manager and CDH QuickStart Guide

This quick start guide describes how to quickly create a new installation of Cloudera Manager 5, CDH 5, and managed services on a cluster of four hosts. The resulting deployment can be used for demonstrations and proof of concept applications, but is not recommended for production.

QuickStart Cluster Host Requirements

The four hosts in the cluster must satisfy the following requirements:

- The hosts must have at least 10 GB RAM
- You must have root or password-less sudo access to the hosts
- If using root, the hosts must accept the same root password
- The hosts must have Internet access to allow the wizard to install software from `archive.cloudera.com`
- Run a supported OS:

    - **RHEL-compatible**

        - Red Hat Enterprise Linux and CentOS, 64-bit

            - 5.10
            - 5.7
            - 6.4
            - 6.5
            - 6.5 in SE Linux mode
            - 6.6
            - 6.6 in SE Linux mode
            - 6.7
            - 7.1

        - Oracle Enterprise Linux (OEL) with Unbreakable Enterprise Kernel (UEK), 64-bit

            - 5.6 (UEK R2)
            - 6.4 (UEK R2)
            - 6.5 (UEK R2, UEK R3)
            - 6.6 (UEK R3)
            - 6.7 (UEK R3)
            - 7.1

        > **Important:** Cloudera supports RHEL 7 with the following limitations:
        >
        > - Only RHEL 7.1 is supported. RHEL 7.0 is not supported.
        > - Only a new installation of RHEL 7.1 is supported. Upgrades from RHEL 6 to RHEL 7.1 are not supported. For more information, see Does Red Hat support upgrades between major versions of Red Hat Enterprise Linux?

    - **SLES** - SUSE Linux Enterprise Server 11, 64-bit. Service Pack 2 or higher is required. The Updates repository must be active and SUSE Linux Enterprise Software Development Kit 11 SP1 is required.
    - **Debian** - Wheezy (7.0 and 7.1), 64-bit
    - **Ubuntu** - Trusty (14.04) and (Precise) 12.04, 64-bit

If your environment does not satisfy these requirements, the procedure described in this guide may not be appropriate for you. For information about other Cloudera Manager installation options and requirements, see Installing Cloudera Manager, CDH, and Managed Services.

Download and Run the Cloudera Manager Server Installer

1. Download the Cloudera Manager installer binary from Cloudera Manager Downloads to the cluster host where you want to install the Cloudera Manager Server.

   a. Click **Download Cloudera Express** or **Download Cloudera Enterprise**. See Cloudera Express and Cloudera Enterprise Features.
   b. Register and click **Submit**.
   c. Download the installer:

   ```
   wget
   https://archive.cloudera.com/cm5/installer/latest/cloudera-manager-installer.bin
   ```

2. Change `cloudera-manager-installer.bin` to have executable permission.

   ```
   $ chmod u+x cloudera-manager-installer.bin
   ```

3. Run the Cloudera Manager Server installer.

   ```
   $ sudo ./cloudera-manager-installer.bin
   ```

4. Read the Cloudera Manager README and then press **Return** or **Enter** to choose **Next**.
5. Read the Cloudera Express License and then press **Return** or **Enter** to choose **Next**. Use the arrow keys and press **Return** or **Enter** to choose **Yes** to confirm you accept the license.
6. Read the Oracle Binary Code License Agreement and then press **Return** or **Enter** to choose **Next**.
7. When the installation completes, the complete URL provided for the Cloudera Manager Admin Console, including the port number, which is 7180 by default. Press **Return** or **Enter** to choose **OK** to continue.
8. Press **Return** or **Enter** to choose **OK** to exit the installer.
9. On RHEL 5 and CentOS 5, install Python 2.6 or 2.7. Download the appropriate repository rpm packages to the Cloudera Manager Server host and then install Python using `yum`. For example, use the following commands:

   ```
   $ su -c 'rpm -Uvh
   http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm'
   ...
   $ yum install python26
   ```

> ▪ **Note:** If the installation is interrupted for some reason, you may need to clean up before you can re-run it. See Uninstalling Cloudera Manager and Managed Software.

Start the Cloudera Manager Admin Console

1. Wait several minutes for the Cloudera Manager Server to complete its startup. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems.
2. In a web browser, enter `http://Server host:7180`, where *Server host* is the fully qualified domain name or IP address of the host where the Cloudera Manager Server is running. The login screen for Cloudera Manager Admin Console displays.
3. Log into Cloudera Manager Admin Console with the credentials: **Username:** `admin` **Password:** `admin`.

4. After logging in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

Install and Configure Software Using the Cloudera Manager Wizard

Installing and configuring Cloudera Manager, CDH, and managed service software on the cluster hosts involves the following three main steps.

Choose Cloudera Manager Edition and Specify Hosts

1. Choose Cloudera Enterprise Data Hub Edition Trial, which does not require a license, but expires after 60 days and cannot be renewed. The trial allows you to create all CDH and managed services supported by Cloudera Manager. Click **Continue**.
2. Information is displayed indicating what edition of Cloudera Manager will be installed and the services you can choose from. Click **Continue**. The Specify hosts for your CDH cluster installation screen displays.
3. Specify the four hosts on which to install CDH and managed services. You can specify hostnames or IP addresses and ranges, for example: 10.1.1.[1-4] or host[1-3].company.com. You can specify multiple addresses and address ranges by separating them by commas, semicolons, tabs, or blank spaces, or by placing them on separate lines.
4. Click **Search**. Cloudera Manager identifies the hosts on your cluster. Verify that the number of hosts shown matches the number of hosts where you want to install services. Deselect host entries that do not exist and deselect the hosts where you do not want to install services. Click **Continue**. The Select Repository screen displays.

Install CDH and Managed Service Software

1. Keep the default distribution method **Use Parcels** and the default version of CDH 5. Leave the Additional Parcels selections at None.
2. For the Cloudera Manager Agent, keep the default **Matched release for this Cloudera Manager Server**. Click **Continue**. The JDK Installation Options screen displays.
3. Select the **Install Oracle Java SE Development Kit (JDK)** checkbox to allow Cloudera Manager to install the JDK on each cluster host or uncheck if you plan to install it yourself. Leave the **Install Java Unlimited Strength Encryption Policy Files** checkbox deselected. Click **Continue**. The Enable Single User Mode screen displays.
4. Leave the **Single User Mode** checkbox deselected and click **Continue**. The Provide SSH login credentials page displays.
5. Specify host SSH login properties:

   a. Keep the default login **root** or enter the user name for an account that has password-less sudo permission.

   b. If you choose to use password authentication, enter and confirm the password.

