

cloudera[®]

Cloudera Installation

Important Notice

© 2010-2021 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, and any other product or service names or slogans contained in this document are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder. If this documentation includes code, including but not limited to, code examples, Cloudera makes this available to you under the terms of the Apache License, Version 2.0, including any required notices. A copy of the Apache License Version 2.0, including any notices, is included herein. A copy of the Apache License Version 2.0 can also be found here: <https://opensource.org/licenses/Apache-2.0>

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. For information about patents covering Cloudera products, see <http://tiny.cloudera.com/patents>.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc.

**395 Page Mill Road
Palo Alto, CA 94306
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com**

Release Information

Version: Cloudera Enterprise 5.3.x
Date: February 3, 2021

Table of Contents

About Cloudera Installation.....	5
---	----------

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH

5.....	6
Cloudera Manager 5 Requirements and Supported Versions.....	6
<i>Supported Operating Systems.....</i>	6
<i>Supported JDK Versions.....</i>	6
<i>Supported Browsers.....</i>	7
<i>Supported Databases.....</i>	7
<i>Supported CDH and Managed Service Versions.....</i>	7
<i>Resource Requirements.....</i>	8
<i>Networking and Security Requirements.....</i>	8
<i>Single User Mode Requirements.....</i>	11
Permission Requirements for Package-based Installations and Upgrades of CDH.....	14
Cloudera Navigator 2 Requirements and Supported Versions.....	16
<i>Cloudera Manager Requirements.....</i>	16
<i>Supported Databases.....</i>	16
<i>Supported Browsers.....</i>	16
<i>Supported CDH and Managed Service Versions.....</i>	16
CDH 5 Requirements and Supported Versions.....	18
<i>Supported Operating Systems.....</i>	18
<i>Supported Databases.....</i>	19
<i>Supported JDK Versions.....</i>	20
<i>Supported Network Protocols.....</i>	20
Supported Configurations with Virtualization and Cloud Platforms.....	21
<i>Microsoft Azure.....</i>	21
<i>VMware.....</i>	21
Ports.....	21
<i>Ports Used by Cloudera Manager and Cloudera Navigator.....</i>	22
<i>Ports Used by Components of CDH 5.....</i>	25
<i>Ports Used by Impala.....</i>	31
<i>Ports Used by Cloudera Search.....</i>	32
<i>Ports Used by Third-Party Components.....</i>	32

Installing Cloudera Manager and CDH.....	34
---	-----------

Cloudera Manager Deployment.....	34
----------------------------------	----

Unmanaged Deployment.....	35
Java Development Kit Installation.....	35
<i>Installing the Oracle JDK.....</i>	36
Installing Cloudera Manager, CDH, and Managed Services.....	36
<i>Cloudera Manager Installation Software.....</i>	37
<i>Cloudera Manager and Managed Service Data Stores.....</i>	38
<i>Managing Software Installation.....</i>	70
<i>Installation Path A - Automated Installation by Cloudera Manager.....</i>	88
<i>Installation Path B - Manual Installation Using Cloudera Manager Packages.....</i>	95
<i>Installation Path C - Manual Installation Using Cloudera Manager Tarballs.....</i>	111
<i>Installing Impala.....</i>	121
<i>Installing Search.....</i>	122
<i>Installing Spark.....</i>	122
<i>Installing KMS (Navigator Key Trustee).....</i>	123
<i>Installing GPL Extras.....</i>	123
<i>Understanding Custom Installation Solutions.....</i>	125
<i>Deploying Clients.....</i>	146
<i>Testing the Installation.....</i>	146
<i>Uninstalling Cloudera Manager and Managed Software.....</i>	147
<i>Uninstalling a CDH Component From a Single Host.....</i>	151
<i>Installing Cloudera Navigator.....</i>	151
<i>Installing and Deploying CDH Using the Command Line.....</i>	153
<i>Before You Install CDH 5 on a Cluster.....</i>	153
<i>Creating a Local Yum Repository.....</i>	154
<i>Installing the Latest CDH 5 Release.....</i>	155
<i>Installing an Earlier CDH 5 Release.....</i>	167
<i>CDH 5 and MapReduce.....</i>	170
<i>Migrating from MapReduce 1 (MRv1) to MapReduce 2 (MRv2, YARN).....</i>	171
<i>Tuning YARN.....</i>	182
<i>Deploying CDH 5 on a Cluster.....</i>	190
<i>Installing CDH 5 Components.....</i>	215
<i>Building RPMs from CDH Source RPMs.....</i>	401
<i>Apache and Third-Party Licenses.....</i>	401
<i>Uninstalling CDH Components.....</i>	402
<i>Viewing the Apache Hadoop Documentation.....</i>	405

Troubleshooting Installation and Upgrade Problems.....406

Appendix: Apache License, Version 2.0.....413

About Cloudera Installation

This guide provides Cloudera software requirements and installation information for production deployments. This guide also provides specific port information for Cloudera software.

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5

This section describes the requirements for installing Cloudera Manager, Cloudera Navigator, and CDH 5.

Cloudera Manager 5 Requirements and Supported Versions

The following sections describe the requirements and supported operating systems, databases, and browsers, including information about which major and minor release version of each entity is supported for Cloudera Manager. After installing each entity, upgrade to the latest patch version and apply any other appropriate updates. An available update may be specific to the operating system on which it is installed. For example, if you are using CentOS in your environment, you could choose 6 as the major version and 4 as the minor version to indicate that you are using CentOS 6.4. After installing this operating system, apply all relevant CentOS 6.4 upgrades and patches. In some cases, such as some browsers, a minor version may not be listed.

For the latest information on compatibility across all Cloudera products, see the [Product Compatibility Matrix](#).

Supported Operating Systems

Cloudera Manager supports the following operating systems:

- **RHEL-compatible**
 - Red Hat Enterprise Linux and CentOS
 - 5.7, 64-bit
 - 6.4, 64-bit
 - 6.4 in SE Linux mode
 - 6.5, 64-bit
 - 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)
- **SLES** - SUSE Linux Enterprise Server 11, 64-bit. Service Pack 2 or later is required for CDH 5, and Service Pack 1 or later is required for CDH 4. To use the embedded PostgreSQL database that is installed when you follow [Installation Path A - Automated Installation by Cloudera Manager](#) on page 88, the Updates repository must be active. The [SUSE Linux Enterprise Software Development Kit 11 SP1](#) is required on hosts running the Cloudera Manager Agents.
- **Debian** - Wheezy (7.0 and 7.1), Squeeze (6.0) (deprecated), 64-bit
- **Ubuntu** - Trusty (14.04), Precise (12.04), Lucid (10.04) (deprecated), 64-bit



Note:

- Debian Squeeze and Ubuntu Lucid are supported only for CDH 4.
- All CDH and Cloudera Manager hosts that make up a logical cluster need to run on the same major OS release to be covered by Cloudera Support.

Supported JDK Versions

Cloudera Manager supports Oracle JDK 1.7.0_67 and 1.8.0_11 when it's managing CDH 5.x and Oracle JDK 1.6.0_31 and 1.7.0_67 when it's managing CDH 4.x. Cloudera Manager supports Oracle JDK 1.7.0_67 and 1.8.0_11 when it's

managing both CDH 4.x and CDH 5.x clusters. Oracle JDK 1.6.0_31 and 1.7.0_67 can be installed during the installation and upgrade. For further information, see [Java Development Kit Installation](#) on page 35.



Important:

There is one exception to the minimum supported and recommended JDK versions in the following table. If Oracle releases a security patch that affects server-side Java before the next minor release of Cloudera products, the Cloudera support policy covers customers using the patch.

Supported Browsers

The Cloudera Manager Admin Console, which you use to install, configure, manage, and monitor services, supports the following browsers:

- Mozilla Firefox 24 and 31
- Google Chrome
- Internet Explorer 9 and higher. Internet Explorer 11 Native Mode.
- Safari 5 and higher

Supported Databases

Cloudera Manager requires several databases. The Cloudera Manager Server stores information about configured services, role assignments, configuration history, commands, users, and running processes in a database of its own. You must also specify a database for the Activity Monitor and Reports Manager management services.



Important: When processes restart, the configuration for each of the services is redeployed using information that is saved in the Cloudera Manager database. If this information is not available, your cluster will not start or function correctly. You must therefore schedule and maintain regular backups of the Cloudera Manager database in order to recover the cluster in the event of the loss of this database.

See [Backing Up Databases](#) on page 56.

The database you use must be configured to support UTF8 character set encoding. The embedded PostgreSQL database that is installed when you follow [Installation Path A - Automated Installation by Cloudera Manager](#) on page 88 automatically provides UTF8 encoding. If you install a custom database, you may need to enable UTF8 encoding. The commands for enabling UTF8 encoding are described in each database topic under [Cloudera Manager and Managed Service Data Stores](#) on page 38.

After installing a database, upgrade to the latest patch version and apply any other appropriate updates. Available updates may be specific to the operating system on which it is installed.

Cloudera Manager and its supporting services can use the following databases:

- MySQL - 5.1, 5.5 and 5.6
- Oracle 11gR2
- PostgreSQL - 8.1, 8.3, 8.4, 9.1, 9.2, 9.3

Cloudera supports the shipped version of MySQL and PostgreSQL for each supported Linux distribution. Each database is supported for all components in Cloudera Manager and CDH subject to the notes in [CDH 4 Supported Databases](#) and [CDH 5 Supported Databases](#).

Supported CDH and Managed Service Versions

The following versions of CDH and managed services are supported:



Warning: Cloudera Manager 5 does not support CDH 3 and you cannot upgrade Cloudera Manager 4 to Cloudera Manager 5 if you have a cluster running CDH 3. Therefore, to upgrade CDH 3 clusters to CDH 4 using Cloudera Manager you must use Cloudera Manager 4.

- **CDH 4 and CDH 5.** The latest released versions of CDH 4 and CDH 5 are strongly recommended. For information on CDH 4 requirements, see [CDH 4 Requirements and Supported Versions](#). For information on CDH 5 requirements, see [CDH 5 Requirements and Supported Versions](#) on page 18.
- **Cloudera Impala** - Cloudera Impala is included with CDH 5. Cloudera Impala 1.2.1 with CDH 4.1.0 or higher. For more information on Impala requirements with CDH 4, see [Impala Requirements](#).
- **Cloudera Search** - Cloudera Search is included with CDH 5. Cloudera Search 1.2.0 with CDH 4.6.0. For more information on Cloudera Search requirements with CDH 4, see [Cloudera Search Requirements](#).
- **Apache Spark** - 0.90 or higher with CDH 4.4.0 or higher.
- **Apache Accumulo** - 1.4.3 with CDH 4.3.0, 1.4.4 with CDH 4.5.0, and 1.6.0 with CDH 4.6.0.

For more information, see the [Product Compatibility Matrix](#).

Resource Requirements

Cloudera Manager requires the following resources:

- **Disk Space**
 - **Cloudera Manager Server**
 - 5 GB on the partition hosting `/var`.
 - 500 MB on the partition hosting `/usr`.
 - For parcels, the space required depends on the number of parcels you download to the Cloudera Manager Server and distribute to Agent hosts. You can download multiple parcels of the same product, of different versions and builds. If you are managing multiple clusters, only one parcel of a product/version/build/distribution is downloaded on the Cloudera Manager Server—not one per cluster. In the local parcel repository on the Cloudera Manager Server, the approximate sizes of the various parcels are as follows:
 - CDH 4.6 - 700 MB per parcel; CDH 5 (which includes Impala and Search) - 1.5 GB per parcel (packed), 2 GB per parcel (unpacked)
 - Cloudera Impala - 200 MB per parcel
 - Cloudera Search - 400 MB per parcel
 - **Cloudera Management Service** - The Host Monitor and Service Monitor databases are stored on the partition hosting `/var`. Ensure that you have at least 20 GB available on this partition. For more information, see [Data Storage for Monitoring Data](#) on page 58.
 - **Agents** - On Agent hosts each unpacked parcel requires about three times the space of the downloaded parcel on the Cloudera Manager Server. By default unpacked parcels are located in `/opt/cloudera/parcels`.
- **RAM** - 4 GB is recommended for most cases and is required when using Oracle databases. 2 GB may be sufficient for non-Oracle deployments with fewer than 100 hosts. However, to run the Cloudera Manager Server on a machine with 2 GB of RAM, you must tune down its maximum heap size (by modifying `-Xmx` in `/etc/default/cloudera-scm-server`). Otherwise the kernel may kill the Server for consuming too much RAM.
- **Python** - Cloudera Manager and CDH 4 require Python 2.4 or higher, but Hue in CDH 5 and package installs of CDH 5 require Python 2.6 or 2.7. All supported operating systems include Python version 2.4 or higher.
- **Perl** - Cloudera Manager requires [perl](#).

Networking and Security Requirements

The hosts in a Cloudera Manager deployment must satisfy the following networking and security requirements:

- Cluster hosts must have a working network name resolution system and correctly formatted `/etc/hosts` file. All cluster hosts must have properly configured forward and reverse host resolution through DNS. The `/etc/hosts` files must
 - Contain consistent information about hostnames and IP addresses across all hosts
 - Not contain uppercase hostnames
 - Not contain duplicate IP addresses

Also, do not use aliases, either in `/etc/hosts` or in configuring DNS. A properly formatted `/etc/hosts` file should be similar to the following example:

```
127.0.0.1      localhost.localdomain  localhost
192.168.1.1   cluster-01.example.com cluster-01
192.168.1.2   cluster-02.example.com cluster-02
192.168.1.3   cluster-03.example.com cluster-03
```

- In most cases, the Cloudera Manager Server must have SSH access to the cluster hosts when you run the installation or upgrade wizard. You must log in using a root account or an account that has password-less sudo permission. For authentication during the installation and upgrade procedures, you must either enter the password or upload a public and private key pair for the root or sudo user account. If you want to use a public and private key pair, the public key must be installed on the cluster hosts before you use Cloudera Manager.


Cloudera Manager uses SSH only during the initial install or upgrade. Once the cluster is set up, you can disable root SSH access or change the root password. Cloudera Manager does not save SSH credentials, and all credential information is discarded when the installation is complete. For more information, see [Permission Requirements for Package-based Installations and Upgrades of CDH](#) on page 14.

- If [single user mode](#) is not enabled, the Cloudera Manager Agent runs as root so that it can make sure the required directories are created and that processes and files are owned by the appropriate user (for example, the `hdfs` and `mapred` users).
- No blocking is done by Security-Enhanced Linux (SELinux).
- IPv6 must be disabled.
- Multihoming CDH or Cloudera Manager is not supported outside specifically certified Cloudera partner appliances. Cloudera finds that current Hadoop architectures combined with modern network infrastructures and security practices remove the need for multihoming. Multihoming, however, is beneficial internally in appliance form factors to take advantage of high-bandwidth InfiniBand interconnects.

Although some subareas of the product may work with unsupported custom multihoming configurations, there are known issues with multihoming. In addition, unknown issues may arise because multihoming is not covered by our test matrix outside the Cloudera-certified partner appliances.

- No blocking by iptables or firewalls; port 7180 must be open because it is used to access Cloudera Manager after installation. Cloudera Manager communicates using specific [ports](#), which must be open.
- For RedHat and CentOS, the `/etc/sysconfig/network` file on each host must contain the hostname you have just set (or verified) for that host.
- Cloudera Manager and CDH use several user accounts and groups to complete their tasks. The set of user accounts and groups varies according to the components you choose to install. Do not delete these accounts or groups and do not modify their permissions and rights. Ensure that no existing systems prevent these accounts and groups from functioning. For example, if you have scripts that delete user accounts not in a whitelist, add these accounts to the list of permitted accounts. Cloudera Manager, CDH, and managed services create and use the following accounts and groups:

Table 1: Users and Groups

Component (Version)	Unix User ID	Groups	Notes
Cloudera Manager (all versions)	cloudera-scm	cloudera-scm	<p>Cloudera Manager processes such as the Cloudera Manager Server and the monitoring roles run as this user.</p> <p>The Cloudera Manager keytab file must be named <code>cmf.keytab</code> since that name is hard-coded in Cloudera Manager.</p> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> Note: Applicable to clusters managed by Cloudera Manager only.</p> </div>
Apache Accumulo (Accumulo 1.4.3 and higher)	accumulo	accumulo	Accumulo processes run as this user.
Apache Avro			No special users.
Apache Flume (CDH 4, CDH 5)	flume	flume	The sink that writes to HDFS as this user must have write privileges.
Apache HBase (CDH 4, CDH 5)	hbase	hbase	The Master and the RegionServer processes run as this user.
HDFS (CDH 4, CDH 5)	hdfs	hdfs, hadoop	The NameNode and DataNodes run as this user, and the HDFS root directory as well as the directories used for edit logs should be owned by it.
Apache Hive (CDH 4, CDH 5)	hive	hive	<p>The HiveServer2 process and the Hive Metastore processes run as this user.</p> <p>A user must be defined for Hive access to its Metastore DB (for example, MySQL or Postgres) but it can be any identifier and does not correspond to a Unix uid. This is <code>javax.jdo.option.ConnectionUserName</code> in <code>hive-site.xml</code>.</p>
Apache HCatalog (CDH 4.2 and higher, CDH 5)	hive	hive	The WebHCat service (for REST access to Hive functionality) runs as the <code>hive</code> user.
HttpFS (CDH 4, CDH 5)	httpfs	httpfs	The HttpFS service runs as this user. See HttpFS Security Configuration for instructions on how to generate the merged <code>httpfs-http.keytab</code> file.
Hue (CDH 4, CDH 5)	hue	hue	Hue services run as this user.
Cloudera Impala (CDH 4.1 and higher, CDH 5)	impala	impala, hadoop, hdfs, hive	Impala services run as this user.
Apache Kafka (Cloudera)	kafka	kafka	Kafka services run as this user.

Component (Version)	Unix User ID	Groups	Notes
Distribution of Kafka 1.2.0)			
KMS (File) (CDH 5.2.1 and higher)	kms	kms	The KMS (File) service runs as this user.
KMS (Navigator Key Trustee) (CDH 5.3 and higher)	kms	kms	The KMS (Navigator Key Trustee) service runs as this user.
Llama (CDH 5)	llama	llama	Llama runs as this user.
Apache Mahout			No special users.
MapReduce (CDH 4, CDH 5)	mapred	mapred, hadoop	Without Kerberos, the JobTracker and tasks run as this user. The LinuxTaskController binary is owned by this user for Kerberos.
Apache Oozie (CDH 4, CDH 5)	oozie	oozie	The Oozie service runs as this user.
Parquet			No special users.
Apache Pig			No special users.
Cloudera Search (CDH 4.3 and higher, CDH 5)	solr	solr	The Solr processes run as this user.
Apache Spark (CDH 5)	spark	spark	The Spark History Server process runs as this user.
Apache Sentry (incubating) (CDH 5.1 and higher)	sentry	sentry	The Sentry service runs as this user.
Apache Sqoop (CDH 4, CDH 5)	sqoop	sqoop	This user is only for the Sqoop1 Metastore, a configuration option that is not recommended.
Apache Sqoop2 (CDH 4.2 and higher, CDH 5)	sqoop2	sqoop, sqoop2	The Sqoop2 service runs as this user.
Apache Whirr			No special users.
YARN (CDH 4, CDH 5)	yarn	yarn, hadoop	Without Kerberos, all YARN services and applications run as this user. The LinuxContainerExecutor binary is owned by this user for Kerberos.
Apache ZooKeeper (CDH 4, CDH 5)	zookeeper	zookeeper	The ZooKeeper processes run as this user. It is not configurable.

Single User Mode Requirements

In a conventional Cloudera Manager deployment, the Cloudera Manager Agent, which manages Hadoop processes on each host, runs as the root user. However, some environments restrict access to the root account.

Cloudera Manager 5.3 provides **single user mode**, which satisfies the requirements of such environments. In single user mode, the Cloudera Manager Agent and *all the processes run by services managed by Cloudera Manager* are started as a single configured user and group. Single user mode prioritizes isolation between Hadoop and the rest of the system over isolation between Hadoop processes running on the system.

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5

Within a Cloudera Manager deployment, single user mode is global and applies to all clusters managed by that instance of Cloudera Manager.

By default, the single user is `cloudera-scm` and the configuration steps described in the following sections assume that user. However, other users are supported. If you choose another user, replace `cloudera-scm` in the following steps with the selected user, and perform the additional steps in [Using a Non-default Single User](#) on page 12.

The following sections describe limitations of single user mode and the required configuration steps for the supported installation scenarios at specific points during the installation process.

Limitations

- Switching between conventional and single user mode is not supported.
- Single user mode is supported for clusters running CDH 5.2 and higher.
- NFS Gateway is not supported in single user mode.

Using a Non-default Single User

When configuring single user mode for a user other than the default (`cloudera-scm`), perform the following configuration steps:

- Make the following directories writable by the single user:
 - `/var/log/cloudera-scm-agent/`
 - `/var/lib/cloudera-scm-agent/`
- Cloudera Manager stores parcels under `/opt/cloudera`, which by default is owned by `cloudera-scm`. Do one of the following:
 - Change `/opt/cloudera` to be writable by the single user.
 - Change the parcel directory location to be writable by the single user:
 1. Go to **Administration > Settings > Parcels**.
 2. Set the **Local Parcel Repository Path** property.
 3. Click **Save Changes**.
- For a single user `username`, create the process limits configuration file at `/etc/security/limits.d/username.conf` with the following settings:

```
username soft nofile 32768
username soft nproc 65536
username hard nofile 1048576
username hard nproc unlimited
username hard memlock unlimited
username soft memlock unlimited
```

Configuration Steps Before Starting Cloudera Manager Agents in Installation Paths B and C

- If you manually install Agent packages, before starting the Agents, configure them to run as `cloudera-scm` by editing the file `/etc/default/cloudera-scm-agent` and uncommenting the line:

```
USER="cloudera-scm"
```

- Configure the parcels directory. Do one of the following:
 - On each host, in the Agent configuration file `/etc/cloudera-scm-agent/config.ini`, set the `parcel_dir` property:

```
# Parcel directory. Unpacked parcels will be stored in this directory.
# Downloaded parcels will be stored in <parcel_dir>/../parcel-cache
# parcel_dir=/opt/cloudera/parcels
```

- 1. Click **Hosts** in the top navigation bar.
 2. Click the **Configuration** tab.
 3. Configure the value of the **Parcel Directory** property. The setting of the `parcel_dir` property in the [Cloudera Manager Agent configuration file](#) overrides this setting.
 4. Click **Save Changes** to commit the changes.
 5. On each host, restart the Cloudera Manager Agent:

```
$ sudo service cloudera-scm-agent restart
```

Configuration Steps Before Running the Installation Wizard

Before configuring a cluster to run in single user mode, the following steps must be performed on *all hosts in the cluster*:

- Give the single user passwordless sudo access. You must create the user if it doesn't exist. One common way of achieving this is to add the user to the configured sudoers group by running the command:

```
usermod -a -G sudo cloudera-scm
```

or adding a new sudo configuration for the `cloudera-scm` group by running the command `visudo` and then adding the following line:

```
%cloudera-scm ALL=(ALL) NOPASSWD: ALL
```

- Sudo must be configured so that `/usr/sbin` is in the path when running `sudo`. One way to achieve this is by adding the following configuration to `sudoers`:

1. Edit the `/etc/sudoers` file using the `visudo` command
2. Add this line to the configuration file:

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

- Set up per user limits for `su` prior to setting up the Agent.

1. Edit `/etc/pam.d/su`.
2. Uncomment:

```
session required pam_limits.so
```

- Roles that run on Tomcat require some directories to exist in non-configurable paths. The following directories must be created and be writable by `cloudera-scm`:
 - **HDFS** (HttpFS role) - `/var/lib/hadoop-https`
 - **Oozie Server** - `/var/lib/oozie`
 - **Sqoop 2 Server** - `/var/lib/sqoop2`
 - **Solr Server** - `/var/lib/solr`
- Cloudera recommends that you create a prefix directory (for example, `/cm`) owned by `cloudera-scm` under which all other service directories will be placed. In single user mode, the Cloudera Manager Agent creates directories under the prefix directory with the correct ownership. If hosts have additional volumes on them that will be used for data directories Cloudera recommends creating a directory on each volume (for example, `/data0/cm` and `/data1/cm`) that is writable by `cloudera-scm`.

Configuration Steps Before Starting the Installation Wizard in Installation Paths B and C

Perform the following steps for the indicated scenarios:

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5

- **Path C** - Do one of the following:
 - Create and change the ownership of `/var/lib/cloudera-scm-server` to the single user.
 - Set the Cloudera Manager Server local storage directory to one owned by the single user:
 1. Go to **Administration > Settings > Advanced**.
 2. Set the **Cloudera Manager Server Local Data Storage Directory** property to a directory owned by the single user.
 3. Click **Save Changes** to commit the changes.
- **Path B and C when using already managed hosts** - Configure single user mode:
 1. Go to **Administration > Settings > Advanced**.
 2. Check the **Single User Mode** checkbox.
 3. Click **Save Changes** to commit the changes.

Configuration Steps While Running the Installation Wizard

When configuring the first cluster in Cloudera Manager using the Installation wizard you'll have the option to set up the cluster in single user mode. This configures the Agents to run as `cloudera-scm`.

During the review configuration step you confirm that all the configured paths are writable by `cloudera-scm`. The directories themselves don't have to exist as long as the parent directory is writable by `cloudera-scm`.

Following the standard review configuration page, an additional paths configuration page shows all the configurable paths for the services that will be created in the cluster. These must also be modified to be locations writable by `cloudera-scm`. In most cases, the paths that need to be modified from their default locations fall under two categories:

- Paths under `/var` - These are `log`, `run`, and `data` directories for the different services.
- Per volume data directories - These are data directory configurations that list a directory per volume. Such configurations are used by HDFS, MapReduce, YARN and Impala.

Configuration for Secure Clusters

You must perform some additional configuration when setting up secure HDFS in single user mode:

- When configuring Kerberos, also refer to [Enabling Kerberos Authentication for Single User Mode or Non-Default Users](#).
- Configure HDFS with [SSL encryption](#).
- Do not configure the DataNode Transceiver port and HTTP Web UI port to use privileged ports.
- Configure DataNode data transfer protection.

Permission Requirements for Package-based Installations and Upgrades of CDH

The following sections describe the permission requirements for package-based installation and upgrades of CDH with and without Cloudera Manager. The permission requirements are not controlled by Cloudera but result from standard UNIX system requirements for the installation and management of packages and running services.

Permission Requirements for Package-Based CDH Installation with Cloudera Manager



Important: Unless otherwise noted, when root or sudo access is required, using another system (such as PowerBroker) that provides root/sudo privileges is acceptable.

Table 2: Permission Requirements with Cloudera Manager

Task	Permissions Required
Install Cloudera Manager (using <code>cloudera-manager-installer.bin</code>)	root or sudo access on a single host
Manually start/stop/restart the Cloudera Manager Server (that is, log onto the host running Cloudera Manager and execute: <code>service cloudera-scm-server action</code>)	root or sudo
Run Cloudera Manager Server.	<code>cloudera-scm</code>
Install CDH components through Cloudera Manager.	<p>One of the following, configured during initial installation of Cloudera Manager:</p> <ul style="list-style-type: none"> • Direct access to root user using the root password. • Direct access to root user using a SSH key file. • Passwordless sudo access for a specific user. This is the same requirement as the installation of CDH components on individual hosts, which is a requirement of the UNIX system in general. <p>You <i>cannot</i> use another system (such as PowerBroker) that provides root/sudo privileges.</p>
Install the Cloudera Manager Agent through Cloudera Manager.	<p>One of the following, configured during initial installation of Cloudera Manager:</p> <ul style="list-style-type: none"> • Direct access to root user using the root password. • Direct access to root user using a SSH key file. • Passwordless sudo access for a specific user. This is the same requirement as the installation of CDH components on individual hosts, which is a requirement of the UNIX system in general. <p>You <i>cannot</i> use another system (such as PowerBroker) that provides root/sudo privileges.</p>
Run the Cloudera Manager Agent.	<p>If single user mode is not enabled, access to the root account during runtime, through one of the following scenarios:</p> <ul style="list-style-type: none"> • During Cloudera Manager and CDH installation, the Agent is automatically started if installation is successful. It is then started using one of the following, as configured during the initial installation of Cloudera Manager: <ul style="list-style-type: none"> – Direct access to root user using the root password – Direct access to root user using a SSH key file – Passwordless sudo access for a specific user <p>Using another system (such as PowerBroker) that provides root/sudo privileges is <i>not</i> acceptable.</p> <ul style="list-style-type: none"> • Through automatic startup during system boot, using <code>init</code>.
Manually start/stop/restart the Cloudera Manager Agent process.	<p>If single user mode is not enabled, root or sudo access.</p> <p>This permission requirement ensures that services managed by the Cloudera Manager Agent assume the appropriate user (that is, the HDFS service assumes the <code>hdfs</code> user) for correct privileges. Any action request for a CDH service managed within Cloudera Manager <i>does not</i> require root or sudo access, because the action is handled by the Cloudera Manager Agent, which is already running under the root user.</p>

Permission Requirements for Package-Based CDH Installation without Cloudera Manager

Table 3: Permission Requirements without Cloudera Manager

Task	Permissions Required
Install CDH products.	root or sudo access for the installation of any RPM-based package during the time of installation and service startup/shut down. Passwordless SSH under the root user is not required for the installation (SSH root keys).
Upgrade a previously installed CDH package.	root or sudo access. Passwordless SSH under the root user is not required for the upgrade process (SSH root keys).
Manually install or upgrade hosts in a CDH ready cluster.	Passwordless SSH as root (SSH root keys), so that scripts can be used to help manage the CDH package and configuration across the cluster.
Change the CDH package (for example: RPM upgrades, configuration changes the require CDH service restarts, addition of CDH services).	root or sudo access to restart any host impacted by this change, which could cause a restart of a given service on each host in the cluster.
Start/stop/restart a CDH service.	root or sudo according to UNIX standards.

Cloudera Navigator 2 Requirements and Supported Versions

The following sections describe various requirements and supported versions of Cloudera Manager, databases, browsers, and CDH and managed service versions for Cloudera Navigator 2.

For more information on compatibility with other components, see the Cloudera [Product Compatibility Matrix](#).

Cloudera Manager Requirements

Cloudera Navigator 2.2 is available with Cloudera Manager 5.3. For information on the requirements for installing Cloudera Manager, see [Cloudera Manager 5 Requirements and Supported Versions](#) on page 6.

Supported Databases

Cloudera Navigator, which stores audit reports, and entity metadata, policies, and user authorization and audit report metadata, supports the following databases:

- MySQL - 5.1, 5.5 and 5.6
- Oracle 11gR2
- PostgreSQL - 8.1, 8.3, 8.4, 9.1, 9.2, 9.3

Supported Browsers

The Cloudera Navigator UI, which you use to create and view audit reports, search and update metadata, and configure Cloudera Navigator user groups, supports the following browsers:

- Mozilla Firefox 24 and higher
- Google Chrome 36 and higher
- Internet Explorer 11
- Safari 5 and higher

Supported CDH and Managed Service Versions

This section describes the CDH and managed service versions supported by the Cloudera Navigator auditing and metadata components.

Cloudera Navigator Auditing Component

This section describes the service versions and audited operations supported by the Cloudera Navigator auditing component.

- **HDFS** - Minimum supported version: CDH 4.0.0.

The captured operations are:

- Operations that access or modify a file's or directory's data or metadata
- Operations denied due to lack of privileges

- **HBase** - Minimum supported version: CDH 4.0.0.

**Note:**

- In CDH versions less than 4.2.0, for grant and revoke operations, the operation in log events is `ADMIN`
- In simple authentication mode, if the HBase Secure RPC Engine property is `false` (the default), the username in log events is `UNKNOWN`. To see a meaningful user name:
 1. Click the HBase service.
 2. Click the **Configuration** tab.
 3. Select **Service-wide > Security**.
 4. Set the HBase Secure RPC Engine property to `true`.
 5. Save the change and restart the service.

- **Hive** - Minimum supported versions: CDH 4.2.0, CDH 4.4.0 for operations denied due to lack of privileges.

The captured operations are:

- Operations (except grant, revoke, and metadata access only) sent to HiveServer2
- Operations denied due to lack of privileges

**Note:**

- Actions taken against Hive using the Hive CLI are *not* audited. Therefore if you have enabled auditing you should disable the Hive CLI to prevent actions against Hive that are not audited.
- In simple authentication mode, the username in log events is the username passed in the HiveServer2 connect command. If you do not pass a username in the connect command, the username in log events is `anonymous`.

- **Hue** - Minimum supported version: CDH 4.2.0.

The captured operations are:

- Operations (except grant, revoke, and metadata access only) sent to Beeswax Server



Note: You do not directly configure the Hue service for auditing. Instead, when you configure the Hive service for auditing, operations sent to the Hive service through Beeswax appear in the Hue service audit log.

- **Cloudera Impala** - Minimum supported version: Cloudera Impala 1.2.1.

The captured operations are:

- Queries denied due to lack of privileges
- Queries that pass analysis

- **Sentry**

The captured operations are:

- Operations sent to the HiveServer2 and Hive Metastore Server roles and Impala service
- Add and delete roles, assign roles to groups and remove roles from groups, create and delete privileges, grant and revoke privileges
- Operations denied due to lack of privileges



Note: You do not directly configure the Sentry service for auditing. Instead, when you configure the Hive and Impala services for auditing, grant, revoke, and metadata operations appear in the Hive or Impala service audit logs.

Cloudera Navigator Metadata Component

This section describes the CDH and managed service versions supported by the Cloudera Navigator metadata component.

CDH 4.4.0 and higher for all components except Pig. For Pig, CDH 4.6.0 and higher. The supported components are:

- HDFS. However, federated HDFS is *not supported*.
- Hive
- MapReduce
- Oozie
- Pig
- Sqoop 1 - all [Cloudera connectors](#) are supported.
- YARN

CDH 5 Requirements and Supported Versions

The following sections describe the requirements and supported versions of operating systems, databases, JDK, and Internet Protocol (IP) for CDH 5.

For the latest information on compatibility across all Cloudera products, see the [Product Compatibility Matrix](#).

Supported Operating Systems

CDH 5 provides packages for Red-Hat-compatible, SLES, Ubuntu, and Debian systems as described below.

Operating System	Version	Packages
Red Hat Enterprise Linux (RHEL)-compatible		
Red Hat Enterprise Linux	5.7	64-bit
	6.2	64-bit
	6.4	64-bit
	6.4 in SE Linux mode	64-bit
	6.5	64-bit
CentOS	5.7	64-bit
	6.2	64-bit
	6.4	64-bit
	6.4 in SE Linux mode	64-bit
	6.5	64-bit

Operating System	Version	Packages
Oracle Enterprise Linux (OEL) with Unbreakable Enterprise Kernel (UEK)	5.6 (UEK R2)	64-bit
	6.4 (UEK R2)	64-bit
	6.5 (UEK R2, UEK R3)	64-bit
SLES		
SLES Linux Enterprise Server (SLES)	11 with Service Pack 2 or later	64-bit
Ubuntu/Debian		
Ubuntu	Precise (12.04) - Long-Term Support (LTS)	64-bit
	Trusty (14.04) - Long-Term Support (LTS)	64-bit
Debian	Wheezy (7.0, 7.1)	64-bit


Note:

- CDH 5 provides *only* 64-bit packages.
- Cloudera has received reports that our RPMs work well on Fedora, but we have not tested this.
- If you are using an operating system that is not supported by Cloudera packages, you can also download source tarballs from [Downloads](#).

Supported Databases

Component	MySQL	SQLite	PostgreSQL	Oracle	Derby - see Note 4
Oozie	5.1, 5.5, 5.6	–	8.1, 8.3, 8.4, 9.1, 9.2, 9.3 See Note 2	11gR2	Default
Flume	–	–	–	–	Default (for the JDBC Channel only)
Hue	5.1, 5.5, 5.6 See Note 6	Default	8.1, 8.3, 8.4, 9.1, 9.2, 9.3 See Note 2	11gR2	–
Hive/Impala	5.1, 5.5, 5.6 See Note 1	–	8.1, 8.3, 8.4, 9.1, 9.2, 9.3 See Note 2	11gR2	Default
Sentry	5.1, 5.5, 5.6 See Note 1	–	8.1, 8.3, 8.4, 9.1, 9.2, 9.3 See Note 2	11gR2	–
Sqoop 1	See Note 3	–	See Note 3	See Note 3	–
Sqoop 2	See Note 4	–	See Note 4	See Note 4	Default



Note:

1. MySQL 5.5 is supported on CDH 5.1. MySQL 5.6 is supported on CDH 5.1 and later. The InnoDB storage engine must be enabled in the MySQL server.
2. PostgreSQL 9.2 is supported on CDH 5.1 and later. PostgreSQL 9.3 is supported on CDH 5.2 and later.
3. For the purposes of transferring data only, Sqoop 1 supports MySQL 5.0 and above, PostgreSQL 8.4 and above, Oracle 10.2 and above, Teradata 13.10 and above, and Netezza TwinFin 5.0 and above. The Sqoop metastore works only with HSQLDB (1.8.0 and higher 1.x versions; the metastore does not work with any HSQLDB 2.x versions).
4. Sqoop 2 can transfer data to and from MySQL 5.0 and above, PostgreSQL 8.4 and above, Oracle 10.2 and above, and Microsoft SQL Server 2012 and above. The Sqoop 2 repository database is supported only on Derby.
5. Derby is supported as shown in the table, but not always recommended. See the pages for individual components in the [#unique_32](#) guide for recommendations.
6. CDH 5 Hue requires the default MySQL version of the operating system on which it is being installed. For example, on RHEL/CentOS 6 you will need MySQL 5.1.

Supported JDK Versions

CDH 5.3.x requires the Oracle JDK, and supports the versions shown in the table that follows.



Important:

- There is one exception to the minimum supported and recommended JDK versions in the following table. If Oracle releases a security patch that affects server-side Java before the next minor release of Cloudera products, the Cloudera support policy covers customers using the patch.
- Client applications using CDH libraries must be running a supported JDK version that matches the JDK version of the CDH cluster they are connecting to.

Table 4: Supported JDK Versions

Recommended Version	Minimum Supported Version	Exceptions
1.7.0_67	1.7.0_67	None
1.8.0_11 or higher	1.8.0_11	None

Supported Network Protocols

- CDH requires IPv4. IPv6 is not supported.
See also [Configuring Network Names](#) on page 190.
- Multihoming CDH or Cloudera Manager is not supported outside specifically certified Cloudera partner appliances. Cloudera finds that current Hadoop architectures combined with modern network infrastructures and security practices remove the need for multihoming. Multihoming, however, is beneficial internally in appliance form factors to take advantage of high-bandwidth InfiniBand interconnects.

Although some subareas of the product may work with unsupported custom multihoming configurations, there are known issues with multihoming. In addition, unknown issues may arise because multihoming is not covered by our test matrix outside the Cloudera-certified partner appliances.

Supported Configurations with Virtualization and Cloud Platforms

This section lists supported configurations for deploying Cloudera software on virtualization and cloud platforms, and provides links to reference architectures for these platforms.

Microsoft Azure

For information on deploying Cloudera software on a Microsoft Azure cloud infrastructure, see the [Reference architecture for deploying on Microsoft Azure](#).

The following limitations and restrictions apply to deploying on Microsoft Azure in the current release:

- Only the D-14 instance type with locally attached disks is supported.
- Only Cloudera Manager 5.x and CDH 5.x are supported.
- The only supported operating system is CentOS 6.5.
- The following services are supported:
 - MRv2 (YARN)
 - Hive
 - Pig
 - Crunch
- The following services are not supported:
 - HBase
 - Impala
 - Spark
 - Solr

VMware

For information on deploying Cloudera software on a VMware-based infrastructure, see the [Reference architecture for deploying on VMware](#).

The following limitations and restrictions apply to deploying on VMware in the current release:

- Use the part of Hadoop Virtual Extensions that has been implemented in HDFS: HADOOP-8468. This will prevent data loss when a physical node goes down that hosts two or more DataNodes.
- Isilon and shared storage are not supported.

Ports

Cloudera Manager, CDH components, managed services, and third-party components use the ports listed in the tables that follow. Before you deploy Cloudera Manager, CDH, and managed services, and third-party components make sure these ports are open on each system. If you are using a firewall, such as iptables, and cannot open all the listed ports, you will need to disable the firewall completely to ensure full functionality.

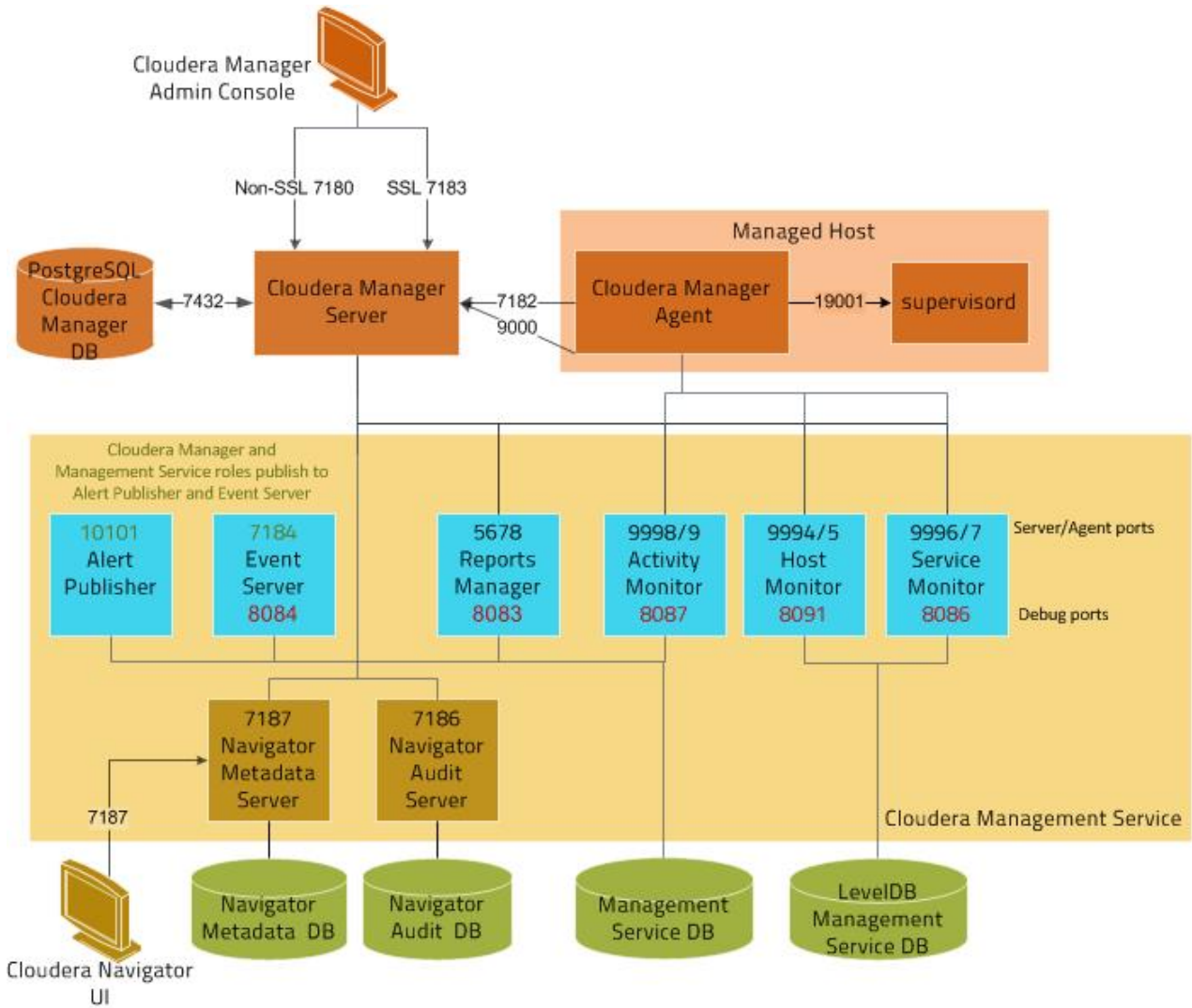


Note:

In the tables in the subsections that follow, the **Access Requirement** for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the nodes (for example the JournalNode ports in an HA configuration); "External" means that the port can be used for either internal or external communication (for example, ports used by the Web UIs for the NodeManager and the JobHistory Server).

Ports Used by Cloudera Manager and Cloudera Navigator

The following diagram provides an overview of the ports used by Cloudera Manager, Cloudera Navigator, and Cloudera Management Service roles:



For further details, see the following table. All ports listed are TCP.

Component	Service	Port	Access Requirement	Configuration	Comment
Cloudera Manager Server	HTTP (Web UI)	7180	External	Administration > Settings > Ports and Addresses	
	HTTPS (Web UI)	7183	External		Used for HTTPS on master, if enabled. HTTP is the default; only one port is open for either HTTP or HTTPS, not both
	Avro (RPC)	7182	Internal		Used for Agent to Server heartbeats
	PostgreSQL database managed by	7432	Internal		The optional embedded PostgreSQL database used for storing configuration

Component	Service	Port	Access Requirement	Configuration	Comment
	cloudera-sm-server service				information for Cloudera Manager Server.
Cloudera Manager Agent	HTTP (Debug)	9000	Internal	/etc/cloudera-sm-agent/config.ini	
	Internal supervisord	localhost:19001	localhost		supervisord status and control port; used for communication between the Agent and supervisord; only open internally (on localhost)
Event Server	Listens for the publication of events.	7184	Internal	Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
	Listens for queries for events.	7185	Internal		
	HTTP (Debug)	8084	Internal		Allows access to debugging and diagnostic information
Alert Publisher	Internal API	10101	Internal	Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
Service Monitor	HTTP (Debug)	8086	Internal	Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
	Listening for Agent messages (private protocol)	9997			
	Internal query API (Avro)	9996			
Activity Monitor	HTTP (Debug)	8087	Internal	Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
	Listening for Agent messages (private protocol)	9999			

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5

Component	Service	Port	Access Requirement	Configuration	Comment
	Internal query API (Avro)	9998			
Host Monitor	HTTP (Debug)	8091	Internal	Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
	Listening for Agent messages (private protocol)	9995			
	Internal query API (Avro)	9994			
Reports Manager	Queries (Thrift)	5678	Internal	Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
	HTTP (Debug)	8083	Internal		
Cloudera Navigator				Cloudera Management Service > Configuration > <i>ServerName</i> Default Group > Ports and Addresses	
Audit Server	HTTP	7186	Internal		
	HTTP (Debug)	8089	Internal		The port where Navigator Audit Server starts a debug web server. Set to -1 to disable debug server.
Metadata Server	HTTP (Web UI)	7187	External		
Task Tracker Plug-in (used for activity monitoring)	HTTP (Debug)	localhost: 4867	localhost		Used only on localhost interface by monitoring agent
Backup and Disaster Recovery	HTTP (Web UI)	7180	External	Administration > Settings page > Ports and Addresses	Used for communication to peer (source) Cloudera Manager.
	HDFS NameNode	8020	External	HDFS > Configuration > NameNode Role Group > Ports and Addresses: NameNode Port	HDFS and Hive replication: communication from destination HDFS and MapReduce hosts to source HDFS NameNode(s). Hive Replication: communication from source Hive hosts to

Component	Service	Port	Access Requirement	Configuration	Comment
					destination HDFS NameNode(s).
	HDFS DataNode	50010	External	HDFS > Configuration > DataNode Role Group(s) > Ports and Addresses: DataNode Transceiver Port	HDFS and Hive replication: communication from destination HDFS and MapReduce hosts to source HDFS DataNode(s). Hive Replication: communication from source Hive hosts to destination HDFS DataNode(s).

Ports Used by Components of CDH 5

All ports listed are TCP.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hadoop HDFS	DataNode		50010	External	dfs.datanode.address	DataNode HTTP server port
	DataNode	Secure	1004	External	dfs.datanode.address	
	DataNode		50075	External	dfs.datanode.http.address	
	DataNode		50475	External	dfs.datanode.https.address	
	DataNode	Secure	1006	External	dfs.datanode.http.address	
	DataNode		50020	External	dfs.datanode.ipc.address	
	NameNode		8020	External	fs.default.name or fs.defaultFS	fs.default.name is deprecated (but still works)
	NameNode		8022	External	dfs.namenode.servicerpc.address	Optional port used by HDFS daemons to avoid sharing the RPC port used by clients (8020). Cloudera recommends using port 8022.
	NameNode		50070	External	dfs.http.address or dfs.namenode.http-address	dfs.http.address is deprecated (but still works)
	NameNode	Secure	50470	External	dfs.https.address or	dfs.https.address is deprecated (but still works)

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
					<code>dfs.namenode.https-address</code>	
	Secondary NameNode		50090	Internal	<code>dfs.secondary.http.address</code> or <code>dfs.namenode.secondary.http-address</code>	<code>dfs.secondary.http.address</code> is deprecated (but still works)
	Secondary NameNode	Secure	50495	Internal	<code>dfs.secondary.https.address</code>	
	JournalNode		8485	Internal	<code>dfs.namenode.shared.edits.dir</code>	
	JournalNode		8480	Internal	<code>dfs.journalnode.http-address</code>	
	JournalNode		8481	Internal	<code>dfs.journalnode.https-address</code>	
	Failover Controller		8019	Internal		Used for NameNode HA
	NFS gateway		2049			<code>nfs</code> <code>port (</code> <code>nfs3.server.port</code> <code>)</code>
	NFS gateway		4242			<code>mountd</code> <code>port (</code> <code>nfs3.mountd.port</code> <code>)</code>
	NFS gateway		111			<code>portmapper</code> or <code>rpcbind</code> <code>port</code>
	HttpFS		14000			
	HttpFS		14001			
Hadoop MapReduce (MRv1)	JobTracker		8021	External	<code>mapred.job.tracker</code>	
	JobTracker		8023	External	<code>mapred.ha.job.tracker</code>	High Availability service protocol port for the JobTracker. The JobTracker listens on a separate port for HA operations.
	JobTracker		50030	External	<code>mapred.job.tracker.http.address</code>	

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	JobTracker	Thrift Plugin	9290	Internal	jobtracker.thrift.address	Required by Hue and Cloudera Manager Activity Monitor
	TaskTracker		50060	External	mapred.task.tracker.http.address	
	TaskTracker		0	Localhost	mapred.task.tracker.report.address	Communicating with child (umbilical)
	Failover Controller		8018	Internal	mapred.ha.zkfc.port	Used for JobTracker HA
Hadoop YARN (MRv2)	ResourceManager		8032	External	yarn.resourcemanager.address	
	ResourceManager		8030	Internal	yarn.resourcemanager.scheduler.address	
	ResourceManager		8031	Internal	yarn.resourcemanager.resource-tracker.address	
	ResourceManager		8033	External	yarn.resourcemanager.admin.address	
	ResourceManager		8088	External	yarn.resourcemanager.webapp.address	
	ResourceManager		8090		yarn.resourcemanager.webapp.https.address	
	NodeManager		8040	Internal	yarn.nodemanager.localizer.address	
	NodeManager		8041	Internal	yarn.nodemanager.address	
	NodeManager		8042	External	yarn.nodemanager.webapp.address	
	NodeManager		8044	External	yarn.nodemanager.webapp.https.address	
	JobHistory Server		10020	Internal	mapreduce.jobhistory.address	
	JobHistory Server		10033	Internal	mapreduce.jobhistory.admin.address	
	Shuffle HTTP		13562	Internal		

Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	JobHistory Server		19888	External	mapreduce.jobhistory.webapp.address	
	JobHistory Server		19890	External	mapreduce.jobhistory.webapp.https.address	
Flume	Flume Agent		41414	External		
Hadoop KMS	Key Management Server		16000	External	kms_http_port	CDH 5.2.1 and higher
	Key Management Server		16001	Localhost	kms_admin_port	CDH 5.2.1 and higher
HBase	Master		60000	External	hbase.master.port	IPC
	Master		60010	External	hbase.master.info.port	HTTP
	RegionServer		60020	External	hbase.regionserver.port	IPC
	RegionServer		60030	External	hbase.regionserver.info.port	HTTP
	HQuorumPeer		2181		hbase.zookeeper.property.clientPort	HBase-managed ZK mode
	HQuorumPeer		2888		hbase.zookeeper.peerport	HBase-managed ZK mode
	HQuorumPeer		3888		hbase.zookeeper.leaderport	HBase-managed ZK mode
	REST	Overridable	8080	External	hbase.rest.port	The default REST port in HBase is 8080. Because this is a commonly used port, Cloudera Manager sets the default to 20550 instead.
	REST	Overridable	20550	External	hbase.rest.port	The default REST port in HBase is 8080. Because this is a commonly used port, Cloudera Manager sets the default to 20550 instead.
	REST UI		8085	External		
	ThriftServer	Thrift Server	9090	External	Pass <code>-p <port></code> on CLI	
	ThriftServer		9095	External		

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
		Avro server	9090	External	Pass <code>--port <port></code> on CLI	
	hbase-solr-indexer	Lily Indexer	11060	External		
Hive	Metastore		9083	External		
	HiveServer2		10000	External	<code>hive.server2.thrift.port</code>	The Beeline command interpreter requires that you specify this port on the command line.
	WebHCat Server		50111	External	<code>templeton.port</code>	
Sentry	Sentry Server		8038	External	<code>sentry.service.server.rpc-port</code>	
	Sentry Server		51000	External	<code>sentry.service.web.port</code>	
Sqoop	Metastore		16000	External	<code>sqoop.metastore.server.port</code>	
Sqoop 2	Sqoop 2 server		8005	Localhost	<code>SQOOP_ADMIN_PORT</code> environment variable	
	Sqoop 2 server		12000	External		
	Sqoop 2		12001	External		Admin port
ZooKeeper	Server (with CDH 5 or Cloudera Manager 5)		2181	External	<code>clientPort</code>	Client port
	Server (with CDH 5 only)		2888	Internal	<code>X in server.N =host:X:Y</code>	Peer
	Server (with CDH 5 only)		3888	Internal	<code>X in server.N =host:X:Y</code>	Peer
	Server (with CDH 5 and Cloudera Manager 5)		3181	Internal	<code>X in server.N =host:X:Y</code>	Peer
	Server (with CDH 5 and Cloudera Manager 5)		4181	Internal	<code>X in server.N =host:X:Y</code>	Peer
	ZooKeeper JMX port		9010	Internal		ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must <i>EITHER</i>

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
						<ul style="list-style-type: none"> Open up all ports when the connection originates from the Cloudera Manager server; <i>OR</i> Do the following: <ol style="list-style-type: none"> Open a non-ephemeral port (such as 9011) in the firewall. Install Oracle Java 7u4 JDK or later. Add the port configuration to the safety valve, for example: oozie.config.jmx.port=9011 Restart ZooKeeper.
Hue	Server		8888	External		
Oozie	Oozie Server		11000	External	OOZIE_HTTP_PORT in oozie-env.sh	HTTP
	Oozie Server	SSL	11443	External		HTTPS
	Oozie Server		11001	localhost	OOZIE_ADMIN_PORT in oozie-env.sh	Shutdown port
Spark	Default Master RPC port		7077	External		
	Default Worker RPC port		7078			
	Default Master web UI port		18080	External		
	Default Worker web UI port		18081	External		
	History Server		18088	External	history.port	

Ports Used by Impala

Impala uses the TCP ports listed in the following table. Before deploying Impala, ensure these ports are open on each system.

Component	Service	Port	Access Requirement	Comment
Impala Daemon	Impala Daemon Frontend Port	21000	External	Used to transmit commands and receive results by <code>impala-shell</code> and version 1.2 of the Cloudera ODBC driver.
Impala Daemon	Impala Daemon Frontend Port	21050	External	Used to transmit commands and receive results by applications, such as Business Intelligence tools, using JDBC, the Beeswax query editor in Hue, and version 2.0 or higher of the Cloudera ODBC driver.
Impala Daemon	Impala Daemon Backend Port	22000	Internal	Internal use only. Impala daemons use this port to communicate with each other.
Impala Daemon	StateStoreSubscriber Service Port	23000	Internal	Internal use only. Impala daemons listen on this port for updates from the statestore daemon.
Catalog Daemon	StateStoreSubscriber Service Port	23020	Internal	Internal use only. The catalog daemon listens on this port for updates from the statestore daemon.
Impala Daemon	Impala Daemon HTTP Server Port	25000	External	Impala web interface for administrators to monitor and troubleshoot.
Impala StateStore Daemon	StateStore HTTP Server Port	25010	External	StateStore web interface for administrators to monitor and troubleshoot.
Impala Catalog Daemon	Catalog HTTP Server Port	25020	External	Catalog service web interface for administrators to monitor and troubleshoot. New in Impala 1.2 and higher.
Impala StateStore Daemon	StateStore Service Port	24000	Internal	Internal use only. The statestore daemon listens on this port for registration/unregistration requests.
Impala Catalog Daemon	Catalog Service Port	26000	Internal	Internal use only. The catalog service uses this port to communicate with the Impala

Component	Service	Port	Access Requirement	Comment
				daemons. New in Impala 1.2 and higher.
Impala Daemon	Llama Callback Port	28000	Internal	Internal use only. Impala daemons use to communicate with Llama. New in CDH 5.0.0 and higher.
Impala Llama ApplicationMaster	Llama Thrift Admin Port	15002	Internal	Internal use only. New in CDH 5.0.0 and higher.
Impala Llama ApplicationMaster	Llama Thrift Port	15000	Internal	Internal use only. New in CDH 5.0.0 and higher.
Impala Llama ApplicationMaster	Llama HTTP Port	15001	External	Llama service web interface for administrators to monitor and troubleshoot. New in CDH 5.0.0 and higher.

Ports Used by Cloudera Search

Component	Service	Port	Protocol	Access Requirement	Comment
Cloudera Search	Solr search/update	8983	http	External	All Solr-specific actions, update/query.
Cloudera Search	Solr (admin)	8984	http	Internal	Solr administrative use.

Ports Used by Third-Party Components

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
Ganglia	ganglia-gmond		8649	UDP/TCP	Internal		
	ganglia-web		80	TCP	External	Via Apache httpd	
Kerberos	KRB5 KDC Server	Secure	88	UDP/TCP	External	kdc_ports and kdc_tcp_ports in either the [kdcdefaults] or [realms] sections of kdc.conf	By default only UDP

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
	KRB5 Admin Server	Secure	749	TCP	Internal	kadmind_port in the [realms] section of kdc.conf	
	kpasswd		464	UDP/TCP	Internal		
SSH	ssh		22	TCP	External		
PostgreSQL			5432	TCP			
MySQL			3306	TCP			
LDAP	LDAP Server		389	TCP			
	LDAP Server over SSL/TLS	SSL/TLS	636	TCP			
	Global Catalog		3268	TCP			
	Global Catalog over SSL/TLS	SSL/TLS	3269	TCP			

Installing Cloudera Manager and CDH

This section introduces options for installing Cloudera Manager, CDH, and managed services. You can install:

- Cloudera Manager, CDH, and managed services in a Cloudera Manager deployment. This is the recommended method for installing CDH and managed services.
- CDH 5 into an unmanaged deployment.

Cloudera Manager Deployment

A Cloudera Manager deployment consists of the following software components:

- Oracle JDK
- Cloudera Manager Server and Agent packages
- Supporting database software
- CDH and managed service software

This section describes the three main installation paths for creating a new Cloudera Manager deployment and the criteria for choosing an installation path. If your cluster already has an installation of a previous version of Cloudera Manager, follow the instructions in [Upgrading Cloudera Manager](#).

The Cloudera Manager installation paths share some common phases, but the variant aspects of each path support different user and cluster host requirements:

- **Demonstration and proof of concept deployments** - There are two installation options:
 - [Installation Path A - Automated Installation by Cloudera Manager](#) on page 88 - Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, and Cloudera Manager Agent, CDH, and managed service software on cluster hosts, and configures databases for the Cloudera Manager Server and Hive Metastore and optionally for Cloudera Management Service roles. This path is recommended for demonstration and proof of concept deployments, but is *not recommended* for production deployments because its not intended to scale and may require database migration as your cluster grows. To use this method, server and cluster hosts must satisfy the following requirements:
 - Provide the ability to log in to the Cloudera Manager Server host using a root account or an account that has password-less sudo permission.
 - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See [Networking and Security Requirements](#) on page 8 for further information.
 - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the necessary installation files.
 - [Installation Path B - Manual Installation Using Cloudera Manager Packages](#) on page 95 - you install the Oracle JDK and Cloudera Manager Server, and embedded PostgreSQL database packages on the Cloudera Manager Server host. You have two options for installing Oracle JDK, Cloudera Manager Agent, CDH, and managed service software on cluster hosts: manually install it yourself or use Cloudera Manager to automate installation. However, in order for Cloudera Manager to automate installation of Cloudera Manager Agent packages or CDH and managed service software, cluster hosts must satisfy the following requirements:
 - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See [Networking and Security Requirements](#) on page 8 for further information.
 - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the necessary installation files.
- **Production deployments** - require you to first manually install and configure a production [database](#) for the Cloudera Manager Server and Hive Metastore. There are two installation options:

- [Installation Path B - Manual Installation Using Cloudera Manager Packages](#) on page 95 - you install the Oracle JDK and Cloudera Manager Server packages on the Cloudera Manager Server host. You have two options for installing Oracle JDK, Cloudera Manager Agent, CDH, and managed service software on cluster hosts: manually install it yourself or use Cloudera Manager to automate installation. However, in order for Cloudera Manager to automate installation of Cloudera Manager Agent packages or CDH and managed service software, cluster hosts must satisfy the following requirements:
 - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See [Networking and Security Requirements](#) on page 8 for further information.
 - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the necessary installation files.
- [Installation Path C - Manual Installation Using Cloudera Manager Tarballs](#) on page 111 - you install the Oracle JDK, Cloudera Manager Server, and Cloudera Manager Agent software as tarballs and use Cloudera Manager to automate installation of CDH and managed service software as parcels.

Unmanaged Deployment

In an unmanaged deployment, you are responsible for managing all phases of the life cycle of CDH and managed service components on each host: installation, configuration, and service life cycle operations such as start and stop. This section describes alternatives for installing CDH 5 software in an unmanaged deployment.

- **Command-line methods:**

- Download and install the CDH 5 "1-click Install" package
- Add the CDH 5 repository
- Build your own CDH 5 repository

If you use one of these command-line methods, the first (downloading and installing the "1-click Install" package) is recommended in most cases because it is simpler than building or adding a repository. See [Installing the Latest CDH 5 Release](#) on page 155 for detailed instructions for each of these options.

- **Tarball** You can download a tarball from [CDH downloads](#). Keep the following points in mind:

- Installing CDH 5 from a tarball installs YARN.
- In CDH 5, there is no separate tarball for MRv1. Instead, the MRv1 binaries, examples, etc., are delivered in the Hadoop tarball. The scripts for running MRv1 are in the `bin-mapreduce1` directory in the tarball, and the MRv1 examples are in the `examples-mapreduce1` directory.

Java Development Kit Installation

Some installation paths require that you install the Oracle Java Development Kit on cluster hosts before deploying Cloudera Manager, CDH, and managed services. To install the Oracle JDK, follow the instructions in [Installing the Oracle JDK](#) on page 36. The completed installation, or any already existing installation, must meet the following requirements.

Requirements

- The JDK must be 64-bit. Do not use a 32-bit JDK.
- Install a supported version:
 - CDH 5 - [Supported JDK Versions](#) on page 20
 - CDH 4 - [Supported JDK Versions](#)
- Install the *same version* of the Oracle JDK on each host.
- Install the JDK in `/usr/java/jdk-version`.



Important:

- You cannot [upgrade from JDK 1.7 to JDK 1.8](#) while upgrading to CDH 5.3. The cluster must already be running CDH 5.3 when you upgrade to JDK 1.8.
- If you are upgrading from a lower major version of the JDK to JDK 1.8 or from JDK 1.6 to JDK 1.7 and you are using AES-256 bit encryption, you must install new encryption policy files. (In a Cloudera Manager deployment, Cloudera Manager offers you an option to automatically install the policy files; for unmanaged deployments, install them manually.) See [If you are Using AES-256 Encryption, install the JCE Policy File](#).

For both managed and unmanaged deployments, you must also ensure that the Java Truststores are retained during the upgrade. (See [Creating Truststores](#).)

- On SLES 11 platforms, do not install or try to use the IBM Java version bundled with the SLES distribution. CDH does not run correctly with that version.

Installing the Oracle JDK

The Oracle JDK installer is available both as an RPM-based installer for RPM-based systems, and as a binary installer for other systems.



















1. Download the `.tar.gz` file for one of the supported versions of the Oracle JDK from [Java SE 8 Downloads](#), [Java SE 7 Downloads](#), or [Java SE 6 Downloads](#). (These links are correct at the time of writing but change frequently.)
2. Extract the JDK to `/usr/java/jdk-version`; for example `/usr/java/jdk.1.7.0_nn` or `/usr/java/jdk.1.8.0_nn`, where `nn` is a supported version.
3. Set `JAVA_HOME` to the directory where the JDK is installed. Add the following line to the specified files:

```
export JAVA_HOME=/usr/java/jdk.1.7.0_nn
```

- Cloudera Manager Server host: `/etc/default/cloudera-scm-server`. This affects only the Cloudera Manager Server process, and does not affect the Cloudera Management Service roles.
- All hosts in an unmanaged deployment: `/etc/default/bigtop-utils`. You do not need to do this for clusters managed by Cloudera Manager.

Installing Cloudera Manager, CDH, and Managed Services

The following diagram illustrates the phases required to install Cloudera Manager and a Cloudera Manager deployment of CDH and managed services. Every phase is required, but you can accomplish each phase in multiple ways, depending on your organization's policies and requirements.

Installation Phases			
Phase 1: Install JDK JDK required by the Server, Management Service, and CDH services is installed.	 Cloudera Manager Installer Installs supported versions of the Oracle JDK in <code>/usr/java</code> .	 JDK Install the same version of the supported versions of the Oracle JDK on each host and set the <code>JAVA_HOME</code> environment variable to the install directory.	Legend  Interactive  Command-Line
Phase 2: Set up DBs Databases required by the Server, Management Service, and optional for some CDH services are installed, configured, and running.	 Cloudera Manager Installer Installs and configures embedded PostgreSQL packages and starts embedded database.	 Embedded PostgreSQL Database <pre> yum install cloudera-manager-server-db-2 service cloudera-manager-server-db start Installs a PostgreSQL daemon on port 7432 in <code>/var/lib/cloudera-scm-server-db</code>. </pre>	 External Database Install and start PostgreSQL, MySQL, or Oracle and create required databases.
Installation Paths			
	A	B	C
Phase 3: Install Server Cloudera Manager Server installed and running on one host.	 Cloudera Manager Installer Installs latest Cloudera Manager Server packages and Server. Requires Internet access and sudo access to Server host.	 Package <pre> yum install cloudera-manager-server cloudera-manager-daemons vi <code>/etc/cloudera-scm-server/db.properties</code> service cloudera-manager-server start </pre>	 Tarball <pre> tar xzf cloudera-manager*.tar.gz -C /opt/cloudera-manager service cloudera-manager-server start </pre>
Phase 4: Install Agents Cloudera Manager Agents installed and running on every host.	 Cloudera Manager Installation Wizard Installs Cloudera Manager Agent package. Requires SSH credentials (password or key) for root or sudo-enabled user.	 Package <pre> yum install cloudera-manager-agent cloudera-manager-daemons vi <code>config.ini</code> service cloudera-manager-agent start </pre>	 Tarball <pre> vi <code>config.ini</code> service cloudera-manager-agent start </pre>
Phase 5: Install CDH and Managed Service SW CDH and managed service software installed on every host.	 Cloudera Manager Installation Wizard Installs choice of CDH and managed service version and repo. Installs parcels or packages.	 Parcel Remote or local repo or manual unpacking. API or UI.	 Package <pre> yum install hadoop zookeeper hue oozie ... </pre>
Phase 6: Create, Configure, and Start CDH and Managed Services CDH and managed services configured and running.	 Cloudera Manager Installation Wizard Creates, configures, and starts selected services, allows assignment of roles to hosts, and setting configuration properties. Auto-configures many options.	 API <pre> POST /api/<version>/cm/deployment Best for scripting pre-configured deployments. </pre>	

The six phases are grouped into three installation paths based on how the Cloudera Manager Server and database software are installed on the Cloudera Manager Server and cluster hosts. The criteria for choosing an installation path are discussed in [Cloudera Manager Deployment](#) on page 34.

Cloudera Manager Installation Software

Cloudera Manager provides the following software for the supported installation paths:

- Installation path A** - A small self-executing Cloudera Manager installation program to install the Cloudera Manager Server and other packages in preparation for host installation. The Cloudera Manager installer, which you install on the host where you want the Cloudera Manager Server to run, performs the following:
 1. Installs the package repositories for Cloudera Manager and the Oracle Java Development Kit (JDK)
 2. Installs the Cloudera Manager packages

3. Installs and configures an embedded PostgreSQL database for use by the Cloudera Manager Server, some Cloudera Management Service roles, some managed services, and Cloudera Navigator roles
- **Installation paths B and C** - Cloudera Manager package repositories for manually installing the Cloudera Manager Server, Agent, and embedded database packages.
 - **All installation paths** - The Cloudera Manager Installation wizard for automating CDH and managed service installation and configuration on the cluster hosts. Cloudera Manager provides two methods for installing CDH and managed services: parcels and packages. Parcels simplify the installation process and allow you to download, distribute, and activate new versions of CDH and managed services from within Cloudera Manager. After you install Cloudera Manager and you connect to the Cloudera Manager Admin Console for the first time, use the Cloudera Manager Installation wizard to:
 1. Discover cluster hosts
 2. Optionally install the Oracle JDK
 3. Optionally install CDH, managed service, and Cloudera Manager Agent software on cluster hosts
 4. Select services
 5. Map service roles to hosts
 6. Edit service configurations
 7. Start services

If you abort the software installation process, the wizard automatically reverts and rolls back the installation process for any uninstalled components. (Installation that has completed successfully on a host is not rolled back on that host.)

Cloudera Manager and Managed Service Data Stores

Cloudera Manager uses databases to store information about the Cloudera Manager configuration, as well as information such as the health of the system or task progress. For quick, simple installations, Cloudera Manager can install and configure an embedded PostgreSQL database as part of the Cloudera Manager installation process. In addition, some CDH services use databases and are automatically configured to use a default database. If you plan to use the embedded and default databases provided during the Cloudera Manager installation, see [Installation Path A - Automated Installation by Cloudera Manager](#) on page 88.

Although the embedded database is useful for getting started quickly, you can also use your own PostgreSQL, MySQL, or Oracle database for the Cloudera Manager Server and services that use databases.

For information about planning, managing, and backing up Cloudera Manager data stores, see [Storage Space Planning for Cloudera Manager](#) on page 60.

Required Databases

The Cloudera Manager Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server all require databases. The type of data contained in the databases and their estimated sizes are as follows:

- Cloudera Manager - Contains all the information about services you have configured and their role assignments, all configuration history, commands, users, and running processes. This relatively small database (<100 MB) is the most important to back up.



Important: When processes restart, the configuration for each of the services is redeployed using information that is saved in the Cloudera Manager database. If this information is not available, your cluster will not start or function correctly. You must therefore schedule and maintain regular backups of the Cloudera Manager database in order to recover the cluster in the event of the loss of this database.

- Activity Monitor - Contains information about past activities. In large clusters, this database can grow large. Configuring an Activity Monitor database is only necessary if a MapReduce service is deployed.
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.

- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Sentry Server - Contains authorization metadata. Relatively small.
- Cloudera Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large.
- Cloudera Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small.

The Cloudera Manager Service Host Monitor and Service Monitor roles have an [internal datastore](#).

Cloudera Manager provides three installation paths:

- Path A automatically installs an embedded PostgreSQL database to meet the requirements of the services. This path reduces the number of installation tasks to complete and choices to make. In Path A you can optionally choose to create external databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server.
- Path B and Path C require you to create databases for the Cloudera Manager Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server.

Using an external database requires more input and intervention as you install databases or gather information about existing ones. These paths also provide greater flexibility in choosing database types and configurations.

Cloudera Manager supports deploying different types of databases in a single environment, but doing so can create unexpected complications. Cloudera recommends choosing one supported database provider for all of the Cloudera databases.

In most cases, you should install databases and services on the same host. For example, if you create the database for Activity Monitor on `myhost1`, then you should typically assign the Activity Monitor role to `myhost1`. You assign the Activity Monitor and Reports Manager roles in the Cloudera Manager wizard during the installation or upgrade process. After completing the installation or upgrade process, you can also modify role assignments in the Management services pages of Cloudera Manager. Although the database location is changeable, before beginning an installation or upgrade, you should decide which hosts to use. The JDBC connector for your database *must* be installed on the hosts where you assign the Activity Monitor and Reports Manager roles.

You can install the database and services on different hosts. Separating databases from services is more likely in larger deployments and in cases where more sophisticated database administrators choose such a configuration. For example, databases and services might be separated if your environment includes Oracle databases that are managed separately by Oracle database administrators.

Setting up the Cloudera Manager Server Database

The Cloudera Manager Server database stores information about service and host configurations. You can use an embedded PostgreSQL database or an external database.

Installing and Starting the Cloudera Manager Server Embedded Database

If you are using [Installation Path B - Manual Installation Using Cloudera Manager Packages](#) on page 95 and you want to use an embedded PostgreSQL database for the Cloudera Management Server, use this procedure to install and start the database:

1. Install the embedded PostgreSQL database packages:

OS	Command
Red Hat-compatible, if you have a yum repo configured	<code>\$ sudo yum install cloudera-manager-server-db-2</code>
Red Hat-compatible, if you're transferring RPMs manually	<code>sudo yum --nogpgcheck localinstall cloudera-manager-server-db-2.noarch.rpm</code>
SLES	<code>\$ sudo zypper install cloudera-manager-server-db-2</code>

OS	Command
Ubuntu or Debian	\$ sudo apt-get install cloudera-manager-server-db-2

2. Start the PostgreSQL database:

```
$ sudo service cloudera-scm-server-db start
```

Preparing a Cloudera Manager Server External Database

Before performing these steps, install and configure a database as described in [MySQL Database](#) on page 48, [Oracle Database](#) on page 53, or [External PostgreSQL Database](#) on page 45.

1. Run the `scm_prepare_database.sh` script on the host where the Cloudera Manager Server package is installed:

- Installer or package install

```
/usr/share/cmf/schema/scm_prepare_database.sh database-type [options] database-name
username password
```

- Tarball install

```
<tarball root>/share/cmf/schema/scm_prepare_database.sh database-type [options]
database-name username password
```

The script prepares the database by:

- Creating the Cloudera Manager Server database configuration file.
- Creating a database for the Cloudera Manager Server to use.
- Setting up a user account for the Cloudera Manager Server.

2. Remove the embedded PostgreSQL properties file if it exists:

- Installer or package install

```
/etc/cloudera-scm-server/db.mgmt.properties
```

- Tarball install

```
<tarball root>/etc/cloudera-scm-server/db.mgmt.properties
```

scm_prepare_database.sh Syntax

```
scm_prepare_database.sh database-type [options] database-name username password
```


 **Note:** You can also run `scm_prepare_database.sh` without options to see the syntax.

Table 5: Required Parameters

Parameter	Description
database-type	One of the supported database types: <ul style="list-style-type: none"> • MySQL - <code>mysql</code> • Oracle - <code>oracle</code> • PostgreSQL - <code>postgresql</code>

Parameter	Description
database-name	The name of the Cloudera Manager Server database to create or use.
username	The username for the Cloudera Manager Server database to create or use.
password	The password for the Cloudera Manager Server database to create or use. If you do not specify the password on the command line, the script prompts you to enter it.

Table 6: Options

Option	Description
-h or --host	The IP address or hostname of the host where the database is installed. The default is to use the local host.
-P or --port	The port number to use to connect to the database. The default port is 3306 for MySQL, 5432 for PostgreSQL, and 1521 for Oracle. This option is used for a remote connection only.
-u or --user	The admin username for the database application. For -u, no space occurs between the option and the provided value. If this option is supplied, the script creates a user and database for the Cloudera Manager Server; otherwise, it uses the user and database you created previously.
-p or --password	The admin password for the database application. The default is no password. For -p, no space occurs between the option and the provided value.
--scm-host	The hostname where the Cloudera Manager Server is installed. Omit if the Cloudera Manager server and the database are installed on the same host.
--config-path	The path to the Cloudera Manager Server configuration files. The default is /etc/cloudera-scm-server.
--schema-path	The path to the Cloudera Manager schema files. The default is /usr/share/cmf/schema (the location of the script).
-f	The script does not stop if an error occurs.
-? or --help	Display help.

Example 1: Running the script when MySQL is installed on another host

This example explains how to run the script on the Cloudera Manager Server host (myhost2) and create and use a temporary MySQL user account to connect to MySQL remotely on the MySQL host (myhost1).

1. At the myhost1 MySQL prompt, create a temporary user who can connect from myhost2:

```
mysql> grant all on *.* to 'temp'@'%' identified by 'temp' with grant option;
Query OK, 0 rows affected (0.00 sec)
```

2. On the Cloudera Manager Server host (myhost2), run the script:

```
$ sudo /usr/share/cmf/schema/scm_prepare_database.sh mysql -h myhost1.sf.cloudera.com
-utemp -ptemp --scm-host myhost2.sf.cloudera.com scm scm scm
Looking for MySQL binary
Looking for schema files in /usr/share/cmf/schema
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/java/jdk1.6.0_31/bin/java -cp
/usr/share/java/mysql-connector-java.jar:/usr/share/cmf/schema/./lib/*
com.cloudera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.properties
com.cloudera.cmf.db.
[ main] DbCommandExecutor INFO Successfully connected to database.
All done, your SCM database is configured correctly!
```

3. On myhost1, delete the temporary user:

```
mysql> drop user 'temp'@'%';
Query OK, 0 rows affected (0.00 sec)
```

Example 2: Running the script to configure Oracle

```
[root@rhel55-6 ~]# /usr/share/cmf/schema/scm_prepare_database.sh -h cm-oracle.example.com
oracle orcl sample_user sample_pass
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/java/jdk1.6.0_31/bin/java -cp
/usr/share/java/mysql-connector-java.jar:/usr/share/cmf/schema/./lib/*
com.cloudera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.properties
com.cloudera.cmf.db.
[ main] DbCommandExecutor INFO Successfully connected to database.
All done, your SCM database is configured correctly!
```

Example 3: Running the script when PostgreSQL is co-located with the Cloudera Manager Server

This example assumes that you have already created the Cloudera Management Server database and database user, naming both `scm`.

```
$ /usr/share/cmf/schema/scm_prepare_database.sh postgresql scm scm scm
```

External Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

You can configure Cloudera Manager to use an external database for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server. If you choose this option, you must create the databases *before* you run the Cloudera Manager installation wizard. For more information, see the instructions in [MySQL Database](#) on page 48, [Oracle Database](#) on page 53, or [External PostgreSQL Database](#) on page 45.

External Databases for Hue and Oozie

Hue and Oozie are automatically configured with databases, but you can configure these services to use external databases after Cloudera Manager is installed.

Configuring an External Database for Hue

By default Hue is configured to use the SQLite database. To use an external database for Hue, see [Using an External Database for Hue](#).

Configuring an External Database for Oozie

By default Oozie is configured to use the Derby database. To use an external database for Oozie, see [Using an External Database for Oozie](#).

Embedded PostgreSQL Database

Installing and Starting the Embedded PostgreSQL Database

This procedure should be used only when creating a demonstration or proof-of-concept deployment. It is *not recommended* for production.

If you are using [Installation Path B - Manual Installation Using Cloudera Manager Packages](#) on page 95 and you want to use an embedded PostgreSQL database for the Cloudera Management Server, use this procedure to install and start the database:

1. Install the embedded PostgreSQL database packages:

OS	Command
Red Hat-compatible, if you have a yum repo configured	<code>\$ sudo yum install cloudera-manager-server-db-2</code>
Red Hat-compatible, if you're transferring RPMs manually	<code>sudo yum --nogpgcheck localinstall cloudera-manager-server-db-2.noarch.rpm</code>
SLES	<code>\$ sudo zypper install cloudera-manager-server-db-2</code>
Ubuntu or Debian	<code>\$ sudo apt-get install cloudera-manager-server-db-2</code>

2. Start the PostgreSQL database:

```
$ sudo service cloudera-scm-server-db start
```

Stopping the Embedded PostgreSQL Database

1. Stop the services that have a dependency on the Hive metastore (Hue, Impala, and Hive) in the following order:
 - Stop the Hue and Impala services.
 - Stop the Hive service.
2. [Stop the Cloudera Management Service.](#)
3. [Stop the Cloudera Manager Server.](#)
4. Stop the Cloudera Manager Server database:

```
sudo service cloudera-scm-server-db stop
```

Changing Embedded PostgreSQL Database Passwords

The embedded PostgreSQL database has generated user accounts and passwords. You can see the generated accounts and passwords during the installation process and you should record them at that time. For example:

Cluster Setup

Database Setup

Configure and test database connections. If using custom databases, create the databases first according to the [Installing and Configuring an External Database](#) section of the [Installation Guide](#) .

Use Custom Databases
 Use Embedded Database

When using the embedded database, passwords are automatically generated. Please copy them down.

Hive

Currently assigned to run on **tcdn53-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	hive	hive	24FLyrjozb

Activity Monitor

Currently assigned to run on **tcdn53-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	amon	amon	2VCic0tDJE

Reports Manager

Currently assigned to run on **tcdn53-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	rman	rman	Mn2l8tEoCH

Navigator Audit Server

Currently assigned to run on **tcdn53-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	nav	nav	P89FAR6e0o

Navigator Metadata Server

Currently assigned to run on **tcdn53-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	navms	navms	29O536GxZp

To find information about the PostgreSQL database account that the Cloudera Manager Server uses, read the `/etc/cloudera-scm-server/db.properties` file:

```
# cat /etc/cloudera-scm-server/db.properties
Auto-generated by scm_prepare_database.sh
#
Sat Oct 1 12:19:15 PDT 201
#
com.cloudera.cmf.db.type=postgresql
com.cloudera.cmf.db.host=localhost:7432
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=TXqEESuhj5
```

To change a password associated with an embedded PostgreSQL database account:

1. Obtain the root password from the `/var/lib/cloudera-scm-server-db/data/generated_password.txt` file:

```
# cat /var/lib/cloudera-scm-server-db/data/generated_password.txt
MnPwGeWaip
```

The password above was generated by `/usr/share/cmf/bin/initialize_embedded_db.sh` (part of the `cloudera-scm-server-db` package) and is the password for the user 'cloudera-scm' for the database in the current directory.

Generated at Fri Jun 29 16:25:43 PDT 2012.

2. On the host on which the Cloudera Manager Server is running, log into PostgreSQL as the root user:

```
psql -U cloudera-scm -p 7432 -h localhost -d postgres
Password for user cloudera-scm: MnPwGeWaip
psql (8.4.18)
Type "help" for help.

postgres=#
```

3. Determine the database and owner names:

```
postgres=# \l
                                List of databases
  Name          | Owner          | Encoding | Collation | Ctype        | Access privileges
-----+-----+-----+-----+-----+-----
 amon           | amon           | UTF8     | en_US.UTF8 | en_US.UTF8   |
 hive          | hive          | UTF8     | en_US.UTF8 | en_US.UTF8   |
 nav           | nav           | UTF8     | en_US.UTF8 | en_US.UTF8   |
 navms         | navms         | UTF8     | en_US.UTF8 | en_US.UTF8   |
 postgres      | cloudera-scm  | UTF8     | en_US.UTF8 | en_US.UTF8   |
 rman          | rman          | UTF8     | en_US.UTF8 | en_US.UTF8   |
 scm           | scm           | UTF8     | en_US.UTF8 | en_US.UTF8   |
 template0     | cloudera-scm  | UTF8     | en_US.UTF8 | en_US.UTF8   | =c/"cloudera-scm"
               |               |         |           |             | :
 "cloudera-scm"=CTc/"cloudera-scm"
 template1     | cloudera-scm  | UTF8     | en_US.UTF8 | en_US.UTF8   | =c/"cloudera-scm"
               |               |         |           |             | :
 "cloudera-scm"=CTc/"cloudera-scm"
(9 rows)
```

4. Set the password for an owner using the `\password` command. For example, to set the password for the `amon` owner, do the following:

```
postgres=# \password amon
Enter new password:
Enter it again:
```

5. Configure the role with the new password:

- In the Cloudera Manager Admin Console, go to the role page.
- Click the **Configuration** tab.
- Expand the **Role Name Default Group > Database** category.
- Set the **Role Name Database Password** property.
- Click **Save Changes** to commit the changes.

External PostgreSQL Database

To use an external PostgreSQL database, follow these procedures.

Installing the External PostgreSQL Server



Note:

- If you already have a PostgreSQL database set up, you can skip to the section [Configuring and Starting the PostgreSQL Server](#) on page 46 to verify that your PostgreSQL configurations meet the requirements for Cloudera Manager.
- Make sure that the data directory, which by default is `/var/lib/postgresql/data/`, is on a partition that has sufficient free space.

1. Use one or more of the following commands to set the locale:

```
export LANGUAGE=en_US.UTF-8
export LANG=en_US.UTF-8
export LC_ALL=en_US.UTF-8
locale-gen en_US.UTF-8
dpkg-reconfigure locales
```

2. Install PostgreSQL packages:

- **Red Hat**

```
$ sudo yum install postgresql-server
```

- **SLES**

```
$ sudo zypper install postgresql91-server
```



Note: This command will install PostgreSQL 9.1. If you want to install a different version, you can use `zypper search postgresql` to search for available versions. You should install version 8.4 or higher.

- **Debian/Ubuntu**

```
$ sudo apt-get install postgresql
```

Configuring and Starting the PostgreSQL Server

By default, PostgreSQL only accepts connections on the loopback interface. You must reconfigure PostgreSQL to accept connections from the Fully Qualified Domain Name (FQDN) of the hosts hosting the management roles. If you do not make these changes, the management processes cannot connect to and use the database on which they depend.

1. Initialize the external PostgreSQL database. For some versions of PostgreSQL, this occurs automatically the first time that you start the PostgreSQL server. In this case, issue the command:

```
$ sudo service postgresql start
```

In other versions, you must explicitly initialize the database using:

```
$ sudo service postgresql initdb
```

See the PostgreSQL documentation for more details.

2. Enable MD5 authentication. Edit `pg_hba.conf`, which is usually found in `/var/lib/pgsql/data` or `/etc/postgresql/8.4/main`. Add the following line:

```
host all all 127.0.0.1/32 md5
```

If the default `pg_hba.conf` file contains the following line:

```
host all all 127.0.0.1/32 ident
```

then the `host` line specifying `md5` authentication shown above must be inserted *before* this `ident` line. Failure to do so may cause an authentication error when running the `scm_prepare_database.sh` script. You can modify the contents of the `md5` line shown above to support different configurations. For example, if you want to access PostgreSQL from a different host, replace `127.0.0.1` with your IP address and update `postgresql.conf`, which is typically found in the same place as `pg_hba.conf`, to include:

```
listen_addresses = '*'
```

3. Configure settings to ensure your system performs as expected. Update these settings in the `/var/lib/pgsql/data/postgresql.conf` or `/var/lib/postgresql/data/postgresql.conf` file. Settings vary based on cluster size and resources as follows:

- Small to mid-sized clusters - Consider the following settings as starting points. If resources are limited, consider reducing the buffer sizes and checkpoint segments further. Ongoing tuning may be required based on each host's resource utilization. For example, if the Cloudera Manager Server is running on the same host as other roles, the following values may be acceptable:
 - `shared_buffers` - 256MB
 - `wal_buffers` - 8MB
 - `checkpoint_segments` - 16
 - `checkpoint_completion_target` - 0.9
- Large clusters - Can contain up to 1000 hosts. Consider the following settings as starting points.
 - `max_connection` - For large clusters, each database is typically hosted on a different host. In general, allow each database on a host 100 maximum connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL, as described at [Connection Settings](#).
 - `shared_buffers` - 1024 MB. This requires that the operating system can allocate sufficient shared memory. See PostgreSQL information on [Managing Kernel Resources](#) for more information on setting kernel resources.
 - `wal_buffers` - 16 MB. This value is derived from the `shared_buffers` value. Setting `wal_buffers` to be approximately 3% of `shared_buffers` up to a maximum of approximately 16 MB is sufficient in most cases.
 - `checkpoint_segments` - 128. The [PostgreSQL Tuning Guide](#) recommends values between 32 and 256 for write-intensive systems, such as this one.
 - `checkpoint_completion_target` - 0.9. This setting is only available in PostgreSQL versions 8.3 and higher, which are highly recommended.

4. Configure the PostgreSQL server to start at boot.

- **Red Hat**

```
$ sudo /sbin/chkconfig postgresql on
$ sudo /sbin/chkconfig --list postgresql
postgresql          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

- **SLES**

```
$ sudo chkconfig --add postgresql
```

- **Debian/Ubuntu**

```
$ sudo chkconfig postgresql on
```

5. Start or restart the PostgreSQL database:

```
$ sudo service postgresql restart
```

Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

Create databases and user accounts for components that require databases:

- If you are not using the [Cloudera Manager installer](#), the Cloudera Manager Server.
- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service)

- Reports Manager

- Each Hive metastore
- Sentry Server
- Cloudera Navigator Audit Server
- Cloudera Navigator Metadata Server

You can create these databases on the host where the Cloudera Manager Server will run, or on any other hosts in the cluster. For performance reasons, you should install each database on the host on which the service runs, as determined by the roles you assign during installation or upgrade. In larger deployments or in cases where database administrators are managing the databases the services use, you can separate databases from services, but use caution.

The database must be configured to support UTF-8 character set encoding.

Record the values you enter for database names, user names, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

1. Connect to PostgreSQL:

```
$ sudo -u postgres psql
```

2. If you are not using the Cloudera Manager installer, create a database for the Cloudera Manager Server. The database name, user name, and password can be any value. Record the names chosen because you will need them later when running the `scm_prepare_database.sh` script.