6. Click **Continue**. Cloudera Manager installs the Oracle JDK and the Cloudera Manager Agent packages on each host and starts the Agent.
7. Click **Continue**. The Installing Selected Parcels screen displays. Cloudera Manager installs CDH. During the parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.
8. Click **Continue**. The Host Inspector runs to validate the installation, and provides a summary of what it finds, including all the versions of the installed components. Click **Finish**. The Cluster Setup screen displays.

Add and Configure Services

1. Click the **All Services** radio button to create HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, Sqoop, HBase, Impala, Solr, Spark, and Key-Value Store Indexer services. Click **Continue**. The Customize Role Assignments screen displays.
2. Configure the following role assignments:

- Click the text field under the HBase Thrift Server role. In the host selection dialog box that displays, select the checkbox next to any host and click **OK** at the bottom right.
- Click the text field under the Server role of the ZooKeeper service. In the host selection dialog box that displays, uncheck the checkbox next to the host assigned by default (the master host) and select checkboxes next to the remaining three hosts. Click **OK** at the bottom right.

Click **Continue**. The Database Setup screen displays.

3. Leave the default setting of **Use Embedded Database** to have Cloudera Manager create and configure all required databases in an embedded PostgreSQL database. Click **Test Connection**. When the test completes, click **Continue**. The Review Changes screen displays.
4. Review the configuration changes to be applied. Click **Continue**. The Command Progress page displays.
5. The wizard performs 32 steps to configure and starts the services. When the startup completes, click **Continue**.
6. A success message displays indicating that the cluster has been successfully started. Click **Finish** to proceed to the **Home** > **Status** tab.

Test the Installation

The **Home** > **Status** tab looks something like this:
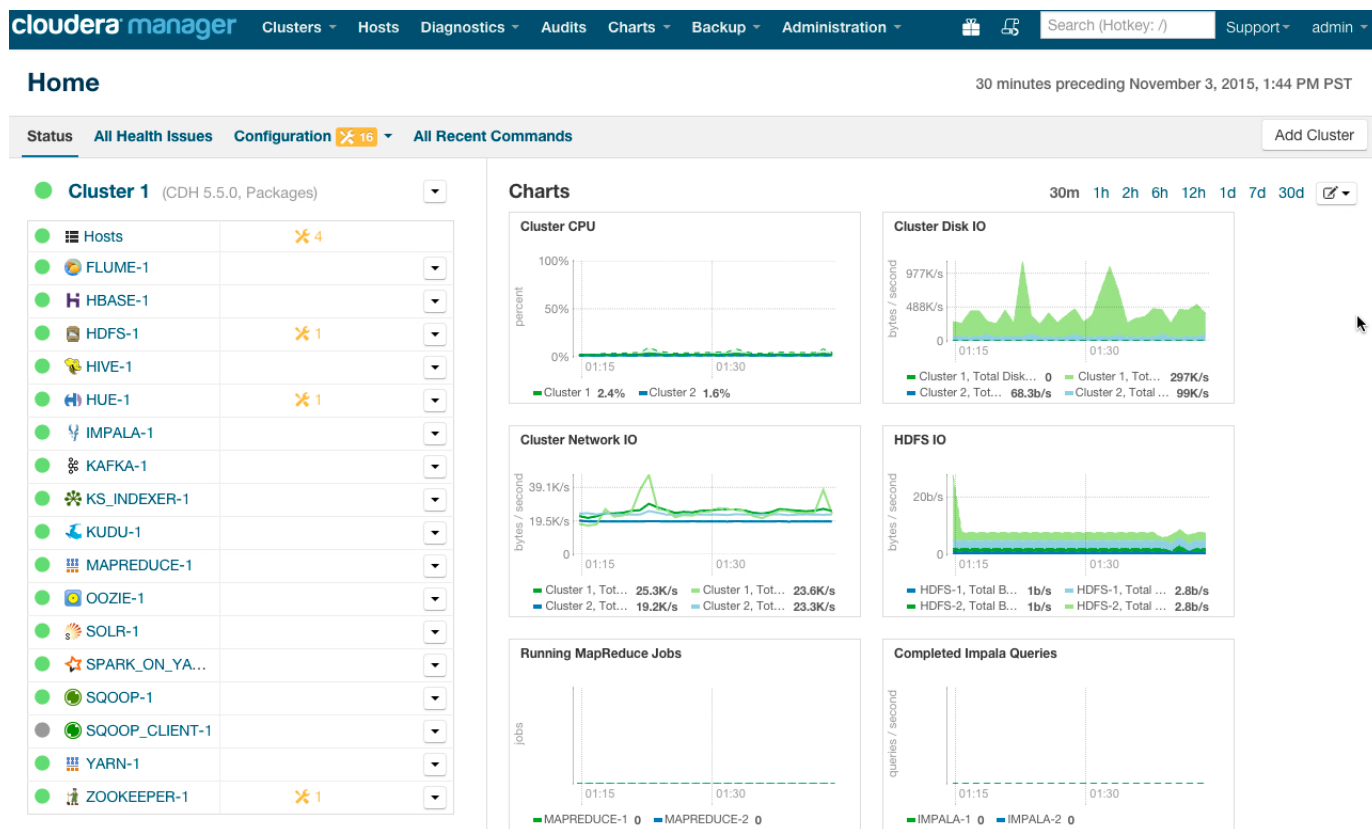
On the left side of the screen is a list of services currently running with their status information. All the services should be running with **Good Health** 🟢, however there may be a small number of configuration warnings indicated by a wrench icon and a number ✖4, which you can ignore.

You can click each service to view more detailed information about the service. You can also test your installation by running a MapReduce job or interacting with the cluster with a Hue application.

Running a MapReduce Job

1. Log into a cluster host.
2. Run the Hadoop PiEstimator example:

```
sudo -u hdfs hadoop jar \
/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar \
pi 10 100
```

3. View the result of running the job by selecting the following from the top navigation bar in the Cloudera Manager Admin Console: **Clusters** > **Cluster 1** > **Activities** > **YARN Applications**. You will see an entry like the following:

05/22/2014 10:45 AM -	Name: **QuasiMonteCarlo**	Pool: **root.hdfs**		
05/22/2014 10:46 AM	Mapper: **QuasiMonteCarlo$QmcMapper**	Reducer: **QuasiMonteCarlo$QmcReducer**		▤ Actions ▾ Details
	Type: **MapReduce** ID: **job_1400700704311_0001** Duration: **54.27s** User: **hdfs** CPU Time: **34.15s**			
	File Bytes Read: **98 B** File Bytes Written: **992.7 KiB** HDFS Bytes Read: **2.7 KiB** HDFS Bytes Written: **215 B**			
	Memory Allocation: **184.7M** Pool: **root.hdfs**			

Testing with Hue

A good way to test the cluster is by running a job. In addition, you can test the cluster by running one of the Hue web applications. Hue is a graphical user interface that allows you to interact with your clusters by running applications that let you browse HDFS, manage a Hive metastore, and run Hive, Impala, and Search queries, Pig scripts, and Oozie workflows.

1. In the Cloudera Manager Admin Console **Home** > **Status** tab, click the Hue service.
2. Click the **Hue Web UI** link, which opens Hue in a new window.
3. Log in with the credentials, **username:** `hdfs`, **password:** `hdfs`.
4. Choose an application in the navigation bar at the top of the browser window.

For more information, see the Hue User Guide.

CDH 5 QuickStart Guide

This guide is for Apache Hadoop developers and system administrators who want to evaluate CDH, the 100% open source platform from Cloudera which contains Apache Hadoop and related projects. It describes how to quickly install Apache Hadoop and CDH components from a Yum, Apt, or zypper/YaST repository on a single Linux node in pseudo-distributed mode.

For more information about installing and configuring CDH 5, and deploying it on a cluster, see Installing and Deploying CDH Using the Command Line.

You can use Cloudera Manager to install and deploy CDH, instead of this guide. Cloudera Manager automates many of the steps and makes the process as a whole much simpler. For more information, see the Cloudera Manager and CDH QuickStart Guide on page 7.

The following sections provide more information and instructions:

Before You Install CDH 5 on a Single Node

> ▪ **Important:**
>
> - If you use Cloudera Manager, do not use these command-line instructions.
> - This information applies specifically to CDH 5.5.x. If you use a lower version of CDH, see the documentation for that version located at Cloudera Documentation.

> ▪ **Note:** When starting, stopping and restarting CDH components, always use the `service (8)` command rather than running `/etc/init.d` scripts directly. This is important because `service` sets the current working directory to / and removes most environment variables (passing only `LANG` and `TERM`) so as to create a predictable environment in which to administer the service. If you run the `/etc/init.d` scripts directly, any environment variables you have set remain in force, and could produce unpredictable results. (If you install CDH from packages, `service` will be installed as part of the Linux Standard Base (LSB).)

Before you install CDH 5 on a single node, there are some important steps you need to do to prepare your system:

1. Verify you are using a supported operating system for CDH 5. See CDH 5 Requirements and Supported Versions.
2. If you haven't already done so, install the JDK before deploying CDH 5.

Installing CDH 5 on a Single Linux Node in Pseudo-distributed Mode

You can evaluate CDH 5 by quickly installing Apache Hadoop and CDH 5 components on a single Linux node in pseudo-distributed mode. In pseudo-distributed mode, Hadoop processing is distributed over all of the cores/processors on a single machine. Hadoop writes all files to the Hadoop Distributed File System (HDFS), and all services and daemons communicate over local TCP sockets for inter-process communication.

> ▪ **Important:**
>
> - If you use Cloudera Manager, do not use these command-line instructions.
> - This information applies specifically to CDH 5.5.x. If you use a lower version of CDH, see the documentation for that version located at Cloudera Documentation.

MapReduce 2.0 (YARN)

MapReduce has undergone a complete overhaul and CDH 5 now includes MapReduce 2.0 (MRv2). The fundamental idea of MRv2's YARN architecture is to split up the two primary responsibilities of the JobTracker — resource management and job scheduling/monitoring — into separate daemons: a global ResourceManager (RM) and per-application ApplicationMasters (AM). With MRv2, the ResourceManager (RM) and per-node NodeManagers (NM), form the data-computation framework. The ResourceManager service effectively replaces the functions of the JobTracker, and NodeManagers run on worker nodes instead of TaskTracker daemons. The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the tasks. For details of the new architecture, see Apache Hadoop NextGen MapReduce (YARN).

See also Migrating from MapReduce 1 (MRv1) to MapReduce 2 (MRv2, YARN).

> **Important:**
>
> For installations in pseudo-distributed mode, there are separate `conf-pseudo` packages for an installation that includes MRv1 (`hadoop-0.20-conf-pseudo`) or an installation that includes YARN (`hadoop-conf-pseudo`). Only one `conf-pseudo` package can be installed at a time: if you want to change from one to the other, you must uninstall the one currently installed.

Installing CDH 5 with MRv1 on a Single Linux Node in Pseudo-distributed mode

> **Important:**
>
> - **Running services:** when starting, stopping and restarting CDH components, always use the `service (8)` command rather than running `/etc/init.d` scripts directly. This is important because `service` sets the current working directory to / and removes most environment variables (passing only `LANG` and `TERM`) so as to create a predictable environment in which to administer the service. If you run the `/etc/init.d` scripts directly, any environment variables you have set remain in force, and could produce unpredictable results. (If you install CDH from packages, `service` will be installed as part of the Linux Standard Base (LSB).)
> - **Java Development Kit:** if you have not already done so, install the Oracle Java Development Kit (JDK) before deploying CDH. Follow these instructions.

> **Important:**
>
> - If you use Cloudera Manager, do not use these command-line instructions.
> - This information applies specifically to CDH 5.5.x. If you use a lower version of CDH, see the documentation for that version located at Cloudera Documentation.

On Red Hat/CentOS/Oracle 5 or Red Hat 6 systems, do the following:

Download the CDH 5 Package

1. Click the entry in the table below that matches your Red Hat or CentOS system, choose **Save File**, and save the file to a directory to which you have write access (it can be your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	RHEL/CentOS/Oracle 5 link
RHEL/CentOS/Oracle 6	RHEL/CentOS/Oracle 6 link
RHEL/CentOS/Oracle 7	RHEL/CentOS/Oracle 7 link

2. Install the RPM.

For Red Hat/CentOS/Oracle 5:

```
$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm
```

For Red Hat/CentOS/Oracle 6 (64-bit):

```
$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm
```

For instructions on how to add a CDH 5 yum repository or build your own CDH 5 yum repository, see Installing CDH 5 On Red Hat-compatible systems.