```
postgres=# CREATE ROLE scm LOGIN PASSWORD 'scm';
postgres=# CREATE DATABASE scm OWNER scm ENCODING 'UTF8';
```

3. Create databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server:

```
postgres=# CREATE ROLE user LOGIN PASSWORD 'password';
postgres=# CREATE DATABASE databaseName OWNER user ENCODING 'UTF8';
```

where *user*, *password*, and *databaseName* can be any value. The examples shown match the default names provided in the Cloudera Manager configuration settings:

Role	Database	User	Password
Activity Monitor	amon	amon	amon_password
Reports Manager	rman	rman	rman_password
Hive Metastore Server	metastore	hive	hive_password
Sentry Server	sentry	sentry	sentry_password
Cloudera Navigator Audit Server	nav	nav	nav_password
Cloudera Navigator Metadata Server	navms	navms	navms_password

For PostgreSQL 8.2.23 or higher, also run:

```
postgres=# ALTER DATABASE Metastore SET standard_conforming_strings = off;
```

MySQL Database

To use a MySQL database, follow these procedures.

Installing the MySQL Server

**Note:**

- If you already have a MySQL database set up, you can skip to the section [Configuring and Starting the MySQL Server](#) on page 49 to verify that your MySQL configurations meet the requirements for Cloudera Manager.
- It is important that the `datadir` directory, which, by default, is `/var/lib/mysql`, is on a partition that has sufficient free space.

1. Install the MySQL database.

OS	Command
RHEL	<code>\$ sudo yum install mysql-server</code>
SLES	<code>\$ sudo zypper install mysql</code> <code>\$ sudo zypper install libmysqlclient_r15</code>
Ubuntu and Debian	<code>\$ sudo apt-get install mysql-server</code>

Note: Some SLES systems encounter errors when using the preceding `zypper install` command. For more information on resolving this issue, see the Novell Knowledgebase topic, [error running chkconfig](#).

After issuing the command to install MySQL, you may need to confirm that you want to complete the installation.

Configuring and Starting the MySQL Server

1. Determine the version of MySQL.
2. Stop the MySQL server if it is running.

OS	Command
RHEL	<code>\$ sudo service mysqld stop</code>
SLES, Ubuntu, and Debian	<code>\$ sudo service mysql stop</code>

3. Move old InnoDB log files `/var/lib/mysql/ib_logfile0` and `/var/lib/mysql/ib_logfile1` out of `/var/lib/mysql/` to a backup location.
4. Determine the location of the [option file](#), `my.cnf`.
5. Update `my.cnf` so that it conforms to the following requirements:
 - To prevent deadlocks, set the isolation level to read committed.
 - Configure the InnoDB engine. Cloudera Manager will not start if its tables are configured with the MyISAM engine. (Typically, tables revert to MyISAM if the InnoDB engine is misconfigured.) To check which engine your tables are using, run the following command from the MySQL shell:

```
mysql> show table status;
```

- The default settings in the MySQL installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the `innodb_flush_method` property to `O_DIRECT`.
- Set the `max_connections` property according to the size of your cluster:
 - Small clusters (fewer than 50 hosts) - You can store more than one database (for example, both the Activity Monitor and Service Monitor) on the same host. If you do this, you should:

- Put each database on its own storage volume.
- Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store five databases on one host (the databases for Cloudera Manager Server, Activity Monitor, Reports Manager, Cloudera Navigator, and Hive metastore), set the maximum connections to 550.
- Large clusters (more than 50 hosts) - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts need not be reserved exclusively for databases, but each database should be on a separate host.
- Binary logging is not a requirement for Cloudera Manager installations. Binary logging provides benefits such as MySQL replication or point-in-time incremental recovery after database restore. Examples of this configuration follow. For more information, see [The Binary Log](#).

Here is an option file with Cloudera recommended settings:

```
[mysqld]
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security risks;
# to do so, uncomment this line:
# symbolic-links = 0

key_buffer_size = 32M
max_allowed_packet = 32M
thread_stack = 256K
thread_cache_size = 64
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M

#log_bin should be on a disk with enough free space. Replace
'/var/lib/mysql/mysql_binary_log' with an appropriate path for your system
#and chown the specified folder to the mysql user.
log_bin=/var/lib/mysql/mysql_binary_log

# For MySQL version 5.1.8 or later. For older versions, reference MySQL documentation
for configuration help.
binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M


# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
innodb_log_file_size = 512M

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

6. If AppArmor is running on the host where MySQL is installed, you might need to configure AppArmor to allow MySQL to write to the binary.
7. Ensure the MySQL server starts at boot.

OS	Command
RHEL	\$ sudo /sbin/chkconfig mysql on \$ sudo /sbin/chkconfig --list mysql

OS	Command
	mysqld 0:off 1:off 2:on 3:on 4:on 5:on 6:off
SLES	\$ sudo chkconfig --add mysql
Ubuntu and Debian	\$ sudo chkconfig mysql on

 **Note:** `chkconfig` may not be available on recent Ubuntu releases. You may need to use Upstart to configure MySQL to start automatically when the system boots. For more information, see the Ubuntu documentation or the [Upstart Cookbook](#).

8. Start the MySQL server:

OS	Command
RHEL	\$ sudo service mysqld start
SLES, Ubuntu, and Debian	\$ sudo service mysql start

9. Set the MySQL root password. In the following example, the current `root` password is blank. Press the **Enter** key when you're prompted for the root password.

```
$ sudo /usr/bin/mysql_secure_installation
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
All done!
```

Installing the MySQL JDBC Driver


Install the JDBC driver on the Cloudera Manager Server host, as well as hosts to which you assign the Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server roles.



Note: If you already have the JDBC driver installed on the hosts that need it, you can skip this section. However, MySQL 5.6 requires a driver version 5.1.26 or higher.

Cloudera recommends that you assign all roles that require databases on the same host and install the driver on that host. Locating all such roles on the same host is recommended but not required. If you install a role, such as Activity Monitor, on one host and other roles on a separate host, you would install the JDBC driver on each host running roles that access the database.

OS	Command
RHEL 5 or 6	1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html .

OS	Command
	<p>2. Extract the JDBC driver JAR file from the downloaded file. For example:</p> <pre>tar zxvf mysql-connector-java-5.1.31.tar.gz</pre> <p>3. Copy the JDBC driver, renamed, to the relevant host. For example:</p> <pre>\$ sudo cp mysql-connector-java-5.1.31/ mysql-connector-java-5.1.31-bin.jar /usr/share/java/ mysql-connector-java.jar</pre> <p>If the target directory does not yet exist on this host, you can create it before copying the JAR file. For example:</p> <pre>\$ sudo mkdir -p /usr/share/java/ \$ sudo cp mysql-connector-java-5.1.31/ mysql-connector-java-5.1.31-bin.jar /usr/share/java/ mysql-connector-java.jar</pre> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> Note: Do not use the <code>yum install</code> command to install the MySQL driver package, because it installs openJDK, and then uses the Linux <code>alternatives</code> command to set the system JDK to be openJDK.</p> </div>
SLES	<pre>\$ sudo zypper install mysql-connector-java</pre>
Ubuntu or Debian	<pre>\$ sudo apt-get install libmysql-java</pre>

Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

Create databases and user accounts for components that require databases:

- If you are not using the [Cloudera Manager installer](#), the Cloudera Manager Server.
- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service)
 - Reports Manager
- Each Hive metastore
- Sentry Server
- Cloudera Navigator Audit Server
- Cloudera Navigator Metadata Server

You can create these databases on the host where the Cloudera Manager Server will run, or on any other hosts in the cluster. For performance reasons, you should install each database on the host on which the service runs, as determined by the roles you assign during installation or upgrade. In larger deployments or in cases where database administrators are managing the databases the services use, you can separate databases from services, but use caution.

The database must be configured to support UTF-8 character set encoding.

Record the values you enter for database names, user names, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

- 1.** Log into MySQL as the root user:

```
$ mysql -u root -p
Enter password:
```

2. Create databases for the Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server:

```
mysql> create database database DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

mysql> grant all on database.* TO 'user'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)
```

database, *user*, and *password* can be any value. The examples match the default names provided in the Cloudera Manager configuration settings:

Role	Database	User	Password
Activity Monitor	amon	amon	amon_password
Reports Manager	rman	rman	rman_password
Hive Metastore Server	metastore	hive	hive_password
Sentry Server	sentry	sentry	sentry_password
Cloudera Navigator Audit Server	nav	nav	nav_password
Cloudera Navigator Metadata Server	navms	navms	navms_password

Backing Up MySQL Databases

To back up the MySQL database, run the `mysqldump` command on the MySQL host, as follows:

```
$ mysqldump -hhostname -uusername -ppassword database > /tmp/database-backup.sql
```

For example, to back up the Activity Monitor database `amon` created in [Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 52, on the local host as the root user, with the password `amon_password`:

```
$ mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database `amon` on remote host `myhost.example.com` as the root user, with the password `amon_password`:

```
$ mysqldump -hmyhost.example.com -uroot -pcloudera amon > /tmp/amon-backup.sql
```

Oracle Database

To use an Oracle database, follow these procedures.

Collecting Oracle Database Information

To configure Cloudera Manager to work with an Oracle database, get the following information from your Oracle DBA:

- Hostname - The DNS name or the IP address of the host where the Oracle database is installed.
- SID - The name of the schema that will store Cloudera Manager information.
- Username - A username for each schema that is storing information. You could have four unique usernames for the four schema.
- Password - A password corresponding to each user name.

Configuring the Oracle Server

Adjusting Oracle Settings to Accommodate Larger Clusters

Cloudera Management services require high write throughput. Depending on the size of your deployments, your DBA may need to modify Oracle settings for monitoring services. These guidelines are for larger clusters and do not apply to the Cloudera Manager configuration database and to smaller clusters. Many factors help determine whether you need to change your database settings, but in most cases, if your cluster has more than 100 hosts, you should consider making the following changes:

- Enable direct and asynchronous I/O by setting the `FILESYSTEMIO_OPTIONS` parameter to `SETALL`.
- Increase the RAM available to Oracle by changing the `MEMORY_TARGET` parameter. The amount of memory to assign depends on the size of the Hadoop cluster.
- Create more redo log groups and spread the redo log members across separate disks or logical unit numbers.
- Increase the size of redo log members to be at least 1 GB.

Modifying the Maximum Number of Oracle Connections

Work with your Oracle database administrator to ensure appropriate values are applied for your Oracle database settings. You must determine the number of connections, transactions, and sessions to be allowed.

Allow 100 maximum connections for each service that requires a database and then add 50 extra connections. For example, for two services, set the maximum connections to 250. If you have five services that require a database on one host (the databases for Cloudera Manager Server, Activity Monitor, Reports Manager, Cloudera Navigator, and Hive metastore), set the maximum connections to 550.

From the maximum number of connections, you can determine the number of anticipated sessions using the following formula:

```
sessions = (1.1 * maximum_connections) + 5
```

For example, if a host has a database for two services, anticipate 250 maximum connections. If you anticipate a maximum of 250 connections, plan for 280 sessions.

Once you know the number of sessions, you can determine the number of anticipated transactions using the following formula:

```
transactions = 1.1 * sessions
```

Continuing with the previous example, if you anticipate 280 sessions, you can plan for 308 transactions.

Work with your Oracle database administrator to apply these derived values to your system.

Using the sample values above, Oracle attributes would be set as follows:

```
alter system set processes=250;  
alter system set transactions=308;  
alter system set sessions=280;
```

Ensure Your Oracle Database Supports UTF8

The database you use must support UTF8 character set encoding. You can implement UTF8 character set encoding in Oracle databases by using the `dbca` utility. In this case, you can use the `characterSet AL32UTF8` option to specify proper encoding. Consult your DBA to ensure UTF8 encoding is properly configured.

Installing the Oracle JDBC Connector

You must install the JDBC connector on the Cloudera Manager Server host and on hosts to which you assign the Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server server roles.

Cloudera recommends that you assign all roles that require a database on the same host and install the connector on that host. Locating all such roles on the same host is recommended but not required. If you install a role, such as Activity Monitor, on one host and other roles on a separate host, you would install the JDBC connector on each host running roles that access the database.

1. Download and install the `ojdbc6.jar` file, which contains the JDBC driver. Download the version that is designed for:
 - Java 6
 - The Oracle database version used in your environment. For example, for an environment using Oracle 11g R2, download the JAR file from <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>.
2. Copy the appropriate JDBC JAR file to `/usr/share/java/oracle-connector-java.jar` for use with the Cloudera Manager databases (for example, for the Activity Monitor, and so on), and for use with Hive.

```
$ mkdir /usr/share/java (if necessary)
$ cp /tmp/ojdbc6.jar /usr/share/java/oracle-connector-java.jar
```

Creating Databases for the Cloudera Manager Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

Create schema and user accounts for components that require databases:

- Cloudera Manager Server (not required if you are using the [Cloudera Manager installer](#))
- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service)
 - Reports Manager
 - Cloudera Navigator Audit Server
 - Cloudera Navigator Metadata Server
- Hive Metastore
- Sentry Server

You can create the Oracle database, schema and users on the host where the Cloudera Manager Server will run, or on any other hosts in the cluster. For performance reasons, you should install each database on the host on which the service runs, as determined by the roles you assign during installation or upgrade. In larger deployments or in cases where database administrators are managing the databases the services use, you can separate databases from services, but use caution.

The database must be configured to support UTF-8 character set encoding.

Record the values you enter for database names, user names, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

1. Log into the Oracle client:

```
sqlplus system@localhost
Enter password: *****
```

2. Create a schema and user for the Cloudera Manager Server:

```
SQL> create user username identified by password;
SQL> grant CREATE SESSION to username;
SQL> grant CREATE ANY TABLE to username;
SQL> grant CREATE ANY SEQUENCE to username;
SQL> grant CREATE ANY INDEX to username;
SQL> grant ALTER ANY TABLE to username;
SQL> grant ALTER ANY INDEX to username;
```

where *username* and *password* are the credentials you specified in [Preparing a Cloudera Manager Server External Database](#) on page 40.

3. Grant a quota on the tablespace (the default tablespace is SYSTEM) where tables will be created:

```
SQL> ALTER USER username quota 100m on tablespace
```

or for unlimited space:

```
SQL> ALTER USER username quota unlimited on tablespace
```

4. Create schema and users for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server: *schema*, *user*, and *password* can be any value. The examples match the default names provided in the Cloudera Manager configuration settings:

Role	Schema	User	Password
Activity Monitor	amon	amon	amon_password
Reports Manager	rman	rman	rman_password
Hive Metastore Server	metastore	hive	hive_password
Sentry Server	sentry	sentry	sentry_password
Cloudera Navigator Audit Server	nav	nav	nav_password
Cloudera Navigator Metadata Server	navms	navms	navms_password

5. For each user in the table in the preceding step, create a user and add privileges for the each user:

```
SQL> create user username identified by password;
SQL> grant CREATE SESSION to username;
SQL> grant CREATE ANY TABLE to username;
SQL> grant CREATE ANY SEQUENCE to username;
SQL> grant CREATE ANY INDEX to username;
SQL> grant ALTER ANY TABLE to username;
SQL> grant ALTER ANY INDEX to username;
```

6. Grant a quota on the tablespace (the default tablespace is SYSTEM) where tables will be created:

```
SQL> ALTER USER username quota 100m on tablespace
```

or for unlimited space:

```
SQL> ALTER USER username quota unlimited on tablespace
```

For further information about Oracle privileges, see [Authorization: Privileges, Roles, Profiles, and Resource Limitations](#).

7. After creating the Cloudera Navigator Audit Server database, set the following additional privileges:

```
GRANT EXECUTE ON sys.dbms_crypto TO nav;
GRANT CREATE VIEW TO nav;
```

where *nav* is the Navigator Audit Server user you specified above when you created the database.

Backing Up Databases

Cloudera recommends that you schedule regular backups of the databases that Cloudera Manager uses to store configuration, monitoring, and reporting data and for managed services that require a database:

- Cloudera Manager - Contains all the information about services you have configured and their role assignments, all configuration history, commands, users, and running processes. This relatively small database (<100 MB) is the most important to back up.



Important: When processes restart, the configuration for each of the services is redeployed using information that is saved in the Cloudera Manager database. If this information is not available, your cluster will not start or function correctly. You must therefore schedule and maintain regular backups of the Cloudera Manager database in order to recover the cluster in the event of the loss of this database.

- Activity Monitor - Contains information about past activities. In large clusters, this database can grow large. Configuring an Activity Monitor database is only necessary if a MapReduce service is deployed.
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.
- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Sentry Server - Contains authorization metadata. Relatively small.
- Cloudera Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large.
- Cloudera Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small.

Backing Up PostgreSQL Databases

To back up a PostgreSQL database, use the same procedure whether the database is embedded or external:

1. Log in to the host where the Cloudera Manager Server is installed.
2. Run the following command as root:

```
cat /etc/cloudera-scm-server/db.properties.
The db.properties file contains:
# Auto-generated by scm_prepare_database.sh
# Mon Jul 27 22:36:36 PDT 2011
com.cloudera.cmf.db.type=postgresql
com.cloudera.cmf.db.host=host:7432
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

3. Run the following command as root using the parameters from the preceding step:

```
# pg_dump -h hostname -p 7432 -U scm > /tmp/scm_server_db_backup.$(date +%Y%m%d)
```

4. Enter the password from the `com.cloudera.cmf.db.password` property in step 2.
5. To back up a database created for one of the roles described in [Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 47, on the local host as the `roleuser` user:

```
# pg_dump -h hostname -p 7432 -U roleuser > /tmp/roledb
```

6. Enter the password specified when the database was created.

Backing Up MySQL Databases

To back up the MySQL database, run the `mysqldump` command on the MySQL host, as follows:

```
$ mysqldump -hhostname -uusername -ppassword database > /tmp/database-backup.sql
```

For example, to back up the Activity Monitor database `amon` created in [Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 52, on the local host as the root user, with the password `amon_password`:

```
$ mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database `amon` on remote host `myhost.example.com` as the root user, with the password `amon_password`:

```
$ mysqldump -hmyhost.example.com -uroot -pcloudera amon > /tmp/amon-backup.sql
```

Backing Up Oracle Databases

For Oracle, work with your database administrator to ensure databases are properly backed up.

Data Storage for Monitoring Data

The Service Monitor and Host Monitor roles in the Cloudera Management Service store time series data, health data, and Impala query and YARN application metadata.

Monitoring Data Migration During Cloudera Manager Upgrade

Cloudera Manager 5 stores Host and Service Monitor data in a local datastore. The Cloudera Manager 4 to Cloudera Manager 5 upgrade wizard automatically migrates data from existing embedded PostgreSQL or external databases to the local datastore. The migration process occurs only once for Host Monitor and Service Monitor, though it can be spread across multiple runs of Host Monitor and Service Monitor if they are restarted before it completes. Resource usage (CPU, memory, and disk) by Host Monitor and Service Monitor are higher than normal during the process.

You can monitor the progress of migrating data from a Cloudera Manager 4 database to the Cloudera Manager 5 datastore in the Host Monitor and Service Monitor [logs](#). Log statements starting with `LDBTimeSeriesDataMigrationTool` identify the upgrade process. The important statements are `Starting DB migration when migration is first started` and `Migration progress: {} total, {} migrated, {} errors` as progress is reported. Progress is reported with partition counts; for example, `3 total, 0 migrated, 0 errors` to start, up to `3 total, 3 migrated, 0 errors` at the end.

After migration completes, the migrated data is summarized in statements such as `Running the LDBTimeSeriesRollupManager at {}, forMigratedData={}` with table names. The external database is never used again by Host Monitor and Service Monitor and the database configurations can be removed (connection information, username, password, and so on).

Configuring Service Monitor Data Storage

The Service Monitor stores time series data and health data, Impala query metadata, and YARN application metadata. By default, the data is stored in `/var/lib/cloudera-service-monitor/` on the Service Monitor host. You can change this by modifying the **Service Monitor Storage Directory** configuration (`firehose.storage.base.directory`). To change this configuration on an active system, see [Moving Monitoring Data on an Active Cluster](#) on page 59.

You can control how much disk space to reserve for the different classes of data the Service Monitor stores by changing the following configuration options:

- Time-series metrics and health data - Time-Series Storage (`firehose_time_series_storage_bytes` - 10 GB default, 10 GB minimum)
- Impala query metadata - Impala Storage (`firehose_impala_storage_bytes` - 1 GB default)
- YARN application metadata - YARN Storage (`firehose_yarn_storage_bytes` - 1 GB default)

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see [Data Granularity and Time-Series Metric Data](#) on page 59.

The default values are small, so you should examine disk usage after several days of activity to determine how much space is needed.

Configuring Host Monitor Data Storage

The Host Monitor stores time series data and health data. By default, the data is stored in `/var/lib/cloudera-host-monitor/` on the Host Monitor host. You can change this by modifying the **Host Monitor Storage Directory** configuration. To change this configuration on an active system, follow the procedure in [Moving Monitoring Data on an Active Cluster](#) on page 59.

You can control how much disk space to reserve for Host Monitor data by changing the following configuration option:

- Time-series metrics and health data: Time Series Storage (`firehose_time_series_storage_bytes` - 10 GB default, 10 GB minimum)

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see [Data Granularity and Time-Series Metric Data](#) on page 59.

The default value is small, so you should examine disk usage after several days of activity to determine how much space they need. The **Charts Library** tab on the Cloudera Management Service page shows the current disk space consumed and its rate of growth, categorized by the type of data stored. For example, you can compare the space consumed by raw metric data to daily summaries of that data.

Viewing Host and Service Monitor Data Storage

The Cloudera Management Service page shows the current disk space consumed and its rate of growth, categorized by the type of data stored. For example, you can compare the space consumed by raw metric data to daily summaries of that data:

1. Select **Clusters > Cloudera Management Service**.
2. Click the **Charts Library** tab.

Data Granularity and Time-Series Metric Data

The Service Monitor and Host Monitor store time-series metric data in a variety of ways. When the data is received, it is written as-is to the metric store. Over time, the raw data is summarized to and stored at various data granularities. For example, after ten minutes, a summary point is written containing the average of the metric over the period as well as the minimum, the maximum, the standard deviation, and a variety of other statistics. This process is summarized to produce hourly, six-hourly, daily, and weekly summaries. This data summarization procedure applies only to metric data. When the Impala query and YARN application monitoring storage limit is reached, the oldest stored records are deleted.

The Service Monitor and Host Monitor internally manage the amount of overall storage space dedicated to each data granularity level. When the limit for a level is reached, the oldest data points at that level are deleted. Metric data for that time period remains available at the lower granularity levels. For example, when an hourly point for a particular time is deleted to free up space, a daily point still exists covering that hour. Because each of these data granularities consumes significantly less storage than the previous summary level, lower granularity levels can be retained for longer periods of time. With the recommended amount of storage, weekly points can often be retained indefinitely.

Some features, such as detailed display of health results, depend on the presence of raw data. Health history is maintained by the event store dictated by its retention policies.

Moving Monitoring Data on an Active Cluster

You can change where monitoring data is stored on a cluster.

Basic: Changing the Configured Directory

1. Stop the Service Monitor or Host Monitor.
2. Save your old monitoring data and then copy the current directory to the new directory (optional).
3. Update the **Storage Directory** configuration option (`firehose.storage.base.directory`) on the corresponding role configuration page.
4. Start the Service Monitor or Host Monitor.

Advanced: High Performance

For the best performance, and especially for a large cluster, Host Monitor and Service Monitor storage directories should have their own dedicated spindles. In most cases, that provides sufficient performance, but you can divide your data further if needed. You cannot configure this directly with Cloudera Manager; instead, you must use symbolic links.

For example, if all your Service Monitor data is located in `/data/1/service_monitor`, and you want to separate your Impala data from your time series data, you could do the following:

1. Stop the Service Monitor.

2. Move the original Impala data in `/data/1/service_monitor/impala` to the new directory, for example `/data/2/impala_data`.
3. Create a symbolic link from `/data/1/service_monitor/impala` to `/data/2/impala_data` with the following command:

```
ln -s /data/2/impala_data /data/1/service_monitor/impala
```

4. Start the Service Monitor.

Host Monitor and Service Monitor Memory Configuration

You can configure Java heap size and non-Java memory size. The memory required or recommended for these configuration options depends on the size of the cluster. In addition to the memory configured, the Host Monitor and Service Monitor use the Linux page cache. Memory available for page caching on the Host Monitor and Service Monitor hosts improves performance.

Table 7: Small Clusters: No More Than 10 Hosts

	Required	Recommended
Java Heap Size	256 MB	512 MB
Non-Java Memory	768 MB	1.5 GB

Table 8: Medium Clusters: Between 11 and 100 Hosts

	Required	Recommended
Java Heap Size	1 GB	2 GB
Non-Java Memory	2 GB	4 GB

Table 9: Large Clusters: More Than 100 Hosts

	Required	Recommended
Java Heap Size	2 GB	4 GB
Non-Java Memory	6 GB	12 GB

Storage Space Planning for Cloudera Manager

Cloudera Manager tracks metrics of services, jobs, and applications in many background processes. All of these metrics require storage. Depending on the size of your organization, this storage may be local or remote, disk-based or in a database, managed by you or by another team in another location.

Most system administrators are aware of common locations like `/var/log/` and the need for these locations to have adequate space. This topic enables you to familiarize yourself with and plan for the storage needs and data storage locations used by the Cloudera Manager Server and the Cloudera Management Service to store metrics and data.

Failing to plan for the storage needs of all components of the Cloudera Manager Server and the Cloudera Management Service can negatively impact your cluster in the following ways:

- The cluster does not have historical operational data to meet internal requirements.
- The cluster is missing critical audit information that was not gathered nor retained for the required length of time.
- Administrators are unable to research past events or health status.
- Administrators do not have historical MR1, YARN, or Impala usage data when they need to reference or report on them later.
- There are gaps in metrics collection and charts.

- The cluster experiences data loss due to filling storage locations to 100% of capacity. The resulting damage from such an event can impact many other components.

There is a main theme here: you need to architect your data storage needs well in advance. You need to inform your operations staff about your critical data storage locations for each host so that they can provision your infrastructure adequately and back it up appropriately. Make sure to document the discovered requirements in your build documentation and run books.

This topic describes both local disk storage and RDBMS storage and these types of storage are labeled within the discussions. This distinction is made both for storage planning and also to inform migration of roles from one host to another, preparing backups, and other lifecycle management events.

The following tables provide details about each individual Cloudera Management service with the goal of enabling Cloudera Manager Administrators to make appropriate storage and lifecycle planning decisions.

Cloudera Manager Server

Table 10: Cloudera Manager Server

Entity	Cloudera Manager Server Configuration
Default Storage Location	<p>RDBMS:</p> <p>Use any supported RDBMS to store the core configuration of your Cloudera Manager database and all cluster, service, and role configurations.</p> <p>See Cloudera Manager and Managed Service Data Stores on page 38.</p> <p>DISK:</p> <p>Cloudera Manager Server Local Data Storage Directory (<code>command_storage_path</code>) on the host where the Cloudera Manager Server is configured to run. This local path is used by Cloudera Manager for storing data, including command result files. Critical configurations are not stored in this location.</p> <p><code>/var/lib/cloudera-scm-server/</code></p>
Storage Configuration Defaults, Minimum, or Maximum	There are no direct storage defaults relevant to this entity.
Where to Control Data Retention or Size	<p>The size of the Cloudera Manager Server database varies depending on the number of managed hosts and the number of discrete commands that have been run in the cluster. To configure the size of the retained command results in the Cloudera Manager Administration Console, select Administration > Settings and edit the following property:</p> <p>Command Eviction Age</p> <p>Length of time after which inactive commands are evicted from the database.</p> <p>Default is two years.</p>
Sizing, Planning & Best Practices	<p>The Cloudera Manager Server database is the most vital configuration store in a Cloudera Manager deployment. This database holds the configuration for clusters, services, roles, and other necessary information that defines a deployment of Cloudera Manager and its managed hosts.</p> <p>You should perform regular, verified, remotely-stored backups of the Cloudera Manager Server database.</p>

Table 11: Cloudera Management Service - Activity Monitor Configuration

Entity	Activity Monitor
Default Storage Location	<p>Any supported RDBMS.</p> <p>See Cloudera Manager and Managed Service Data Stores on page 38.</p>
Storage Configuration Defaults / Minimum / Maximum	Default: 14 Days worth of MRv1 jobs/tasks
Where to Control Data Retention or Size	<p>You control Activity Monitor storage usage by configuring the number of days or hours of data to retain. Older data are purged.</p> <p>To configure data retention in the Cloudera Manager Administration Console, navigate to Home > Cloudera Management Service > Configuration > Activity Monitor and edit the following properties:</p> <p>Purge Activities Data at This Age</p> <p>In Activity Monitor, purge data about MapReduce jobs and aggregate activities when the data reaches this age in hours. By default, Activity Monitor keeps data about activities for 336 hours (14 days).</p> <p>Purge Attempts Data at This Age</p> <p>In the Activity Monitor, purge data about MapReduce attempts when the data reaches this age in hours. Because attempt data may consume large amounts of database space, you may wish to purge it more frequently than activity data. By default, Activity Monitor keeps data about attempts for 336 hours (14 days).</p> <p>Purge MapReduce Service Data at This Age</p> <p>The number of hours of past service-level data to keep in the Activity Monitor database, such as total slots running. The default is to keep data for 336 hours (14 days).</p>
Sizing, Planning, and Best Practices	<p>The Activity Monitor only monitors MapReduce (MRv1) jobs, and does not monitor not YARN applications. If you no longer use MapReduce (MRv1) in your cluster, the Activity Monitor is not required for Cloudera Manager 5 (or later) or CDH 5 (or higher).</p> <p>The amount of storage space needed for 14 days worth of MapReduce (MRv1) activities can vary greatly and directly depends on the size of your cluster and the level of activity that uses MapReduce (MRv1). It may be necessary to adjust and readjust the amount of storage as you determine the "stable state" and "burst state" of the MapReduce activity in your cluster.</p> <p>For example, consider the following test cluster and usage:</p> <ul style="list-style-type: none"> • A simulated 1000-host cluster, each host with 32 slots • Synthetic MapReduce (MRv1) jobs with 200 attempts (tasks) per activity (job) <p>Sizing observations for this cluster:</p> <ul style="list-style-type: none"> • Each attempt takes 10 minutes to complete. • This usage results in roughly 20 thousand jobs a day with some 5 million total attempts. • For a retention period of 7 days, this Activity Monitor database required 200 GB.

Table 12: Cloudera Management Service - Service Monitor Configuration

Entity	Service Monitor Configuration
Default Storage Location	/var/lib/cloudera-service-monitor/ on the host where the Service Monitor role is configured to run.
Storage Configuration Defaults / Minimum / Maximum	<ul style="list-style-type: none"> • 10 GiB Services Time Series Storage + • 1 GiB Impala Query Storage + • 1 GiB YARN Application Storage <p>Total: ~12 GiB Minimum (No Maximum)</p>
Where to Control Data Retention or Size	<p>Service Monitor data growth is controlled by configuring the maximum amount of storage space it may use.</p> <p>To configure data retention in Cloudera Manager Administration Console, select Home > Cloudera Management Service > Configuration > Service Monitor Default Group and edit the following properties:</p> <p>Time-Series Storage</p> <p>The approximate amount of disk space dedicated to storing time series and health data. When the store has reached its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data once it reaches the limit.</p> <p>Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. The Service Monitor stores metric data not only as raw data points but also as ten-minute, hourly, six-hourly, daily, and weekly summary data points. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time while weekly summary points are unlikely to ever be deleted.</p> <p>Select Cloudera Management Service > Charts Library tab in Cloudera Manager for information about how space is consumed within the Service Monitor. These pre-built charts also show information about the amount of data retained and time window covered by each data granularity.</p> <p>Impala Storage</p> <p>The approximate amount of disk space dedicated to storing Impala query data. When the store reaches its maximum size, it deletes older to make room for newer queries. The disk usage is approximate because the store only begins deleting data when it reaches the limit.</p> <p>YARN Storage</p> <p>The approximate amount of disk space dedicated to storing YARN application data. Once the store reaches its maximum size, it deletes older data to make room for newer applications. The disk usage is approximate because Cloudera Manager only begins deleting data when it reaches the limit.</p>
Sizing, Planning, and Best Practices	The Service Monitor gathers metrics about configured roles and services in your cluster and also runs active health tests. These health tests run regardless of idle and use periods, because they are always relevant. The Service Monitor gathers metrics and health test results regardless of the level of activity in the cluster. This data continues to grow, even in an idle cluster.

Table 13: Cloudera Management Service - Host Monitor

Entity	Host Monitor
Default Storage Location	<code>/var/lib/cloudera-host-monitor/</code> on the hoist where the Host Monitor role is configured to run
Storage Configuration Defaults / Minimum/ Maximum	Default + Minimum: 10 GiB Host Time Series Storage
Where to Control Data Retention or Size	<p>Host Monitor data growth is controlled by configuring the maximum amount of storage space it may use.</p> <p>See Data Storage for Monitoring Data on page 58.</p> <p>To configure these data retention in Cloudera Manager Administration Console, select : Home > Cloudera Management Service > Configuration > Host Monitor Default Group</p> <p>Time-Series Storage</p> <p>The approximate amount of disk space dedicated to storing time series and health data. When the store reaches its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data when it reaches the limit.</p> <p>Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. Host Monitor stores metric data not only as raw data points but also ten-minutely, hourly, six-hourly, daily, and weekly summary data points. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time, while weekly summary points are unlikely to ever be deleted.</p> <p>See the Cloudera Management Service > Charts Library tab in Cloudera Manager for information on how space is consumed within the Host Monitor. These pre-built charts also show information about the amount of data retained and the time window covered by each data granularity.</p>
Sizing, Planning and Best Practices	The Host Monitor gathers metrics about host-level items of interest (for example: disk space usage, RAM, CPU usage, swapping, etc) and also informs host health tests. The Host Monitor gathers metrics and health test results regardless of the level of activity in the cluster. This data continues to grow fairly linearly, even in an idle cluster.

Table 14: Cloudera Management Service - Event Server

Entity	Event Server
Default Storage Location	<code>/var/lib/cloudera-scm-eventserver/</code> on the host where the Event Server role is configured to run
Storage Configuration Defaults	5,000,000 events retained
Where to Control Data Retention or Minimum /Maximum	<p>The amount of storage space the Event Server uses is influenced by configuring how many discrete events it may retain.</p> <p>To configure data retention in Cloudera Manager Administration Console, select Home > Cloudera Management Service > Configuration > Event Server Group and edit the following property:</p>


Entity	Event Server
	<p>Maximum Number of Events in the Event Server Store</p> <p>The maximum size of the Event Server store, in events. Once this size is exceeded, events are deleted starting with the oldest first until the size of the store is below this threshold</p>
Sizing, Planning, and Best Practices	<p>The Event Server is a managed Lucene index that collects relevant events that happen within your cluster, such as results of health tests, log events that are created when a log entry matches a set of rules for identifying messages of interest and makes them available for searching, filtering and additional action. You can view and filter events on the Diagnostics > Events tab of the Cloudera Manager Administration Console. You can also poll this data using the Cloudera Manager API.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Note: The Management Service role named Alert Publisher sources all the content for its work by regularly polling the Event Server for entries that are marked to be sent out via SNMP or SMTP(S). The Alert Publisher is not discussed because it has no noteworthy storage requirements of its own.</p> </div>

Table 15: Cloudera Management Service - Reports Manager

Entity	Reports Manager
Default Storage Location	<p>RDBMS:</p> <p>Any Supported RDBMS.</p> <p>See Installing and Configuring Databases.</p> <p>DISK:</p> <p><code>/var/lib/cloudera-scm-headlamp/</code> on the host where the Reports Manager role is configured to run.</p>
Storage Configuration Defaults	<p>RDBMS:</p> <p>There are no exposed defaults or configurations to directly cull or purge the size of this data set.</p> <p>DISK:</p> <p>There are no configuration defaults to influence the size of this location. The size of the data in this location depends not only on the size of the HDFS fsimage, but also on the HDFS path complexity.</p>
Where to Control Data Retention or Minimum / Maximum	<p>The Reports Manager uses space in two main locations, one local on the host where Reports Manager runs, and the other in the RDBMS provided to it for its historical aggregation. The RDBMS is not required to be on the same host where the Reports Manager runs.</p>
Sizing, Planning, and Best Practices	<p>Reports Manager downloads the fsimage from the Namenode every 60 minutes (default) and stores it locally to perform operations against, including indexing the HDFS filesystem structure represented in the fsimage. A larger fsimage, or more deep and complex paths within HDFS consume more disk space.</p> <p>Reports Manager has no control over the size of the fsimage. If your total HDFS usage trends upward notably or you add excessively long paths in HDFS, it may be necessary to revisit and adjust the amount of space allocated to the Reports</p>

Entity	Reports Manager
	Manager for its local storage. Periodically monitor, review and readjust the local storage allocation.

Cloudera Navigator

Table 16: Cloudera Navigator - Navigator Audit Server

Entity	Navigator Audit Server
Default Storage Location	Any Supported RDBMS. See Installing and Configuring Databases .
Storage Configuration Defaults	Default: 90 Days retention
Where to Control Data Retention or Min/Max	<p>Navigator Audit Server storage usage is controlled by configuring how many days of data it may retain. Any older data are purged.</p> <p>To configure data retention in the Cloudera Manager Administration Console, select Home > Cloudera Management Service > Configuration > Navigator Audit Server Default Group and edit the following property:</p> <p>Navigator Audit Server Data Expiration Period</p> <p>In Navigator Audit Server, purge audit data of various auditable services when the data reaches this age in days. By default, Navigator Audit Server keeps data about audits for 90 days.</p>
Sizing, Planning, and Best Practices	<p>The size of the Navigator Audit Server database directly depends on the number of audit events the cluster's audited services generate. Normally the volume of HDFS audits exceed the volume of other audits (all other components like MRv1, Hive and Impala read from HDFS, which generates additional audit events).</p> <p>The average size of a discrete HDFS audit event is ~1 KB. For a busy cluster of 50 hosts with ~100K audit events generated per hour, the Navigator Audit Server database would consume ~2.5 GB per day. To retain 90 days of audits at that level, plan for a database size of around 250 GB. If other configured cluster services generate roughly the same amount of data as the HDFS audits, plan for the Navigator Audit Server database to require around 500 GB of storage for 90 days of data.</p> <p>Notes:</p> <ul style="list-style-type: none"> • Individual Hive and Impala queries themselves can be very large. Since the query itself is part of an audit event, such audit events consume space in proportion to the length of the query. • The amount of space required increases as activity on the cluster increases. In some cases, Navigator Audit Server databases can exceed 1TB for 90 days of audit events. Benchmark your cluster periodically and adjust accordingly. <p>Use this table to map Product Compatibility Matrix for Cloudera Navigator versions to Cloudera Manager versions.</p>

Table 17: Cloudera Navigator - Navigator Metadata Server

Entity	Navigator Metadata Server
Default Storage Location	RDBMS:

Entity	Navigator Metadata Server
	<p>Any Supported RDBMS.</p> <p>See Installing and Configuring Databases.</p> <p>DISK:</p> <p><code>/var/lib/cloudera-scm-navigator/</code> on the host where the Navigator Metadata Server role is configured to run,</p>
Storage Configuration Defaults	<p>RDBMS:</p> <p>There are no exposed defaults or configurations to directly cull or purge the size of this data set.</p> <p>DISK:</p> <p>There are no configuration defaults to influence the size of this location. You can change the location itself with the Navigator Metadata Server Storage Dir property. The size of the data in this location depends on the amount of metadata in the system (HDFS fsimage size, Hive Metastore size) and activity on the system (the number of MapReduce Jobs run, Hive queries executed, etc).</p>
Where to Control Data Retention or Min/Max	<p>RDBMS:</p> <p>There is no maximum size of this data and no way to purge data that is old.</p> <p>DISK:</p> <p>There is no maximum size of this data. As data in the cluster grows its metadata is captured and stored in the location specified by the Navigator Metadata Server Storage Dir property.</p>
Sizing, Planning, and Best Practices	<p>RDBMS:</p> <p>The database is used to store policies and authorization data. The dataset is small, but this database is also used during a Solr schema upgrade, where Solr documents are extracted and inserted again in Solr. This has same space requirements as above use case, but the space is only used temporarily during product upgrades.</p> <p>Use this matrix to map Cloudera Navigator and Cloudera Manager versions.</p> <p>DISK:</p> <p>This filesystem location contains all the metadata that is extracted from managed clusters. The data is stored in Solr, so this is the location where Solr stores its index and documents. Depending on the size of the cluster, this data can occupy tens of gigabytes. A guideline is to look at the size of HDFS fsimage and allocate two to three times that size as the initial size. The data here is incremental and continues to grow as activity is performed on the cluster. The rate of growth can be on order of tens of megabytes per day.</p>

General Performance Notes

When possible:

- For entities that use an RDBMS, install the database on the same host as the service.
- Provide a dedicated spindle to the RDBMS or datastore data directory to avoid disk contention with other read/write activity.

Cluster Lifecycle Management with Cloudera Manager

Cloudera Manager clusters that use parcels to provide CDH and other components require adequate disk space in the following locations:

Table 18: Parcel Lifecycle Management

Parcel Lifecycle Path (default)	Notes
Local Parcel Repository Path <code>/opt/cloudera/parcel-repo</code>	<p>This path exists only on the host where Cloudera Manager Server (cloudera-scm-server) runs. The Cloudera Manager Server stages all new parcels in this location as it fetches them from any external repositories. Cloudera Manager Agents are then instructed to fetch the parcels from this location when the administrator distributes the parcel via the Cloudera Manager Administration Console or the Cloudera Manager API.</p> <p>Sizing and Planning</p> <p>The default location is <code>/opt/cloudera/parcel-repo</code> but you can configure another local filesystem location on the host where Cloudera Manager Server runs. See Parcel Configuration Settings on page 78.</p> <p>Provide sufficient space to hold all the parcels you download from all configured Remote Parcel Repository URLs (See Parcel Configuration Settings on page 78). Cloudera Manager deployments that manage multiple clusters store all applicable parcels for all clusters.</p> <p>Parcels are provided for each operating system, so be aware that heterogeneous clusters (distinct operating systems represented in the cluster) require more space than clusters with homogeneous operating systems.</p> <p>For example, a cluster with both RHEL5.x and 6.x hosts must hold -el5 and -el6 parcels in the Local Parcel Repository Path, which requires twice the amount of space.</p> <p>Lifecycle Management and Best Practices</p> <p>Delete any parcels that are no longer in use from the Cloudera Manager Administration Console, (never delete them manually from the command line) to recover disk space in the Local Parcel Repository Path and simultaneously across all managed cluster hosts which hold the parcel.</p> <p>Backup Considerations</p> <p>Perform regular backups of this path, and consider it a non-optional accessory to backing up Cloudera Manager Server. If you migrate Cloudera Manager Server to a new host or restore it from a backup (for example, after a hardware failure), recover the full content of this path to the new host, in the <code>/opt/cloudera/parcel-repo</code> directory before starting any <code>cloudera-scm-agent</code> or <code>cloudera-scm-server</code> processes.</p>
Parcel Cache <code>/opt/cloudera/parcel-cache</code>	<p>Managed Hosts running a Cloudera Manager Agent stage distributed parcels into this path (as <code>.parcel</code> files, unextracted). Do not manually manipulate this directory or its files.</p> <p>Sizing and Planning</p> <p>Provide sufficient space per-host to hold all the parcels you distribute to each host.</p> <p>You can configure Cloudera Manager to remove these cached <code>.parcel</code> files after they are extracted and placed in <code>/opt/cloudera/parcels/</code>. It is not mandatory to keep these temporary files but keeping them avoids the need to transfer the</p>

Parcel Lifecycle Path (default)	Notes
	<p>.parcel file from the Cloudera Manager Server repository should you need to extract the parcel again for any reason.</p> <p>To configure this behavior in the Cloudera Manager Administration Console, select Administration > Settings > Parcels > Retain Downloaded Parcel Files</p>
<p>Host Parcel Directory</p> <p>/opt/cloudera/parcels</p>	<p>Managed cluster hosts running a Cloudera Manager Agent extract parcels from the /opt/cloudera/parcel-cache directory into this path upon parcel activation. Many critical system symlinks point to files in this path and you should never manually manipulate its contents.</p> <p>Sizing and Planning</p> <p>Provide sufficient space on each host to hold all the parcels you distribute to each host. Be aware that the typical CDH parcel size is slightly larger than 1 GB per parcel. If you maintain various versions of parcels staged before and after upgrading, be aware of the disk space implications.</p> <p>You can configure Cloudera Manager to automatically remove older parcels once they are no longer in use. As an administrator you can always manually delete parcel versions not in use, but configuring these settings can handle the deletion automatically, in case you forget.</p> <p>To configure this behavior in the Cloudera Manager Administration Console, select Administration > Settings > Parcels and configure the following property:</p> <p>Automatically Remove Old Parcels</p> <p>This parameter controls whether parcels for old versions of an activated product should be removed from a cluster when they are no longer in use.</p> <p>The default value is Disabled.</p> <p>Number of Old Parcel Versions to Retain</p> <p>If you enable Automatically Remove Old Parcels, this setting specifies the number of old parcels to keep. Any old parcels beyond this value are removed. If this property is set to zero, no old parcels are retained.</p> <p>The default value is 3.</p>

Table 19: Management Service Lifecycle - Space Reclamation Tasks

Task	Description
Activity Monitor (One-time)	The Activity Monitor only works against a MapReduce (MR1) service, not YARN. So if your deployment has fully migrated to YARN and no longer uses a MapReduce (MR1) service, your Activity Monitor database is no longer growing. If you have waited longer than the default Activity Monitor retention period (14 days) to address this point, then the Activity Monitor has already purged it all for you and your database is mostly empty. If your deployment meets these conditions, consider cleaning up by dropping the Activity Monitor database (again, only when you are satisfied that you no longer need the data or have confirmed that it is no longer in use) and the Activity Monitor role.
Service Monitor and Host Monitor (One-time)	For those who used Cloudera Manager version 4.x and have now upgraded to version 5.x: The Service Monitor and Host Monitor were migrated from their previously-configured RDBMS into a dedicated time series store used solely by each of these roles respectively. After this happens, there is still legacy database

Task	Description
	<p>connection information in the configuration for these roles. This was used to allow for the initial migration but is no longer being used for any active work.</p> <p>After the above migration has taken place, the RDBMS databases previously used by the Service Monitor and Host Monitor are no longer used. Space occupied by these databases is now recoverable. If appropriate in your environment (and you are satisfied that you have long-term backups or do not need the data on disk any longer), you can drop those databases.</p>
Ongoing Space Reclamation	<p>Cloudera Management Services are automatically rolling up, purging or otherwise consolidating aged data for you in the background. Configure retention and purging limits per-role to control how and when this occurs. These configurations are discussed per-entity above. Adjust the default configurations to meet your space limitations or retention needs.</p>

Conclusion

Keep this information in mind for planning and architecting the deployment of a cluster managed by Cloudera Manager. If you already have a live cluster, find lifecycle and backup information that can help you keep critical monitoring, auditing and metadata sources safe and properly backed up.

Managing Software Installation

A major function of Cloudera Manager is to install CDH and managed service software in your cluster. Cloudera Manager supports two software distribution formats: packages and parcels.

A **package** is a binary distribution format that contains compiled code and meta-information such as a package description, version, and dependencies. Package management systems evaluate this meta-information to allow package searches, perform upgrades to a newer version, and ensure that all dependencies of a package are fulfilled. Cloudera Manager uses the native "system package manager" for each supported OS.

A **parcel** is a binary distribution format containing the program files, along with additional metadata used by Cloudera Manager. There are a few notable differences between parcels and packages:

- Parcels are self-contained and installed in a versioned directory, which means that multiple versions of a given parcel can be installed side-by-side. You can then designate one of these installed versions as the active one. With packages, only one package can be installed at a time so there's no distinction between what's installed and what's active.
- Parcels can be installed at any location in the filesystem and by default are installed in `/opt/cloudera/parcels`. In contrast, packages are installed in `/usr/lib`.
- Parcel handling automatically downloads, distributes, and activates the correct parcel for the operating system running on each host in the cluster. All CDH and Cloudera Manager hosts that make up a logical cluster need to run on the same major OS release to be covered by Cloudera Support.



Note: If you choose to install CDH manually using these instructions, you cannot use Cloudera Manager to install additional parcels, you must use the packages option in Cloudera Manager. See [Managing Software Installation](#) on page 70.

Parcels

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

A **parcel** is a binary distribution format containing the program files, along with additional metadata used by Cloudera Manager. There are a few notable differences between parcels and packages:

- Parcels are self-contained and installed in a versioned directory, which means that multiple versions of a given parcel can be installed side-by-side. You can then designate one of these installed versions as the active one. With

packages, only one package can be installed at a time so there's no distinction between what's installed and what's active.

- Parcels can be installed at any location in the filesystem and by default are installed in `/opt/cloudera/parcels`. In contrast, packages are installed in `/usr/lib`.
- Parcel handling automatically downloads, distributes, and activates the correct parcel for the operating system running on each host in the cluster. All CDH and Cloudera Manager hosts that make up a logical cluster need to run on the same major OS release to be covered by Cloudera Support.

Parcels are available for CDH 4.1.3 or later, and for Impala, Search, Spark, Accumulo, Kafka, Key Trustee KMS, and Sqoop Connectors.



Important: You cannot install software using both parcels and packages in the same cluster.

Advantages of Parcels

As a consequence of their unique properties, parcels offer a number of advantages over packages:

- **CDH is distributed as a single object** - In contrast to having a separate package for each part of CDH, when using parcels there is just a single object to install. This is especially useful when managing a cluster that isn't connected to the Internet.
- **Internal consistency** - All CDH components are matched so there isn't a danger of different parts coming from different versions of CDH.
- **Installation outside of `/usr`** - In some environments, Hadoop administrators do not have privileges to install system packages. In the past, these administrators had to fall back to CDH tarballs, which deprived them of a lot of infrastructure that packages provide. With parcels, administrators can install to `/opt` or anywhere else without having to step through all the additional manual steps of regular tarballs.



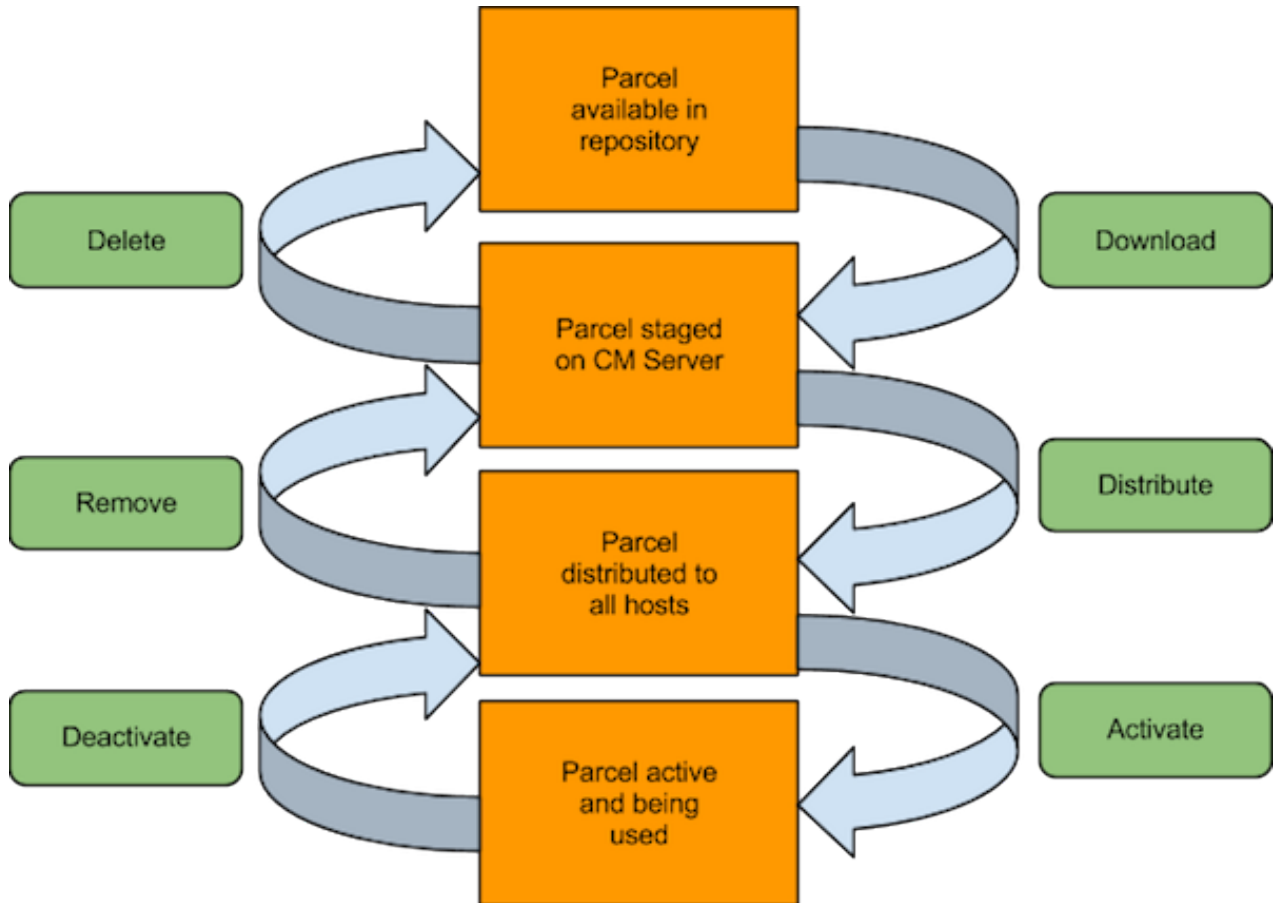
Note: With parcel software distribution, the path to the CDH libraries is `/opt/cloudera/parcels/CDH/lib` instead of the usual `/usr/lib`. You should not link `/usr/lib/` elements to parcel deployed paths, as such links may confuse scripts that distinguish between the two paths.

- **Installation of CDH without `sudo`** - Parcel installation is handled by the Cloudera Manager Agent running as root so it's possible to install CDH without needing `sudo`.
- **Decouples distribution from activation** - Due to side-by-side install capabilities, it is possible to stage a new version of CDH across the cluster in advance of switching over to it. This allows the longest running part of an upgrade to be done ahead of time without affecting cluster operations, consequently reducing the downtime associated with upgrade.
- **Rolling upgrades** - These are only possible with parcels, due to their side-by-side nature. Packages require shutting down the old process, upgrading the package, and then starting the new process. This can be hard to recover from in the event of errors and requires extensive integration with the package management system to function seamlessly. When a new version is staged side-by-side, switching to a new minor version is simply a matter of changing which version of CDH is used when restarting each process. It then becomes practical to do upgrades with [rolling restarts](#), where service roles are restarted in the right order to switch over to the new version with minimal service interruption. Your cluster can continue to run on the existing installed components while you stage a new version across your cluster, without impacting your current operations. Note that major version upgrades (for example, CDH 4 to CDH 5) require full service restarts due to the substantial changes between the versions. Finally, you can upgrade individual parcels, or multiple parcels at the same time.
- **Upgrade management** - Cloudera Manager can fully manage all the steps involved in a CDH version upgrade. In contrast, with packages, Cloudera Manager can only help with initial installation.
- **Distributing additional components** - Parcels are not limited to CDH. Cloudera Impala, Cloudera Search, LZO, and [add-on service](#) parcels are also available.
- **Compatibility with other distribution tools** - If there are specific reasons to use other tools for download and/or distribution, you can do so, and Cloudera Manager will work alongside your other tools. For example, you can

handle distribution with Puppet. Or, you can download the parcel to Cloudera Manager Server manually (perhaps because your cluster has no Internet connectivity) and then have Cloudera Manager distribute the parcel to the cluster.

Parcel Life Cycle

To enable upgrades and additions with minimal disruption, parcels participate in six phases: download, distribute, activate, deactivate, remove, and delete.



- **Downloading** a parcel copies the appropriate software to a local parcel repository on the Cloudera Manager Server, where it is available for distribution to the other hosts in any of your clusters managed by this Cloudera Manager Server. You can have multiple parcels for a given product downloaded to your Cloudera Manager Server. Once a parcel has been downloaded to the Server, it will be available for distribution on all clusters managed by the Server. A downloaded parcel will appear in the cluster-specific section for every cluster managed by this Cloudera Manager Server.
- **Distributing** a parcel copies the parcel to the member hosts of a cluster and unpacks it. Distributing a parcel does not actually upgrade the components running on your cluster; the current services continue to run unchanged. You can have multiple parcels distributed on your cluster.



Note: The distribute process does not require Internet access; rather the Cloudera Manager Agent on each cluster member downloads the parcels from the local parcel repository on the Cloudera Manager Server.

- **Activating** a parcel causes the Cloudera Manager to link to the new components, ready to run the new version *upon the next restart*. Activation does not automatically stop the current services or perform a restart — you have the option to restart the service(s) after activation, or you can allow the system administrator to determine the appropriate time to perform those operations.

- **Deactivating** a parcel causes Cloudera Manager to unlink from the parcel components. A parcel cannot be deactivated while it is still in use on one or more hosts.
- **Removing** a parcel causes Cloudera Manager to remove the parcel components from the hosts.
- **Deleting** a parcel causes Cloudera Manager to remove the parcel components from the local parcel repository.

For example, the following screenshot:

Parcels

The screenshot displays the Cloudera Manager interface for managing parcels. It is divided into two main sections: 'Downloadable' and 'Cluster 1 - CDH4'.

Downloadable Section:

- IMPALA 1.1.1-1.p0.17:** This parcel is currently being downloaded. The progress bar shows 11% completion. A 'Cancel' button is visible below the progress bar.

Cluster 1 - CDH4 Section:

- CDH 4.4.0-1.cd4.4.0.p0.39:** This parcel is being distributed. The progress bar shows 0% completion. A 'Cancel' button is visible below the progress bar.
- CDH 4.3.0-1.cd4.3.0.p0.22:** This parcel is already activated. An 'Actions' dropdown menu is visible below the parcel name.
- SOLR 1.0.0-1.cd4.3.0.p0.4:** This parcel is distributed and ready to be activated. An 'Activate' button is visible below the parcel name. A badge with the numbers 1, 2, and 3 is located at the bottom of this parcel's card.

shows:

- One activated CDH parcel
- One SOLR parcel distributed and ready to activate
- One Impala parcel being downloaded
- One CDH parcel being distributed

Cloudera Manager detects when new parcels are available. The parcel indicator in the Admin Console navigation bar (📦) indicates how many parcels are eligible for downloading or distribution. For example, CDH parcels older than the active one do not contribute to the count if you are already using the latest version. If no parcels are eligible, or if all parcels have been activated, then the indicator will not have a number badge. You can configure Cloudera Manager to download and distribute parcels automatically, if desired.



Important: If you plan to upgrade CDH you should follow the instructions in [Upgrading CDH and Managed Services Using Cloudera Manager](#) because steps in addition to activating the parcel must be performed in order to successfully upgrade.

Parcel Locations

The default location for the local parcel directory on the Cloudera Manager Server host is `/opt/cloudera/parcel-repo`. To change this location, follow the instructions in [Configuring Cloudera Manager Server Parcel Settings](#) on page 78.


The default location for the distributed parcels on the managed hosts is `/opt/cloudera/parcels`. To change this location, set the `parcel_dir` property in `/etc/cloudera-scm-agent/config.ini` file of the Cloudera Manager Agent and restart the Cloudera Manager Agent or by following the instructions in [Configuring the Host Parcel Directory](#) on page 79.



Note: With parcel software distribution, the path to the CDH libraries is `/opt/cloudera/parcels/CDH/lib` instead of the usual `/usr/lib`. You should not link `/usr/lib/` elements to parcel deployed paths, as such links may confuse scripts that distinguish between the two paths.

Managing Parcels

Through the Parcels interface in Cloudera Manager, you can determine what software versions are running across your clusters. You access the Parcels page by doing one of the following:


- Clicking the parcel indicator in the Admin Console navigation bar 
- Clicking the **Hosts** in the top navigation bar, then the **Parcels** tab.

The Parcels page is divided into several sections. The top section, labeled **Downloadable**, shows you all the parcels that are available for download from the configured parcel repositories.

Below the Downloadable section, each cluster managed by this Cloudera Manager Server has a section that shows the parcels that have been downloaded, distributed, or activated on that cluster.

When you download a parcel, it appears under every cluster, if you are managing more than one. However, this just indicates that the parcel is available for distribution on those clusters — in fact there is only one copy of the downloaded parcel, residing on the Cloudera Manager Server. Only after you distribute the parcel to a cluster will copies of it be placed on the hosts in that cluster.

Downloading a Parcel

1. Click the parcel indicator  in the top navigation bar. This takes you to the **Hosts** page, **Parcels** tab. By default, any parcels available for download are shown in the **Downloadable** section of the Parcels page. Parcels available for download will display a **Download** button.

If the parcel you want is not shown here — for example, you want to upgrade to version of CDH that is not the most current version — you can make additional remote parcel repositories available through the Administration Settings page. You can also configure the location of the local parcel repository and other settings. See [Parcel Configuration Settings](#) on page 78.

2. Click **Download** to initiate the download of the parcel from the remote parcel repository to your local repository.

When the parcel has been downloaded, the button label changes to Distribute.



Note: The parcel download is done at the Cloudera Manager Server, so with multiple clusters, the downloaded parcels are shown as available to *all* clusters managed by the Cloudera Manager Server. However, distribution (to a specific cluster's hosts) must be selected on a cluster-by-cluster basis.

Distributing a Parcel

Parcels that have been downloaded can be distributed to the hosts in your cluster, available for activation.

From the Parcels tab, click the **Distribute** button for the parcel you want to distribute. This starts the distribution process to the hosts in the cluster.

Distribution does not require Internet access; rather the Cloudera Manager Agent on each cluster member downloads the parcel from the local parcel repository hosted on the Cloudera Manager Server.

If you have a large number of hosts to which the parcels should be distributed, you can control how many concurrent uploads Cloudera Manager will perform. You can configure this setting on the **Administration** page, **Properties** tab under the Parcels section.

You can delete a parcel that is ready to be distributed; click the triangle at the right end of the Distribute button to access the Delete command. This will delete the downloaded parcel from the local parcel repository.

Distributing parcels to the hosts in the cluster does not affect the current running services.

Activating a Parcel

Parcels that have been distributed to the hosts in a cluster are ready to be activated.

1. From the Parcels tab, click the **Activate** button for the parcel you want to activate. This will update Cloudera Manager to point to the new software, ready to be run the next time a service is restarted.
2. A pop-up warns you that your currently running process will not be affected until you restart, and gives you the option to perform a restart. If you do not want to restart at this time, click **Close**.

If you elect not to restart services as part of the Activation process, you can instead go to the **Clusters** tab and restart your services at a later time. Until you restart services, the current software will continue to run. This allows you to restart your services at a time that is convenient based on your maintenance schedules or other considerations.

Activating a new parcel also deactivates the previously active parcel (if any) for the product you've just upgraded. However, until you restart the services, the previously active parcel will have the link **Still in use** and you will not be able to remove the parcel until it is no longer being used.



Note: Under some situations, such as doing a major release upgrade (for example, CDH 4 to CDH 5) additional upgrade steps may be necessary. In this case, instead of **Activate**, the button may instead say **Upgrade**. This indicates that there may be additional steps involved in the upgrade.

Deactivating a Parcel

You can deactivate an active parcel; this will update Cloudera Manager to point to the previous software version, ready to be run the next time a service is restarted. To deactivate a parcel, click **Actions** on an activated parcel and select **Deactivate**.

To use the previous version of the software, go to the **Clusters** tab and restart your services.



Note: If you did your original installation from parcels, and there is only one version of your software installed (that is, no packages, and no previous parcels have been activated and started) then when you attempt to restart after deactivating the current version, your roles will be stopped but will not be able to restart.

Removing a Parcel

To remove a parcel, click the down arrow to the right of an **Activate** button and select **Remove from Hosts**.

Deleting a Parcel

To delete a parcel, click the down arrow to the right of a **Distribute** button and select **Delete**.


Troubleshooting

If you experience an error while performing parcel operations, click on the red 'X' icons on the parcel page to display a message that will identify the source of the error.

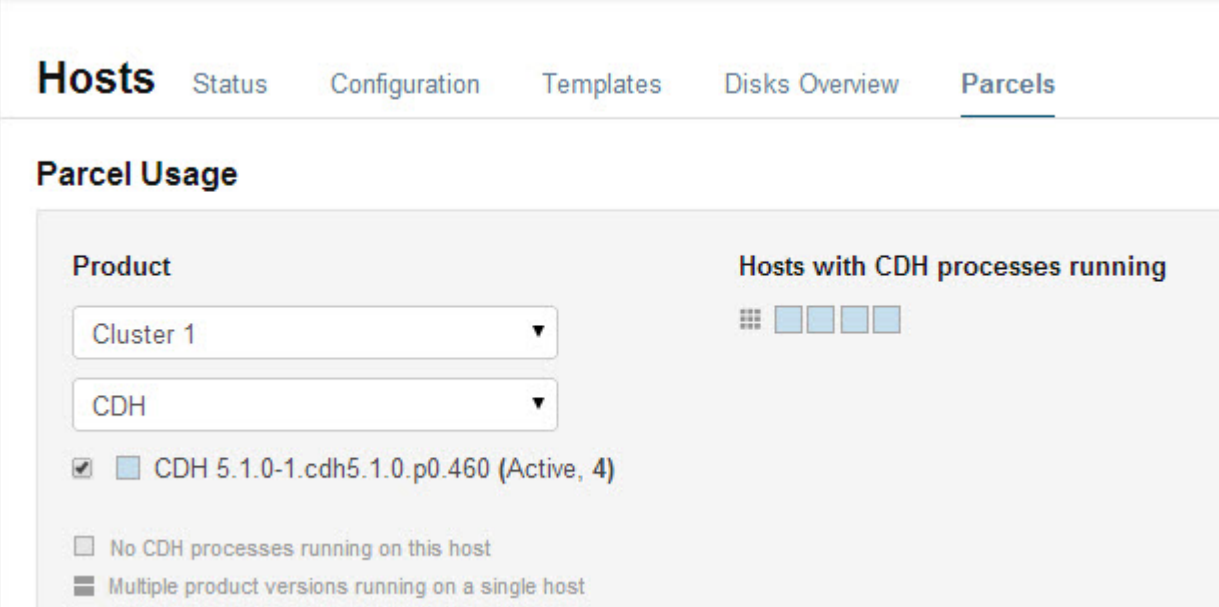
If you have a parcel distributing but never completing, make sure you have enough free space in the [parcel download directories](#), as Cloudera Manager will retry to downloading and unpacking parcels even if there is insufficient space.

Viewing Parcel Usage

The **Parcel Usage** page shows you which parcels are in current use in your clusters. This is particularly useful in a large deployment where it may be difficult to keep track of what versions are installed across the cluster, especially if some hosts were not available when you performed an installation or upgrade, or were added later. To display the Parcel Usage page:

1. Do one of the following:
 - Click  in the top navigation bar
 - Click **Hosts** in the top navigation bar and click the **Parcels** tab.
2. Click the **Parcel Usage** button.


This page only shows the usage of parcels, not components that were installed as packages. If you select a cluster running packages (for example, a CDH 4 cluster) the cluster is not displayed, and instead you will see a message indicating the cluster is not running parcels. If you have individual hosts running components installed as packages, they will appear as "empty."



You can view parcel usage by cluster, or by product.

You can also view just the hosts running only the active parcels, or just hosts running older parcels (not the currently active parcels) or both.

The "host map" at the right shows each host in the cluster with the status of the parcels on that host. If the host is actually running the processes from the currently activated parcels, the host is indicated in blue. A black square indicates that a parcel has been activated, but that all the running processes are from an earlier version of the software. This can happen, for example, if you have not restarted a service or role after activating a new parcel.

Move the cursor over the  icon to see the rack to which the hosts are assigned. Hosts on different racks are displayed in separate rows.

To view the exact versions of the software running on a given host, you can click on the square representing the host. This pops up a display showing the parcel versions installed on that host.

Hosts Status Configuration Templates Disks Overview **Parcels**

Parcel Usage

Product

Cluster 1


CDH

CDH 5.1.0-1.cdh5.1.0.p0.460

No CDH processes running

Multiple product versions running

Hosts with CDH processes running



[tcdn501-1.ent.cloudera.com](#)


Versions used by running roles

CDH 5.1.0-1.cdh5.1.0.p0.460 Active

[Hive Metastore Server](#)
[HiveServer2](#)
[JobHistory Server](#)
[NameNode](#)
[Oozie Server](#)
[ResourceManager](#)
[SecondaryNameNode](#)
[Server](#)

Other products in use by host

For CDH 4.4, Impala 1.1.1, and Solr 0.9.3 or higher, the pop-up lists the roles running on the selected host that are part of the listed parcel. Clicking a role opens the Cloudera Manager page for that role. It also shows whether the parcel is active or not.

If a host is running various software versions, the square representing the host is a four-square icon . When you move the cursor over that host, both the active and inactive components are shown. For example, in the image below, the older CDH parcel has been deactivated, but only the HDFS service has been restarted.

Hosts Status Configuration Templates Disks Overview **Parcels**

Parcel Usage

Product

Cluster 1

CDH

CDH 5.1.0-1.cdh5.0.1.p0.460

CDH 5.0.1-1.cdh5.0.1.p0.47

No CDH processes running

Multiple product versions running

Hosts with CDH processes running

[tcdn501-1.ent.cloudera.com](#)

Versions used by running roles

CDH 5.0.1-1.cdh5.0.1.p0.47 Inactive

[Hive Metastore Server](#) [HiveServer2](#) [Hue Server](#) [JobHistory Server](#)

[Oozie Server](#) [ResourceManager](#) [Server](#) [Sqoop 2 Server](#)

CDH 5.1.0-1.cdh5.1.0.p0.460 Active

[NameNode](#) [SecondaryNameNode](#)


Other products in use by host

Parcel Configuration Settings

You can configure where parcels are stored on the Cloudera Manager Server host, the URLs of parcel repositories, the properties of a proxy server through which parcels are downloaded, and where parcels distributed to cluster hosts are stored.

Configuring Cloudera Manager Server Parcel Settings

1. Use one of the following methods to open the parcel settings page:

- **Navigation bar**
 1. Click  in the top navigation bar
 2. Click the **Edit Settings** button.
- **Menu**
 1. Select **Administration > Settings**.
 2. Click the **Parcels** category.
- **Tab**
 1. Click the **Hosts** tab.
 2. Click the **Configuration** tab.
 3. Click the **Parcels** category.
 4. Click the **Edit Settings** button.

2. Specify a property:

- **Local Parcel Repository Path** defines the path on the Cloudera Manager Server host where downloaded parcels are stored.
- **Remote Parcel Repository URLs** is a list of repositories that Cloudera Manager should check for parcels. Initially this points to the latest released CDH 4, CDH 5, Impala, and Solr repositories but you can add your own repository locations to the list. You can use this mechanism to add Cloudera repositories that are not listed by default, such as older versions of CDH, or the Sentry parcel for CDH 4.3. You can also use this to add your own [custom repositories](#). The locations of the Cloudera parcel repositories are `https://archive.cloudera.com/product/parcels/version`, where *product* is `cdh4`, `cdh5`, `gplextras5`, `impala`, `search`, and `sentry`, and *version* is a specific product version or `latest`.

To add a parcel repository:

1. In the **Remote Parcel Repository URLs** list, click **+** to open an additional row.
2. Enter the path to the repository.

3. Click **Save Changes**.

You can also:

- Set the frequency with which Cloudera Manager will check for new parcels.
- Configure a proxy to access to the remote repositories.
- Configure whether downloads and distribution of parcels should occur automatically whenever new ones are detected. If automatic downloading/distribution are not enabled (the default), you must go to the **Parcels** page to initiate these actions.
- Control which products can be downloaded if automatic downloading is enabled.
- Control whether to retain downloaded parcels.
- Control whether to retain old parcel version and how many parcel versions to retain

You can configure the bandwidth limits and the number of concurrent uploads, to tune the load that parcel distribution puts on your network. The defaults are up to 50 concurrent parcel uploads and 50 MiB/s aggregate bandwidth.

- The concurrent upload count (**Maximum Parcel Uploads**) doesn't matter, theoretically, if all hosts have the same speed Ethernet. In general, 50 concurrent uploads is an acceptable setting in most cases. However, in a scenario where the server has more bandwidth (say 10Gbe while the normal hosts are using 1Gbe), then the count is important to maximize bandwidth, and would need to be at least the difference in speeds (10x in this case).
- The bandwidth limit (**Parcel Distribution Rate Limit**) should be your Ethernet speed (in MiB/seconds) divided by approximately 16. You can use a higher limit if you have QoS set up to prevent starving other services, or if you are willing accept a higher risk of higher bandwidth load.

Configuring a Proxy Server

To configure a proxy server through which parcels are downloaded, follow the instructions in [Configuring Network Settings](#).

Configuring the Host Parcel Directory

To configure the location of distributed parcels:

1. Click **Hosts** in the top navigation bar.
2. Click the **Configuration** tab.
3. Configure the value of the **Parcel Directory** property. The setting of the `parcel_dir` property in the [Cloudera Manager Agent configuration file](#) overrides this setting.
4. Click **Save Changes** to commit the changes.
5. On each host, restart the Cloudera Manager Agent:

```
$ sudo service cloudera-scm-agent restart
```



Installing Cloudera Manager and CDH

Migrating from Packages to Parcels

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

Managing software distribution using parcels offers many [advantages](#) over packages. To migrate from packages to the *same version* parcel, perform the following steps. To upgrade to a different version, see [Upgrading CDH and Managed Services Using Cloudera Manager](#).


Download, Distribute, and Activate Parcels

1. In the Cloudera Manager Admin Console, click the Parcels indicator in the top navigation bar ( or .
2. Click **Download** for the version that matches the CDH or service version of the currently installed packages. If the parcel you want is not shown here—for example, if you want to use a version of CDH that is not the most current version—you can add parcel repositories through the [Parcel Configuration Settings](#) on page 78 page:
 - **CDH 5** - Impala, Spark, and Search are included in the CDH parcel.
 - CDH - `https://username:password@archive.cloudera.com/p/cdh5/parcels/`
 - GPL Extras - `https://archive.cloudera.com/p/gplextras5/parcels/`
 - **Other services**
 - Accumulo - `https://username:password@archive.cloudera.com/p/accumulo/parcels/`
 - Sqoop connectors - `https://username:password@archive.cloudera.com/p/sqoop-connectors/parcels/`

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel file(s) and put them into a repository. See [Creating and Using a Parcel Repository for Cloudera Manager](#) on page 126 for more details.


3. When the download has completed, click **Distribute** for the version you downloaded.
4. When the parcel has been distributed and unpacked, the button will change to say **Activate**.
5. Click **Activate**.

Restart the Cluster and Deploy Client Configuration

1. Restart the cluster:
 - a. On the Home page, click  to the right of the cluster name and select **Restart**.
 - b. Click **Restart** that appears in the next screen to confirm. The **Command Details** window shows the progress of stopping services.

When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

You can optionally perform a [rolling restart](#).

2. Redeploy client configurations:
 - a. On the Home page, click  to the right of the cluster name and select **Deploy Client Configuration**.
 - b. Click **Deploy Client Configuration**.

Uninstall Packages

1. If your Hue service uses the embedded SQLite DB, back up `/var/lib/hue/desktop.db` to a location that is not `/var/lib/hue` as this directory is removed when the packages are removed.

2. Uninstall the CDH packages on each host:



Warning: If you are running Key HSM, do *not* uninstall `bigtop-utils` because it is a requirement for the `keytrustee-keyhsm` package.

- **Not including Impala and Search**

Operating System	Command
RHEL	<code>\$ sudo yum remove bigtop-utils bigtop-jsvc bigtop-tomcat hue-common sqoop2-client</code>
SLES	<code>\$ sudo zypper remove bigtop-utils bigtop-jsvc bigtop-tomcat hue-common sqoop2-client</code>
Ubuntu or Debian	<code>\$ sudo apt-get purge bigtop-utils bigtop-jsvc bigtop-tomcat hue-common sqoop2-client</code>

- **Including Impala and Search**

Operating System	Command
RHEL	<code>\$ sudo yum remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc</code>
SLES	<code>\$ sudo zypper remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc</code>
Ubuntu or Debian	<code>\$ sudo apt-get purge 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc</code>

3. Restart all the Cloudera Manager Agents to force an update of the symlinks to point to the newly installed components on each host:

```
$ sudo service cloudera-scm-agent restart
```

4. If your Hue service uses the embedded SQLite DB, restore the DB you backed up:

- Stop the Hue service.
- Copy the backup from the temporary location to the newly created Hue database directory, `/var/lib/hue`.
- Start the Hue service.

Restart Cloudera Manager Agents

Restart all the Cloudera Manager Agents to force an update of the symlinks to point to the newly installed components. On each host run:

```
$ sudo service cloudera-scm-agent restart
```

Update Applications to Reference Parcel Paths

With parcel software distribution, the path to the CDH libraries is `/opt/cloudera/parcels/CDH/lib` instead of the usual `/usr/lib/`. You should not link `/usr/lib/` elements to parcel deployed paths, as such links may confuse scripts that distinguish between the two paths. Instead you should update your applications to reference the new library locations.

Migrating from Parcels to Packages

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

Installing Cloudera Manager and CDH

To migrate from a parcel to the *same version* packages, perform the following steps. To upgrade to a different version, see [Upgrading CDH and Managed Services Using Cloudera Manager](#).

Install Packages

Install CDH and Managed Service Packages



For more information about manually installing CDH packages, see [CDH 4 Installation Guide](#) or [Cloudera Installation Guide](#).


1. Choose a repository strategy:

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems.
- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see [Understanding Custom Installation Solutions](#) on page 125. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

2. Install packages:

CDH Version	Procedure						
CDH 5	<ul style="list-style-type: none">• Red Hat<ol style="list-style-type: none">1. Download and install the "1-click Install" package.<ol style="list-style-type: none">a. Download the CDH 5 "1-click Install" package.<p>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 (for example, your home directory).</p><table border="1"><thead><tr><th>OS Version</th><th>Click this Link</th></tr></thead><tbody><tr><td>Red Hat/CentOS/Oracle 5</td><td>Red Hat/CentOS/Oracle 5 link</td></tr><tr><td>Red Hat/CentOS/Oracle 6</td><td>Red Hat/CentOS/Oracle 6 link</td></tr></tbody></table><ol style="list-style-type: none">b. Install the RPM:<ul style="list-style-type: none">• Red Hat/CentOS/Oracle 5<pre>\$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm</pre>• Red Hat/CentOS/Oracle 6<pre>\$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm</pre>2. (Optionally) add a repository key:<ul style="list-style-type: none">• Red Hat/CentOS/Oracle 5<pre>\$ sudo rpm --import https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera</pre>	OS Version	Click this Link	Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link	Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 link
OS Version	Click this Link						
Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link						
Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 link						

CDH Version	Procedure
	<ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 6 <pre data-bbox="451 289 1455 342" style="border: 1px dashed #ccc; padding: 5px;">\$ sudo rpm --import https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p data-bbox="451 394 756 422">3. Install the CDH packages:</p> <pre data-bbox="451 457 1442 625" style="border: 1px dashed #ccc; padding: 5px;">\$ sudo yum clean all \$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-https hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="521 657 1425 772" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div> • SLES <ol style="list-style-type: none"> 1. Download and install the "1-click Install" package. <ol style="list-style-type: none"> a. Download the CDH 5 "1-click Install" package. <p data-bbox="532 953 1466 1014">Click this link, choose Save File, and save it to a directory to which you have write access (for example, your home directory).</p> b. Install the RPM: <pre data-bbox="451 1098 1040 1125" style="border: 1px dashed #ccc; padding: 5px;">\$ sudo rpm -i cloudera-cdh-5-0.x86_64.rpm</pre> c. Update your system package index by running: <pre data-bbox="451 1220 753 1247" style="border: 1px dashed #ccc; padding: 5px;">\$ sudo zypper refresh</pre> 2. (Optionally) add a repository key: <pre data-bbox="451 1360 1442 1413" style="border: 1px dashed #ccc; padding: 5px;">\$ sudo rpm --import https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> 3. Install the CDH packages: <pre data-bbox="451 1507 1442 1675" style="border: 1px dashed #ccc; padding: 5px;">\$ sudo zypper clean --all \$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-https hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="521 1707 1425 1822" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div> • Ubuntu and Debian <ol style="list-style-type: none"> 1. Download and install the "1-click Install" package

CDH Version	Procedure								
	<p>a. Download the CDH 5 "1-click Install" package:</p> <table border="1" data-bbox="537 279 1463 474"> <thead> <tr> <th>OS Version</th> <th>Click this Link</th> </tr> </thead> <tbody> <tr> <td>Wheezy</td> <td>Wheezy link</td> </tr> <tr> <td>Precise</td> <td>Precise link</td> </tr> <tr> <td>Trusty</td> <td>Trusty link</td> </tr> </tbody> </table> <p>b. Install the package by doing one of the following:</p> <ul style="list-style-type: none"> 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: <pre data-bbox="435 699 1463 758">sudo dpkg -i cdh5-repository_1.0_all.deb</pre> <p>2. Optionally add a repository key:</p> <ul style="list-style-type: none"> Debian Wheezy <pre data-bbox="435 905 1463 984">\$ curl -s https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> Ubuntu Precise <pre data-bbox="435 1052 1463 1131">\$ curl -s https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key sudo apt-key add -</pre> <p>3. Install the CDH packages:</p> <pre data-bbox="435 1220 1463 1409">\$ sudo apt-get update \$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="518 1430 1425 1541" style="border: 1px solid black; padding: 5px;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div>	OS Version	Click this Link	Wheezy	Wheezy link	Precise	Precise link	Trusty	Trusty link
OS Version	Click this Link								
Wheezy	Wheezy link								
Precise	Precise link								
Trusty	Trusty link								
<p>CDH 4, Impala, and Solr</p>	<ul style="list-style-type: none"> Red Hat-compatible <ol style="list-style-type: none"> Click the entry in the table at CDH Download Information that matches your Red Hat or CentOS system. Navigate to the repo file (<code>cloudera-cdh4.repo</code>) for your system and save it in the <code>/etc/yum.repos.d/</code> directory. Optionally add a repository key: 								

CDH Version	Procedure
	<ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 5 <pre data-bbox="435 281 1442 348">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <ul style="list-style-type: none"> • Red Hat/CentOS 6 <pre data-bbox="435 436 1442 491">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p data-bbox="451 541 971 571">4. Install packages on every host in your cluster:</p> <p data-bbox="505 590 784 619">a. Install CDH 4 packages:</p> <pre data-bbox="435 646 1442 751">\$ sudo yum -y install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-https hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <p data-bbox="505 783 1468 846">b. To install the <code>hue-common</code> package and all Hue applications on the Hue host, install the hue meta-package:</p> <pre data-bbox="435 877 764 907">\$ sudo yum install hue</pre> <p data-bbox="451 955 927 984">5. (Requires CDH 4.2 or later) Install Impala</p> <p data-bbox="505 1003 1468 1066">a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your Red Hat or CentOS system.</p> <p data-bbox="505 1073 1406 1136">b. Navigate to the repo file for your system and save it in the <code>/etc/yum.repos.d/</code> directory.</p> <p data-bbox="505 1142 1133 1171">c. Install Impala and the Impala Shell on Impala machines:</p> <pre data-bbox="435 1203 1036 1232">\$ sudo yum -y install impala impala-shell</pre> <p data-bbox="451 1281 927 1310">6. (Requires CDH 4.3 or later) Install Search</p> <p data-bbox="505 1329 1468 1392">a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Red Hat or CentOS system.</p> <p data-bbox="505 1398 1406 1461">b. Navigate to the repo file for your system and save it in the <code>/etc/yum.repos.d/</code> directory.</p> <p data-bbox="505 1467 1273 1497">c. Install the Solr Server on machines where you want Cloudera Search.</p> <pre data-bbox="435 1528 922 1558">\$ sudo yum -y install solr-server</pre> <ul style="list-style-type: none"> • SLES <p data-bbox="451 1665 797 1694">1. Run the following command:</p> <pre data-bbox="435 1726 1458 1789">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/cdh4/sles/11/x86_64/cdh/cloudera-cdh4.repo</pre> <p data-bbox="451 1812 987 1841">2. Update your system package index by running:</p> <pre data-bbox="435 1873 753 1902">\$ sudo zypper refresh</pre>

CDH Version	Procedure
	<p>3. Optionally add a repository key:</p> <pre data-bbox="435 281 1446 352">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p>4. Install packages on every host in your cluster:</p> <p>a. Install CDH 4 packages:</p> <pre data-bbox="435 485 1446 590">\$ sudo zypper install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-https hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <p>b. To install the hue-common package and all Hue applications on the Hue host, install the hue meta-package:</p> <pre data-bbox="435 716 808 743">\$ sudo zypper install hue</pre> <p>c. (Requires CDH 4.2 or later) Install Impala</p> <p>a. Run the following command:</p> <pre data-bbox="435 884 1446 940">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/impala/sles/11/x86_64/impala/cloudera-impala.repo</pre> <p>b. Install Impala and the Impala Shell on Impala machines:</p> <pre data-bbox="435 1031 1036 1058">\$ sudo zypper install impala impala-shell</pre> <p>d. (Requires CDH 4.3 or later) Install Search</p> <p>a. Run the following command:</p> <pre data-bbox="435 1220 1446 1276">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/search/sles/11/x86_64/search/cloudera-search.repo</pre> <p>b. Install the Solr Server on machines where you want Cloudera Search.</p> <pre data-bbox="435 1367 922 1394">\$ sudo zypper install solr-server</pre> <p>• Ubuntu or Debian</p> <ol style="list-style-type: none"> In the table at CDH Version and Packaging Information, click the entry that matches your Ubuntu or Debian system. Navigate to the list file (<code>cloudera.list</code>) for your system and save it in the <code>/etc/apt/sources.list.d/</code> directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your <code>cloudera.list</code> file should look like: <pre data-bbox="435 1717 1446 1822">deb [arch=amd64] https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib deb-src https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib</pre> <p>3. Optionally add a repository key:</p>

CDH Version	Procedure
	<ul style="list-style-type: none"> • Ubuntu Lucid <pre data-bbox="435 279 1474 359">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> • Ubuntu Precise <pre data-bbox="435 436 1474 516">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/ubuntu/precise/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> • Debian Squeeze <pre data-bbox="435 594 1474 674">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/debian/squeeze/amd64/cdh/archive.key sudo apt-key add -</pre> <p data-bbox="451 688 974 720">4. Install packages on every host in your cluster:</p> <ul style="list-style-type: none"> a. Install CDH 4 packages: <pre data-bbox="435 783 1474 905">\$ sudo apt-get install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop-hadoop-hdfs hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <ul style="list-style-type: none"> b. To install the <code>hue-common</code> package and all Hue applications on the Hue host, install the <code>hue</code> meta-package: <pre data-bbox="435 1014 824 1056">\$ sudo apt-get install hue</pre> <ul style="list-style-type: none"> c. (Requires CDH 4.2 or later) Install Impala <ul style="list-style-type: none"> a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your Ubuntu or Debian system. b. Navigate to the list file for your system and save it in the <code>/etc/apt/sources.list.d/</code> directory. c. Install Impala and the Impala Shell on Impala machines: <pre data-bbox="435 1329 1052 1371">\$ sudo apt-get install impala impala-shell</pre> <ul style="list-style-type: none"> d. (Requires CDH 4.3 or later) Install Search <ul style="list-style-type: none"> a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Ubuntu or Debian system. b. Install Solr Server on machines where you want Cloudera Search: <pre data-bbox="435 1581 938 1623">\$ sudo apt-get install solr-server</pre>

Deactivate Parcels

When you deactivate a parcel, Cloudera Manager points to the installed packages, ready to be run the next time a service is restarted. To deactivate parcels,

1. Go to the Parcels page by doing one of the following:

Installing Cloudera Manager and CDH

- Clicking the parcel indicator in the Admin Console navigation bar (📦)
- Clicking the **Hosts** in the top navigation bar, then the **Parcels** tab.

2. Click **Actions** on the activated CDH and managed service parcels and select **Deactivate**.

Restart the Cluster

1. On the Home page, click



to the right of the cluster name and select **Restart**.

2. Click **Restart** that appears in the next screen to confirm. The **Command Details** window shows the progress of stopping services.

When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

You can optionally perform a [rolling restart](#).

Remove and Delete Parcels

Removing a Parcel

To remove a parcel, click the down arrow to the right of an **Activate** button and select **Remove from Hosts**.

Deleting a Parcel

To delete a parcel, click the down arrow to the right of a **Distribute** button and select **Delete**.

Installation Path A - Automated Installation by Cloudera Manager

Before proceeding with this path for a new installation, review [Cloudera Manager Deployment](#) on page 34. If you are upgrading a Cloudera Manager existing installation, see [Upgrading Cloudera Manager](#).

The general steps in the procedure for Installation Path A follow.

Before You Begin

In certain circumstances you may need to perform optional installation and configuration steps.

Install and Configure External Databases

Read [Cloudera Manager and Managed Service Data Stores](#) on page 38. If you are using an external database for services or Cloudera Management Service roles, install and configure it following the instructions in [External Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 42.

Perform Configuration Required by Single User Mode

If you choose to create a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 11.

(CDH 5 only) On RHEL 5 and CentOS 5, Install Python 2.6 or 2.7

CDH 5 Hue will only work with the default Python version of the operating system on which it is being installed. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue. However, RHEL 5 and CentOS 5 users will have to download Python 2.6 from the EPEL repository as described below.

To install packages from the EPEL repository, download the appropriate repository rpm packages to your machine and then install Python using `yum`. For example, use the following commands for RHEL 5 or CentOS 5:

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