Install CDH 5

1. (Optionally) add a repository key. Add the Cloudera Public GPG Key to your repository by executing one of the following commands:

   - **For Red Hat/CentOS/Oracle 5 systems:**

   ```
   $ sudo rpm --import
   https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera
   ```

   - **For Red Hat/CentOS/Oracle 6 systems:**

   ```
   $ sudo rpm --import
   https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
   ```

2. Install Hadoop in pseudo-distributed mode:

   To install Hadoop with MRv1:

   ```
   $ sudo yum install hadoop-0.20-conf-pseudo
   ```

On SLES systems, do the following:

Download and install the CDH 5 package

1. Download the CDH 5 "1-click Install" package.

   Download the rpm file, choose **Save File**, and save it to a directory to which you have write access (for example, your home directory).

2. Install the RPM:

   ```
   $ sudo rpm -i cloudera-cdh-5-0.x86_64.rpm
   ```

   For instructions on how to add a CDH 5 SLES repository or build your own CDH 5 SLES repository, see Installing CDH 5 On SLES systems.

Install CDH 5

1. (Optionally) add a repository key. Add the Cloudera Public GPG Key to your repository by executing the following command:

   - **For all SLES systems:**

   ```
   $ sudo rpm --import
   https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
   ```

2. Install Hadoop in pseudo-distributed mode:

To install Hadoop with MRv1:

```
$ sudo zypper install hadoop-0.20-conf-pseudo
```

On Ubuntu and other Debian systems, do the following:

Download and install the package

1. Download the CDH 5 "1-click Install" package:

OS Version	Package Link
Wheezy	Wheezy package
Precise	Precise package
Trusty	Trusty package

2. Install the package by doing one of the following:

   - Choose **Open with** in the download window to use the package manager.
   - Choose **Save File**, save the package to a directory to which you have write access (for example, your home directory), and install it from the command line. For example:

     ```
     sudo dpkg -i cdh5-repository_1.0_all.deb
     ```

For instructions on how to add a CDH 5 Debian repository or build your own CDH 5 Debian repository, see Installing CDH 5 on Ubuntu or Debian systems.

Install CDH 5

1. (Optionally) add a repository key. Add the Cloudera Public GPG Key to your repository by executing the following command:

   - **For Ubuntu Lucid systems:**

     ```
     $ curl -s https://archive.cloudera.com/cdh5/ubuntu/lucid/amd64/cdh/archive.key
       | sudo apt-key add -
     ```

   - **For Ubuntu Precise systems:**

     ```
     $ curl -s https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key
       | sudo apt-key add -
     ```

   - **For Debian Squeeze systems:**

     ```
     $ curl -s https://archive.cloudera.com/cdh5/debian/squeeze/amd64/cdh/archive.key
       | sudo apt-key add -
     ```

2. Install Hadoop in pseudo-distributed mode:

   To install Hadoop with MRv1:

   ```
   $ sudo apt-get update
   $ sudo apt-get install hadoop-0.20-conf-pseudo
   ```

Starting Hadoop and Verifying it is Working Properly:

For MRv1, a pseudo-distributed Hadoop installation consists of one node running all five Hadoop daemons: `namenode`, `jobtracker`, `secondarynamenode`, `datanode`, and `tasktracker`.

To verify the `hadoop-0.20-conf-pseudo` packages on your system.

- To view the files on Red Hat or SLES systems:

```
$ rpm -ql hadoop-0.20-conf-pseudo
```

- To view the files on Ubuntu systems:

```
$ dpkg -L hadoop-0.20-conf-pseudo
```

The new configuration is self-contained in the `/etc/hadoop/conf.pseudo.mr1` directory.

The Cloudera packages use the `alternatives` framework for managing which Hadoop configuration is active. All Hadoop components search for the Hadoop configuration in `/etc/hadoop/conf`.

To start Hadoop, proceed as follows.

Step 1: Format the NameNode.

Before starting the NameNode for the first time you **must** format the file system.

Make sure you perform the format of the NameNode as user `hdfs`. If you are not using Kerberos, you can do this as part of the command string, using `sudo -u hdfs` as in the command above.

```
$ sudo -u hdfs hdfs namenode -format
```

If Kerberos is enabled, do not use commands in the form `sudo -u <user> <command>`; they will fail with a security error. Instead, use the following commands: `$ kinit <user>` (if you are using a password) *or* `$ kinit -kt <keytab> <principal>` (if you are using a `keytab`) and then, for each command executed by this user, `$ <command>`

> **Important:**
>
> In earlier releases, the `hadoop-conf-pseudo` package automatically formatted HDFS on installation. In CDH 5, you must do this explicitly.

Step 2: Start HDFS

```
for x in `cd /etc/init.d ; ls hadoop-hdfs-*` ; do sudo service $x start ; done
```

To verify services have started, you can check the web console. The NameNode provides a web console `http://localhost:50070/` for viewing your Distributed File System (DFS) capacity, number of DataNodes, and logs. In this pseudo-distributed configuration, you should see one live DataNode named `localhost`.

Step 3: Create the directories needed for Hadoop processes.

Issue the following command to create the directories needed for all installed Hadoop processes with the appropriate permissions.

```
$ sudo /usr/lib/hadoop/libexec/init-hdfs.sh
```

Step 4: Verify the HDFS File Structure

```
$ sudo -u hdfs hadoop fs -ls -R /
```

You should see output similar to the following excerpt:

```
...
drwxrwxrwt - hdfs supergroup 0 2012-04-19 15:14 /tmp
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib/hadoop-hdfs
drwxr-xr-x - hdfs supergroup 0 2012-04-19 15:16 /var/lib/hadoop-hdfs/cache
drwxr-xr-x - mapred supergroup 0 2012-04-19 15:19 /var/lib/hadoop-hdfs/cache/mapred
drwxr-xr-x - mapred supergroup 0 2012-04-19 15:29
/var/lib/hadoop-hdfs/cache/mapred/mapred
drwxrwxrwt - mapred supergroup 0 2012-04-19 15:33
/var/lib/hadoop-hdfs/cache/mapred/mapred/staging
...
```

Step 5: Start MapReduce

```
for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-*` ; do sudo service $x start ;
done
```

To verify services have started, you can check the web console. The `JobTracker` provides a web console `http://localhost:50030/` for viewing and running completed and failed jobs with logs.

Step 6: Create User Directories

Create a home directory for each MapReduce user. It is best to do this on the NameNode; for example:

```
$ sudo -u hdfs hadoop fs -mkdir -p /user/<user>
$ sudo -u hdfs hadoop fs -chown <user> /user/<user>
```

where <user> is the Linux username of each user.

Alternatively, you can log in as each Linux user (or write a script to do so) and create the home directory as follows:

```
$ sudo -u hdfs hadoop fs -mkdir -p /user/$USER
$ sudo -u hdfs hadoop fs -chown $USER /user/$USER
```

Running an example application with MRv1

1. Create a home directory on HDFS for the user who will be running the job (for example, `joe`):

   ```
   sudo -u hdfs hadoop fs -mkdir -p /user/joe
   sudo -u hdfs hadoop fs -chown joe /user/joe
   ```

   Do the following steps as the user `joe`.

2. Make a directory in HDFS called `input` and copy some XML files into it by running the following commands:

   ```
   $ hadoop fs -mkdir input
   $ hadoop fs -put /etc/hadoop/conf/*.xml input
   $ hadoop fs -ls input
   Found 3 items:
   -rw-r--r-- 1 joe supergroup 1348 2012-02-13 12:21 input/core-site.xml
   -rw-r--r-- 1 joe supergroup 1913 2012-02-13 12:21 input/hdfs-site.xml
   -rw-r--r-- 1 joe supergroup 1001 2012-02-13 12:21 input/mapred-site.xml
   ```

3. Run an example Hadoop job to grep with a regular expression in your input data.

   ```
   $ /usr/bin/hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar grep input
     output 'dfs[a-z.]+'
   ```

4. After the job completes, you can find the output in the HDFS directory named `output` because you specified that output directory to Hadoop.

```
$ hadoop fs -ls
Found 2 items
drwxr-xr-x - joe supergroup 0 2009-08-18 18:36 /user/joe/input
drwxr-xr-x - joe supergroup 0 2009-08-18 18:38 /user/joe/output
```

You can see that there is a new directory called `output`.

5. List the output files.

```
$ hadoop fs -ls output
Found 2 items
drwxr-xr-x - joe supergroup 0 2009-02-25 10:33 /user/joe/output/_logs
-rw-r--r-- 1 joe supergroup 1068 2009-02-25 10:33 /user/joe/output/part-00000
-rw-r--r- 1 joe supergroup 0 2009-02-25 10:33 /user/joe/output/_SUCCESS
```

6. Read the results in the output file; for example:

```
$ hadoop fs -cat output/part-00000 | head
1 dfs.datanode.data.dir
1 dfs.namenode.checkpoint.dir
1 dfs.namenode.name.dir
1 dfs.replication
1 dfs.safemode.extension
1 dfs.safemode.min.datanodes
```

Installing CDH 5 with YARN on a Single Linux Node in Pseudo-distributed mode

> **Important:**
>
> - If you use Cloudera Manager, do not use these command-line instructions.
> - This information applies specifically to CDH 5.5.x. If you use a lower version of CDH, see the documentation for that version located at Cloudera Documentation.

Before you start, uninstall MRv1 if necessary

If you have already installed MRv1 following the steps in the previous section, you now need to uninstall `hadoop-0.20-conf-pseudo` before running YARN. Proceed as follows.

1. Stop the daemons:

```
$ for x in `cd /etc/init.d ; ls hadoop-hdfs-*` ; do sudo service $x stop ; done
$ for x in 'cd /etc/init.d ; ls hadoop-0.20-mapreduce-*' ; do sudo service $x stop
 ; done
```

2. Remove `hadoop-0.20-conf-pseudo`:

- On Red Hat-compatible systems:

```
$ sudo yum remove hadoop-0.20-conf-pseudo hadoop-0.20-mapreduce-*
```

- On SLES systems:

```
$ sudo zypper remove hadoop-0.20-conf-pseudo hadoop-0.20-mapreduce-*
```

- On Ubuntu or Debian systems:

```
$ sudo apt-get remove hadoop-0.20-conf-pseudo hadoop-0.20-mapreduce-*
```

In this case (after uninstalling `hadoop-0.20-conf-pseudo`) you can skip the package download steps below.

> ▪ **Important:**
>
> If you have not already done so, install the Oracle Java Development Kit (JDK) before deploying CDH 5. Follow these instructions.

On Red Hat/CentOS/Oracle 5 or Red Hat 6 systems, do the following:

Download the CDH 5 Package

1. Click the entry in the table below that matches your Red Hat or CentOS system, choose **Save File**, and save the file to a directory to which you have write access (it can be your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	RHEL/CentOS/Oracle 5 link
RHEL/CentOS/Oracle 6	RHEL/CentOS/Oracle 6 link
RHEL/CentOS/Oracle 7	RHEL/CentOS/Oracle 7 link

2. Install the RPM.

   For Red Hat/CentOS/Oracle 5:

   ```
   $ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm
   ```

   For Red Hat/CentOS/Oracle 6 (64-bit):

   ```
   $ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm
   ```

   For instructions on how to add a CDH 5 yum repository or build your own CDH 5 yum repository, see Installing CDH 5 On Red Hat-compatible systems.

Install CDH 5

1. (Optionally) add a repository key. Add the Cloudera Public GPG Key to your repository by executing the following command:

   - **For Red Hat/CentOS/Oracle 5 systems:**

     ```
     $ sudo rpm --import
     https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera
     ```

   - **For Red Hat/CentOS/Oracle 6 systems:**

     ```
     $ sudo rpm --import
     https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
     ```

2. Install Hadoop in pseudo-distributed mode: **To install Hadoop with YARN:**

   ```
   $ sudo yum install hadoop-conf-pseudo
   ```

On SLES systems, do the following:

Download and install the CDH 5 package

1. Download the CDH 5 "1-click Install" package.

Download the rpm file, choose **Save File**, and save it to a directory to which you have write access (for example, your home directory).