```
...
$ yum install python26
```

Configure an HTTP Proxy

The Cloudera Manager installer accesses `archive.cloudera.com` by using `yum` on RHEL systems, `zypper` on SLES systems, or `apt-get` on Debian/Ubuntu systems. If your hosts access the Internet through an HTTP proxy, you can configure `yum`, `zypper`, or `apt-get`, system-wide, to access `archive.cloudera.com` through a proxy. To do so, modify the system configuration on the Cloudera Manager Server host and on every cluster host as follows:

OS	File	Property
RHEL-compatible	<code>/etc/yum.conf</code>	<code>proxy=http://server:port/</code>
SLES	<code>/root/.curlrc</code>	<code>--proxy=http://server:port/</code>
Ubuntu or Debian	<code>/etc/apt/apt.conf</code>	<code>Acquire::http::Proxy "http://server:port";</code>

Download and Run the Cloudera Manager Server Installer

- Download the Cloudera Manager installer binary from [Cloudera Manager 5.3.7 Downloads](#) to the cluster host where you want to install the Cloudera Manager Server.
 - Click **Download Cloudera Express** or **Download Cloudera Enterprise**. See [Cloudera Express and Cloudera Enterprise Features](#).
 - Optionally register and click **Submit** or click the Just take me to the **download page** link. The `cloudera-manager-installer.bin` file downloads.
- Change `cloudera-manager-installer.bin` to have executable permission.

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

- Run the Cloudera Manager Server installer:
 - Install Cloudera Manager packages from the Internet - `sudo ./cloudera-manager-installer.bin`
 - Install Cloudera Manager packages from a [local repository](#) - `sudo ./cloudera-manager-installer.bin --skip_repo_package=1`
- Read the Cloudera Manager README and then press **Return** or **Enter** to choose **Next**.
- 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.
- Read the Oracle Binary Code License Agreement 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 Oracle Binary Code License Agreement. The following occurs:
 - The installer installs the Oracle JDK and the Cloudera Manager repository files.
 - The installer installs the Cloudera Manager Server and embedded PostgreSQL packages.
 - The installer starts the Cloudera Manager Server and embedded PostgreSQL database.
- 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.
- Press **Return** or **Enter** to choose **OK** to exit the installer.



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](#) on page 147.

Installing Cloudera Manager and CDH

Start and Log into the Cloudera Manager Admin Console

The Cloudera Manager Server URL takes the following form `http://Server host:port`, where *Server host* is the fully-qualified domain name or IP address of the host where the Cloudera Manager Server is installed and *port* is the port configured for the Cloudera Manager Server. The default port is 7180.

1. Wait several minutes for the Cloudera Manager Server to complete its startup. To observe the startup process you can perform `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](#) on page 406.
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. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can [change the password](#) using Cloudera Manager after you run the installation wizard. While you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.

Use the Cloudera Manager Wizard for Software Installation and Configuration

The following instructions describe how to use the Cloudera Manager installation wizard to do an initial installation and configuration. The wizard lets you:

- Select the edition of Cloudera Manager to install
- Find the cluster hosts you specify using hostname and IP address ranges
- Connect to each host with SSH to install the Cloudera Manager Agent and other components
- Optionally installs the Oracle JDK on the cluster hosts. If you choose not to have the JDK installed, you must install it on all clusters according to the following instructions prior to running the wizard:
 - CDH 5 - [Java Development Kit Installation](#) on page 35.
 - CDH 4 - [Java Development Kit Installation](#).
- Install CDH and managed service packages or parcels
- Configure CDH and managed services automatically and start the services



Important: All hosts in the cluster must have some way to access installation files using one of the following methods:

- Internet access to allow the wizard to install software packages or parcels from `archive.cloudera.com`.
- A custom internal repository that the host(s) can access. For example, for a Red Hat host, you could set up a Yum repository. See [Creating and Using a Package Repository for Cloudera Manager](#) on page 129 [Creating and Using a Package Repository for Cloudera Manager](#) on page 129 for more information about this option.

Choose Cloudera Manager Edition and Hosts

1. Choose which [edition](#) to install:
 - Cloudera Express, which does not require a license, but provides a somewhat limited set of features.
 - Cloudera Enterprise Data Hub Edition Trial, which does not require a license, but expires after 60 days and cannot be renewed
 - Cloudera Enterprise with one of the following license types:
 - Basic Edition
 - Flex Edition
 - Data Hub Edition

If you choose Cloudera Express or Cloudera Enterprise Data Hub Edition Trial, you can elect to upgrade the license at a later time. See [Managing Licenses](#).

2. If you have elected Cloudera Enterprise, install a license:
 - a. Click **Upload License**.
 - b. Click the document icon to the left of the **Select a License File** text field.
 - c. Navigate to the location of your license file, click the file, and click **Open**.
 - d. Click **Upload**.

Click **Continue** to proceed with the installation.

3. Information is displayed indicating what the CDH installation includes. At this point, you can access online Help or the Support Portal if you wish. Click **Continue** to proceed with the installation. The Specify hosts for your CDH cluster installation page displays.
4. To enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the cluster hostnames or IP addresses. You can also specify hostname and IP address ranges. For example:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.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. Use this technique to make more specific searches instead of searching overly wide ranges. The scan results will include all addresses scanned, but only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. If you don't know the IP addresses of all of the hosts, you can enter an address range that spans over unused addresses and then deselect the hosts that do not exist (and are not discovered) later in this procedure. However, keep in mind that wider ranges will require more time to scan.

5. Click **Search**. Cloudera Manager identifies the hosts on your cluster to allow you to configure them for services. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking **Abort Scan**. To find additional hosts, click **New Search**, add the host names or IP addresses and click **Search** again. Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts. Common causes of loss of connectivity are firewalls and interference from SELinux.
6. 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.

Choose Software Installation Method and Install Software



Important: You cannot install software using both parcels and packages in the same cluster.

1. Select the repository type to use for the installation: parcels or packages.

- **Use Parcels:**

1. Choose the parcels to install. The choices you see depend on the repositories you have chosen – a repository may contain multiple parcels. Only the parcels for the latest supported service versions are configured by default.

You can add additional parcels for previous versions by specifying custom repositories. For example, you can find the locations of the previous CDH 4 parcels at

<https://archive.cloudera.com/cdh4/parcels/>. Or, if you are installing CDH 4.3 and want to use [policy-file authorization](#), you can add the Sentry parcel using this mechanism.

1. To specify the parcel directory, local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:

- **Parcel Directory and Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host.
- **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the **+** button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the [Configuring Cloudera Manager Server Parcel Settings](#) on page 78 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you will see all the unique parcels contained in all your repositories.
- **Proxy Server** - Specify the properties of a proxy server.

2. Click **OK**.

- **Use Packages:**

1. Select the major release of CDH to install.
 2. Select the specific release of CDH to install.
 3. Select the specific releases of Impala and Solr to install, assuming you have selected an appropriate CDH version. You can choose either the latest version or use a custom repository. Choose **None** if you do not want to install that service.
2. Select the release of Cloudera Manager Agent to install. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories. 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 leave deselected if you plan to install it yourself. If checked, your local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, select the **Install Java Unlimited Strength Encryption Policy Files** checkbox. Click **Continue**. The Enable Single User Mode screen displays.
 4. (Optional) Select **Single User Mode** to configure the Cloudera Manager Agent and all service processes to run as the same user. This mode requires [extra configuration steps](#) that must be done manually on all hosts in the cluster. If you have not performed the steps, directory creation will fail in the installation wizard. In most cases, you can create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**. The Provide SSH login credentials screen displays.
5. Specify SSH login properties:
 - Select **root** or enter the user name for an account that has password-less sudo permission.
 - Select an authentication method:
 - If you choose to use password authentication, enter and confirm the password.
 - If you choose to use public-key authentication provide a passphrase and path to the required key files.
 - You can choose to specify an alternate SSH port. The default value is 22.
 - You can specify the maximum number of host installations to run at once. The default value is 10.

Click **Continue**. Cloudera Manager performs the following:

- **Parcels** - installs the Oracle JDK and the Cloudera Manager Agent packages and starts the Agent. Click **Continue**. During the parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels you will see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.
- **Packages** - configures package repositories, installs the Oracle JDK, CDH and managed service and the Cloudera Manager Agent packages, and starts the Agent. When the **Continue** button at the bottom of the screen turns

blue, the installation process is completed. If the installation has completed successfully on some hosts but failed on others, you can click **Continue** if you want to skip installation on the failed hosts and continue to the next screen to start configuring services on the successful hosts.

While packages are being installed, the status of installation on each host is displayed. You can click the **Details** link for individual hosts to view detailed information about the installation and error messages if installation fails on any hosts. If you click the **Abort Installation** button while installation is in progress, it will halt any pending or in-progress installations and roll back any in-progress installations to a clean state. The **Abort Installation** button does not affect host installations that have already completed successfully or already failed.

6. 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. If the validation is successful, click **Finish**. The Cluster Setup screen displays.

Add Services

1. In the first page of the Add Services wizard you choose the combination of services to install and whether to install Cloudera Navigator:

- Click the radio button next to the combination of services to install:

CDH 4	CDH 5
<ul style="list-style-type: none"> • Core Hadoop - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • Core with HBase • Core with Impala • All Services - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • Custom Services - Any combination of services. 	<ul style="list-style-type: none"> • Core Hadoop - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, and Sqoop • Core with HBase • Core with Impala • Core with Search • Core with Spark • All Services - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, Sqoop, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • Custom Services - Any combination of services.

As you select the services, keep the following in mind:

- Some services depend on other services; for example, HBase requires HDFS and ZooKeeper. Cloudera Manager tracks dependencies and installs the correct combination of services.
- In a Cloudera Manager deployment of a CDH 4 cluster, the MapReduce service is the default MapReduce computation framework. Choose **Custom Services** to install YARN or use the Add Service functionality to add YARN after installation completes.



Note: You can create a YARN service in a CDH 4 cluster, but it is not considered production ready.

- In a Cloudera Manager deployment of a CDH 5 cluster, the YARN service is the default MapReduce computation framework. Choose **Custom Services** to install MapReduce or use the Add Service functionality to add MapReduce after installation completes.



Note: In CDH 5, the MapReduce service has been deprecated. However, the MapReduce service is fully supported for backward compatibility through the CDH 5 life cycle.

- The Flume service can be added only after your cluster has been set up.
- If you have chosen Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See the [Cloudera Navigator Documentation](#).

Click **Continue**. The Customize Role Assignments screen displays.

2. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. These assignments are typically acceptable, but you can reassign them if necessary.

Click a field below a role to display a dialog containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts or **Custom** to display the pageable hosts dialog.

The following shortcuts for specifying hostname patterns are supported:

- Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

3. When you are satisfied with the assignments, click **Continue**. The Database Setup screen displays.

4. Configure database settings:

- a. Choose the database type:

- Leave the default setting of **Use Embedded Database** to have Cloudera Manager create and configure required databases. Make a note of the auto-generated passwords.

Cluster Setup

Database Setup

Configure and test database connections. If using custom databases, create the databases first according to the [Installing and Configuring an External Database](#) section of the [Installation Guide](#).

Use Custom Databases
 Use Embedded Database

When using the embedded database, passwords are automatically generated. Please copy them down.

Hive				
Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	hive	hive	24FLyrj0zb
Activity Monitor				
Currently assigned to run on tcdn53-1.ent.cloudera.com.				
Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	amon	amon	2VCic0tDJE
Reports Manager				
Currently assigned to run on tcdn53-1.ent.cloudera.com.				
Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	rman	rman	Mn2i8IEoCH
Navigator Audit Server				
Currently assigned to run on tcdn53-1.ent.cloudera.com.				
Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	nav	nav	P89RAR6e0o
Navigator Metadata Server				
Currently assigned to run on tcdn53-1.ent.cloudera.com.				
Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn53-1.ent.cloudera.com:7432	PostgreSQL	navms	navms	29O536GxZp

- Select **Use Custom Databases** to specify external databases.
 1. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
 - b. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.) The Review Changes screen displays.
5. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed.



Warning: DataNode data directories should not be placed on NAS devices.

Click **Continue**. The wizard starts the services.

6. When all of the services are started, click **Continue**. You will see a success message indicating that your cluster has been successfully started.
7. Click **Finish** to proceed to the [Cloudera Manager Admin Console Home Page](#).

Change the Default Administrator Password

As soon as possible after running the wizard and beginning to use Cloudera Manager, change the default administrator password:

1. Right-click the logged-in username at the far right of the top navigation bar and select **Change Password**.
2. Enter the current password and a new password twice, and then click **Update**.

Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 146.

Installation Path B - Manual Installation Using Cloudera Manager Packages

Before proceeding with this path for a new installation, review [Cloudera Manager Deployment](#) on page 34. If you are upgrading a Cloudera Manager existing installation, see [Upgrading Cloudera Manager](#).

To install the Cloudera Manager Server using packages, follow the instructions in this section. You can also use Puppet or Chef to install the packages. The general steps in the procedure for Installation Path B follow.

During Cloudera Manager installation you can choose to install CDH and managed service as parcels or packages. For packages, you can choose to have Cloudera Manager install the packages or install them yourself.



Important: As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.

This can affect new installations, upgrades, adding new hosts to a cluster, and adding a new cluster.

For more information, see [Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication](#).



Important: You cannot install software using both parcels and packages in the same cluster.

Installing Cloudera Manager and CDH

Before You Begin

Install and Configure Databases

Read [Cloudera Manager and Managed Service Data Stores](#) on page 38. If you are using an external database, install and configure a database as described in [MySQL Database](#) on page 48, [Oracle Database](#) on page 53, or [External PostgreSQL Database](#) on page 45.

Perform Configuration Required by Single User Mode

If you choose to create a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 11.

(CDH 5 only) On RHEL 5 and CentOS 5, Install Python 2.6 or 2.7

CDH 5 Hue will only work with the default Python version of the operating system on which it is being installed. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue. However, RHEL 5 and CentOS 5 users will have to download Python 2.6 from the EPEL repository as described below.

To install packages from the EPEL repository, download the appropriate repository rpm packages to your machine and then install Python using `yum`. For example, use the following commands for RHEL 5 or CentOS 5:

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

Establish Your Cloudera Manager Repository Strategy

Cloudera recommends installing products using package management tools such as `yum` for Red Hat compatible systems, `zypper` for SLES, and `apt-get` for Debian/Ubuntu. These tools depend on access to repositories to install software. For example, Cloudera maintains Internet-accessible repositories for CDH and Cloudera Manager installation files. Strategies for installing Cloudera Manager include:

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems. For Cloudera Manager repository locations and client repository files, see [Cloudera Manager Version and Download Information](#).
- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see [Understanding Custom Installation Solutions](#) on page 125. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

Red Hat-compatible

1. Save the appropriate Cloudera Manager repo file (`cloudera-manager.repo`) for your system:

OS Version	Repo URL
Red Hat/CentOS/Oracle 5	https://username:password@archive.cloudera.com/p/cm5/redhat/5/x86_64/cm/cloudera-manager.repo
Red Hat/CentOS 6	https://username:password@archive.cloudera.com/p/cm5/redhat/6/x86_64/cm/cloudera-manager.repo

2. Copy the repo file to the `/etc/yum.repos.d/` directory.

SLES

1. Run the following command:

```
$ sudo zypper addrepo -f
https://username:password@archive.cloudera.com/p/cm5/sles/11/x86_64/cm/cloudera-manager.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

Ubuntu or Debian

1. Save the appropriate Cloudera Manager list file (`cloudera.list`) for your system:

OS Version	Repo URL
Ubuntu Trusty (14.04)	https://username:password@archive.cloudera.com/p/cm5/ubuntu/trusty/amd64/cm/cloudera.list
Ubuntu Precise (12.04)	https://username:password@archive.cloudera.com/p/cm5/ubuntu/precise/amd64/cm/cloudera.list
Ubuntu Lucid (10.04)	https://username:password@archive.cloudera.com/p/cm5/ubuntu/lucid/amd64/cm/cloudera.list
Debian Wheezy (7.0 and 7.1)	https://username:password@archive.cloudera.com/p/cm5/debian/wheezy/amd64/cm/cloudera.list
Debian Squeeze (6.0)	https://username:password@archive.cloudera.com/p/cm5/debian/squeeze/amd64/cm/cloudera.list

2. Copy the content of that file and append it to the content of the `cloudera.list` in the `/etc/apt/sources.list.d/` directory.
3. Update your system package index by running:

```
$ sudo apt-get update
```

Install the Oracle JDK

Install the Oracle Java Development Kit (JDK) on the Cloudera Manager Server host.

The JDK is included in the Cloudera Manager 5 repositories. After downloading and editing the repo or list file, install the JDK as follows:

OS	Command
RHEL	<code>\$ sudo yum install oracle-j2sdk1.7</code>
SLES	<code>\$ sudo zypper install oracle-j2sdk1.7</code>
Ubuntu or Debian	<code>\$ sudo apt-get install oracle-j2sdk1.7</code>

Install the Cloudera Manager Server Packages

1. Install the Cloudera Manager Server packages either on the host where the database is installed, or on a host that has access to the database. This host need not be a host in the cluster that you want to manage with Cloudera Manager. On the Cloudera Manager Server host, type the following commands to install the Cloudera Manager packages.

OS	Command
RHEL, if you have a yum repo configured	<pre>\$ sudo yum install cloudera-manager-daemons cloudera-manager-server</pre>
RHEL, if you're manually transferring RPMs	<pre>\$ sudo yum --nogpgcheck localinstall cloudera-manager-daemons-*.rpm \$ sudo yum --nogpgcheck localinstall cloudera-manager-server-*.rpm</pre>
SLES	<pre>\$ sudo zypper install cloudera-manager-daemons cloudera-manager-server</pre>
Ubuntu or Debian	<pre>\$ sudo apt-get install cloudera-manager-daemons cloudera-manager-server</pre>

2. If you choose an Oracle database for use with Cloudera Manager, edit the `/etc/default/cloudera-scm-server` file on the Cloudera Manager server host. Locate the line that begins with `export CM_JAVA_OPTS` and change the `-Xmx2G` option to `-Xmx4G`.

Set up a Database for the Cloudera Manager Server

Depending on whether you are using an external database, or the embedded PostgreSQL database, do one of the following:

- External database - Prepare the Cloudera Manager Server database as described in [Preparing a Cloudera Manager Server External Database](#) on page 40.
- Embedded database - Install an embedded PostgreSQL database as described in [Installing and Starting the Cloudera Manager Server Embedded Database](#) on page 39.

(Optional) Install Cloudera Manager Agent, CDH, and Managed Service Software

You can use Cloudera Manager to install Cloudera Manager Agent packages, CDH, and managed service software, or you can install them manually.

To use Cloudera Manager to install the software (in [Choose the Software Installation Method and Install Software](#) on page 108), you must meet the requirements described in [Cloudera Manager Deployment](#) on page 34. If you use Cloudera Manager to install software, go to [Start the Cloudera Manager Server](#) on page 106. Otherwise, proceed with the following sections.

Install the Oracle JDK

Install the Oracle JDK on the cluster hosts. Cloudera Manager 5 can manage both CDH 5 and CDH 4, and the required JDK version varies accordingly:

- CDH 5 - [Java Development Kit Installation](#) on page 35.
- CDH 4 - [Java Development Kit Installation](#).

Install Cloudera Manager Agent Packages

To install the packages manually, do the following on every Cloudera Manager Agent host (including those that will run one or more of the Cloudera Management Service roles: Service Monitor, Activity Monitor, Event Server, Alert Publisher, or Reports Manager):

1. Use one of the following commands to install the Cloudera Manager Agent packages:

OS	Command
RHEL, if you have a yum repo configured:	<pre>\$ sudo yum install cloudera-manager-agent cloudera-manager-daemons</pre>
RHEL, if you're manually transferring RPMs:	<pre>\$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons</pre>

OS	Command
SLES	\$ sudo zypper install cloudera-manager-agent cloudera-manager-daemons
Ubuntu or Debian	\$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons

- On every Cloudera Manager Agent host, configure the Cloudera Manager Agent to point to the Cloudera Manager Server by setting the following properties in the `/etc/cloudera-scm-agent/config.ini` configuration file:

Property	Description
<code>server_host</code>	Name of the host where Cloudera Manager Server is running.
<code>server_port</code>	Port on the host where Cloudera Manager Server is running.

For more information on Agent configuration options, see [Agent Configuration File](#).

Install CDH and Managed Service Packages


For more information about manually installing CDH packages, see [CDH 4 Installation Guide](#) or [Cloudera Installation Guide](#).



- Choose a repository strategy:

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems.
- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see [Understanding Custom Installation Solutions](#) on page 125. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

- Install packages:

CDH Version	Procedure						
CDH 5	<ul style="list-style-type: none"> Red Hat <ol style="list-style-type: none"> Download and install the "1-click Install" package. <ol style="list-style-type: none"> Download the CDH 5 "1-click Install" package. <p>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 (for example, your home directory).</p> <table border="1"> <thead> <tr> <th>OS Version</th> <th>Click this Link</th> </tr> </thead> <tbody> <tr> <td>Red Hat/CentOS/Oracle 5</td> <td>Red Hat/CentOS/Oracle 5 link</td> </tr> <tr> <td>Red Hat/CentOS/Oracle 6</td> <td>Red Hat/CentOS/Oracle 6 link</td> </tr> </tbody> </table> Install the RPM: <ul style="list-style-type: none"> Red Hat/CentOS/Oracle 5 <pre>\$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm</pre> 	OS Version	Click this Link	Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link	Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 link
OS Version	Click this Link						
Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link						
Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 link						

CDH Version	Procedure
	<ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 6 <pre data-bbox="435 279 1370 317" style="border: 1px dashed gray; padding: 5px;">\$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm</pre> <p data-bbox="451 388 846 415">2. (Optionally) add a repository key:</p> <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 5 <pre data-bbox="435 493 1455 546" style="border: 1px dashed gray; padding: 5px;">\$ sudo rpm --import https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 6 <pre data-bbox="435 640 1455 693" style="border: 1px dashed gray; padding: 5px;">\$ sudo rpm --import https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p data-bbox="451 743 756 770">3. Install the CDH packages:</p> <pre data-bbox="435 804 1443 974" style="border: 1px dashed gray; padding: 5px;">\$ sudo yum clean all \$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="516 1005 1425 1121" style="border: 1px solid gray; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div> <ul style="list-style-type: none"> • SLES <p data-bbox="451 1207 1019 1234">1. Download and install the "1-click Install" package.</p> <ul style="list-style-type: none"> a. Download the CDH 5 "1-click Install" package. <p data-bbox="532 1304 1468 1360">Click this link, choose Save File, and save it to a directory to which you have write access (for example, your home directory).</p> b. Install the RPM: <pre data-bbox="435 1444 1040 1472" style="border: 1px dashed gray; padding: 5px;">\$ sudo rpm -i cloudera-cdh-5-0.x86_64.rpm</pre> <p data-bbox="451 1509 1037 1537">c. Update your system package index by running:</p> <pre data-bbox="435 1570 753 1598" style="border: 1px dashed gray; padding: 5px;">\$ sudo zypper refresh</pre> <p data-bbox="451 1648 846 1675">2. (Optionally) add a repository key:</p> <pre data-bbox="435 1707 1438 1759" style="border: 1px dashed gray; padding: 5px;">\$ sudo rpm --import https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p data-bbox="451 1793 756 1820">3. Install the CDH packages:</p> <pre data-bbox="435 1854 1382 1927" style="border: 1px dashed gray; padding: 5px;">\$ sudo zypper clean --all \$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase</pre>

CDH Version	Procedure								
	<pre>hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="521 380 1425 495" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div> <ul style="list-style-type: none"> • Ubuntu and Debian <ol style="list-style-type: none"> 1. Download and install the "1-click Install" package <ol style="list-style-type: none"> a. Download the CDH 5 "1-click Install" package: <table border="1" data-bbox="537 674 1463 867" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0070c0; color: white;">OS Version</th> <th style="background-color: #0070c0; color: white;">Click this Link</th> </tr> </thead> <tbody> <tr> <td>Wheezy</td> <td>Wheezy link</td> </tr> <tr> <td>Precise</td> <td>Precise link</td> </tr> <tr> <td>Trusty</td> <td>Trusty link</td> </tr> </tbody> </table> <ol style="list-style-type: none"> b. Install the package by doing one of the following: <ul style="list-style-type: none"> • 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: <div data-bbox="435 1094 1474 1150" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>sudo dpkg -i cdh5-repository_1.0_all.deb</pre> </div> 2. Optionally add a repository key: <ul style="list-style-type: none"> • Debian Wheezy <div data-bbox="435 1297 1474 1375" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>\$ curl -s https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/archive.key sudo apt-key add -</pre> </div> • Ubuntu Precise <div data-bbox="435 1451 1474 1528" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>\$ curl -s https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key sudo apt-key add -</pre> </div> 3. Install the CDH packages: <div data-bbox="435 1612 1474 1801" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>\$ sudo apt-get update \$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> </div> <div data-bbox="521 1822 1425 1938" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div>	OS Version	Click this Link	Wheezy	Wheezy link	Precise	Precise link	Trusty	Trusty link
OS Version	Click this Link								
Wheezy	Wheezy link								
Precise	Precise link								
Trusty	Trusty link								

CDH Version	Procedure
CDH 4, Impala, and Solr	<ul style="list-style-type: none"> • Red Hat-compatible <ol style="list-style-type: none"> 1. Click the entry in the table at CDH Download Information that matches your Red Hat or CentOS system. 2. Navigate to the repo file (<code>cloudera-cdh4.repo</code>) for your system and save it in the <code>/etc/yum.repos.d/</code> directory. 3. Optionally add a repository key: <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 5 <pre data-bbox="435 533 1446 590">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> • Red Hat/CentOS 6 <pre data-bbox="435 684 1446 741">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> 4. Install packages on every host in your cluster: <ol style="list-style-type: none"> a. Install CDH 4 packages: <pre data-bbox="435 894 1446 993">\$ sudo yum -y install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> b. To install the <code>hue-common</code> package and all Hue applications on the Hue host, install the <code>hue</code> meta-package: <pre data-bbox="435 1121 768 1148">\$ sudo yum install hue</pre> 5. (Requires CDH 4.2 or later) Install Impala <ol style="list-style-type: none"> a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your Red Hat or CentOS system. b. Navigate to the repo file for your system and save it in the <code>/etc/yum.repos.d/</code> directory. c. Install Impala and the Impala Shell on Impala machines: <pre data-bbox="435 1444 1040 1472">\$ sudo yum -y install impala impala-shell</pre> 6. (Requires CDH 4.3 or later) Install Search <ol style="list-style-type: none"> a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Red Hat or CentOS system. b. Navigate to the repo file for your system and save it in the <code>/etc/yum.repos.d/</code> directory. c. Install the Solr Server on machines where you want Cloudera Search. <pre data-bbox="435 1766 927 1793">\$ sudo yum -y install solr-server</pre> • SLES

CDH Version	Procedure
	<p>1. Run the following command:</p> <pre data-bbox="435 279 1455 352">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/cdh4/sles/11/x86_64/cdh/cloudera-cdh4.repo</pre> <p>2. Update your system package index by running:</p> <pre data-bbox="435 426 753 464">\$ sudo zypper refresh</pre> <p>3. Optionally add a repository key:</p> <pre data-bbox="435 548 1438 621">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p>4. Install packages on every host in your cluster:</p> <p>a. Install CDH 4 packages:</p> <pre data-bbox="435 741 1425 856">\$ sudo zypper install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <p>b. To install the hue-common package and all Hue applications on the Hue host, install the hue meta-package:</p> <pre data-bbox="435 968 808 1005">\$ sudo zypper install hue</pre> <p>c. (Requires CDH 4.2 or later) Install Impala</p> <p>a. Run the following command:</p> <pre data-bbox="435 1136 1455 1209">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/impala/sles/11/x86_64/impala/cloudera-impala.repo</pre> <p>b. Install Impala and the Impala Shell on Impala machines:</p> <pre data-bbox="435 1283 1036 1320">\$ sudo zypper install impala impala-shell</pre> <p>d. (Requires CDH 4.3 or later) Install Search</p> <p>a. Run the following command:</p> <pre data-bbox="435 1472 1455 1545">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/search/sles/11/x86_64/search/cloudera-search.repo</pre> <p>b. Install the Solr Server on machines where you want Cloudera Search.</p> <pre data-bbox="435 1619 922 1656">\$ sudo zypper install solr-server</pre> <p>• Ubuntu or Debian</p> <p>1. In the table at CDH Version and Packaging Information, click the entry that matches your Ubuntu or Debian system.</p>

CDH Version	Procedure
	<p>2. Navigate to the list file (<code>cloudera.list</code>) for your system and save it in the <code>/etc/apt/sources.list.d/</code> directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your <code>cloudera.list</code> file should look like:</p> <pre data-bbox="451 352 1458 457">deb [arch=amd64] https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib deb-src https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib</pre> <p>3. Optionally add a repository key:</p> <ul style="list-style-type: none"> • Ubuntu Lucid <pre data-bbox="451 604 1458 657">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> • Ubuntu Precise <pre data-bbox="451 751 1458 804">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/ubuntu/precise/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> • Debian Squeeze <pre data-bbox="451 898 1458 951">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/debian/squeeze/amd64/cdh/archive.key sudo apt-key add -</pre> <p>4. Install packages on every host in your cluster:</p> <ol style="list-style-type: none"> a. Install CDH 4 packages: <pre data-bbox="451 1108 1458 1203">\$ sudo apt-get install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-https hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <ol style="list-style-type: none"> b. To install the <code>hue-common</code> package and all Hue applications on the Hue host, install the hue meta-package: <pre data-bbox="451 1339 824 1371">\$ sudo apt-get install hue</pre> <ol style="list-style-type: none"> c. (Requires CDH 4.2 or later) Install Impala <ol style="list-style-type: none"> a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your Ubuntu or Debian system. b. Navigate to the list file for your system and save it in the <code>/etc/apt/sources.list.d/</code> directory. c. Install Impala and the Impala Shell on Impala machines: <pre data-bbox="451 1644 1052 1675">\$ sudo apt-get install impala impala-shell</pre> d. (Requires CDH 4.3 or later) Install Search <ol style="list-style-type: none"> a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Ubuntu or Debian system. b. Install Solr Server on machines where you want Cloudera Search: <pre data-bbox="451 1896 938 1927">\$ sudo apt-get install solr-server</pre>

(Optional) Install Key Trustee Key Provider

If you want to use Cloudera Navigator Key Trustee Server as the underlying key store for [HDFS Data At Rest Encryption](#), you must install the Key Trustee Key Provider.



Important: Following these instructions will install the required software to add the **KMS (Navigator Key Trustee)** service to your cluster; this enables you to use an existing Cloudera Navigator Key Trustee Server as the underlying key store for [HDFS Data At Rest Encryption](#). This *does not* install Cloudera Navigator Key Trustee Server. Contact your account team for assistance installing Cloudera Navigator Key Trustee Server.

To install the Key Trustee Key Provider:

1. Identify the appropriate repository for your operating system, and copy the repository URL:

OS Version	Repository URL
RHEL-compatible 6	RHEL 6 Repository
RHEL-compatible 5	RHEL 5 Repository
SLES 11	SLES 11 Repository
Ubuntu Trusty (14.04)	Ubuntu Trusty Repository
Ubuntu Precise (12.04)	Ubuntu Precise Repository
Debian Wheezy (7.0 and 7.1)	Debian Wheezy Repository

2. Add the repository to your system, using the appropriate procedure for your operating system:

- **RHEL-compatible**

Download the repository and copy it to the `/etc/yum.repos.d/` directory. Refresh the package index by running `sudo yum clean all`.

- **SLES**

Add the repository to your system using the following command:

```
$ sudo zypper addrepo -f <repository_url>
```

Refresh the package index by running `sudo zypper refresh`.

- **Ubuntu or Debian**

Copy the content of the appropriate `cloudera.list` file from the above repository table and append it to the `/etc/apt/sources.list.d/cloudera.list` file. Create the file if it does not exist. Refresh the package index by running `sudo apt-get update`.

3. Install the `keytrustee-keyprovider` package, using the appropriate command for your operating system:

- **RHEL-compatible**

```
$ sudo yum install keytrustee-keyprovider
```

- **SLES**

```
$ sudo zypper install keytrustee-keyprovider
```

- **Ubuntu or Debian**

```
$ sudo apt-get install keytrustee-keyprovider
```

Start the Cloudera Manager Server



Important: When you start the Cloudera Manager Server and Agents, Cloudera Manager assumes you are not already running HDFS and MapReduce. If these services are running:

1. Shut down HDFS and MapReduce. See [Stopping Services](#) (CDH 4) or [Stopping CDH Services Using the Command Line](#) (CDH 5) for the commands to stop these services.
2. Configure the init scripts to *not* start on boot. Use commands similar to those shown in [Configuring init to Start Core Hadoop System Services](#) (CDH 4) or [Configuring init to Start Hadoop System Services](#) (CDH 5), but *disable* the start on boot (for example, `$ sudo chkconfig hadoop-hdfs-namenode off`).

Contact Cloudera Support for help converting your existing Hadoop configurations for use with Cloudera Manager.

1. Run this command on the Cloudera Manager Server host:

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

If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 406.

(Optional) Start the Cloudera Manager Agents

If you installed the Cloudera Manager Agent packages in [Install Cloudera Manager Agent Packages](#) on page 98, run this command on each Agent host:

```
sudo service cloudera-scm-agent start
```

When the Agent starts, it contacts the Cloudera Manager Server. If communication fails between a Cloudera Manager Agent and Cloudera Manager Server, see [Troubleshooting Installation and Upgrade Problems](#) on page 406.

When the Agent hosts reboot, `cloudera-scm-agent` starts automatically.

Start and Log into the Cloudera Manager Admin Console

The Cloudera Manager Server URL takes the following form `http://Server host:port`, where *Server host* is the fully-qualified domain name or IP address of the host where the Cloudera Manager Server is installed and *port* is the port configured for the Cloudera Manager Server. The default port is 7180.

1. Wait several minutes for the Cloudera Manager Server to complete its startup. To observe the startup process you can perform `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](#) on page 406.
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. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can [change the password](#) using Cloudera Manager after you run the installation wizard. While you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.

Choose Cloudera Manager Edition and Hosts

You can use the Cloudera Manager wizard to choose which edition of Cloudera Manager you are using and which hosts will run CDH and managed services.

1. When you start the Cloudera Manager Admin Console, the install wizard starts up. Click **Continue** to get started.
2. Choose which [edition](#) to install:

- Cloudera Express, which does not require a license, but provides a somewhat limited set of features.
- Cloudera Enterprise Data Hub Edition Trial, which does not require a license, but expires after 60 days and cannot be renewed
- Cloudera Enterprise with one of the following license types:
 - Basic Edition
 - Flex Edition
 - Data Hub Edition

If you choose Cloudera Express or Cloudera Enterprise Data Hub Edition Trial, you can elect to upgrade the license at a later time. See [Managing Licenses](#).

3. If you have elected Cloudera Enterprise, install a license:
 - a. Click **Upload License**.
 - b. Click the document icon to the left of the **Select a License File** text field.
 - c. Navigate to the location of your license file, click the file, and click **Open**.
 - d. Click **Upload**.

Click **Continue** to proceed with the installation.

4. Click **Continue** in the next screen. The **Specify Hosts** page displays.
5. Do one of the following:

- If you installed Cloudera Agent packages in [Install Cloudera Manager Agent Packages](#) on page 98, choose from among hosts with the packages installed:
 1. Click the **Currently Managed Hosts** tab.
 2. Choose the hosts to add to the cluster.
- Search for and choose hosts:
 1. To enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the cluster hostnames or IP addresses. You can also specify hostname and IP address ranges. For example:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.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. Use this technique to make more specific searches instead of searching overly wide ranges. The scan results will include all addresses scanned, but only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. If you don't know the IP addresses of all of the hosts, you can enter an address range that spans over unused addresses and then deselect the hosts that do not exist (and are not discovered) later in this procedure. However, keep in mind that wider ranges will require more time to scan.

2. Click **Search**. Cloudera Manager identifies the hosts on your cluster to allow you to configure them for services. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking **Abort Scan**. To find additional hosts, click **New Search**, add the host names or IP addresses and click **Search** again. Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts. Common causes of loss of connectivity are firewalls and interference from SELinux.

3. 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.

6. Click **Continue**. The **Select Repository** page displays.

Choose the Software Installation Method and Install Software

The following instructions describe how to use the Cloudera Manager wizard to install Cloudera Manager Agent, CDH, and managed service software.

1. Install CDH and managed service software using either packages or parcels:

- **Use Packages** - If you *did not* install packages in [Install CDH and Managed Service Packages](#) on page 99, click the package versions to install. Otherwise, select the CDH version (CDH 4 or CDH 5) that matches the packages that you installed manually.

- **Use Parcels**

1. Choose the parcels to install. The choices you see depend on the repositories you have chosen – a repository may contain multiple parcels. Only the parcels for the latest supported service versions are configured by default.

You can add additional parcels for previous versions by specifying custom repositories. For example, you can find the locations of the previous CDH 4 parcels at

`https://username:password@archive.cloudera.com/p/cdh4/parcels/`. Or, if you are installing CDH 4.3 and want to use [policy-file authorization](#), you can add the Sentry parcel using this mechanism.

1. To specify the parcel directory, local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:

- **Parcel Directory** and **Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host. If you change the default value for **Parcel Directory** and have already installed and started Cloudera Manager Agents, restart the Agents:

```
$ sudo service cloudera-scm-agent restart
```

- **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the **+** button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the [Configuring Cloudera Manager Server Parcel Settings](#) on page 78 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you will see all the unique parcels contained in all your repositories.
- **Proxy Server** - Specify the properties of a proxy server.

2. Click **OK**.

2. If you *did not* install Cloudera Manager Agent packages in [Install Cloudera Manager Agent Packages](#) on page 98, do the following:

- a. Select the release of Cloudera Manager Agent to install. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories. 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 leave deselected if you plan to install it yourself. If checked, your local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, select the **Install Java Unlimited Strength Encryption Policy Files** checkbox. Click **Continue**. The Enable Single User Mode screen displays.

4. (Optional) Select **Single User Mode** to configure the Cloudera Manager Agent and all service processes to run as the same user. This mode requires [extra configuration steps](#) that must be done manually on all hosts in the cluster.

If you have not performed the steps, directory creation will fail in the installation wizard. In most cases, you can create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**. The Provide SSH login credentials screen displays.

5. If you chose to have Cloudera Manager install packages, specify host installation properties:
 - Select **root** or enter the user name for an account that has password-less sudo permission.
 - Select an authentication method:
 - If you choose to use password authentication, enter and confirm the password.
 - If you choose to use public-key authentication provide a passphrase and path to the required key files.
 - You can choose to specify an alternate SSH port. The default value is 22.
 - You can specify the maximum number of host installations to run at once. The default value is 10.
6. Click **Continue**. If you *did not* install packages in [\(Optional\) Install Cloudera Manager Agent, CDH, and Managed Service Software](#) on page 98, Cloudera Manager installs the Oracle JDK, Cloudera Manager Agent, packages and CDH and managed service packages or parcels. During the parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels you will see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed. Click **Continue**.
7. 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. If the validation is successful, click **Finish**. The Cluster Setup screen displays.

Add Services

The following instructions describe how to use the Cloudera Manager wizard to configure and start CDH and managed services.

1. In the first page of the Add Services wizard you choose the combination of services to install and whether to install Cloudera Navigator:
 - Click the radio button next to the combination of services to install:

CDH 4	CDH 5
<ul style="list-style-type: none"> • Core Hadoop - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • Core with HBase • Core with Impala • All Services - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • Custom Services - Any combination of services. 	<ul style="list-style-type: none"> • Core Hadoop - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, and Sqoop • Core with HBase • Core with Impala • Core with Search • Core with Spark • All Services - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, Sqoop, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • Custom Services - Any combination of services.

As you select the services, keep the following in mind:

- Some services depend on other services; for example, HBase requires HDFS and ZooKeeper. Cloudera Manager tracks dependencies and installs the correct combination of services.
- In a Cloudera Manager deployment of a CDH 4 cluster, the MapReduce service is the default MapReduce computation framework. Choose **Custom Services** to install YARN or use the Add Service functionality to add YARN after installation completes.



Note: You can create a YARN service in a CDH 4 cluster, but it is not considered production ready.

- In a Cloudera Manager deployment of a CDH 5 cluster, the YARN service is the default MapReduce computation framework. Choose **Custom Services** to install MapReduce or use the Add Service functionality to add MapReduce after installation completes.



Note: In CDH 5, the MapReduce service has been deprecated. However, the MapReduce service is fully supported for backward compatibility through the CDH 5 life cycle.

- The Flume service can be added only after your cluster has been set up.
- If you have chosen Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See the [Cloudera Navigator Documentation](#).

Click **Continue**. The Customize Role Assignments screen displays.

2. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. These assignments are typically acceptable, but you can reassign them if necessary.

Click a field below a role to display a dialog containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts or **Custom** to display the pageable hosts dialog.

The following shortcuts for specifying hostname patterns are supported:

- Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

3. When you are satisfied with the assignments, click **Continue**. The Database Setup screen displays.
4. On the Database Setup page, configure settings for required databases:
 - a. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
 - b. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.) The Review Changes screen displays.
5. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed.



Warning: DataNode data directories should not be placed on NAS devices.

Click **Continue**. The wizard starts the services.

6. When all of the services are started, click **Continue**. You will see a success message indicating that your cluster has been successfully started.

7. Click **Finish** to proceed to the [Cloudera Manager Admin Console Home Page](#).

Change the Default Administrator Password

As soon as possible after running the wizard and beginning to use Cloudera Manager, change the default administrator password:

1. Right-click the logged-in username at the far right of the top navigation bar and select **Change Password**.
2. Enter the current password and a new password twice, and then click **Update**.

Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 146.

Installation Path C - Manual Installation Using Cloudera Manager Tarballs

Before proceeding with this path for a new installation, review [Cloudera Manager Deployment](#) on page 34. If you are upgrading a Cloudera Manager existing installation, see [Upgrading Cloudera Manager](#).

To avoid using system packages, and to use tarballs and parcels instead, follow the instructions in this section.



Important: As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.

This can affect new installations, upgrades, adding new hosts to a cluster, and adding a new cluster.

For more information, see [Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication](#).

Before You Begin

Install the Oracle JDK

See [Java Development Kit Installation](#) on page 35.

Install and Configure Databases

Read [Cloudera Manager and Managed Service Data Stores](#) on page 38. If you are using an external database, install and configure a database as described in [MySQL Database](#) on page 48, [Oracle Database](#) on page 53, or [External PostgreSQL Database](#) on page 45.

(CDH 5 only) On RHEL 5 and CentOS 5, Install Python 2.6 or 2.7

CDH 5 Hue will only work with the default Python version of the operating system on which it is being installed. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue. However, RHEL 5 and CentOS 5 users will have to download Python 2.6 from the EPEL repository as described below.

To install packages from the EPEL repository, download the appropriate repository rpm packages to your machine and then install Python using `yum`. For example, use the following commands for RHEL 5 or CentOS 5:

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

Install the Cloudera Manager Server and Agents

Tarballs contain both the Cloudera Manager Server and Cloudera Manager Agent in a single file. Download tarballs from the locations listed in [Cloudera Manager Version and Download Information](#). Copy the tarballs and unpack them

on all hosts on which you intend to install Cloudera Manager Server and Cloudera Manager Agents, in a directory of your choosing. If necessary, create a new directory to accommodate the files you extract from the tarball. For instance, if `/opt/cloudera-manager` does not exist, create it using a command similar to:

```
$ sudo mkdir /opt/cloudera-manager
```

When you have a directory to which to extract the contents of the tarball, extract the contents. For example, to copy a tar file to your home directory and extract the contents of all tar files to the `/opt/` directory, use a command similar to the following:

```
$ sudo tar xzf cloudera-manager*.tar.gz -C /opt/cloudera-manager
```

The files are extracted to a subdirectory named according to the Cloudera Manager version being extracted. For example, files could extract to `/opt/cloudera-manager/cm-5.0/`. This full path is needed later and is referred to as *tarball_root* directory.

Perform Configuration Required by Single User Mode

If you choose to create a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 11.

Create Users

The Cloudera Manager Server and managed services need a user account to complete tasks. When installing Cloudera Manager from tarballs, you must create this user account on all hosts manually. Because Cloudera Manager Server and managed services are configured to use the user account `cloudera-scm` by default, creating a user with this name is the simplest approach. After creating such a user, it is automatically used after installation is complete.

To create a user `cloudera-scm`, use a command such as the following:

```
$ sudo useradd --system --home=/opt/cloudera-manager/cm-5.0/run/cloudera-scm-server  
--no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm
```

For the preceding `useradd` command, ensure the `--home` argument path matches your environment. This argument varies according to where you place the tarball and the version number varies among releases. For example, the `--home` location could be `/opt/cm-5.0/run/cloudera-scm-server`.

Create the Cloudera Manager Server Local Data Storage Directory

1. Create the following directory: `/var/lib/cloudera-scm-server`.
2. Change the owner of the directory so that the `cloudera-scm` user and group have ownership of the directory. For example:

```
$ sudo mkdir /var/log/cloudera-scm-server  
$ sudo chown cloudera-scm:cloudera-scm /var/log/cloudera-scm-server
```

Configure Cloudera Manager Agents

- On every Cloudera Manager Agent host, configure the Cloudera Manager Agent to point to the Cloudera Manager Server by setting the following properties in the `tarball_root/etc/cloudera-scm-agent/config.ini` configuration file:

Property	Description
<code>server_host</code>	Name of the host where Cloudera Manager Server is running.
<code>server_port</code>	Port on the host where Cloudera Manager Server is running.

- By default, a tarball install has a `var` subdirectory where state is stored that in a non-tarball install is stored in `/var`. Cloudera recommends that you reconfigure the tarball install to use an external directory as the `/var` equivalent (`/var` or any other directory outside the tarball) so that when you upgrade Cloudera Manager, the new tarball installation can access this state. Configure the install to use an external directory for storing state by editing `tarball_root/etc/default/cloudera-scm-agent` and setting the `CMF_VAR` variable to the location of the `/var` equivalent. If you don't reuse the state directory between different tarball installations, the potential exists for duplicate Cloudera Manager Agent entries to occur in the Cloudera Manager database.

Custom Cloudera Manager Users and Directories

Cloudera Manager is built to use a default set of directories and user accounts. You can use the default locations and accounts, but there is also the option to change these settings. In some cases, changing these settings is required. For most installations, you can skip ahead to [Configure a Database for the Cloudera Manager Server](#) on page 114. By default, Cloudera Manager services creates directories in `/var/log` and `/var/lib`. The directories the Cloudera Manager installer attempts to create are:

- `/var/log/cloudera-scm-headlamp`
- `/var/log/cloudera-scm-firehose`
- `/var/log/cloudera-scm-alertpublisher`
- `/var/log/cloudera-scm-eventserver`
- `/var/lib/cloudera-scm-headlamp`
- `/var/lib/cloudera-scm-firehose`
- `/var/lib/cloudera-scm-alertpublisher`
- `/var/lib/cloudera-scm-eventserver`
- `/var/lib/cloudera-scm-server`

If you are using a custom user and directory for Cloudera Manager, you must create these directories on the Cloudera Manager Server host and assign ownership of these directories to your user manually. Issues might arise if any of these directories already exist. The Cloudera Manager installer makes no changes to existing directories. In such a case, Cloudera Manager is unable to write to any existing directories for which it does not have proper permissions and services may not perform as expected. To resolve such situations, do one of the following:

- **Change ownership of existing directories:**

1. Change the directory owner to the Cloudera Manager user. If the Cloudera Manager user and group are `cloudera-scm` and you needed to take ownership of the headlamp log directory, you would issue a command similar to the following:

```
$ sudo chown -R cloudera-scm:cloudera-scm /var/log/cloudera-scm-headlamp
```

2. Repeat the process of using `chown` to change ownership for all existing directories to the Cloudera Manager user.

- **Use alternate directories for services:**

1. If the directories you plan to use do not exist, create them now. For example to create `/var/cm_logs/cloudera-scm-headlamp` for use by the `cloudera-scm` user, you might use the following commands:

```
sudo mkdir /var/cm_logs/cloudera-scm-headlamp
sudo chown cloudera-scm /var/cm_logs/cloudera-scm-headlamp
```

2. Connect to the Cloudera Manager Admin Console.
3. Select **Clusters > Cloudera Management Service**
4. Select **Scope > role name**.
5. Click the **Configuration** tab.
6. Enter a term in the **Search** field to find the settings to be changed. For example, you might enter `/var` or `directory`.

7. Update each value with the new locations for Cloudera Manager to use.



Note: The configuration property for the **Cloudera Manager Server Local Data Storage Directory** (default value is: `/var/lib/cloudera-scm-server`) is located on a different page:

1. Select **Administration > Settings**.
2. Type `directory` in the Search box.
3. Enter the directory path in the **Cloudera Manager Server Local Data Storage Directory** property.

8. Click **Save Changes** to commit the changes.

Configure a Database for the Cloudera Manager Server

Depending on whether you are using an external database, or the embedded PostgreSQL database, do one of the following:

- External database - Prepare the Cloudera Manager Server database as described in [Preparing a Cloudera Manager Server External Database](#) on page 40.
- Embedded database - Install an embedded PostgreSQL database as described in [Installing and Starting the Cloudera Manager Server Embedded Database](#) on page 39.

Create Parcel Directories

1. On the Cloudera Manager Server host, create a parcel repository directory:

```
$ sudo mkdir -p /opt/cloudera/parcel-repo
```

2. Change the directory ownership to be the username you are using to run Cloudera Manager:

```
$ sudo chown username:groupname /opt/cloudera/parcel-repo
```

where *username* and *groupname* are the user and group names (respectively) you are using to run Cloudera Manager. For example, if you use the default username `cloudera-scm`, you would give the command:

```
$ chown cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo
```

3. On each cluster host, create a parcels directory:

```
$ sudo mkdir -p /opt/cloudera/parcels
```

4. Change the directory ownership to be the username you are using to run Cloudera Manager:

```
$ sudo chown username:groupname /opt/cloudera/parcels
```

where *username* and *groupname* are the user and group names (respectively) you are using to run Cloudera Manager. For example, if you use the default username `cloudera-scm`, you would give the command:

```
$ sudo chown cloudera-scm:cloudera-scm /opt/cloudera/parcels
```

Start the Cloudera Manager Server



Important: When you start the Cloudera Manager Server and Agents, Cloudera Manager assumes you are not already running HDFS and MapReduce. If these services are running:

1. Shut down HDFS and MapReduce. See [Stopping Services](#) (CDH 4) or [Stopping CDH Services Using the Command Line](#) (CDH 5) for the commands to stop these services.
2. Configure the init scripts to *not* start on boot. Use commands similar to those shown in [Configuring init to Start Core Hadoop System Services](#) (CDH 4) or [Configuring init to Start Hadoop System Services](#) (CDH 5), but *disable* the start on boot (for example, `$ sudo chkconfig hadoop-hdfs-namenode off`).

Contact Cloudera Support for help converting your existing Hadoop configurations for use with Cloudera Manager.

The way in which you start the Cloudera Manager Server varies according to what account you want the server to run under:

- As root:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-server start
```

- As another user. If you run as another user, ensure the user you created for Cloudera Manager owns the location to which you extracted the tarball including the newly created database files. If you followed the earlier examples and created the directory `/opt/cloudera-manager` and the user `cloudera-scm`, you could use the following command to change ownership of the directory:

```
$ sudo chown -R cloudera-scm:cloudera-scm /opt/cloudera-manager
```

Once you have established ownership of directory locations, you can start Cloudera Manager Server using the user account you chose. For example, you might run the Cloudera Manager Server as `cloudera-service`. In such a case there are following options:

- Run the following command:

```
$ sudo -u cloudera-service tarball_root/etc/init.d/cloudera-scm-server start
```

- Edit the configuration files so the script internally changes the user. Then run the script as root:

1. Remove the following line from `tarball_root/etc/default/cloudera-scm-server`:

```
export CMF_SUDO_CMD=" "
```

2. Change the user and group in `tarball_root/etc/init.d/cloudera-scm-server` to the user you want the server to run as. For example, to run as `cloudera-service`, change the user and group as follows:

```
USER=cloudera-service
GROUP=cloudera-service
```

3. Run the server script as root:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-server start
```

- To start the Cloudera Manager Server automatically after a reboot:
 1. Run the following commands on the Cloudera Manager Server host:

- **RHEL-compatible and SLES**

```
$ cp tarball_root/etc/init.d/cloudera-scm-server /etc/init.d/cloudera-scm-server
$ chkconfig cloudera-scm-server on
```

- **Debian/Ubuntu**

```
$ cp tarball_root/etc/init.d/cloudera-scm-server /etc/init.d/cloudera-scm-server
$ update-rc.d cloudera-scm-server defaults
```

2. On the Cloudera Manager Server host, open the `/etc/init.d/cloudera-scm-server` file and change the value of `CMF_DEFAULTS` from `/${CMF_DEFAULTS:-/etc/default}` to `tarball_root/etc/default`.

If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 406.

Start the Cloudera Manager Agents

The way in which you start the Cloudera Manager Agent varies according to what account you want the Agent to run under:

- To start the Cloudera Manager Agent, run this command on each Agent host:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent start
```

When the Agent starts, it contacts the Cloudera Manager Server.

- If you are running [single user mode](#), start Cloudera Manager Agent using the user account you chose. For example, you might run the Cloudera Manager Agent as `cloudera-scm`. In such a case there are following options:
 - Run the following command:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent start
```

- Edit the configuration files so the script internally changes the user, then run the script as root:

1. Remove the following line from `tarball_root/etc/default/cloudera-scm-agent`:

```
export CMF_SUDO_CMD=" "
```

2. Change the user and group in `tarball_root/etc/init.d/cloudera-scm-agent` to the user you want the Agent to run as. For example, to run as `cloudera-scm`, change the user and group as follows:

```
USER=cloudera-scm
GROUP=cloudera-scm
```

3. Run the Agent script as root:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent start
```

- To start the Cloudera Manager Agents automatically after a reboot:

1. Run the following commands on each Agent host:

- **RHEL-compatible and SLES**

```
$ cp tarball_root/etc/init.d/cloudera-scm-agent /etc/init.d/cloudera-scm-agent
$ chkconfig cloudera-scm-agent on
```

- **Debian/Ubuntu**

```
$ cp tarball_root/etc/init.d/cloudera-scm-agent /etc/init.d/cloudera-scm-agent
$ update-rc.d cloudera-scm-agent defaults
```

2. On each Agent, open the `tarball_root/etc/init.d/cloudera-scm-agent` file and change the value of `CMF_DEFAULTS` from `${CMF_DEFAULTS:-/etc/default}` to `tarball_root/etc/default`.

Install Dependencies

When installing with tarballs and parcels, some services may require additional dependencies that are not provided by Cloudera. On each host, install required packages:

- **Red-hat compatible**
 - chkconfig
 - python (2.7 required for CDH 5)
 - bind-utils
 - psmisc
 - libxslt
 - zlib
 - sqlite
 - cyrus-sasl-plain
 - cyrus-sasl-gssapi
 - fuse
 - portmap
 - fuse-libs
 - redhat-lsb
- **SLES**
 - chkconfig
 - python (2.7 required for CDH 5)
 - bind-utils
 - psmisc
 - libxslt
 - zlib
 - sqlite
 - cyrus-sasl-plain
 - cyrus-sasl-gssapi
 - fuse
 - portmap
 - python-xml
 - libfuse2
- **Debian/Ubuntu**
 - lsb-base
 - psmisc
 - bash
 - libsasl2-modules
 - libsasl2-modules-gssapi-mit
 - zlib1g
 - libxslt1.1
 - libsqlite3-0

Installing Cloudera Manager and CDH

- libfuse2
- fuse-utils or fuse
- rpcbind

Start and Log into the Cloudera Manager Admin Console

The Cloudera Manager Server URL takes the following form `http://Server host:port`, where *Server host* is the fully-qualified domain name or IP address of the host where the Cloudera Manager Server is installed and *port* is the port configured for the Cloudera Manager Server. The default port is 7180.

1. Wait several minutes for the Cloudera Manager Server to complete its startup. To observe the startup process you can perform `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](#) on page 406.
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. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can [change the password](#) using Cloudera Manager after you run the installation wizard. While you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.

Choose Cloudera Manager Edition and Hosts

1. When you start the Cloudera Manager Admin Console, the install wizard starts up. Click **Continue** to get started.
2. Choose which [edition](#) to install:
 - Cloudera Express, which does not require a license, but provides a somewhat limited set of features.
 - Cloudera Enterprise Data Hub Edition Trial, which does not require a license, but expires after 60 days and cannot be renewed
 - Cloudera Enterprise with one of the following license types:
 - Basic Edition
 - Flex Edition
 - Data Hub Edition

If you choose Cloudera Express or Cloudera Enterprise Data Hub Edition Trial, you can elect to upgrade the license at a later time. See [Managing Licenses](#).

3. If you have elected Cloudera Enterprise, install a license:
 - a. Click **Upload License**.
 - b. Click the document icon to the left of the **Select a License File** text field.
 - c. Navigate to the location of your license file, click the file, and click **Open**.
 - d. Click **Upload**.

Click **Continue** to proceed with the installation.

4. Click **Continue** in the next screen. The **Specify Hosts** page displays.
5. Click the **Currently Managed Hosts** tab.
6. Choose the hosts to add to the cluster.
7. Click **Continue**. The **Select Repository** page displays.

Choose Software Installation Method and Install Software

1. Click **Use Parcels** to install CDH and managed services using parcels and then do the following:
 - a. **Use Parcels**

- a. Choose the parcels to install. The choices you see depend on the repositories you have chosen – a repository may contain multiple parcels. Only the parcels for the latest supported service versions are configured by default.

You can add additional parcels for previous versions by specifying custom repositories. For example, you can find the locations of the previous CDH 4 parcels at

<https://username:password@archive.cloudera.com/p/cdh4/parcels/>. Or, if you are installing CDH 4.3 and want to use [policy-file authorization](#), you can add the Sentry parcel using this mechanism.

1. To specify the parcel directory, local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:

- **Parcel Directory and Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host. If you change the default value for **Parcel Directory** and have already installed and started Cloudera Manager Agents, restart the Agents:

```
$ sudo service cloudera-scm-agent restart
```

- **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the **+** button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the [Configuring Cloudera Manager Server Parcel Settings](#) on page 78 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you will see all the unique parcels contained in all your repositories.
- **Proxy Server** - Specify the properties of a proxy server.

2. Click **OK**.

- b. Click **Continue**. Cloudera Manager installs the CDH and managed service parcels. During the parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels you will see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed. Click **Continue**.

2. 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. If the validation is successful, click **Finish**. The Cluster Setup screen displays.

Add Services

The following instructions describe how to use the Cloudera Manager wizard to configure and start CDH and managed services.

1. In the first page of the Add Services wizard you choose the combination of services to install and whether to install Cloudera Navigator:

- Click the radio button next to the combination of services to install:

CDH 4	CDH 5
<ul style="list-style-type: none"> • Core Hadoop - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • Core with HBase • Core with Impala • All Services - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • Custom Services - Any combination of services. 	<ul style="list-style-type: none"> • Core Hadoop - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, and Sqoop • Core with HBase • Core with Impala • Core with Search • Core with Spark • All Services - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, Sqoop, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • Custom Services - Any combination of services.

As you select the services, keep the following in mind:

- Some services depend on other services; for example, HBase requires HDFS and ZooKeeper. Cloudera Manager tracks dependencies and installs the correct combination of services.
- In a Cloudera Manager deployment of a CDH 4 cluster, the MapReduce service is the default MapReduce computation framework. Choose **Custom Services** to install YARN or use the Add Service functionality to add YARN after installation completes.



Note: You can create a YARN service in a CDH 4 cluster, but it is not considered production ready.

- In a Cloudera Manager deployment of a CDH 5 cluster, the YARN service is the default MapReduce computation framework. Choose **Custom Services** to install MapReduce or use the Add Service functionality to add MapReduce after installation completes.



Note: In CDH 5, the MapReduce service has been deprecated. However, the MapReduce service is fully supported for backward compatibility through the CDH 5 life cycle.

- The Flume service can be added only after your cluster has been set up.
- If you have chosen Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See the [Cloudera Navigator Documentation](#).

Click **Continue**. The Customize Role Assignments screen displays.

2. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. These assignments are typically acceptable, but you can reassign them if necessary.

Click a field below a role to display a dialog containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts or **Custom** to display the pageable hosts dialog.

The following shortcuts for specifying hostname patterns are supported:

- Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

3. When you are satisfied with the assignments, click **Continue**. The Database Setup screen displays.
4. On the Database Setup page, configure settings for required databases:
 - a. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
 - b. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.) The Review Changes screen displays.

- Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed.



Warning: DataNode data directories should not be placed on NAS devices.

Click **Continue**. The wizard starts the services.

- When all of the services are started, click **Continue**. You will see a success message indicating that your cluster has been successfully started.
- Click **Finish** to proceed to the [Cloudera Manager Admin Console Home Page](#).

(Optional) Change the Cloudera Manager User

After configuring your services, the installation wizard attempts to automatically start the Cloudera Management Service under the assumption that it will run using `cloudera-scm`. If you configured this service to run using a user other than `cloudera-scm`, then the Cloudera Management Service roles do not start automatically. In such a case, change the service configuration to use the user account that you selected:

- Connect to the Cloudera Manager Admin Console.
- Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the Home page, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
- Click the **Configuration** tab.
- Use the search box to find the property to be changed. For example, you might enter "system" to find the **System User** and **System Group** properties.
- Make any changes required to the System User and System Group to ensure Cloudera Manager uses the proper user accounts.
- Click **Save Changes**.

After making this configuration change, manually start the Cloudera Management Service roles.

Change the Default Administrator Password

As soon as possible after running the wizard and beginning to use Cloudera Manager, change the default administrator password:

- Right-click the logged-in username at the far right of the top navigation bar and select **Change Password**.
- Enter the current password and a new password twice, and then click **Update**.

Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 146.

Installing Impala

Cloudera Impala is included with CDH 5. To use Cloudera Impala with CDH 4, you must install both CDH and Impala on the hosts that will run Impala.



Note:

- See [Supported CDH and Managed Service Versions](#) on page 7 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 34.

Installing Cloudera Manager and CDH

Installing Impala after Upgrading Cloudera Manager

If you have just upgraded Cloudera Manager from a version that did not support Impala, the Impala software is not installed automatically. (Upgrading Cloudera Manager does not automatically upgrade CDH or other managed services). You can add Impala using parcels; go to the **Hosts** tab, and select the **Parcels** tab. If you have installed CDH 4 you should see at least one Impala parcel available for download. See [Parcels](#) on page 70 for detailed instructions on using parcels to install or upgrade Impala. If you do not see any Impala parcels available, click the **Edit Settings** button on the **Parcels** page to go to the Parcel configuration settings and verify that the Impala parcel repo URL (<https://archive.cloudera.com/impala/parcels/latest/>) has been configured in the **Parcels** configuration page. See [Parcel Configuration Settings](#) on page 78 for more details.

Post Installation Configuration

See [The Impala Service](#) for instructions on configuring the Impala service.

Installing Search

Cloudera Search is provided by the Solr service. The Solr service is included with CDH 5. To use Cloudera Search with CDH 4, you must install both CDH and Search on the hosts that will run Search.



Note:

- See [Supported CDH and Managed Service Versions](#) on page 7 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 34.

Installing Search after Upgrading Cloudera Manager

If you have just upgraded Cloudera Manager from a version that did not support Search, the Search software is not installed automatically. (Upgrading Cloudera Manager does not automatically upgrade CDH or other managed services). You can add Search using parcels; go to the **Hosts** tab, and select the **Parcels** tab. You should see at least one Solr parcel available for download. See [Parcels](#) on page 70 for detailed instructions on using parcels to install or upgrade Solr. If you do not see any Solr parcels available, click the **Edit Settings** button on the **Parcels** page to go to the Parcel configuration settings and verify that the Search parcel repo URL. The URL should point to the subdirectory of <https://archive.cloudera.com/cdh5/parcels/> that corresponds to the release configured in the **Parcels** configuration page. See [Parcel Configuration Settings](#) on page 78 for more details.

Post Installation Configuration

See [The Solr Service](#) for instructions on configuring Cloudera Search.

Installing Spark

[Apache Spark](#) is included with CDH 5. To use Apache Spark with CDH 4, you must install both CDH and Spark on the hosts that will run Spark.



Note:

- See [Supported CDH and Managed Service Versions](#) on page 7 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 34.

Installing Spark after Upgrading Cloudera Manager

If you have just upgraded Cloudera Manager from a version that did not support Spark, the Spark software is not installed automatically. (Upgrading Cloudera Manager does not automatically upgrade CDH or other managed services).

You can add Spark using parcels; go to the **Hosts** tab, and select the **Parcels** tab. You should see at least one Spark parcel available for download. See [Parcels](#) on page 70 for detailed instructions on using parcels to install or upgrade Spark. If you do not see any Spark parcels available, click the **Edit Settings** button on the **Parcels** page to go to the Parcel configuration settings and verify that the Spark parcel repo URL (<https://archive.cloudera.com/spark/parcels/latest/>) has been configured in the **Parcels** configuration page. See [Parcel Configuration Settings](#) on page 78 for more details.

Post Installation Configuration

See [Managing Spark Using Cloudera Manager](#) for instructions on adding the Spark service.

Installing KMS (Navigator Key Trustee)

KMS (Navigator Key Trustee) is a new service in Cloudera Manager 5.3.0 and CDH 5.3.0. It is a custom Key Management Service (KMS) that uses Cloudera Navigator Key Trustee Server as the underlying key store, rather than the file-based Java KeyStore (JKS) used by the default Hadoop KMS.

To use the **KMS (Navigator Key Trustee)** service, you must first install the Key Trustee binaries.



Note:

- See [Supported CDH and Managed Service Versions](#) on page 7 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 34.

Installing KMS (Navigator Key Trustee) after Upgrading Cloudera Manager



Important: Following these instructions will install the required software to add the **KMS (Navigator Key Trustee)** service to your cluster; this enables you to use an existing Cloudera Navigator Key Trustee Server as the underlying key store for [HDFS Data At Rest Encryption](#). This *does not* install Cloudera Navigator Key Trustee Server. Contact your account team for assistance installing Cloudera Navigator Key Trustee Server.

If you have just upgraded Cloudera Manager from a version that did not support KMS (Navigator Key Trustee), the Key Trustee binaries are not installed automatically. (Upgrading Cloudera Manager does not automatically upgrade CDH or other managed services). You can add the Key Trustee binaries using parcels; go to the **Hosts** tab, and select the **Parcels** tab. You should see at least one Key Trustee parcel available for download. See [Parcels](#) on page 70 for detailed instructions on using parcels to install or upgrade Key Trustee. If you do not see any Key Trustee parcels available, click the **Edit Settings** button on the **Parcels** page to go to the Parcel configuration settings and verify that the Key Trustee parcel repo URL (<https://archive.cloudera.com/navigator-keytrustee5/parcels/latest/>) has been configured in the **Parcels** configuration page. See [Parcel Configuration Settings](#) on page 78 for more details.

If your cluster is installed using packages, see [\(Optional\) Install Key Trustee Key Provider](#) on page 105 for instructions on how to install the required software.

Post Installation Configuration

Contact your account team for assistance configuring **KMS (Navigator Key Trustee)** to communicate with an existing Key Trustee Server, or for assistance installing and configuring a new Key Trustee Server.

Installing GPL Extras

GPL Extras contains LZO functionality.

To install the GPL Extras parcel:

1. Add the appropriate repository to the Cloudera Manager list of [parcel repositories](#). The public repositories can be found at:

- **CDH 5** - <https://archive.cloudera.com/gplextras5/parcels/latest>
- **CDH 4** - <https://archive.cloudera.com/gplextras/parcels/latest>

If you are using LZO with Impala, you must choose a specific version of the GPL Extras parcel for the Impala version according to the following tables:

Table 20: CDH 5

Impala Version	Parcels Version Subdirectory	GPL Extras Parcel Version
CDH 5.x.y	5.x.y/	GPLEXTRAS-5.x.y

Table 21: CDH 4

Impala Version	Parcels Version Subdirectory	GPL Extras Parcel Version
2.1.0	0.4.15.101/	HADOOP_LZO-0.4.15-1.gplextras.p0.101
2.0.0	0.4.15.101/	HADOOP_LZO-0.4.15-1.gplextras.p0.101
1.4.0	0.4.15.85/	HADOOP_LZO-0.4.15-1.gplextras.p0.85
1.3.1	0.4.15.64/	HADOOP_LZO-0.4.15-1.gplextras.p0.64
1.2.4	0.4.15.58/	HADOOP_LZO-0.4.15-1.gplextras.p0.58
1.2.3	0.4.15.39/	HADOOP_LZO-0.4.15-1.gplextras.p0.39
1.2.2	0.4.15.37/	HADOOP_LZO-0.4.15-1.gplextras.p0.37
1.2.1	0.4.15.33/	HADOOP_LZO-0.4.15-1.gplextras.p0.33

To create the repository URL, append the version directory to the URL (CDH 4)

<https://archive.cloudera.com/gplextras/parcels/> or (CDH 5)

<https://archive.cloudera.com/gplextras5/parcels/> respectively. For example:

<https://archive.cloudera.com/gplextras5/parcels/5.0.2>.

2. Download, distribute, and activate the parcel.
3. If not already installed, on all cluster hosts, install the `lzo` package on RHEL or the `liblzo2-2` package on SLES, Debian, or Ubuntu:

RedHat:

```
sudo yum install lzo
```

Debian or Ubuntu:

```
sudo apt-get install liblzo2-2
```

SLES:

```
sudo zypper install liblzo2-2
```

Understanding Custom Installation Solutions

Cloudera hosts two types of software repositories that you can use to install products such as Cloudera Manager or CDH—parcel repositories and RHEL and SLES RPM and Debian/Ubuntu package repositories.

These repositories are effective solutions in most cases, but custom installation solutions are sometimes required. Using the software repositories requires client access over the Internet and results in the installation of the latest version of products. An alternate solution is required if:

- You need to install older product versions. For example, in a CDH cluster, all hosts must run the same CDH version. After completing an initial installation, you may want to add hosts. This could be to increase the size of your cluster to handle larger tasks or to replace older hardware.
- The hosts on which you want to install Cloudera products are not connected to the Internet, so they are unable to reach the Cloudera repository. (For a parcel installation, only the Cloudera Manager Server needs Internet access, but for a package installation, all cluster members need access to the Cloudera repository). Some organizations choose to partition parts of their network from outside access. Isolating segments of a network can provide greater assurance that valuable data is not compromised by individuals out of maliciousness or for personal gain. In such a case, the isolated computers are unable to access Cloudera repositories for new installations or upgrades.

In both of these cases, using a custom repository solution allows you to meet the needs of your organization, whether that means installing older versions of Cloudera software or installing any version of Cloudera software on hosts that are disconnected from the Internet.

Understanding Parcels

Parcels are a packaging format that facilitate upgrading software from within Cloudera Manager. You can download, distribute, and activate a new software version all from within Cloudera Manager. Cloudera Manager downloads a parcel to a local directory. Once the parcel is downloaded to the Cloudera Manager Server host, an Internet connection is no longer needed to deploy the parcel. Parcels are available for CDH 4.1.3 and onwards. For detailed information about parcels, see [Parcels](#) on page 70.

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel files and put them into a parcel repository. See [Creating and Using a Parcel Repository for Cloudera Manager](#) on page 126.

Understanding Package Management

Before getting into the details of how to configure a custom package management solution in your environment, it can be useful to have more information about:

- Package management tools
- Package repositories

Package Management Tools

Packages (`rpm` or `deb` files) help ensure that installations complete successfully by encoding each package's dependencies. That means that if you request the installation of a solution, all required elements can be installed at the same time. For example, `hadoop-0.20-hive` depends on `hadoop-0.20`. Package management tools, such as `yum` (RHEL), `zypper` (SLES), and `apt-get` (Debian/Ubuntu) are tools that can find and install any required packages. For example, for RHEL, you might enter `yum install hadoop-0.20-hive`. `yum` would inform you that the `hive` package requires `hadoop-0.20` and offers to complete that installation for you. `zypper` and `apt-get` provide similar functionality.

Package Repositories

Package management tools operate on package repositories.

Repository Configuration Files

Information about package repositories is stored in configuration files, the location of which varies according to the package management tool.

- RedHat/CentOS `yum - /etc/yum.repos.d`
- SLES `zypper - /etc/zypp/zypper.conf`

Installing Cloudera Manager and CDH

- Debian/Ubuntu `apt-get - /etc/apt/apt.conf` (Additional repositories are specified using `*.list` files in the `/etc/apt/sources.list.d/` directory.)

For example, on a typical CentOS system, you might find:

```
[user@localhost ~]$ ls -l /etc/yum.repos.d/
total 24
-rw-r--r-- 1 root root 2245 Apr 25 2010 CentOS-Base.repo
-rw-r--r-- 1 root root 626 Apr 25 2010 CentOS-Media.repo
```

The `.repo` files contain pointers to one or many repositories. There are similar pointers inside configuration files for `zypper` and `apt-get`. In the following snippet from `CentOS-Base.repo`, there are two repositories defined: one named `Base` and one named `Updates`. The `mirrorlist` parameter points to a website that has a list of places where this repository can be downloaded.

```
# ...
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
# ...
```

Listing Repositories

You can list the repositories you have enabled. The command varies according to operating system:

- RedHat/CentOS - `yum repolist`
- SLES - `zypper repos`
- Debian/Ubuntu - `apt-get` does not include a command to display sources, but you can determine sources by reviewing the contents of `/etc/apt/sources.list` and any files contained in `/etc/apt/sources.list.d/`.

The following shows an example of what you might find on a CentOS system in `repolist`:

```
[root@localhost yum.repos.d]$ yum repolist
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* addons: mirror.san.fastserv.com
* base: centos.eecs.wsu.edu
* extras: mirrors.ecvps.com
* updates: mirror.5ninesolutions.com
repo id                repo name                status
addons                 CentOS-5 - Addons       enabled:
0
base                   CentOS-5 - Base         enabled: 3,434
extras                 CentOS-5 - Extras       enabled: 296
updates                CentOS-5 - Updates      enabled: 1,137
repolist: 4,867
```

Creating and Using a Parcel Repository for Cloudera Manager

This topic describes how to create a repository and then how to direct hosts in your environment to use that repository. To create a repository, you simply put the parcel files you want to host in one directory. Then publish the resulting repository on a website. There are two options for publishing the directory:

- [Install a Web Server](#) on page 127
- [Create a Temporary Local Repository](#) on page 128



Important: As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.

This can affect new installations, upgrades, adding new hosts to a cluster, and adding a new cluster.

For more information, see [Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication](#).

Install a Web Server

The repository is typically hosted using HTTP on a host inside your network. If you already have a web server in your organization, you can move the repository directory, which will include both the RPMs and the `repodata/` subdirectory, to a location hosted by the web server. An easy web server to install is the Apache HTTPD. If you are able to use an existing web server, then note the URL and skip to [Download Parcel and Publish Files](#) on page 127.

Installing Apache HTTPD

You may need to respond to some prompts to confirm you want to complete the installation.

OS	Command
RHEL	<code>[root@localhost yum.repos.d]\$ yum install httpd</code>
SLES	<code>[root@localhost zypp]\$ zypper install httpd</code>
Ubuntu or Debian	<code>[root@localhost apt]\$ apt-get install httpd</code>

Starting Apache HTTPD

OS	Command
RHEL	<code>[root@localhost tmp]\$ service httpd start</code> Starting httpd: [OK]
SLES	<code>[root@localhost tmp]\$ service apache2 start</code> Starting httpd: [OK]
Ubuntu or Debian	<code>[root@localhost tmp]\$ service apache2 start</code> Starting httpd: [OK]

Download Parcel and Publish Files

1. Download the parcel and `manifest.json` files for your OS distribution from

- **CDH 5** - Impala, Spark, and Search are included in the CDH parcel.
 - CDH - <https://username:password@archive.cloudera.com/p/cdh5/parcels/>
 - GPL Extras - <https://archive.cloudera.com/p/gplextras5/parcels/>
- **Other services**
 - Accumulo - <https://username:password@archive.cloudera.com/p/accumulo/parcels/>
 - Sqoop connectors - <https://username:password@archive.cloudera.com/p/sqoop-connectors/parcels/>

2. Move the `.parcel` and `manifest.json` files to the web server directory, and modify file permissions. For example, you might use the following commands:

```
[root@localhost tmp]$ mkdir /var/www/html/cdh4.6
[root@localhost tmp]$ mv CDH-4.6.0-1.cdh4.6.0.p0.26-lucid.parcel /var/www/html/cdh4.6
[root@localhost tmp]$ mv manifest.json /var/www/html/cdh4.6
[root@localhost tmp]$ chmod -R ugo+rX /var/www/html/cdh4.6
```

After moving the files and changing permissions, visit `http://hostname:80/cdh4.6/` to verify that you can access the parcel. Apache may have been configured to not show indexes, which is also acceptable.

Create a Temporary Local Repository

Alternatively you can quickly create a temporary local repository to deploy a parcel once. It is convenient to perform this on the same host that runs Cloudera Manager, or a gateway role. In this example, [python SimpleHTTPServer](#) is used from a directory of your choosing.

1. Download the patched `.parcel` and `manifest.json` files as provided in a secure link from Cloudera Support.
2. Copy the `.parcel` and `manifest.json` files to a location of your choosing on your server. This is the directory from which the python SimpleHTTPServer will serve the files. For example:

```
$ mkdir /tmp/parcel
$ cp /home/user/Downloads/patchparcel/CDH-4.5.0.p234.parcel /tmp/parcel/
$ cp /home/user/Downloads/patchparcel/manifest.json /tmp/parcel/
```

3. Determine a port that your system is not listening on (for example, port 8900).
4. Change to the directory containing the `.parcel` and `manifest.json` files.

```
$ cd /tmp/parcel
```

5. Start a python SimpleHTTPServer to host these two files:


```
$ python -m SimpleHTTPServer 8900
Serving HTTP on 0.0.0.0 port 8900 ...
```

6. Confirm you can get to this hosted parcel directory by going to `http://server:8900` in your browser. You should see links for the hosted files.

Configure the Cloudera Manager Server to Use the Parcel URL

1. Use one of the following methods to open the parcel settings page:

- **Navigation bar**


1. Click  in the top navigation bar
2. Click the **Edit Settings** button.

- **Menu**

1. Select **Administration > Settings**.
2. Click the **Parcels** category.

- **Tab**

1. Click the **Hosts** tab.
2. Click the **Configuration** tab.
3. Click the **Parcels** category.
4. Click the **Edit Settings** button.

2. In the **Remote Parcel Repository URLs** list, click  to open an additional row.
3. Enter the path to the parcel. For example, `http://hostname:port/cdh4.6/`.
4. Click **Save Changes** to commit the changes.

Creating and Using a Package Repository for Cloudera Manager

This topic describes how to create a package repository and then how to direct hosts in your environment to use that repository. To create a repository, you simply put the repo files you want to host in one directory. Then publish the resulting repository on a website. There are two options for publishing the directory:

- [Install a Web Server](#) on page 129
- [Create a Temporary Local Repository](#) on page 130



Important: As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.

This can affect new installations, upgrades, adding new hosts to a cluster, and adding a new cluster.

For more information, see [Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication](#).

Install a Web Server

The repository is typically hosted using HTTP on a host inside your network. If you already have a web server in your organization, you can move the repository directory, which will include both the RPMs and the `repodata/` subdirectory, to some a location hosted by the web server. An easy web server to install is the Apache HTTPD. If you are able to use an existing web server, then note the URL and skip to [Download Tarball and Publish Repository Files](#) on page 129.

Installing Apache HTTPD

You may need to respond to some prompts to confirm you want to complete the installation.

OS	Command
RHEL	<code>[root@localhost yum.repos.d]\$ yum install httpd</code>
SLES	<code>[root@localhost zypp]\$ zypper install httpd</code>
Ubuntu or Debian	<code>[root@localhost apt]\$ apt-get install httpd</code>

Starting Apache HTTPD

OS	Command
RHEL	<code>[root@localhost tmp]\$ service httpd start</code> Starting httpd: [OK]
SLES	<code>[root@localhost tmp]\$ service apache2 start</code> Starting httpd: [OK]
Ubuntu or Debian	<code>[root@localhost tmp]\$ service apache2 start</code> Starting httpd: [OK]

Download Tarball and Publish Repository Files

1. Download the tarball for your OS distribution from the [repo as tarball archive](#).
2. Unpack the tarball, move the files to the web server directory, and modify file permissions. For example, you might use the following commands:

```
[root@localhost tmp]$ gunzip cm5.0.0-centos6.tar.gz
[root@localhost tmp]$ tar xvf cm5.0.0-centos6.tar
```

```
[root@localhost tmp]$ mv cm /var/www/html
[root@localhost tmp]$ chmod -R ugo+rX /var/www/html/cm
```

After moving files and changing permissions, visit `http://hostname:port/cm` to verify that you see an index of files. Apache may have been configured to not show indexes, which is also acceptable.

Create a Temporary Local Repository

Alternatively you can quickly create a temporary local repository to deploy a package once. It is convenient to perform this on the same host that runs Cloudera Manager, or a gateway role. In this example, [python SimpleHTTPServer](#) is used from a directory of your choosing.

1. Download the tarball for your OS distribution from the [repo as tarball archive](#).
2. Unpack the tarball and modify file permissions. For example, you might use the following commands:

```
[root@localhost tmp]$ gunzip cm5.0.0-centos6.tar.gz
[root@localhost tmp]$ tar xvf cm5.0.0-centos6.tar
[root@localhost tmp]$ chmod -R ugo+rX /tmp/cm
```

3. Determine a port that your system is not listening on (for example, port 8900).
4. Change to the directory containing the files.

```
$ cd /tmp/cm
```

5. Start a python SimpleHTTPServer to host these two files:

```
$ python -m SimpleHTTPServer 8900
Serving HTTP on 0.0.0.0 port 8900 ...
```

6. Confirm you can get to this hosted package directory by going to `http://server:8900/cm` in your browser. You should see links for the hosted files.

Modify Clients to Find Repository

Having established the repository, modify the clients so they find the repository.

OS	Command
RHEL	<p>Create files on client systems with the following information and format, where <i>hostname</i> is the name of the web server:</p> <pre>[myrepo] name=myrepo baseurl=http://hostname/cm/5 enabled=1 gpgcheck=0</pre> <p>See <code>man yum.conf</code> for more details. Put that file into <code>/etc/yum.repos.d/myrepo.repo</code> on all of your hosts to enable them to find the packages that you are hosting.</p>
SLES	<p>Use the <code>zypper</code> utility to update client system repo information by issuing the following command:</p> <pre>\$ zypper addrepo http://hostname/cm alias</pre>
Ubuntu or Debian	<p>Add a new <code>list</code> file to <code>/etc/apt/sources.list.d/</code> on client systems. For example, you might create the file <code>/etc/apt/sources.list.d/my-private-cloudera-repo.list</code>. In that file, create an entry to your newly created repository. For example:</p> <pre>\$ cat /etc/apt/sources.list.d/my-private-cloudera-repo.list deb http://hostname/cm cloudera</pre>

OS	Command
	<p>After adding your <code>.list</code> file, ensure <code>apt-get</code> uses the latest information by issuing the following command:</p> <pre>\$ sudo apt-get update</pre>

After completing these steps, you have established the environment necessary to install a previous version of Cloudera Manager or install Cloudera Manager to hosts that are not connected to the Internet. Proceed with the installation process, being sure to target the newly created repository with your package management tool.

Configuring a Custom Java Home Location

Java, which Cloudera services require, may be installed at a custom location. Follow the installation instructions in:

- CDH 5 - [Java Development Kit Installation](#) on page 35.
- CDH 4 - [Java Development Kit Installation](#).

If you choose to use a custom Java location, modify the host configuration to ensure the JDK can be found:

1. Open the Cloudera Manager Admin Console.
2. In the main navigation bar, click the **Hosts** tab and optionally click a specific host link.
3. Click the **Configuration** tab.
4. In the **Advanced** category, click the **Java Home Directory** property.
5. Set the property to the custom location.
6. Click **Save Changes**.
7. Restart all services.

If you do not update the configuration, Cloudera services will be unable to find this resource and will not start.

Installing Older Versions of Cloudera Manager 5

When you install Cloudera Manager—for example, by using the installer downloadable from the Cloudera Downloads website—the most recent version is installed by default. This ensures that you install the latest features and bug fixes. In some cases, however, you may want to install a previous version.

For example, you might install a previous version if you want to expand an existing cluster. In this case, follow the instructions in [Adding a Host to the Cluster](#).

You can also add a cluster to be managed by the same instance of Cloudera Manager by using the **Add Cluster** feature from the Services page in the Cloudera Manager Admin Console. Follow the instructions in [Adding a Cluster](#).

You may also want to install a previous version of the Cloudera Manager Server on a new cluster if, for example, you have validated a specific version and want to deploy that version on additional clusters. Installing an older version of Cloudera Manager requires several manual steps to install and configure the database and the correct version of the Cloudera Manager Server. After completing these steps, run the Installation wizard to complete the installation of Cloudera Manager and CDH.

Before You Begin

Install and Configure Databases

Cloudera Manager Server, Cloudera Management Service, and the Hive metastore data are stored in a database. Install and configure required databases following the instructions in [Cloudera Manager and Managed Service Data Stores](#) on page 38.

(CDH 5 only) On RHEL 5 and CentOS 5, Install Python 2.6 or 2.7

CDH 5 Hue will only work with the default Python version of the operating system on which it is being installed. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue. However, RHEL 5 and CentOS 5 users will have to download Python 2.6 from the EPEL repository as described below.

Installing Cloudera Manager and CDH

To install packages from the EPEL repository, download the appropriate repository rpm packages to your machine and then install Python using `yum`. For example, use the following commands for RHEL 5 or CentOS 5:

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

Establish Your Cloudera Manager Repository Strategy

- **Download and Edit the Repo File for RHEL-compatible OSs or SLES**
 1. Download the Cloudera Manager repo file (`cloudera-manager.repo`) for your OS version using the links provided on the [Cloudera Manager Version and Download Information](#) page. For example, for Red Hat/CentOS 6, the file is located at `https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo`.
 2. Edit the file to change `baseurl` to point to the version of Cloudera Manager you want to download. For example, to install Cloudera Manager version 5.0.1, change:
`baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/` to
`baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5.0.1/`.
 3. Save the edited file:
 - For Red Hat or CentOS, save it in `/etc/yum.repos.d/`.
 - For SLES, save it in `/etc/zypp/repos.d`.
- **Download and Edit the `cloudera.list` file for Debian or Apt**
 1. Download the Cloudera Manager list file (`cloudera.list`) using the links provided at [Cloudera Manager Version and Download Information](#). For example, for Ubuntu 10.04 (lucid), this file is located at `https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm/cloudera.list`.
 2. Edit the file to change the second-to-last element to specify the version of Cloudera Manager you want to install. For example, with Ubuntu lucid, if you want to install Cloudera Manager version 5.0.1, change: `deb https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm lucid-cm5 contrib` to `deb https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm lucid-cm5.0.1 contrib`.
 3. Save the edited file in the directory `/etc/apt/sources.list.d/`.

Install the Oracle JDK

Install the Oracle Java Development Kit (JDK) on the Cloudera Manager Server host.

The JDK is included in the Cloudera Manager 5 repositories. After downloading and editing the repo or list file, install the JDK as follows:

OS	Command
RHEL	<code>\$ sudo yum install oracle-j2sdk1.7</code>
SLES	<code>\$ sudo zypper install oracle-j2sdk1.7</code>
Ubuntu or Debian	<code>\$ sudo apt-get install oracle-j2sdk1.7</code>

Install the Cloudera Manager Server Packages

1. Install the Cloudera Manager Server packages either on the host where the database is installed, or on a host that has access to the database. This host need not be a host in the cluster that you want to manage with Cloudera Manager. On the Cloudera Manager Server host, type the following commands to install the Cloudera Manager packages.

OS	Command
RHEL, if you have a yum repo configured	<code>\$ sudo yum install cloudera-manager-daemons cloudera-manager-server</code>

OS	Command
RHEL, if you're manually transferring RPMs	<pre>\$ sudo yum --nogpgcheck localinstall cloudera-manager-daemons-*.rpm \$ sudo yum --nogpgcheck localinstall cloudera-manager-server-*.rpm</pre>
SLES	<pre>\$ sudo zypper install cloudera-manager-daemons cloudera-manager-server</pre>
Ubuntu or Debian	<pre>\$ sudo apt-get install cloudera-manager-daemons cloudera-manager-server</pre>

- If you choose an Oracle database for use with Cloudera Manager, edit the `/etc/default/cloudera-scm-server` file on the Cloudera Manager server host. Locate the line that begins with `export CM_JAVA_OPTS` and change the `-Xmx2G` option to `-Xmx4G`.

Set up a Database for the Cloudera Manager Server

Depending on whether you are using an external database, or the embedded PostgreSQL database, do one of the following:

- External database - Prepare the Cloudera Manager Server database as described in [Preparing a Cloudera Manager Server External Database](#) on page 40.
- Embedded database - Install an embedded PostgreSQL database as described in [Installing and Starting the Cloudera Manager Server Embedded Database](#) on page 39.

(Optional) Install Cloudera Manager Agent, CDH, and Managed Service Software

You can use Cloudera Manager to install Cloudera Manager Agent packages, CDH, and managed service software, or you can install them manually.

To use Cloudera Manager to install the software (in [Choose the Software Installation Method and Install Software](#) on page 108), you must meet the requirements described in [Cloudera Manager Deployment](#) on page 34. If you use Cloudera Manager to install software, go to [Start the Cloudera Manager Server](#) on page 106. Otherwise, proceed with the following sections.

Install the Oracle JDK

Install the Oracle JDK on the cluster hosts. Cloudera Manager 5 can manage both CDH 5 and CDH 4, and the required JDK version varies accordingly:

- CDH 5 - [Java Development Kit Installation](#) on page 35.
- CDH 4 - [Java Development Kit Installation](#).

Install Cloudera Manager Agent Packages

To install the packages manually, do the following on every Cloudera Manager Agent host (including those that will run one or more of the Cloudera Management Service roles: Service Monitor, Activity Monitor, Event Server, Alert Publisher, or Reports Manager):

- Use one of the following commands to install the Cloudera Manager Agent packages:

OS	Command
RHEL, if you have a yum repo configured:	<pre>\$ sudo yum install cloudera-manager-agent cloudera-manager-daemons</pre>
RHEL, if you're manually transferring RPMs:	<pre>\$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons</pre>
SLES	<pre>\$ sudo zypper install cloudera-manager-agent cloudera-manager-daemons</pre>

OS	Command
Ubuntu or Debian	\$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons

2. On every Cloudera Manager Agent host, configure the Cloudera Manager Agent to point to the Cloudera Manager Server by setting the following properties in the `/etc/cloudera-scm-agent/config.ini` configuration file:

Property	Description
server_host	Name of the host where Cloudera Manager Server is running.
server_port	Port on the host where Cloudera Manager Server is running.

For more information on Agent configuration options, see [Agent Configuration File](#).

Install CDH and Managed Service Packages


For more information about manually installing CDH packages, see [CDH 4 Installation Guide](#) or [Cloudera Installation Guide](#).



1. Choose a repository strategy:

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems.
- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see [Understanding Custom Installation Solutions](#) on page 125. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

2. Install packages:

CDH Version	Procedure						
CDH 5	<ul style="list-style-type: none"> • Red Hat <ol style="list-style-type: none"> 1. Download and install the "1-click Install" package. <ol style="list-style-type: none"> a. Download the CDH 5 "1-click Install" package. 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 (for example, your home directory). <table border="1" data-bbox="537 1404 1463 1680"> <thead> <tr> <th>OS Version</th> <th>Click this Link</th> </tr> </thead> <tbody> <tr> <td>Red Hat/CentOS/Oracle 5</td> <td>Red Hat/CentOS/Oracle 5 link</td> </tr> <tr> <td>Red Hat/CentOS/Oracle 6</td> <td>Red Hat/CentOS/Oracle 6 link</td> </tr> </tbody> </table> b. Install the RPM: <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 5 <pre style="border: 1px dashed black; padding: 5px;">\$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm</pre> 	OS Version	Click this Link	Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link	Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 link
OS Version	Click this Link						
Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link						
Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 link						

CDH Version	Procedure
	<ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 6 <pre data-bbox="435 281 1370 323">\$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm</pre> <p data-bbox="451 390 846 422">2. (Optionally) add a repository key:</p> <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 5 <pre data-bbox="435 495 1455 548">\$ sudo rpm --import https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 6 <pre data-bbox="435 642 1455 695">\$ sudo rpm --import https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p data-bbox="451 747 756 779">3. Install the CDH packages:</p> <pre data-bbox="435 810 1442 978">\$ sudo yum clean all \$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="516 1010 1425 1125" style="border: 1px solid black; padding: 5px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div> <ul style="list-style-type: none"> • SLES <p data-bbox="451 1209 1019 1241">1. Download and install the "1-click Install" package.</p> <ul style="list-style-type: none"> a. Download the CDH 5 "1-click Install" package. <p data-bbox="532 1304 1466 1367">Click this link, choose Save File, and save it to a directory to which you have write access (for example, your home directory).</p> b. Install the RPM: <pre data-bbox="435 1440 1040 1472">\$ sudo rpm -i cloudera-cdh-5-0.x86_64.rpm</pre> <p data-bbox="500 1514 1036 1545">c. Update your system package index by running:</p> <pre data-bbox="435 1566 753 1598">\$ sudo zypper refresh</pre> <p data-bbox="451 1650 846 1682">2. (Optionally) add a repository key:</p> <pre data-bbox="435 1713 1442 1766">\$ sudo rpm --import https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p data-bbox="451 1797 756 1829">3. Install the CDH packages:</p> <pre data-bbox="435 1860 1382 1934">\$ sudo zypper clean --all \$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase</pre>

CDH Version	Procedure								
	<pre>hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> <div data-bbox="516 380 1425 493" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div> <ul style="list-style-type: none"> • Ubuntu and Debian <ol style="list-style-type: none"> 1. Download and install the "1-click Install" package <ol style="list-style-type: none"> a. Download the CDH 5 "1-click Install" package: <table border="1" data-bbox="537 669 1463 865" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0070c0; color: white;">OS Version</th> <th style="background-color: #0070c0; color: white;">Click this Link</th> </tr> </thead> <tbody> <tr> <td>Wheezy</td> <td>Wheezy link</td> </tr> <tr> <td>Precise</td> <td>Precise link</td> </tr> <tr> <td>Trusty</td> <td>Trusty link</td> </tr> </tbody> </table> b. Install the package by doing one of the following: <ul style="list-style-type: none"> • 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: <div data-bbox="435 1094 1468 1150" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>sudo dpkg -i cdh5-repository_1.0_all.deb</pre> </div> 2. Optionally add a repository key: <ul style="list-style-type: none"> • Debian Wheezy <div data-bbox="435 1293 1468 1373" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>\$ curl -s https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/archive.key sudo apt-key add -</pre> </div> • Ubuntu Precise <div data-bbox="435 1451 1468 1530" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>\$ curl -s https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key sudo apt-key add -</pre> </div> 3. Install the CDH packages: <div data-bbox="435 1608 1468 1803" style="border: 1px dashed #ccc; padding: 5px; margin: 5px 0;"> <pre>\$ sudo apt-get update \$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr</pre> </div> <div data-bbox="516 1822 1425 1936" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  Note: Installing these packages also installs all the other CDH packages required for a full CDH 5 installation. </div>	OS Version	Click this Link	Wheezy	Wheezy link	Precise	Precise link	Trusty	Trusty link
OS Version	Click this Link								
Wheezy	Wheezy link								
Precise	Precise link								
Trusty	Trusty link								

CDH Version	Procedure
CDH 4, Impala, and Solr	<ul style="list-style-type: none"> • Red Hat-compatible <ol style="list-style-type: none"> 1. Click the entry in the table at CDH Download Information that matches your Red Hat or CentOS system. 2. Navigate to the repo file (<code>cloudera-cdh4.repo</code>) for your system and save it in the <code>/etc/yum.repos.d/</code> directory. 3. Optionally add a repository key: <ul style="list-style-type: none"> • Red Hat/CentOS/Oracle 5 <pre data-bbox="435 527 1446 600">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> • Red Hat/CentOS 6 <pre data-bbox="435 680 1446 753">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> 4. Install packages on every host in your cluster: <ol style="list-style-type: none"> a. Install CDH 4 packages: <pre data-bbox="435 890 1446 989">\$ sudo yum -y install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> b. To install the <code>hue-common</code> package and all Hue applications on the Hue host, install the <code>hue</code> meta-package: <pre data-bbox="435 1121 764 1146">\$ sudo yum install hue</pre> 5. (Requires CDH 4.2 or later) Install Impala <ol style="list-style-type: none"> a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your Red Hat or CentOS system. b. Navigate to the repo file for your system and save it in the <code>/etc/yum.repos.d/</code> directory. c. Install Impala and the Impala Shell on Impala machines: <pre data-bbox="435 1444 1037 1470">\$ sudo yum -y install impala impala-shell</pre> 6. (Requires CDH 4.3 or later) Install Search <ol style="list-style-type: none"> a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Red Hat or CentOS system. b. Navigate to the repo file for your system and save it in the <code>/etc/yum.repos.d/</code> directory. c. Install the Solr Server on machines where you want Cloudera Search. <pre data-bbox="435 1766 924 1791">\$ sudo yum -y install solr-server</pre> • SLES

CDH Version	Procedure
	<p>1. Run the following command:</p> <pre data-bbox="435 281 1458 352">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/cdh4/sles/11/x86_64/cdh/cloudera-cdh4.repo</pre> <p>2. Update your system package index by running:</p> <pre data-bbox="435 428 753 464">\$ sudo zypper refresh</pre> <p>3. Optionally add a repository key:</p> <pre data-bbox="435 554 1442 625">\$ sudo rpm --import https://username:password@archive.cloudera.com/p/cdh4/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera</pre> <p>4. Install packages on every host in your cluster:</p> <p>a. Install CDH 4 packages:</p> <pre data-bbox="435 743 1425 856">\$ sudo zypper install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <p>b. To install the hue-common package and all Hue applications on the Hue host, install the hue meta-package:</p> <pre data-bbox="435 974 808 1010">\$ sudo zypper install hue</pre> <p>c. (Requires CDH 4.2 or later) Install Impala</p> <p>a. Run the following command:</p> <pre data-bbox="435 1142 1458 1213">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/impala/sles/11/x86_64/impala/cloudera-impala.repo</pre> <p>b. Install Impala and the Impala Shell on Impala machines:</p> <pre data-bbox="435 1289 1036 1325">\$ sudo zypper install impala impala-shell</pre> <p>d. (Requires CDH 4.3 or later) Install Search</p> <p>a. Run the following command:</p> <pre data-bbox="435 1478 1458 1549">\$ sudo zypper addrepo -f https://username:password@archive.cloudera.com/p/search/sles/11/x86_64/search/cloudera-search.repo</pre> <p>b. Install the Solr Server on machines where you want Cloudera Search.</p> <pre data-bbox="435 1625 922 1661">\$ sudo zypper install solr-server</pre> <p>• Ubuntu or Debian</p> <p>1. In the table at CDH Version and Packaging Information, click the entry that matches your Ubuntu or Debian system.</p>

CDH Version	Procedure
	<p>2. Navigate to the list file (<code>cloudera.list</code>) for your system and save it in the <code>/etc/apt/sources.list.d/</code> directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your <code>cloudera.list</code> file should look like:</p> <pre data-bbox="435 352 1458 464">deb [arch=amd64] https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib deb-src https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib</pre> <p>3. Optionally add a repository key:</p> <ul style="list-style-type: none"> • Ubuntu Lucid <pre data-bbox="435 604 1458 653">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/ubuntu/lucid/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> • Ubuntu Precise <pre data-bbox="435 751 1458 800">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/ubuntu/precise/amd64/cdh/archive.key sudo apt-key add -</pre> <ul style="list-style-type: none"> • Debian Squeeze <pre data-bbox="435 898 1458 947">\$ curl -s https://username:password@archive.cloudera.com/p/cdh4/debian/squeeze/amd64/cdh/archive.key sudo apt-key add -</pre> <p>4. Install packages on every host in your cluster:</p> <ul style="list-style-type: none"> a. Install CDH 4 packages: <pre data-bbox="435 1108 1458 1199">\$ sudo apt-get install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-https hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper</pre> <ul style="list-style-type: none"> b. To install the <code>hue-common</code> package and all Hue applications on the Hue host, install the hue meta-package: <pre data-bbox="435 1339 824 1367">\$ sudo apt-get install hue</pre> <ul style="list-style-type: none"> c. (Requires CDH 4.2 or later) Install Impala <ul style="list-style-type: none"> a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your Ubuntu or Debian system. b. Navigate to the list file for your system and save it in the <code>/etc/apt/sources.list.d/</code> directory. c. Install Impala and the Impala Shell on Impala machines: <pre data-bbox="435 1640 1052 1667">\$ sudo apt-get install impala impala-shell</pre> d. (Requires CDH 4.3 or later) Install Search <ul style="list-style-type: none"> a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Ubuntu or Debian system. b. Install Solr Server on machines where you want Cloudera Search: <pre data-bbox="435 1898 938 1925">\$ sudo apt-get install solr-server</pre>

(Optional) Install Key Trustee Key Provider

If you want to use Cloudera Navigator Key Trustee Server as the underlying key store for [HDFS Data At Rest Encryption](#), you must install the Key Trustee Key Provider.



Important: Following these instructions will install the required software to add the **KMS (Navigator Key Trustee)** service to your cluster; this enables you to use an existing Cloudera Navigator Key Trustee Server as the underlying key store for [HDFS Data At Rest Encryption](#). This *does not* install Cloudera Navigator Key Trustee Server. Contact your account team for assistance installing Cloudera Navigator Key Trustee Server.

To install the Key Trustee Key Provider:

1. Identify the appropriate repository for your operating system, and copy the repository URL:

OS Version	Repository URL
RHEL-compatible 6	RHEL 6 Repository
RHEL-compatible 5	RHEL 5 Repository
SLES 11	SLES 11 Repository
Ubuntu Trusty (14.04)	Ubuntu Trusty Repository
Ubuntu Precise (12.04)	Ubuntu Precise Repository
Debian Wheezy (7.0 and 7.1)	Debian Wheezy Repository

2. Add the repository to your system, using the appropriate procedure for your operating system:

- **RHEL-compatible**

Download the repository and copy it to the `/etc/yum.repos.d/` directory. Refresh the package index by running `sudo yum clean all`.

- **SLES**

Add the repository to your system using the following command:

```
$ sudo zypper addrepo -f <repository_url>
```

Refresh the package index by running `sudo zypper refresh`.

- **Ubuntu or Debian**

Copy the content of the appropriate `cloudera.list` file from the above repository table and append it to the `/etc/apt/sources.list.d/cloudera.list` file. Create the file if it does not exist. Refresh the package index by running `sudo apt-get update`.

3. Install the `keytrustee-keyprovider` package, using the appropriate command for your operating system:

- **RHEL-compatible**

```
$ sudo yum install keytrustee-keyprovider
```

- **SLES**

```
$ sudo zypper install keytrustee-keyprovider
```

- **Ubuntu or Debian**

```
$ sudo apt-get install keytrustee-keyprovider
```

Start the Cloudera Manager Server



Important: When you start the Cloudera Manager Server and Agents, Cloudera Manager assumes you are not already running HDFS and MapReduce. If these services are running:

1. Shut down HDFS and MapReduce. See [Stopping Services](#) (CDH 4) or [Stopping CDH Services Using the Command Line](#) (CDH 5) for the commands to stop these services.
2. Configure the init scripts to *not* start on boot. Use commands similar to those shown in [Configuring init to Start Core Hadoop System Services](#) (CDH 4) or [Configuring init to Start Hadoop System Services](#) (CDH 5), but *disable* the start on boot (for example, `$ sudo chkconfig hadoop-hdfs-namenode off`).

Contact Cloudera Support for help converting your existing Hadoop configurations for use with Cloudera Manager.

1. Run this command on the Cloudera Manager Server host:

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

If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 406.

(Optional) Start the Cloudera Manager Agents

If you installed the Cloudera Manager Agent packages in [Install Cloudera Manager Agent Packages](#) on page 98, run this command on each Agent host:

```
sudo service cloudera-scm-agent start
```

When the Agent starts, it contacts the Cloudera Manager Server. If communication fails between a Cloudera Manager Agent and Cloudera Manager Server, see [Troubleshooting Installation and Upgrade Problems](#) on page 406.

When the Agent hosts reboot, `cloudera-scm-agent` starts automatically.

Start and Log into the Cloudera Manager Admin Console

The Cloudera Manager Server URL takes the following form `http://Server host:port`, where *Server host* is the fully-qualified domain name or IP address of the host where the Cloudera Manager Server is installed and *port* is the port configured for the Cloudera Manager Server. The default port is 7180.

1. Wait several minutes for the Cloudera Manager Server to complete its startup. To observe the startup process you can perform `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](#) on page 406.
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. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can [change the password](#) using Cloudera Manager after you run the installation wizard. While you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.

Choose Cloudera Manager Edition and Hosts

You can use the Cloudera Manager wizard to choose which edition of Cloudera Manager you are using and which hosts will run CDH and managed services.

1. When you start the Cloudera Manager Admin Console, the install wizard starts up. Click **Continue** to get started.
2. Choose which [edition](#) to install:

- Cloudera Express, which does not require a license, but provides a somewhat limited set of features.
- Cloudera Enterprise Data Hub Edition Trial, which does not require a license, but expires after 60 days and cannot be renewed
- Cloudera Enterprise with one of the following license types:
 - Basic Edition
 - Flex Edition
 - Data Hub Edition

If you choose Cloudera Express or Cloudera Enterprise Data Hub Edition Trial, you can elect to upgrade the license at a later time. See [Managing Licenses](#).

3. If you have elected Cloudera Enterprise, install a license:
 - a. Click **Upload License**.
 - b. Click the document icon to the left of the **Select a License File** text field.
 - c. Navigate to the location of your license file, click the file, and click **Open**.
 - d. Click **Upload**.

Click **Continue** to proceed with the installation.

4. Click **Continue** in the next screen. The **Specify Hosts** page displays.
5. Do one of the following:

- If you installed Cloudera Agent packages in [Install Cloudera Manager Agent Packages](#) on page 98, choose from among hosts with the packages installed:
 1. Click the **Currently Managed Hosts** tab.
 2. Choose the hosts to add to the cluster.
- Search for and choose hosts:
 1. To enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the cluster hostnames or IP addresses. You can also specify hostname and IP address ranges. For example:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.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. Use this technique to make more specific searches instead of searching overly wide ranges. The scan results will include all addresses scanned, but only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. If you don't know the IP addresses of all of the hosts, you can enter an address range that spans over unused addresses and then deselect the hosts that do not exist (and are not discovered) later in this procedure. However, keep in mind that wider ranges will require more time to scan.

2. Click **Search**. Cloudera Manager identifies the hosts on your cluster to allow you to configure them for services. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking **Abort Scan**. To find additional hosts, click **New Search**, add the host names or IP addresses and click **Search** again. Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts. Common causes of loss of connectivity are firewalls and interference from SELinux.

3. 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.

6. Click **Continue**. The **Select Repository** page displays.

Choose the Software Installation Method and Install Software

The following instructions describe how to use the Cloudera Manager wizard to install Cloudera Manager Agent, CDH, and managed service software.

1. Install CDH and managed service software using either packages or parcels:

- **Use Packages** - If you *did not* install packages in [Install CDH and Managed Service Packages](#) on page 99, click the package versions to install. Otherwise, select the CDH version (CDH 4 or CDH 5) that matches the packages that you installed manually.

- **Use Parcels**

1. Choose the parcels to install. The choices you see depend on the repositories you have chosen – a repository may contain multiple parcels. Only the parcels for the latest supported service versions are configured by default.

You can add additional parcels for previous versions by specifying custom repositories. For example, you can find the locations of the previous CDH 4 parcels at

`https://username:password@archive.cloudera.com/p/cdh4/parcels/`. Or, if you are installing CDH 4.3 and want to use [policy-file authorization](#), you can add the Sentry parcel using this mechanism.

1. To specify the parcel directory, local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:

- **Parcel Directory** and **Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host. If you change the default value for **Parcel Directory** and have already installed and started Cloudera Manager Agents, restart the Agents:

```
$ sudo service cloudera-scm-agent restart
```

- **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the **+** button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the [Configuring Cloudera Manager Server Parcel Settings](#) on page 78 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you will see all the unique parcels contained in all your repositories.
- **Proxy Server** - Specify the properties of a proxy server.

2. Click **OK**.

2. If you *did not* install Cloudera Manager Agent packages in [Install Cloudera Manager Agent Packages](#) on page 98, do the following:

- a. Select the release of Cloudera Manager Agent to install. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories. 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 leave deselected if you plan to install it yourself. If checked, your local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, select the **Install Java Unlimited Strength Encryption Policy Files** checkbox. Click **Continue**. The Enable Single User Mode screen displays.

4. (Optional) Select **Single User Mode** to configure the Cloudera Manager Agent and all service processes to run as the same user. This mode requires [extra configuration steps](#) that must be done manually on all hosts in the cluster.

If you have not performed the steps, directory creation will fail in the installation wizard. In most cases, you can create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**. The Provide SSH login credentials screen displays.

5. If you chose to have Cloudera Manager install packages, specify host installation properties:
 - Select **root** or enter the user name for an account that has password-less sudo permission.
 - Select an authentication method:
 - If you choose to use password authentication, enter and confirm the password.
 - If you choose to use public-key authentication provide a passphrase and path to the required key files.
 - You can choose to specify an alternate SSH port. The default value is 22.
 - You can specify the maximum number of host installations to run at once. The default value is 10.
6. Click **Continue**. If you *did not* install packages in [\(Optional\) Install Cloudera Manager Agent, CDH, and Managed Service Software](#) on page 98, Cloudera Manager installs the Oracle JDK, Cloudera Manager Agent, packages and CDH and managed service packages or parcels. During the parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels you will see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed. Click **Continue**.
7. 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. If the validation is successful, click **Finish**. The Cluster Setup screen displays.

Add Services

The following instructions describe how to use the Cloudera Manager wizard to configure and start CDH and managed services.

1. In the first page of the Add Services wizard you choose the combination of services to install and whether to install Cloudera Navigator:
 - Click the radio button next to the combination of services to install:

CDH 4	CDH 5
<ul style="list-style-type: none"> • Core Hadoop - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • Core with HBase • Core with Impala • All Services - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • Custom Services - Any combination of services. 	<ul style="list-style-type: none"> • Core Hadoop - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, and Sqoop • Core with HBase • Core with Impala • Core with Search • Core with Spark • All Services - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, Sqoop, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • Custom Services - Any combination of services.

As you select the services, keep the following in mind:

- Some services depend on other services; for example, HBase requires HDFS and ZooKeeper. Cloudera Manager tracks dependencies and installs the correct combination of services.
- In a Cloudera Manager deployment of a CDH 4 cluster, the MapReduce service is the default MapReduce computation framework. Choose **Custom Services** to install YARN or use the Add Service functionality to add YARN after installation completes.



Note: You can create a YARN service in a CDH 4 cluster, but it is not considered production ready.

- In a Cloudera Manager deployment of a CDH 5 cluster, the YARN service is the default MapReduce computation framework. Choose **Custom Services** to install MapReduce or use the Add Service functionality to add MapReduce after installation completes.



Note: In CDH 5, the MapReduce service has been deprecated. However, the MapReduce service is fully supported for backward compatibility through the CDH 5 life cycle.

- The Flume service can be added only after your cluster has been set up.
- If you have chosen Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See the [Cloudera Navigator Documentation](#).

Click **Continue**. The Customize Role Assignments screen displays.

2. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. These assignments are typically acceptable, but you can reassign them if necessary.

Click a field below a role to display a dialog containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts or **Custom** to display the pageable hosts dialog.

The following shortcuts for specifying hostname patterns are supported:

- Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

3. When you are satisfied with the assignments, click **Continue**. The Database Setup screen displays.
4. On the Database Setup page, configure settings for required databases:
 - a. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
 - b. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.) The Review Changes screen displays.
5. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed.



Warning: DataNode data directories should not be placed on NAS devices.

Click **Continue**. The wizard starts the services.

6. When all of the services are started, click **Continue**. You will see a success message indicating that your cluster has been successfully started.

7. Click **Finish** to proceed to the [Cloudera Manager Admin Console Home Page](#).

Change the Default Administrator Password

As soon as possible after running the wizard and beginning to use Cloudera Manager, change the default administrator password:

1. Right-click the logged-in username at the far right of the top navigation bar and select **Change Password**.
2. Enter the current password and a new password twice, and then click **Update**.

Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 146.

Deploying Clients

Client configuration files are generated automatically by Cloudera Manager based on the services you install.

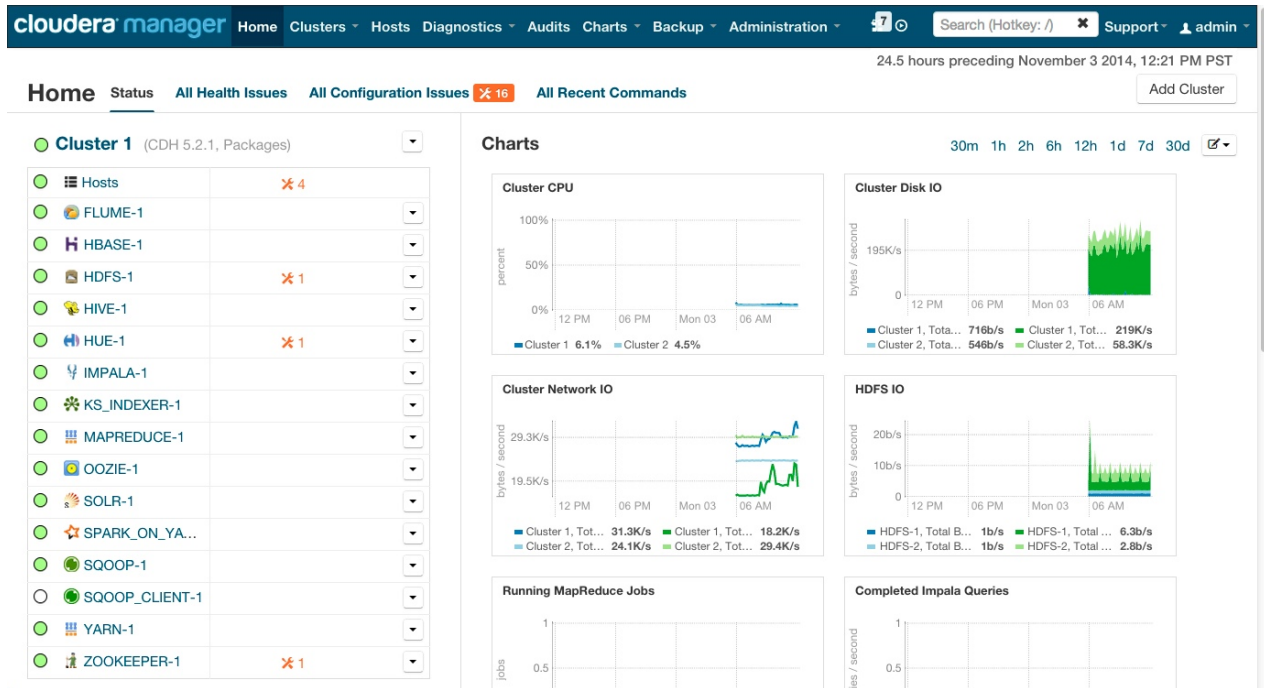
Cloudera Manager deploys these configurations automatically at the end of the installation workflow. You can also download the client configuration files to deploy them manually.

If you modify the configuration of your cluster, you may need to redeploy the client configuration files. If a service's status is "Client configuration redeployment required," you need to redeploy those files.

See [Client Configuration Files](#) for information on downloading client configuration files, or redeploying them through Cloudera Manager.

Testing the Installation

To begin testing, [start the Cloudera Manager Admin Console](#). Once you've logged in, the Home page should look 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** (green circle). You can click on each service to view more detailed information about each service. You can also test your installation by either checking each Host's heartbeats, running a MapReduce job, or interacting with the cluster with an existing Hue application.

Checking Host Heartbeats

One way to check whether all the Agents are running is to look at the time since their last heartbeat. You can do this by clicking the **Hosts** tab where you can see a list of all the Hosts along with the value of their **Last Heartbeat**. By default, every Agent must heartbeat successfully every 15 seconds. A recent value for the **Last Heartbeat** means that the Server and Agents are communicating successfully.

Running a MapReduce Job

1. Log into a host in the cluster.
2. Run the Hadoop PiEstimator example using one of the following commands:
 - **Parcel** - `sudo -u hdfs hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100`
 - **Package** - `sudo -u hdfs hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100`

or create and run the WordCount v1.0 application described in [Hadoop Tutorial](#).

3. Depending on whether your cluster is configured to run MapReduce jobs on the YARN or MapReduce service, view the results of running the job by selecting one of the following from the top navigation bar in the Cloudera Manager Admin Console :

- **Clusters** > **ClusterName** > **yarn Applications**
- **Clusters** > **ClusterName** > **mapreduce Activities**

If you run the PiEstimator job on the YARN service (the default) you will see an entry like the following in **yarn Applications**:

05/22/2014 10:45 AM	-	Name: QuasiMonteCarlo	Pool: root.hdfs	<input type="button" value="Actions"/>	<input type="button" value="Details"/>
05/22/2014 10:46 AM		Mapper: QuasiMonteCarlo\$QmcMapper	Reducer: QuasiMonteCarlo\$QmcReducer		
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 page, 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](#).

Uninstalling Cloudera Manager and Managed Software

Use the following instructions to uninstall the Cloudera Manager Server, Agents, managed software, and databases.

Reverting an Incomplete Installation

If you have come to this page because your installation did not complete (for example, if it was interrupted by a virtual machine timeout), and you want to proceed with the installation, do the following before reinstalling:

1. Remove files and directories:

```
$ sudo rm -rf /usr/share/cmfs /var/lib/cloudera* /var/cache/yum/cloudera*
```

Uninstalling Cloudera Manager and Managed Software

Follow the steps in this section to remove software and data.

Record User Data Paths

The user data paths listed [Remove User Data](#) on page 151, `/var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/zookeeper data_drive_path/dfs data_drive_path/mapred data_drive_path/yarn`, are the default settings. However, at some point they may have been reconfigured in Cloudera Manager. If you want to remove all user data from the cluster and have changed the paths, either when you installed CDH and managed services or at some later time, note the location of the paths by checking the configuration in each service.

Stop all Services

1. For each cluster managed by Cloudera Manager:

- a. On the Home page, click



to the right of the cluster name and select **Stop**.

- b. Click **Stop** in the confirmation screen. The **Command Details** window shows the progress of stopping services. When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.

- c. On the Home page, click




to the right of the Cloudera Management Service entry and select **Stop**. The **Command Details** window shows the progress of stopping services. When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.

2. [Stop the Cloudera Management Service](#).

Deactivate and Remove Parcels

If you installed using packages, skip this step and go to [Uninstall the Cloudera Manager Server](#) on page 148; you will remove packages in [Uninstall Cloudera Manager Agent and Managed Software](#) on page 149. If you installed using parcels remove them as follows:

1. Click the parcel indicator  in the main navigation bar.
2. For each activated parcel, select **Actions** > **Deactivate**. When this action has completed, the parcel button changes to **Activate**.
3. For each activated parcel, select **Actions** > **Remove from Hosts**. When this action has completed, the parcel button changes to **Distribute**.
4. For each activated parcel, select **Actions** > **Delete**. This removes the parcel from the local parcel repository.

There may be multiple parcels that have been downloaded and distributed, but that are not active. If this is the case, you should also remove those parcels from any hosts onto which they have been distributed, and delete the parcels from the local repository.

Delete the Cluster

On the **Home** page, Click the drop-down list next to the cluster you want to delete and select **Delete**.

Uninstall the Cloudera Manager Server

The commands for uninstalling the Cloudera Manager Server depend on the method you used to install it. Refer to steps below that correspond to the method you used to install the Cloudera Manager Server.

- **If you used the cloudera-manager-installer.bin file** - Run the following command on the Cloudera Manager Server host:

```
$ sudo /usr/share/cmf/uninstall-cloudera-manager.sh
```



Note: If the `uninstall-cloudera-manager.sh` is not installed on your cluster, use the following instructions to uninstall the Cloudera Manager Server.

- **If you did not use the `cloudera-manager-installer.bin` file** - If you installed the Cloudera Manager Server using a different installation method such as Puppet, run the following commands on the Cloudera Manager Server host.

1. Stop the Cloudera Manager Server and its database:

```
sudo service cloudera-scm-server stop
sudo service cloudera-scm-server-db stop
```

- 2. Uninstall the Cloudera Manager Server and its database.** This process described also removes the embedded PostgreSQL database software, if you installed that option. If you did not use the embedded PostgreSQL database, omit the `cloudera-manager-server-db` steps.

Red Hat systems:

```
sudo yum remove cloudera-manager-server
sudo yum remove cloudera-manager-server-db-2
```

SLES systems:

```
sudo zypper -n rm --force-resolution cloudera-manager-server
sudo zypper -n rm --force-resolution cloudera-manager-server-db-2
```

Debian/Ubuntu systems:

```
sudo apt-get remove cloudera-manager-server
sudo apt-get remove cloudera-manager-server-db-2
```

Uninstall Cloudera Manager Agent and Managed Software

Do the following on all Agent hosts:

- 1. Stop the Cloudera Manager Agent.**

Red Hat/SLES systems:

```
$ sudo service cloudera-scm-agent hard_stop
```

Debian/Ubuntu systems:

```
$ sudo /usr/sbin/service cloudera-scm-agent hard_stop
```

- 2. Uninstall software:**

OS	Parcel Install	Package Install
Red Hat	<pre>\$ sudo yum remove 'cloudera-manager-*</pre>	<ul style="list-style-type: none"> • CDH 5 <pre>\$ sudo yum remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-https hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>

OS	Parcel Install	Package Install
SLES	<pre>\$ sudo zypper remove 'cloudera-manager-*</pre>	<ul style="list-style-type: none"> • CDH 5 <pre>\$ sudo zypper remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>
Debian/Ubuntu	<pre>\$ sudo apt-get purge 'cloudera-manager-*</pre>	<ul style="list-style-type: none"> • CDH 5 <pre>\$ sudo apt-get purge 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>

3. Run the `clean` command:

Red Hat

```
$ sudo yum clean all
```

SLES

```
$ sudo zypper clean
```

Debian/Ubuntu

```
$ sudo apt-get clean
```

Remove Cloudera Manager and User Data

Kill Cloudera Manager and Managed Processes

On all Agent hosts, kill any running Cloudera Manager and managed processes:

```
$ for u in cloudera-scm flume hadoop hdfs hbase hive httpfs hue impala llama mapred
oozie solr spark sqoop sqoop2 yarn zookeeper; do sudo kill $(ps -u $u -o pid=); done
```



Note: This step should not be necessary if you stopped all the services and the Cloudera Manager Agent correctly.

Remove Cloudera Manager Data

This step permanently removes Cloudera Manager data. If you want to be able to access any of this data in the future, you must back it up before removing it. If you used an embedded PostgreSQL database, that data is stored in `/var/lib/cloudera-scm-server-db`. On all Agent hosts, run the following command:

```
$ sudo umount cm_processes
$ sudo rm -Rf /usr/share/cmF /var/lib/cloudera* /var/cache/yum/cloudera*
/var/log/cloudera* /var/run/cloudera*
```

Remove the Cloudera Manager Lock File

On all Agent hosts, run this command to remove the Cloudera Manager lock file:

```
$ sudo rm /tmp/.scm_prepare_node.lock
```

Remove User Data

This step permanently removes all user data. To preserve the data, copy it to another cluster using the `distcp` command before starting the uninstall process. On all Agent hosts, run the following commands:

```
$ sudo rm -Rf /var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/lib/navigator
/var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/zookeeper
```

Run the following command on each data drive on all Agent hosts (adjust the paths for the data drives on each host):

```
$ sudo rm -Rf data_drive_path/dfs data_drive_path/mapred data_drive_path/yarn
```



Note: For additional information about uninstalling CDH, including clean-up of CDH files, see the entry on Uninstalling CDH Components in or [Cloudera Installation Guide](#).

Stop and Remove External Databases

If you chose to store Cloudera Manager or user data in an [external database](#), see the database vendor documentation for details on how to remove the databases.

Uninstalling a CDH Component From a Single Host

The following procedure removes CDH software components from a single host that is managed by Cloudera Manager.

1. In the Cloudera Manager Administration Console, select the **Hosts** tab.

A list of hosts in the cluster displays.

2. Select the host where you want to uninstall CDH software.
3. Click the **Actions for Selected** button and select **Remove From Cluster**.

Cloudera Manager removes the roles and host from the cluster.

4. (Optional) Manually delete the `krb5.conf` file used by Cloudera Manager.

Installing Cloudera Navigator

Minimum Required Role: [Full Administrator](#)

Cloudera Navigator is implemented as two roles in the [Cloudera Management Service](#): Navigator Audit Server and Navigator Metadata Server. You can add Cloudera Navigator roles while installing Cloudera Manager for the first time or into an existing Cloudera Manager installation. For information on compatible Cloudera Navigator and Cloudera Manager versions, see the [Product Compatibility Matrix for Cloudera Navigator](#) product compatibility matrix.

Installing Cloudera Manager and CDH

Configuring a Database for the Cloudera Navigator

When you install Cloudera Navigator you choose a database to store audit events and policy, role, and audit report metadata. You can choose either an embedded PostgreSQL database or an external database. For information on supported databases, see [Supported Databases](#) on page 16. For information on setting up an external database, see [Cloudera Manager and Managed Service Data Stores](#) on page 38.

Adding Cloudera Navigator Roles in a New Cloudera Manager Installation

1. Install Cloudera Manager following the instructions in [Cloudera Manager Deployment](#) on page 34.
2. In the first page of the Cloudera Manager installation wizard, choose one of the license options that support Cloudera Navigator:
 - Cloudera Enterprise Data Hub Edition Trial
 - Cloudera Enterprise
 - Flex Edition
 - Data Hub Edition
3. If you have elected Cloudera Enterprise, install a license:
 - a. Click **Upload License**.
 - b. Click the document icon to the left of the **Select a License File** text field.
 - c. Navigate to the location of your license file, click the file, and click **Open**.
 - d. Click **Upload**.Click **Continue** to proceed with the installation.
4. In the first page of the Add Services procedure, check the **Include Cloudera Navigator** checkbox.
5. If you have chosen to use an external database, provide the Cloudera Navigator Audit Server and Metadata Server database properties in the **Database Setup** page.

Adding Cloudera Navigator Roles in an Existing Cloudera Manager Installation

1. Add and start the Cloudera Navigator roles:
 - [Adding the Navigator Audit Server Role](#)
 - [Adding the Navigator Metadata Server Role](#)

Related Information

- [Cloudera Navigator 2 Overview](#)
- [Upgrading Cloudera Navigator](#)
- [Cloudera Navigator Administration](#)
- [Cloudera Data Management](#)
- [Configuring Authentication in Cloudera Navigator](#)
- [Configuring SSL for Cloudera Navigator](#)
- [Cloudera Navigator User Roles](#)

Installing and Deploying CDH Using the Command Line

Before You Install CDH 5 on a Cluster



Important:

- When starting, stopping and restarting CDH components, always use the `service (8)` command rather than running scripts in `/etc/init.d` directly. This is important because `service` sets the current working directory to `/` and removes most environment variables (passing only `LANG` and `TERM`), to create a predictable environment for the service. If you run the scripts in `/etc/init.d`, locally-set environment variables could produce unpredictable results. If you install CDH from RPMs, `service` will be installed as part of the Linux Standard Base (LSB).
- On SLES 11 platforms, do not install or try to use the IBM Java version bundled with the SLES distribution; Hadoop will not run correctly with that version. Install the Oracle JDK following directions under [Java Development Kit Installation](#).
- If you are migrating from MapReduce v1 (MRv1) to MapReduce v2 (MRv2, YARN), see [Migrating from MapReduce 1 \(MRv1\) to MapReduce 2 \(MRv2, YARN\)](#) on page 171 for important information and instructions.

Before you install CDH 5 on a cluster, 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](#) on page 18.
2. If you haven't already done so, install the Oracle Java Development Kit. For instructions and recommendations, see [Java Development Kit Installation](#).

Scheduler Defaults

Note the following differences between MRv1 (MapReduce) and MRv2 (YARN).

- MRv1 (MapReduce v1):
 - Cloudera Manager and CDH 5 set the default to FIFO.

FIFO is set as the default for backward-compatibility purposes, but Cloudera recommends Fair Scheduler. Capacity Scheduler is also available.
- MRv2 (YARN):
 - Cloudera Manager and CDH 5 set the default to Fair Scheduler.

Cloudera recommends Fair Scheduler because Impala and Llama are optimized for it. FIFO and Capacity Scheduler are also available.

High Availability

In CDH 5, you can configure high availability both for the NameNode and the JobTracker or ResourceManager.

- For more information and instructions on setting up a new HA configuration, see [High Availability](#).



Important:

If you configure [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`. After completing the [HDFS HA software configuration](#), follow the installation instructions in [Deploying HDFS High Availability](#).

Creating a Local Yum Repository



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).



Important: As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.

This can affect new installations, upgrades, adding new hosts to a cluster, and adding a new cluster.

For more information, see [Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication](#).

This section explains how to set up a local yum repository to install CDH on the machines in your cluster. There are a number of reasons you might want to do this, for example:

- The computers in your cluster may not have Internet access. You can still use yum to do an installation on those machines by creating a local yum repository.
- You may want to keep a stable local repository to ensure that any new installations (or re-installations on existing cluster members) use exactly the same bits.
- Using a local repository may be the most efficient way to distribute the software to the cluster members.

To set up your own internal mirror, follow the steps below. You need an internet connection for the steps that require you to download packages and create the repository itself. You will also need an internet connection in order to download updated RPMs to your local repository.

1. Click the entry in the table below that matches your RHEL or CentOS system, navigate to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.

For OS Version	Click this Link
RHEL/CentOS/Oracle 5	RHEL/CentOS/Oracle 5 link
RHEL/CentOS/Oracle 6 (64-bit)	RHEL/CentOS/Oracle 6 link

2. Install a web server such as `apache/lighttpd` on the machine which will serve the RPMs. The default configuration should work. HTTP access must be allowed to pass through any firewalls between this server and the internet connection.
3. On the server with the web server,, install the `yum-utils` and `createrepo` RPM packages if they are not already installed. The `yum-utils` package includes the `reposync` command, which is required to create the local Yum repository.

```
sudo yum install yum-utils createrepo
```

4. On the same computer as in the previous steps, download the yum repository into a temporary location. On RHEL/CentOS 6, you can use a command such as:

```
reposync -r cloudera-cdh5
```

You can replace with any alpha-numeric string. It will be the name of your local repository, used in the header of the repo file other systems will use to connect to your repository. You can now disconnect your server from the internet.

5. Put all the RPMs into a directory served by your web server, such as `/var/www/html/cdh/5/RPMS/noarch/` (or `x86_64` or `i386` instead of `noarch`). The directory structure `5/RPMS/noarch` is required. Make sure you can remotely access the files in the directory using HTTP, using a URL similar to `http://<yourwebserver>/cdh/5/RPMS/`.
6. On your web server, issue the following command from the `5/` subdirectory of your RPM directory:

```
createrepo .
```

This will create or update the metadata required by the `yum` command to recognize the directory as a repository. The command creates a new directory called `repodata`. If necessary, adjust the permissions of files and directories in your entire repository directory to be readable by the web server user.

7. Edit the repo file you downloaded in step 1 and replace the line starting with `baseurl=` or `mirrorlist=` with `baseurl=http://<yourwebserver>/cdh/5/`, using the URL from step 5. Save the file back to `/etc/yum/repos.d/`.
8. While disconnected from the internet, issue the following commands to install CDH from your local `yum` repository.

Example:

```
yum update
yum install hadoop
```

Once you have confirmed that your internal mirror works, you can distribute this modified repo file to any system which can connect to your repository server. Those systems can now install CDH from your local repository without internet access. Follow the instructions under [Installing the Latest CDH 5 Release](#) on page 155, starting at Step 2 (you have already done Step 1).

Installing the Latest CDH 5 Release



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x. If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

Ways To Install CDH 5

You can install CDH 5 in any of the following ways:

- Install Cloudera Manager, CDH, and managed services in a [Cloudera Manager Deployment](#) on page 34.



Note: Cloudera recommends that you use this automated method if possible.

- Or use one of the manual methods described below:
 - Download and install the CDH 5 "1-click Install" package; *OR*
 - Add the CDH 5 repository; *OR*
 - Build your own CDH 5 repository

If you use one of these manual methods rather than Cloudera Manager, the first (downloading and installing the "1-click Install" package) is recommended in most cases because it is simpler than building or adding a repository.

- Install from a CDH 5 tarball — see, the next topic, "How Packaging Affects CDH 5 Deployment".

Installing Cloudera Manager and CDH

How Packaging Affects CDH 5 Deployment

Installing from Packages

- To install and deploy YARN, follow the directions on this page and proceed with [Deploying MapReduce v2 \(YARN\) on a Cluster](#).
- To install and deploy MRv1, follow the directions on this page and then proceed with [Deploying MapReduce v1 \(MRv1\) on a Cluster](#).

Installing from a Tarball



Note: The instructions in this Installation Guide are tailored for a package installation, as described in the sections that follow, and do not cover installation or deployment from tarballs.

- If you install CDH 5 from a [tarball](#), you will install YARN.
- In CDH 5, there is no separate tarball for MRv1. Instead, the MRv1 binaries, examples, etc., are delivered in the Hadoop tarball itself. The scripts for running MRv1 are in the `bin-mapreduce1` directory in the tarball, and the MRv1 examples are in the `examples-mapreduce1` directory.

Before You Begin Installing CDH 5 Manually

- For a list of supported operating systems, see [CDH 5 Requirements and Supported Versions](#) on page 18.
- These instructions assume that the `sudo` command is configured on the hosts where you will be doing the installation. If this is not the case, you will need the root user (superuser) to configure it.



Note:

If you are migrating from MapReduce v1 (MRv1) to MapReduce v2 (MRv2, YARN), see [Migrating from MapReduce 1 \(MRv1\) to MapReduce 2 \(MRv2, YARN\)](#) on page 171 for important information and instructions.



Important: Running Services

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



Important: Java Development Kit:

- if you have not already done so, **install the Oracle Java Development Kit (JDK)**; see [Java Development Kit Installation](#).

High Availability

In CDH 5, you can configure high availability both for the NameNode and the JobTracker or ResourceManager.

- For more information and instructions on setting up a new HA configuration, see [High Availability](#).

**Important:**

If you configure [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`. After completing the [HDFS HA software configuration](#), follow the installation instructions in [Deploying HDFS High Availability](#).

Steps to Install CDH 5 Manually

Step 1: Add or Build the CDH 5 Repository or Download the "1-click Install" package.

- If you are installing CDH 5 on a [Red Hat](#) system, you can download Cloudera packages using `yum` or your web browser.
- If you are installing CDH 5 on a [SLES](#) system, you can download the Cloudera packages using `zypper` or YaST or your web browser.
- If you are installing CDH 5 on an [Ubuntu or Debian](#) system, you can download the Cloudera packages using `apt` or your web browser.

On Red Hat-compatible Systems

Use one of the following methods to add or build the CDH 5 repository or download the package on Red Hat-compatible systems.

**Note:**

Use only one of the three methods.

- [Download and install the CDH 5 "1-click Install" package](#) *OR*
- [Add the CDH 5 repository](#) *OR*
- [Build a Yum Repository](#)

Do this on all the systems in the cluster.

To download and install the CDH 5 "1-click Install" 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	Click this Link
Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link
Red Hat/CentOS/Oracle 6	Red Hat/CentOS/Oracle 6 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
```

Now continue with [Step 1a: Optionally Add a Repository Key](#), and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.