2. Install the RPM:

```
$ sudo rpm -i cloudera-cdh-5-0.x86_64.rpm
```

For instructions on how to add a CDH 5 SLES repository or build your own CDH 5 SLES repository, see Installing CDH 5 On SLES systems.

Install CDH 5

1. (Optionally) add a repository key. Add the Cloudera Public GPG Key to your repository by executing the following command:

- **For all SLES systems:**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
```

2. Install Hadoop in pseudo-distributed mode: **To install Hadoop with YARN:**

```
$ sudo zypper install hadoop-conf-pseudo
```

On Ubuntu and other Debian systems, do the following:

Download and install the package

1. Download the CDH 5 "1-click Install" package:

OS Version	Package Link
Wheezy	Wheezy package
Precise	Precise package
Trusty	Trusty package

2. Install the package by doing one of the following:

- Choose **Open with** in the download window to use the package manager.
- Choose **Save File**, save the package to a directory to which you have write access (for example, your home directory), and install it from the command line. For example:

```
sudo dpkg -i cdh5-repository_1.0_all.deb
```

> **Note:**
>
> For instructions on how to add a CDH 5 Debian repository or build your own CDH 5 Debian repository, see Installing CDH 5 On Ubuntu or Debian systems.

Install CDH 5

1. (Optionally) add a repository key. Add the Cloudera Public GPG Key to your repository by executing the following command:

- **For Ubuntu Lucid systems:**

```
$ curl -s https://archive.cloudera.com/cdh5/ubuntu/lucid/amd64/cdh/archive.key
  | sudo apt-key add -
```

- **For Ubuntu Precise systems:**

```
$ curl -s https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key
  | sudo apt-key add -
```

- **For Debian Squeeze systems:**

```
$ curl -s https://archive.cloudera.com/cdh5/debian/squeeze/amd64/cdh/archive.key
  | sudo apt-key add -
```

2. Install Hadoop in pseudo-distributed mode: **To install Hadoop with YARN:**

```
$ sudo apt-get update
$ sudo apt-get install hadoop-conf-pseudo
```

Starting Hadoop and Verifying it is Working Properly

For YARN, a pseudo-distributed Hadoop installation consists of one node running all five Hadoop daemons: `namenode`, `secondarynamenode`, `resourcemanager`, `datanode`, and `nodemanager`.

- To view the files on Red Hat or SLES systems:

```
$ rpm -ql hadoop-conf-pseudo
```

- To view the files on Ubuntu systems:

```
$ dpkg -L hadoop-conf-pseudo
```

The new configuration is self-contained in the `/etc/hadoop/conf.pseudo` directory.

The Cloudera packages use the `alternative` framework for managing which Hadoop configuration is active. All Hadoop components search for the Hadoop configuration in `/etc/hadoop/conf`.

To start Hadoop, proceed as follows.

Step 1: Format the NameNode.

Before starting the NameNode for the first time you **must** format the file system.

```
$ sudo -u hdfs hdfs namenode -format
```

Make sure you perform the format of the NameNode as user `hdfs`. You can do this as part of the command string, using `sudo -u hdfs` as in the command above.

> - **Important:**
>
> In earlier releases, the `hadoop-conf-pseudo` package automatically formatted HDFS on installation. In CDH 5, you must do this explicitly.

Step 2: Start HDFS

```
$ for x in `cd /etc/init.d ; ls hadoop-hdfs-*` ; do sudo service $x start ; done
```

To verify services have started, you can check the web console. The NameNode provides a web console `http://localhost:50070/` for viewing your Distributed File System (DFS) capacity, number of DataNodes, and logs. In this pseudo-distributed configuration, you should see one live DataNode named `localhost`.

Step 3: Create the directories needed for Hadoop processes.

Issue the following command to create the directories needed for all installed Hadoop processes with the appropriate permissions.

```
$ sudo /usr/lib/hadoop/libexec/init-hdfs.sh
```

Step 4: Verify the HDFS File Structure:

Run the following command:

```
$ sudo -u hdfs hadoop fs -ls -R /
```

You should see output similar to the following excerpt:

```
...
drwxrwxrwt - hdfs supergroup 0 2012-05-31 15:31 /tmp
drwxr-xr-x - hdfs supergroup 0 2012-05-31 15:31 /tmp/hadoop-yarn
drwxrwxrwt - mapred mapred 0 2012-05-31 15:31 /tmp/hadoop-yarn/staging
drwxr-xr-x - mapred mapred 0 2012-05-31 15:31 /tmp/hadoop-yarn/staging/history
drwxrwxrwt - mapred mapred 0 2012-05-31 15:31
/tmp/hadoop-yarn/staging/history/done_intermediate
drwxr-xr-x - hdfs supergroup 0 2012-05-31 15:31 /var
drwxr-xr-x - hdfs supergroup 0 2012-05-31 15:31 /var/log
drwxr-xr-x - yarn mapred 0 2012-05-31 15:31 /var/log/hadoop-yarn
...
```

Step 5: Start YARN

```
$ sudo service hadoop-yarn-resourcemanager start
$ sudo service hadoop-yarn-nodemanager start
$ sudo service hadoop-mapreduce-historyserver start
```

Step 6: Create User Directories

Create a home directory for each MapReduce user. It is best to do this on the NameNode; for example:

```
$ sudo -u hdfs hadoop fs -mkdir /user/<user>
$ sudo -u hdfs hadoop fs -chown <user> /user/<user>
```

where <user> is the Linux username of each user.

Alternatively, you can log in as each Linux user (or write a script to do so) and create the home directory as follows:

```
$ sudo -u hdfs hadoop fs -mkdir /user/$USER
$ sudo -u hdfs hadoop fs -chown $USER /user/$USER
```

Running an example application with YARN

1. Create a home directory on HDFS for the user who will be running the job (for example, `joe`):

   ```
   $ sudo -u hdfs hadoop fs -mkdir /user/joe
   $ sudo -u hdfs hadoop fs -chown joe /user/joe
   ```

   Do the following steps as the user `joe`.

2. Make a directory in HDFS called `input` and copy some XML files into it by running the following commands in pseudo-distributed mode:

```
$ hadoop fs -mkdir input
$ hadoop fs -put /etc/hadoop/conf/*.xml input
$ hadoop fs -ls input
Found 3 items:
-rw-r--r-- 1 joe supergroup 1348 2012-02-13 12:21 input/core-site.xml
-rw-r--r-- 1 joe supergroup 1913 2012-02-13 12:21 input/hdfs-site.xml
-rw-r--r-- 1 joe supergroup 1001 2012-02-13 12:21 input/mapred-site.xml
```

3. Set `HADOOP_MAPRED_HOME` for user `joe`:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
```

4. Run an example Hadoop job to `grep` with a regular expression in your input data.

```
$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep input
output23 'dfs[a-z.]+'
```

5. After the job completes, you can find the output in the HDFS directory named `output23` because you specified that output directory to Hadoop.

```
$ hadoop fs -ls
Found 2 items
drwxr-xr-x - joe supergroup 0 2009-08-18 18:36 /user/joe/input
drwxr-xr-x - joe supergroup 0 2009-08-18 18:38 /user/joe/output23
```

You can see that there is a new directory called `output23`.

6. List the output files.

```
$ hadoop fs -ls output23
Found 2 items
drwxr-xr-x - joe supergroup 0 2009-02-25 10:33 /user/joe/output23/_SUCCESS
-rw-r--r-- 1 joe supergroup 1068 2009-02-25 10:33 /user/joe/output23/part-r-00000
```

7. Read the results in the output file.

```
$ hadoop fs -cat output23/part-r-00000 | head
1 dfs.safemode.min.datanodes
1 dfs.safemode.extension
1 dfs.replication
1 dfs.permissions.enabled
1 dfs.namenode.name.dir
1 dfs.namenode.checkpoint.dir
1 dfs.datanode.data.dir
```

Components That Require Additional Configuration

The following CDH components require additional configuration after installation.

- HBase. For more information, see HBase Installation
- ZooKeeper. For more information, see ZooKeeper Installation
- Snappy. For more information, see Snappy Installation
- Hue. For more information, see Hue Installation
- Oozie. For more information, see Oozie Installation

Next Steps After QuickStart

- Learn more about installing and configuring CDH 5. See Installation.

- Learn how to deploy CDH 5 in fully-distributed mode on a cluster of machines. See Deploying CDH 5 on a Cluster.
- Watch the Cloudera training videos and work through the published exercises to learn how to write your first MapReduce job. See training videos and exercises at Cloudera University.
- Learn how to quickly and easily use Whirr to run CDH 5 clusters on cloud providers' clusters, such as Amazon Elastic Compute Cloud (Amazon EC2). See Whirr Installation.
- Get help from the Cloudera Support team. Cloudera can help you install, configure, optimize, tune, and run Hadoop for large-scale data processing and analysis. Cloudera supports Hadoop whether you run our distribution on servers in your own datacenter, or on hosted infrastructure services such as Amazon EC2, Rackspace, SoftLayer, or VMware's vCloud. For more information, see Cloudera Support.
- Get help from the community. You can also send a message to the CDH user's list.

Cloudera Search QuickStart Guide

This guide shows how to establish and use a sample deployment to query a real data set. At a high level, you set up a cluster, enable search, run a script to create an index and load data, and then run queries.

Prerequisites for Cloudera Search QuickStart Scenarios

Before installing Search, install Cloudera Manager and a CDH cluster. The scenario in this guide works with CDH 5.5.x and Cloudera Manager 5.5.x. The `quickstart.sh` script and supporting files are included with CDH. Install Cloudera Manager, CDH, and Solr using the Cloudera Manager and CDH QuickStart Guide on page 7.

The primary services that the Search Quick Start depends on are:

- HDFS: Stores data. Deploy on all hosts.
- ZooKeeper: Coordinates Solr hosts. Deploy on one host. Use default port 2181. The examples refer to a machine named `search-zk`. You may want to give your Zookeeper machine this name to simplify reusing content exactly as it appears in this document. If you choose a different name, you must adjust some commands accordingly.
- Solr with SolrCloud: Provides search services such as document indexing and querying. Deploy on two hosts.
- Hue: Includes the Search application, which you can use to complete search queries. Deploy Hue on one host.

After you have completed the installation processes outlined in the Cloudera Manager Quick Start Guide, you can Load and Index Data in Search on page 25.

Load and Index Data in Search

Execute the script found in a subdirectory of the following locations. The path for the script often includes the product version, such as Cloudera Manager 5.5.x, so path details vary. To address this issue, use wildcards.

- Packages: `/usr/share/doc`. If Search for CDH 5.5.4 is installed to the default location using packages, the Quick Start script is found in `/usr/share/doc/search-*/quickstart`.
- Parcels: `/opt/cloudera/parcels/CDH/share/doc`. If Search for CDH 5.5.4 is installed to the default location using parcels, the Quick Start script is found in `/opt/cloudera/parcels/CDH/share/doc/search-*/quickstart`.

The script uses several defaults that you might want to modify:

Table 1: Script Parameters and Defaults

Parameter	Default	Notes
NAMENODE_CONNECT	`` `hostname`:8020 ``	For use on an HDFS HA cluster. If you use NAMENODE_CONNECT, do not use NAMENODE_HOST or NAMENODE_PORT.
NAMENODE_HOST	`` `hostname` ``	If you use NAMENODE_HOST and NAMENODE_PORT, do not use NAMENODE_CONNECT.
NAMENODE_PORT	8020	If you use NAMENODE_HOST and NAMENODE_PORT, do not use NAMENODE_CONNECT.
ZOOKEEPER_ENSEMBLE	`` `hostname`:2181/solr ``	Zookeeper ensemble to point to. For example: ```zk1,zk2,zk3:2181/solr```

Parameter	Default	Notes
		If you use `ZOOKEEPER_ENSEMBLE`, do not use `ZOOKEEPER_HOST` or `ZOOKEEPER_PORT`, `ZOOKEEPER_ROOT`.
`ZOOKEEPER_HOST`	`` `hostname` ``	
`ZOOKEEPER_PORT`	`2181`	
`ZOOKEEPER_ROOT`	`/solr`	
`HDFS_USER`	`${HDFS_USER:="${USER}"}`	
`SOLR_HOME`	`/opt/cloudera/parcels/SOLR/lib/solr`	

By default, the script is configured to run on the NameNode host, which is also running ZooKeeper. Override these defaults with custom values when you start `quickstart.sh`. For example, to use an alternate NameNode and HDFS user ID, you could start the script as follows:

```
$ NAMENODE_HOST=nnhost HDFS_USER=jsmith ./quickstart.sh
```

The first time the script runs, it downloads required files such as the Enron data and configuration files. If you run the script again, it uses the Enron information already downloaded, as opposed to downloading this information again. On such subsequent runs, the existing data is used to re-create the `enron-email-collection` SolrCloud collection.

> **Note:** Downloading the data from its server, expanding the data, and uploading the data can be time consuming. Although your connection and CPU speed determine the time these processes require, fifteen minutes is typical and longer is not uncommon.

The script also generates a Solr configuration and creates a collection in SolrCloud. The following sections describes what the script does and how you can complete these steps manually, if desired. The script completes the following tasks:

1. Set variables such as hostnames and directories.
2. Create a directory to which to copy the Enron data and then copy that data to this location. This data is about 422 MB and in some tests took about five minutes to download and two minutes to untar.
3. Create directories for the current user in HDFS, change ownership of that directory to the current user, create a directory for the Enron data, and load the Enron data to that directory. In some tests, it took about a minute to copy approximately 3 GB of untarred data.
4. Use `solrctl` to create a template of the instance directory.
5. Use `solrctl` to create a new Solr collection for the Enron mail collection.
6. Create a directory to which the MapReduceBatchIndexer can write results. Ensure that the directory is empty.
7. Use the MapReduceIndexerTool to index the Enron data and push the result live to `enron-mail-collection`. In some tests, it took about seven minutes to complete this task.

Using Search to Query Loaded Data

After loading data into Search as described in Load and Index Data in Search on page 25, you can use Hue to query data.

Hue must have admin privileges to query loaded data. This is because querying requires Hue import collections or indexes, and these processes can only be completed with admin permissions on the Solr service.

1. Connect to Cloudera Manager and click the **Hue** service, which is often named something like HUE-1. Click **Hue Web UI**.
2. Click the **Search** menu.

3. Select the Enron collection for import.
4. (Optional) Click the Enron collection to configure how the search results display. For more information, see Hue Configuration.
5. Type a search string in the **Search...** text box and press **Enter**.
6. Review the results of your Search.

Cloudera Docker Container

The Cloudera Docker image is a single-host deployment of the Cloudera distribution, including CDH and Cloudera Manager. You can use this environment to learn Hadoop, try new ideas, and test and demonstrate your application.

Docker is different from other virtual machines that isolate or simulate access to the host's hardware, so that entire guest operating systems can run on them. Docker uses a Linux container, which partitions resources of the host operating system; the container has its own view of the filesystem and other resources, but it runs on the same kernel. Docker provides tooling, a packaging format, and infrastructure around Linux containers and related technologies.

Docker is supported in several recent Linux distributions. For example, on Ubuntu 14.04, you can install Docker using the following command:

```
sudo apt-get install docker.io
```

Importing the Cloudera QuickStart Image

You can import the Docker image by pulling it from the Docker Hub:

```
docker pull cloudera/quickstart:latest
```

You can also download the image from the Cloudera website. After the file is downloaded and on your host, you can import it into Docker:

```
tar xzf cloudera-quickstart-vm-*-docker.tar.gz
docker import - cloudera/quickstart:latest < cloudera-quickstart-vm-*-docker/*.tar
```

Running a Cloudera QuickStart Container

To run a container using the image, you must know the name or hash of the image. If you followed the import instructions above, the name is cloudera/quickstart:latest. The hash is also printed in the terminal when you import, or you can look up the hashes of all imported images with:

```
docker images
```

Once you know the name or hash of the image, you can run it:

```
docker run --hostname=quickstart.cloudera --privileged=true -t -i [OPTIONS] [IMAGE]
/usr/bin/docker-quickstart
```

The required flags and other options are described in the following table:

Option	Description
`--hostname=quickstart.cloudera`	Required: Pseudo-distributed configuration assumes this hostname.
`--privileged=true`	Required: For HBase, MySQL-backed Hive metastore, Hue, Oozie, Sentry, and Cloudera Manager.
`-t`	Required: Allocate a pseudoterminal. Once services are started, a Bash shell takes over. This switch starts a terminal emulator to run the services.
`-i`	Required: If you want to use the terminal, either immediately or connect to the terminal later.
`-p 8888`	Recommended: Map the Hue port in the guest to another port on the host.

Option	Description
`-p [PORT]`	Optional: Map any other ports (for example, 7180 for Cloudera Manager, 80 for a guided tutorial).
`-d`	Optional: Run the container in the background.

Use `/usr/bin/docker-quickstart` to start all CDH services, and then run a Bash shell. You can directly run `/bin/bash` instead if you want to start services manually.

See Networking for details about port mapping.

Connecting to the Docker Shell

If you do not pass the `-d` flag to `docker run`, your terminal automatically attaches to the container.

A container dies when you exit the shell, but you can disconnect and leave the container running by typing `Ctrl+p` followed by `Ctrl+q`.

If you disconnect from the shell or passed the `-d` flag on startup, you can connect to the shell later using the following command:

`docker attach [CONTAINER HASH]`

You can look up the hashes of running containers using the following command:

`docker ps`

When attaching to a container, you might need to press **Enter** to see the shell prompt. To disconnect from the terminal without the container exiting, type `Ctrl+p` followed by `Ctrl+q`.

Networking

To make a port accessible outside the container, pass the `-p <port>` flag. Docker maps this port to another port on the host system. You can look up the interface to which it binds and the port number it maps to using the following command:

`docker port [CONTAINER HASH] [GUEST PORT]`

To interact with the Cloudera QuickStart image from other systems, make sure `quickstart.cloudera` resolves to the IP address of the machine where the image is running. You might also want to set up port forwarding so that the port you would normally connect to on a real cluster is mapped to the corresponding port.

When you are mapping ports like this, services are not aware and might provide links or other references to specific ports that are no longer available on your client.

Other Notes

The Cloudera stack is designed to run on a distributed cluster. Pausing and stopping a Docker image is like pausing an entire datacenter—some services might shut down because they seem to be out of touch with the rest of the cluster.

Cloudera Manager is not started by default. To see options for starting Cloudera Manager, run the following command:

`/home/cloudera/cloudera-manager`

See Cloudera documentation and the Cloudera website for other information, including the license agreement associated with the Docker image.