Note: Make sure your repositories are up to date

Before proceeding, make sure the repositories on each system are up to date:

```
sudo yum clean all
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

OR: To add the CDH 5 repository:

Click the entry in the table below that matches your RHEL or CentOS system, navigate to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.

For OS Version	Click this Link
RHEL/CentOS/Oracle 5	RHEL/CentOS/Oracle 5 link
RHEL/CentOS/Oracle 6 (64-bit)	RHEL/CentOS/Oracle 6 link

Now continue with [Step 2: Optionally Add a Repository Key](#) on page 162, and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.



Note: Make sure your repositories are up to date

Before proceeding, make sure the repositories on each system are up to date:

```
sudo yum clean all
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

OR: To build a Yum repository:

If you want to create your own `yum` repository, download the appropriate repo file, create the repo, distribute the repo file and set up a web server, as described under [Creating a Local Yum Repository](#).

Now continue with [Step 2: Optionally Add a Repository Key](#) on page 162, and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.



Note: Make sure your repositories are up to date

Before proceeding, make sure the repositories on each system are up to date:

```
sudo yum clean all
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

On SLES Systems

Use one of the following methods to download the CDH 5 repository or package on SLES systems.

**Note:**

Use only one of the three methods.

- [Download and install the CDH 5 "1-click Install" Package](#) *OR*
- [Add the CDH 5 repository](#) *OR*
- [Build a SLES Repository](#)

To download and install the CDH 5 "1-click Install" package:

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

Click [this link](#), 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
```

3. Update your system package index by running:

```
$ sudo zypper refresh
```

Now continue with [Step 2: Optionally Add a Repository Key](#) on page 162, and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.

OR: To add the CDH 5 repository:

1. Run the following command:

```
$ sudo zypper addrepo -f
https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

Now continue with [Step 2: Optionally Add a Repository Key](#) on page 162, and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.

**Note: Make sure your repositories are up to date**

Before proceeding to the next step, make sure the repositories on each system are up to date:

```
sudo zypper clean --all
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

OR: To build a SLES repository:

If you want to create your own SLES repository, create a mirror of [the CDH SLES directory](#) by following [these instructions](#) that explain how to create a SLES repository from the mirror.

Now continue with [Step 2: Optionally Add a Repository Key](#) on page 162, and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.



Note: Make sure your repositories are up to date

Before proceeding to the next step, make sure the repositories on each system are up to date:

```
sudo zypper clean --all
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

On Ubuntu or Debian Systems

Use one of the following methods to download the CDH 5 repository or package.

- [Download and install the CDH 5 "1-click Install" Package](#) *OR*
- [Add the CDH 5 repository](#) *OR*
- [Build a Debian Repository](#)



Note:

- Use only one of the three methods.
- There is an extra step if you are adding a repository on Ubuntu Trusty, as described below.
- Unless you are adding a repository on Ubuntu Trusty, don't forget to run `apt-get update` after downloading, adding, or building the repository.

To download and install the CDH 5 "1-click Install" package:

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

OS Version	Click this Link
Wheezy	Wheezy link
Precise	Precise link
Trusty	Trusty link

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: Make sure your repositories are up to date

Before proceeding to the next step, make sure the repositories on each system are up to date:

```
sudo apt-get update
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

Now continue with [Step 2: Optionally Add a Repository Key](#) on page 162, and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.

OR: To add the CDH 5 repository:

- Download the appropriate `cloudera.list` file by issuing one of the following commands. You can use another HTTP client if `wget` is not available, but the syntax may be different.

**Important: Ubuntu 14.04 (Trusty)**

If you are running Ubuntu Trusty, you need to perform an additional step after adding the repository. See "Additional Step for Trusty" below.

OS Version	Command
Debian Wheezy	<pre>\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/wheezy/amd64/cdh/cloudera.list' \ -o /etc/apt/sources.list.d/cloudera.list</pre>
Ubuntu Precise	<pre>\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/wheezy/amd64/cdh/cloudera.list' \ -o /etc/apt/sources.list.d/cloudera.list</pre>
Ubuntu Lucid	<pre>\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/lucid/amd64/cdh/cloudera.list' \ -o /etc/apt/sources.list.d/cloudera.list</pre>
Ubuntu Trusty	<pre>\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/cloudera.list' \ -o /etc/apt/sources.list.d/cloudera.list</pre>

**Note: Make sure your repositories are up to date**

Unless you are adding a repository on Ubuntu Trusty, make sure the repositories on each system are up to date before proceeding to the next step:

```
sudo apt-get update
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

Additional step for Trusty

This step ensures that you get the right ZooKeeper package for the current CDH release. You need to prioritize the Cloudera repository you have just added, such that you install the CDH version of ZooKeeper rather than the version that is bundled with Ubuntu Trusty.

To do this, create a file at `/etc/apt/preferences.d/cloudera.pref` with the following contents:

```
Package: *
Pin: release o=Cloudera, l=Cloudera
Pin-Priority: 501
```

**Note:**

You *do not* need to run `apt-get update` after creating this file.

Now continue with [Step 1a: Optionally Add a Repository Key](#), and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.

OR: To build a Debian repository:

If you want to create your own apt repository, create a mirror of [the CDH Debian directory](#) and then [create an apt repository from the mirror](#).



Note: Make sure your repositories are up to date

Before proceeding to the next step, make sure the repositories on each system are up to date:

```
sudo apt-get update
```

This ensures that the system repositories contain the latest software (it does not actually install anything).

Now continue with [Step 1a: Optionally Add a Repository Key](#), and then choose [Step 3: Install CDH 5 with YARN](#) on page 163, or [Step 4: Install CDH 5 with MRv1](#) on page 164; or do both steps if you want to install both implementations.

Step 2: Optionally Add a Repository Key

Before installing YARN or MRv1: (Optionally) add a repository key on each system in the cluster. 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
```

- **For all SLES systems:**

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

- **For Ubuntu or Debian systems:**

OS Version	Command
Debian Wheezy	<pre>\$ wget https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/archive.key -O archive.key \$ sudo apt-key add archive.key</pre>
Ubuntu Precise	<pre>\$ wget https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key -O archive.key \$ sudo apt-key add archive.key</pre>
Ubuntu Lucid	<pre>\$ wget https://archive.cloudera.com/cdh5/ubuntu/lucid/amd64/cdh/archive.key -O archive.key \$ sudo apt-key add archive.key</pre>
Ubuntu Trusty	<pre>\$ wget https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/archive.key -O archive.key \$ sudo apt-key add archive.key</pre>

This key enables you to verify that you are downloading genuine packages.

Step 3: Install CDH 5 with YARN

**Note:**

Skip this step if you intend to use *only* MRv1. Directions for installing MRv1 are in [Step 4](#).

To install CDH 5 with YARN:

**Note:**

If you decide to configure [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`. After completing the [HA software configuration](#), follow the installation instructions under [Deploying HDFS High Availability](#).

1. Install and deploy ZooKeeper.

**Important:**

Cloudera recommends that you install (or update) and start a ZooKeeper cluster before proceeding. This is a **requirement** if you are deploying high availability (HA) for the NameNode.

Follow instructions under [ZooKeeper Installation](#).

2. Install each type of daemon package on the appropriate systems(s), as follows.

Where to install	Install commands
Resource Manager host (analogous to MRv1 JobTracker) running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-yarn-resourcemanager</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-yarn-resourcemanager</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get update; sudo apt-get install hadoop-yarn-resourcemanager</code>
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-hdfs-namenode</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-hdfs-namenode</code>
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode</code>

Where to install	Install commands
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-hdfs-secondarynamenode</code>
All cluster hosts except the Resource Manager running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
One host in the cluster running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
All client hosts running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-client</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-client</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-client</code>

**Note:**

The `hadoop-yarn` and `hadoop-hdfs` packages are installed on each system automatically as dependencies of the other packages.

Step 4: Install CDH 5 with MRv1

**Note:**

If you are also installing YARN, you can skip any packages you have already installed in [Step 3: Install CDH 5 with YARN](#) on page 163.

Skip this step and go to [Step 3: Install CDH 5 with YARN](#) on page 163 if you intend to use *only* YARN.



Important: Before proceeding, you need to decide:

- Whether to configure High Availability (HA) for the NameNode or JobTracker; see the [High Availability](#) for more information and instructions.
- Where to deploy the NameNode, Secondary NameNode, and JobTracker daemons. As a general rule:
 - The NameNode and JobTracker run on the same "master" host unless the cluster is large (more than a few tens of nodes), and the master host (or hosts) should not run the Secondary NameNode (if used), DataNode or TaskTracker services.
 - In a large cluster, it is especially important that the Secondary NameNode (if used) runs on a separate machine from the NameNode.
 - Each node in the cluster **except the master host(s)** should run the DataNode and TaskTracker services.

If you decide to configure [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`. After completing the [HA software configuration](#), follow the installation instructions under [Deploying HDFS High Availability](#).

First, install and deploy ZooKeeper.



Important:

Cloudera recommends that you install (or update) and start a ZooKeeper cluster before proceeding. This is a **requirement** if you are deploying high availability (HA) for the NameNode or JobTracker.

Follow instructions under [ZooKeeper Installation](#). Make sure you create the `myid` file in the data directory, as instructed, if you are starting a ZooKeeper ensemble after a fresh install.

Next, install packages.

Install each type of daemon package on the appropriate systems(s), as follows.



Note:

On Ubuntu systems, Ubuntu may try to start the service immediately after you install it. This should fail harmlessly, but if you want to prevent it, there is advice [here](#).

Where to install	Install commands
JobTracker host running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-jobtracker</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-jobtracker</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-jobtracker</code>
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-hdfs-namenode</code>

Where to install	Install commands
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-hdfs-namenode</code>
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-hdfs-secondarynamenode</code>
All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
All client hosts running:	
<i>Red Hat/CentOS compatible</i>	<code>sudo yum clean all; sudo yum install hadoop-client</code>
<i>SLES</i>	<code>sudo zypper clean --all; sudo zypper install hadoop-client</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-client</code>

Step 5: (Optional) Install LZO

If you decide to install LZO (Lempel–Ziv–Oberhumer compression), proceed as follows. For information about choosing a compression format, see [Choosing a Data Compression Format](#)




Note:

If you are upgrading to a new version of LZO, rather than installing it for the first time, you must first remove the old version; for example, on a RHEL system:

```
yum remove hadoop-lzo
```


1. Add the repository on each host in the cluster. Follow the instructions for your OS version:

For OS Version	Do this
Red Hat/CentOS/Oracle 5	Navigate to this link and save the file in the <code>/etc/yum.repos.d/</code> directory.
Red Hat/CentOS 6	Navigate to this link and save the file in the <code>/etc/yum.repos.d/</code> directory.
SLES	<ol style="list-style-type: none"> Run the following command: <pre>\$ sudo zypper addrepo -f https://archive.cloudera.com/gplextras5/sles/11/x86_64/gplextras/ cloudera-gplextras5.repo</pre> Update your system package index by running: <pre>\$ sudo zypper refresh</pre>
Ubuntu or Debian	Navigate to this link and save the file as <code>/etc/apt/sources.list.d/gplextras.list</code> . <div style="border: 1px solid #f0e68c; padding: 10px; margin-top: 10px;">  Important: Make sure you do not let the file name default to <code>cloudera.list</code>, as that will overwrite your existing <code>cloudera.list</code>. </div>

2. Install the package on each host as follows:

For OS version	Install commands
Red Hat/CentOS compatible	<code>sudo yum install hadoop-lzo</code>
SLES	<code>sudo zypper install hadoop-lzo</code>
Ubuntu or Debian	<code>sudo apt-get install hadoop-lzo</code>

3. Continue with installing and deploying CDH. As part of the deployment, you will need to do some additional configuration for LZO, as shown under [Configuring LZO](#) on page 202.

 **Important:** Make sure you do this configuration *after* you have [copied the default configuration files](#) to a custom location and set alternatives to point to it.

Step 6: Deploy CDH and Install Components

Now proceed with:

- [deploying CDH 5](#)
- [installing components](#).

Installing an Earlier CDH 5 Release

Follow these instructions to install a CDH 5 release that is **earlier than the current CDH 5 release**.

A common reason for doing this would be that you need to add new nodes to an existing cluster that is not running the most recent version of CDH 5. For example your cluster might be running CDH 5.0.1 when the most recent release is CDH 5.1.0; in this case, you will want to install CDH 5.0.1 on the new nodes, not CDH 5.1.0. These instructions are tailored for a fresh install (rather than an upgrade), in a cluster not being managed by Cloudera Manager,



Warning:

Do not attempt to use these instructions to roll your cluster back to a previous release. Use them only to expand an existing cluster that you do not want to upgrade to the latest release, or to create a new cluster running a version of CDH 5 that is earlier than the current CDH 5 release.

Installing Cloudera Manager and CDH

Downloading and Installing an Earlier Release

Choose your Linux version and proceed as follows to install an earlier release:

- [On Red Hat-compatible systems](#)
- [On SLES systems](#)
- [On Ubuntu and Debian systems](#)

On Red Hat-compatible systems

Step 1. Download and save the Yum repo file

Click the entry in the table below that matches your Red Hat or CentOS system, navigate to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.

For OS Version	Click this Link
Red Hat/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link
Red Hat/CentOS 6 (64-bit)	Red Hat/CentOS 6 link

Step 2. Edit the repo file

Open the repo file you have just saved and change the 5 at the end of the line that begins `baseurl=` to the version number you want.

For example, if you have saved the file for [Red Hat 6](#), it will look like this when you open it for editing:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/
gpgkey = https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

If you want to install CDH 5.0.1, for example, change

```
baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/ to
```

```
baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.0.1/
```

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.0.1/
gpgkey = https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

Step 3: Proceed with the installation

1. Go to <http://www.cloudera.com/content/cloudera/en/documentation.html>.
2. Use the `Select Version` scroller to find the release you want, for example, select CDH and 5.0.x
3. Find the CDH Installation Guide for your release.
4. Follow the instructions for Red Hat on the "Installing CDH 5" page, starting with the instructions for optionally adding a repository key. (This comes immediately before the steps for installing CDH 5 with MRv1 or YARN, and is usually Step 2.)

On SLES systems

Step 1. Add the Cloudera repo

1. Run the following command:

```
$ sudo zypper addrepo -f
https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

Step 2. Edit the repo file

Open the repo file that you have just added to your system and change the 5 at the end of the line that begins `baseurl=` to the version number you want.

The file should look like this when you open it for editing:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/
gpgkey = https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

If you want to install CDH5.0.1, for example, change

```
baseurl=https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/ to
```

```
baseurl= https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.0.1/
```

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.0.1/
gpgkey = https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

Step 3: Proceed with the installation

1. Go to <http://www.cloudera.com/content/cloudera/en/documentation.html>.
2. Use the Select a Product Version scroller to find the release you want, for example CDH 5.0.x
3. Find the CDH Installation Guide for your release.
4. Follow the instructions for SLES on the "Installing CDH 5" page, starting with the instructions for optionally adding a repository key. (This comes immediately before the steps for installing CDH 5 with MRv1 or YARN, and is usually Step 2.)

On Ubuntu and Debian systems

Proceed as follows to add the Cloudera repo for your operating-system version and the Cloudera release you need.

Step 1: Create the repo File

Create a new file `/etc/apt/sources.list.d/cloudera.list` with the following contents:

- For Ubuntu systems:

```
deb [arch=amd64] https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5
contrib deb-src https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5
contrib
```

Installing Cloudera Manager and CDH

- For Debian systems:

```
deb https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib deb-src  
https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib
```

where: <OS-release-arch> is `debian/wheezy/amd64/cdh` or `ubuntu/precise/amd64/cdh`, and <RELEASE> is the name of your distribution, which you can find by running `lsb_release -c`.

Now replace `-cdh5` near the end of each line (before `contrib`) with the CDH release you need to install. Here are some examples using CDH5.0.1:

For 64-bit Ubuntu Precise:

```
deb [arch=amd64] https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh  
precise-cdh5.0.1 contrib  
deb-src https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh precise-cdh5.0.1  
contrib
```

For Debian Wheezy:

```
deb https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.0.1 contrib  
deb-src https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.0.1 contrib
```

Step 2: Proceed with the installation

1. Go to <http://www.cloudera.com/content/cloudera/en/documentation.html>.
2. Use the Select a Product Version scroller to find the release you want, for example CDH 5.0.x
3. Find the CDH Installation Guide for your release.
4. Follow the instructions for Ubuntu or Debian on the "Installing CDH 5" page, starting with the instructions for optionally adding a repository key. (This comes immediately before the steps for installing CDH5 with MRv1 or YARN, and is usually Step 2.)

CDH 5 and MapReduce

CDH 5 supports two versions of the MapReduce computation framework: MRv1 and MRv2. The default installation in CDH 5 is MapReduce (MRv2) built on the YARN framework. In this document, we refer to this new MapReduce version as YARN. You can use the instructions later in this section to install:

- YARN *or*
- MapReduce (MRv1) *or*
- both implementations.



Important: MapReduce MRv1 and YARN share a common set of configuration files, so it is safe to *configure* both of them. Cloudera does not recommend running MapReduce MRv1 and YARN daemons on the same hosts at the same time. If you want to easily switch between MapReduce MRv1 and YARN, consider using Cloudera Manager [features](#) for managing these services.

MapReduce MRv2 (YARN)

The MRv2 YARN architecture splits 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 hosts instead of TaskTracker daemons. The per-application ApplicationMaster is, in effect, a framework-specific library and negotiates resources from the ResourceManager and works with the NodeManagers to execute and monitor the tasks. For details of this architecture, see [Apache Hadoop NextGen MapReduce \(YARN\)](#).

See also [Migrating from MapReduce 1 \(MRv1\) to MapReduce 2 \(MRv2, YARN\)](#) on page 171.

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

This is a guide to migrating from Apache MapReduce 1 (MRv1) to the Next Generation MapReduce (MRv2 or YARN).

Introduction

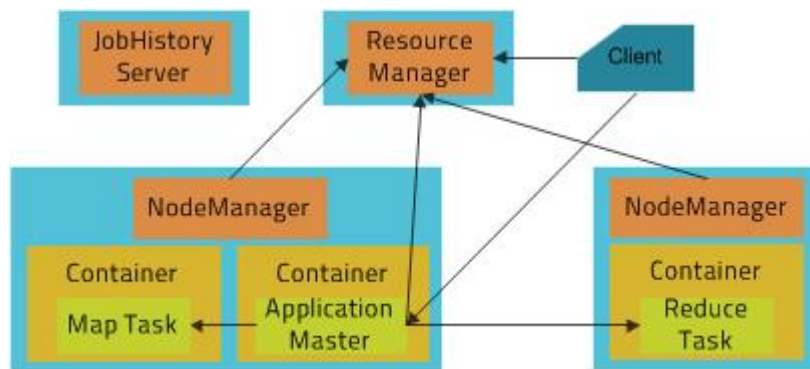
MapReduce 2, or Next Generation MapReduce, is a long needed upgrade to the way that scheduling, resource management, and execution occur in Hadoop. At their core, the improvements separate cluster resource management capabilities from MapReduce-specific logic. They enable Hadoop to share resources dynamically between MapReduce and other parallel processing frameworks, such as Impala, allow more sensible and finer-grained resource configuration for better cluster utilization, and permit it to scale to accommodate more and larger jobs.

This document provides a guide to both the architectural and user-facing changes, so that both cluster operators and MapReduce programmers can easily make the transition.

Terminology and Architecture

MapReduce from Hadoop 1 (MapReduce MRv1) has been split into two components. The cluster resource management capabilities have become YARN (Yet Another Resource Negotiator), while the MapReduce-specific capabilities remain MapReduce. In the MapReduce MRv1 architecture, the cluster was managed by a service called the JobTracker. TaskTracker services lived on each host and would launch tasks on behalf of jobs. The JobTracker would serve information about completed jobs.

In MapReduce MRv2, the functions of the JobTracker have been split between three services. The Resource Manager is a persistent YARN service that receives and runs applications (a MapReduce job is an application) on the cluster. It contains the scheduler, which, as previously, is pluggable. The MapReduce-specific capabilities of the JobTracker have been moved into the MapReduce Application Master, one of which is started to manage each MapReduce job and terminated when the job completes. The JobTracker function of serving information about completed jobs has been moved to the JobHistory Server. The TaskTracker has been replaced with the Node Manager, a YARN service that manages resources and deployment on a host. It is responsible for launching containers, each of which can house a map or reduce task.



The new architecture has its advantages. First, by breaking up the JobTracker into a few different services, it avoids many of the scaling issues faced by MapReduce in Hadoop 1. More importantly, it makes it possible to run frameworks other than MapReduce on a Hadoop cluster. For example, Impala can also run on YARN and [share resources](#) with MapReduce.

For MapReduce Programmers: Writing and Running Jobs

Nearly all jobs written for MRv1 will be able to run without any modifications on an MRv2 cluster.

Java API Compatibility

MRv2 supports both the old (`mapred`) and new (`mapreduce`) MapReduce APIs used for MRv1, with a few caveats. The difference between the old and new APIs, which concerns user-facing changes, should not be confused with the difference between MRv1 and MRv2, which concerns changes to the underlying framework. CDH 4 and CDH 5 both support the new and old MapReduce APIs.

In general, applications that use `@Public/@Stable` APIs will be binary-compatible from CDH 4, meaning that compiled binaries should be able to run without modifications on the new framework. Source compatibility may be broken for applications that make use of a few obscure APIs that are technically public, but rarely needed and primarily exist for internal use. These APIs are detailed below. Source incompatibility means that code changes will be required to compile. It is orthogonal to binary compatibility - binaries for an application that is binary-compatible, but not source-compatible, will continue to run fine on the new framework, but code changes will be required to regenerate those binaries.

	Binary Incompatibilities	Source Incompatibilities
CDH 4 MRv1 to CDH 5 MRv1	None	None
CDH 4 MRv1 to CDH 5 MRv2	None	Rare
CDH 5 MRv1 to CDH 5 MRv2	None	Rare

The following are the known source incompatibilities:

- `KeyValueLineRecordReader#getProgress` and `LineRecordReader#getProgress` now throw `IOExceptions` in both the old and new APIs. Their superclass method, `RecordReader#getProgress`, already did this, but source compatibility will be broken for the rare code that used it without a `try/catch` block.
- `FileOutputCommitter#abortTask` now throws an `IOException`. Its superclass method always did this, but source compatibility will be broken for the rare code that used it without a `try/catch` block. This was fixed in CDH 4.3 MRv1 to be compatible with MRv2.
- `Job#getDependentJobs`, an API marked `@Evolving`, now returns a `List` instead of an `ArrayList`.

Compiling Jobs Against MRv2

If you are using Maven, compiling against MRv2 requires including the same artifact, `hadoop-client`. Changing the version to Hadoop 2 version (for example, using `2.2.0-cdh5.0.0` instead of `2.0.0-mr1-cdh4.3.0`) should be enough. If you are not using Maven, compiling against all the Hadoop JARs is recommended. A comprehensive list of Hadoop Maven artifacts is available at: [Using the CDH 5 Maven Repository](#).

Job Configuration

As in MRv1, job configuration options can be specified on the command line, in Java code, or in the `mapred-site.xml` on the client machine in the same way they previously were. The vast majority of job configuration options that were available in MRv1 work in MRv2 as well. For consistency and clarity, many options have been given new names. The older names are deprecated, but will still work for the time being. The exceptions to this are `mapred.child.ulimit` and all options relating to JVM reuse, as these are no longer supported.

Submitting and Monitoring Jobs

The MapReduce command line interface remains entirely compatible. Use of the Hadoop command line tool to run MapReduce related commands (`pipes`, `job`, `queue`, `classpath`, `historyserver`, `distcp`, `archive`) is deprecated, but still works. The `mapred` command line tool is preferred for these commands.

Selecting Appropriate JAR files for your Jobs

The following table shows the names and locations of the JAR files used in MRv1 and the corresponding names and locations in YARN:

Name	MapReduce MRv1 location	YARN location
Streaming	<code>/usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh<version>.jar</code>	<code>/usr/lib/hadoop-mapreduce/hadoop-streaming.jar</code>
Rumen	N/A	<code>/usr/lib/hadoop-mapreduce/hadoop-rumen.jar</code>
Hadoop Examples	<code>/usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar</code>	<code>/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar</code>
DistCp v1	<code>/usr/lib/hadoop-0.20-mapreduce/hadoop-tools.jar</code>	<code>/usr/lib/hadoop-mapreduce/hadoop-extras.jar</code>

Name	MapReduce MRv1 location	YARN location
DistCp v2	N/A	/usr/lib/hadoop-mapreduce/hadoop-distcp.jar
Hadoop archives	/usr/lib/hadoop-0.20-mapreduce/hadoop-tools.jar	/usr/lib/hadoop-mapreduce/hadoop-archives.jar

Requesting Resources

A MapReduce job submission includes the amount of resources to reserve for each map and reduce task. As in MapReduce 1, the amount of memory requested is controlled by the `mapreduce.map.memory.mb` and `mapreduce.reduce.memory.mb` properties.

MapReduce 2 adds additional parameters that control how much processing power to reserve for each task as well. The `mapreduce.map.cpu.vcores` and `mapreduce.reduce.cpu.vcores` properties express how much parallelism a map or reduce task can take advantage of. These should remain at their default value of 1 unless your code is explicitly spawning extra compute-intensive threads.

For Administrators: Configuring and Running MRv2 Clusters Configuration Migration

Since MapReduce 1 functionality has been split into two components, MapReduce cluster configuration options have been split into YARN configuration options, which go in `yarn-site.xml`, and MapReduce configuration options, which go in `mapred-site.xml`. Many have been given new names to reflect the shift. As JobTrackers and TaskTrackers no longer exist in MRv2, all configuration options pertaining to them no longer exist, although many have corresponding options for the ResourceManager, NodeManager, and JobHistoryServer.

A minimal configuration required to run MRv2 jobs on YARN is:

- `yarn-site.xml` configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>you.hostname.com</value>
  </property>

  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

- `mapred-site.xml` configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

See [Deploying MapReduce v2 \(YARN\) on a Cluster](#) on page 204 for instructions for a full deployment.

Resource Configuration

One of the larger changes in MRv2 is the way that resources are managed. In MRv1, each host was configured with a fixed number of map slots and a fixed number of reduce slots. Under YARN, there is no distinction between resources available for maps and resources available for reduces - all resources are available for both. Second, the notion of slots has been discarded, and resources are now configured in terms of amounts of memory (in megabytes) and CPU (in “virtual cores”, which are described below). Resource configuration is an inherently difficult topic, and the added

flexibility that YARN provides in this regard also comes with added complexity. Cloudera Manager will pick sensible values automatically, but if you are setting up your cluster manually or just interested in the details, read on.

Configuring Memory Settings for YARN and MRv2

The memory configuration for YARN and MRv2 memory is important to get the best performance from your cluster. Several different settings are involved. The table below shows the default settings, as well as the settings that Cloudera recommends, for each configuration option. See [Managing MapReduce and YARN](#) for more configuration specifics and, for detailed tuning advice with sample configurations, see [Tuning YARN](#) on page 182.

Table 22: YARN and MRv2 Memory Configuration

Cloudera Manager Property Name	CDH Property Name	Default Configuration	Cloudera Tuning Guidelines
Container Memory Minimum	<code>yarn.scheduler.minimum-allocation-mb</code>	1 GB	0
Container Memory Maximum	<code>yarn.scheduler.maximum-allocation-mb</code>	64 GB	amount of memory on largest node
Container Memory Increment	<code>yarn.scheduler.increment-allocation-mb</code>	512 MB	Use a fairly large value, such as 128 MB
Container Memory	<code>yarn.nodemanager.resource.memory-mb</code>	8 GB	8 GB
Map Task Memory	<code>mapreduce.map.memory.mb</code>	1 GB	1 GB
Reduce Task Memory	<code>mapreduce.reduce.memory.mb</code>	1 GB	1 GB
Map Task Java Opts Base	<code>mapreduce.map.java.opts</code>	<code>-Djava.net.preferIPv4Stack=true</code>	<code>-Djava.net.preferIPv4Stack=true</code> <code>-Xmx768m</code>
Reduce Task Java Opts Base	<code>mapreduce.reduce.java.opts</code>	<code>-Djava.net.preferIPv4Stack=true</code>	<code>-Djava.net.preferIPv4Stack=true</code> <code>-Xmx768m</code>
ApplicationMaster Memory	<code>yarn.app.mapreduce.am.resource.mb</code>	1 GB	1 GB
ApplicationMaster Java Opts Base	<code>yarn.app.mapreduce.am.command-opts</code>	<code>-Djava.net.preferIPv4Stack=true</code>	<code>-Djava.net.preferIPv4Stack=true</code> <code>-Xmx768m</code>

Resource Requests

From the perspective of a developer requesting resource allocations for a job's tasks, nothing needs to be changed. Map and reduce task memory requests still work and, additionally, tasks that will use multiple threads can request more than 1 core with the `mapreduce.map.cpu.vcores` and `mapreduce.reduce.cpu.vcores` properties.

Configuring Host Capacities

In MRv1, the `mapred.tasktracker.map.tasks.maximum` and `mapred.tasktracker.reduce.tasks.maximum` properties dictated how many map and reduce slots each TaskTracker had. These properties no longer exist in YARN. Instead, YARN uses `yarn.nodemanager.resource.memory-mb` and `yarn.nodemanager.resource.cpu-vcores`, which control the amount of memory and CPU on each host, both available to both maps and reduces. If you were using Cloudera Manager to configure these automatically, Cloudera Manager will take care of it in MRv2 as well. If configuring these manually, simply set these to the amount of memory and number of cores on the machine after subtracting out resources needed for other services.

Virtual Cores

To better handle varying CPU requests, YARN supports virtual cores (vcores), a resource meant to express parallelism. The “virtual” in the name is somewhat misleading - on the NodeManager, vcores should be configured equal to the

number of physical cores on the machine. Tasks should be requested with `vcores` equal to the number of cores they can saturate at once. Currently `vcores` are very coarse - tasks will rarely want to ask for more than one of them, but a complementary axis that represents processing power may be added in the future to enable finer-grained resource configuration.

Rounding Request Sizes

Also noteworthy are the `yarn.scheduler.minimum-allocation-mb`, `yarn.scheduler.minimum-allocation-vcores`, `yarn.scheduler.increment-allocation-mb`, and `yarn.scheduler.increment-allocation-vcores` properties, which default to 1024, 1, 512, and 1 respectively. If tasks are submitted with resource requests lower than the minimum-allocation values, their requests will be set to these values. If tasks are submitted with resource requests that are not multiples of the increment-allocation values, their requests will be rounded up to the nearest increments.

To make all of this more concrete, let's use an example. Each host in the cluster has 24 GB of memory and 6 cores. Other services running on the nodes require 4 GB and 1 core, so we set `yarn.nodemanager.resource.memory-mb` to 20480 and `yarn.nodemanager.resource.cpu-vcores` to 5. If you leave the map and reduce task defaults of 1024 MB and 1 virtual core intact, you will have at most 5 tasks running at the same time. If you want each of your tasks to use 5 GB, set their `mapreduce.(map|reduce).memory.mb` to 5120, which would limit you to 4 tasks running at the same time.

Scheduler Configuration

Cloudera recommends using the Fair Scheduler in MRv2. (FIFO and Capacity Scheduler are also available.) Fair Scheduler allocation files require changes in light of the new way that resources work. The `minMaps`, `maxMaps`, `minReduces`, and `maxReduces` queue properties have been replaced with a `minResources` property and a `maxProperties`. Instead of taking a number of slots, these properties take a value like "1024 MB, 3 vcores". By default, the MRv2 Fair Scheduler will attempt to equalize memory allocations in the same way it attempted to equalize slot allocations in MRv1. The MRv2 Fair Scheduler contains a number of new features including hierarchical queues and fairness based on multiple resources.

Administration Commands

The `jobtracker` and `tasktracker` commands, which start the JobTracker and TaskTracker, are no longer supported because these services no longer exist. They are replaced with "`yarn resourcemanager`" and "`yarn nodemanager`", which start the ResourceManager and NodeManager respectively. "`hadoop mradmin`" is no longer supported. Instead, "`yarn radmin`" should be used. The new admin commands mimic the functionality of the MRv1 names, allowing nodes, queues, and ACLs to be refreshed while the ResourceManager is running.

Security

The following section outlines the additional changes needed to migrate a secure cluster.

New YARN Kerberos service principals should be created for the ResourceManager and NodeManager, using the pattern used for other Hadoop services, that is, `yarn@HOST`. The `mapred` principal should still be used for the JobHistory Server. If you are using Cloudera Manager to configure security, this will be taken care of automatically.

As in MRv1, a configuration must be set to have the user that submits a job own its task processes. The equivalent of MRv1's `LinuxTaskController` is the `LinuxContainerExecutor`. In a secure setup, NodeManager configurations should set `yarn.nodemanager.container-executor.class` to `org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor`. Properties set in the `taskcontroller.cfg` configuration file should be migrated to their analogous properties in the `container-executor.cfg` file.

In secure setups, configuring `hadoop-policy.xml` allows administrators to set up access control lists on internal protocols. The following is a table of MRv1 options and their MRv2 equivalents:

MRv1	MRv2	Comment
<code>security.task.umbilical.protocol.acl</code>	<code>security.job.task.protocol.acl</code>	As in MRv1, this should never be set to anything other than *

MRv1	MRv2	Comment
<code>security.inter.tracker.protocol.acl</code>	<code>security.resourcetracker.protocol.acl</code>	
<code>security.job.submission.protocol.acl</code>	<code>security.applicationclient.protocol.acl</code>	
<code>security.admin.operations.protocol.acl</code>	<code>security.resourcemanagement-administration.protocol.acl</code>	
	<code>security.applicationmaster.protocol.acl</code>	No MRv1 equivalent
	<code>security.containermanagement.protocol.acl</code>	No MRv1 equivalent
	<code>security.resourcelocalizer.protocol.acl</code>	No MRv1 equivalent
	<code>security.job.client.protocol.acl</code>	No MRv1 equivalent

Queue access control lists (ACLs) are now placed in the Fair Scheduler configuration file instead of the JobTracker configuration. A list of users and groups that can submit jobs to a queue can be placed in `aclSubmitApps` in the queue's configuration. The queue administration ACL is no longer supported, but will be in a future release.

Ports

The following is a list of default ports used by MRv2 and YARN, as well as the configuration properties used to configure them.

Port	Use	Property
8032	ResourceManager Client RPC	<code>yarn.resourcemanager.address</code>
8030	ResourceManager Scheduler RPC (for ApplicationMasters)	<code>yarn.resourcemanager.scheduler.address</code>
8033	ResourceManager Admin RPC	<code>yarn.resourcemanager.admin.address</code>
8088	ResourceManager Web UI and REST APIs	<code>yarn.resourcemanager.webapp.address</code>
8031	ResourceManager Resource Tracker RPC (for NodeManagers)	<code>yarn.resourcemanager.resource-tracker.address</code>
8040	NodeManager Localizer RPC	<code>yarn.nodemanager.localizer.address</code>
8042	NodeManager Web UI and REST APIs	<code>yarn.nodemanager.webapp.address</code>
10020	Job History RPC	<code>mapreduce.jobhistory.address</code>
19888	Job History Web UI and REST APIs	<code>mapreduce.jobhistory.webapp.address</code>
13562	Shuffle HTTP	<code>mapreduce.shuffle.port</code>

High Availability

YARN supports ResourceManager HA to make a YARN cluster highly-available; the underlying architecture of active-standby pair is similar to JobTracker HA in MRv1. A major improvement over MRv1 is: in YARN, the completed tasks of in-flight MapReduce jobs are not re-run on recovery after the ResourceManager is restarted or failed over. Further, the configuration and setup has also been simplified. The main differences are:

1. Failover controller has been moved from a separate ZKFC daemon to be a part of the ResourceManager itself. So, there is no need to run an additional daemon.
2. Clients, applications, and NodeManagers do not require configuring a proxy-provider to talk to the active ResourceManager.

Below is a table with HA-related configurations used in MRv1 and their equivalents in YARN:

MRv1	YARN / MRv2	Comment
<code>mapred.jobtrackers.name</code>	<code>yarn.resourcemanager.ha.rm-ids</code>	

MRv1	YARN / MRv2	Comment
<code>mapred.ha.jobtracker.id</code>	<code>yarn.resourcemanager.ha.id</code>	Unlike in MRv1, this must be configured in YARN.
<code>mapred.jobtracker.rpc-address.name.id</code>	See	YARN/ MRv2 has different RPC ports for different functionalities. Each port-related configuration must be suffixed with an id. Note that there is no <name> in YARN.
<code>mapred.ha.jobtracker.rpc-address.name.id</code>	<code>yarn.resourcemanager.ha.admin.address</code>	See Configuring YARN (MRv2) ResourceManager High Availability Using the Command Line
<code>mapred.ha.fencing.methods</code>	<code>yarn.resourcemanager.ha.fencer</code>	Not required to be specified
<code>mapred.client.failover.*</code>	None	Not required
	<code>yarn.resourcemanager.ha.enabled</code>	Enable HA
<code>mapred.jobtracker.restart.recover</code>	<code>yarn.resourcemanager.recovery.enabled</code>	Enable recovery of jobs after failover
	<code>yarn.resourcemanager.store.class</code>	<code>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</code>
<code>mapred.ha.automatic-failover.enabled</code>	<code>yarn.resourcemanager.ha.automatic-failover.enabled</code>	Enable automatic failover
<code>mapred.ha.zkfc.port</code>	<code>yarn.resourcemanager.ha.automatic-failover.port</code>	
<code>mapred.job.tracker</code>	<code>yarn.resourcemanager.cluster.id</code>	Cluster name

Upgrading an MRv1 Installation with Cloudera Manager

See [Importing MapReduce Configurations to YARN](#) for instructions.

Manually Upgrading an MRv1 Installation

The following packages are no longer used in MRv2 and should be uninstalled: `hadoop-0.20-mapreduce`, `hadoop-0.20-mapreduce-jobtracker`, `hadoop-0.20-mapreduce-tasktracker`, `hadoop-0.20-mapreduce-zkfc`, `hadoop-0.20-mapreduce-jobtrackerha`

The following additional packages must be [installed](#): `hadoop-yarn`, `hadoop-mapreduce`, `hadoop-mapreduce-historyserver`, `hadoop-yarn-resourcemanager`, `hadoop-yarn-nodemanager`.

The next step is to look at all the service configurations placed in `mapred-site.xml` and replace them with their corresponding YARN configuration. Configurations starting with `yarn` should be placed inside `yarn-site.xml`, not `mapred-site.xml`. Refer to the Resource Configuration section above for best practices on how to convert TaskTracker slot capacities (`mapred.tasktracker.map.tasks.maximum` and `mapred.tasktracker.reduce.tasks.maximum`) to NodeManager resource capacities (`yarn.nodemanager.resource.memory-mb` and

yarn.nodemanager.resource.cpu-vcores), as well as how to convert configurations in the Fair Scheduler allocations file, fair-scheduler.xml.

Finally, you can start the ResourceManager, NodeManagers and the JobHistoryServer.

Web UI

In MRv1, the JobTracker Web UI served detailed information about the state of the cluster and the jobs (recent and current) running on it. It also contained the job history page, which served information from disk about older jobs.

The MRv2 Web UI provides the same information structured in the same way, but has been revamped with a new look and feel. The ResourceManager’s UI, which includes information about running applications and the state of the cluster, is now located by default at <ResourceManager host>:8088. The JobHistory UI is now located by default at <JobHistoryServer host>:19888. Jobs can be searched and viewed there just as they could in MapReduce 1.

Because the ResourceManager is meant to be agnostic to many of the concepts in MapReduce, it cannot host job information directly. Instead, it proxies to a Web UI that can. If the job is running, this proxy is the relevant MapReduce Application Master; if the job has completed, then this proxy is the JobHistoryServer. Thus, the user experience is similar to that of MapReduce 1, but the information is now coming from different places.

Summary of Configuration Changes

The following tables summarize the changes in configuration parameters between MRv1 and MRv2.

JobTracker Properties and ResourceManager Equivalents

MRv1	YARN / MRv2
mapred.jobtracker.taskScheduler	yarn.resourcemanager.scheduler.class
mapred.jobtracker.completeuserjobs.maximum	yarn.resourcemanager.max-completed-applications
mapred.jobtracker.restart.recover	yarn.resourcemanager.recovery.enabled
mapred.job.tracker	yarn.resourcemanager.hostname or all of the following: yarn.resourcemanager.address yarn.resourcemanager.scheduler.address yarn.resourcemanager.resource-tracker.address yarn.resourcemanager.admin.address
mapred.job.tracker.http.address	yarn.resourcemanager.webapp.address or yarn.resourcemanager.hostname
mapred.job.tracker.handler.count	yarn.resourcemanager.resource-tracker.client.thread-count
mapred.hosts	yarn.resourcemanager.nodes.include-path
mapred.hosts.exclude	yarn.resourcemanager.nodes.exclude-path
mapred.cluster.max.map.memory.mb	yarn.scheduler.maximum-allocation-mb
mapred.cluster.max.reduce.memory.mb	yarn.scheduler.maximum-allocation-mb
mapred.acls.enabled	yarn.acl.enable
mapreduce.cluster.acls.enabled	yarn.acl.enable

JobTracker Properties and JobHistoryServer Equivalents

MRv1	YARN / MRv2	Comment
mapred.job.tracker.retiredjobs.cache.size	mapreduce.jobhistory.joblist.cache.size	
mapred.job.tracker.jobhistory.lru.cache.size	mapreduce.jobhistory.loadedjobs.cache.size	

MRv1	YARN / MRv2	Comment
mapred.job.tracker.history.completed.location	mapreduce.jobhistory.done-dir	Local FS in MR1; stored in HDFS in MR2
hadoop.job.history.user.location	mapreduce.jobhistory.done-dir	
hadoop.job.history.location	mapreduce.jobhistory.done-dir	

JobTracker Properties and MapReduce ApplicationMaster Equivalents

MRv1	YARN / MRv2	Comment
mapreduce.jobtracker.staging.root.dir	yarn.app.mapreduce.am.staging-dir	Now configurable per job

TaskTracker Properties and NodeManager Equivalents

MRv1	YARN / MRv2
mapred.tasktracker.map.tasks.maximum	yarn.nodemanager.resource.memory-mb and yarn.nodemanager.resource.cpu-vcores
mapred.tasktracker.reduce.tasks.maximum	yarn.nodemanager.resource.memory-mb and yarn.nodemanager.resource.cpu-vcores
mapred.tasktracker.expiry.interval	yarn.nm.liveliness-monitor.expiry-interval-ms
mapred.tasktracker.resourcecalculatorplugin	yarn.nodemanager.container-monitor.resource-calculator.class
mapred.tasktracker.taskmemorymanager.monitoring-interval	yarn.nodemanager.container-monitor.interval-ms
mapred.tasktracker.tasks.sleep-time-before-sigkill	yarn.nodemanager.sleep-delay-before-sigkill.ms
mapred.task.tracker.task-controller	yarn.nodemanager.container-executor.class
mapred.local.dir	yarn.nodemanager.local-dirs
mapreduce.cluster.local.dir	yarn.nodemanager.local-dirs
mapred.disk.healthChecker.interval	yarn.nodemanager.disk-health-checker.interval-ms
mapred.healthChecker.script.path	yarn.nodemanager.health-checker.script.path
mapred.healthChecker.interval	yarn.nodemanager.health-checker.interval-ms
mapred.healthChecker.script.timeout	yarn.nodemanager.health-checker.script.timeout-ms
mapred.healthChecker.script.args	yarn.nodemanager.health-checker.script.opts
local.cache.size	yarn.nodemanager.localizer.cache.target-size-mb
mapreduce.tasktracker.cache.local.size	yarn.nodemanager.localizer.cache.target-size-mb

TaskTracker Properties and Shuffle Service Equivalents

The table that follows shows TaskTracker properties and their equivalents in the auxiliary shuffle service that runs inside NodeManagers.

MRv1	YARN / MRv2
tasktracker.http.threads	mapreduce.shuffle.max.threads
mapred.task.tracker.http.address	mapreduce.shuffle.port

MRv1	YARN / MRv2
mapred.tasktracker.indexcache.mb	mapred.tasktracker.indexcache.mb

Per-Job Configuration Properties

Many of these properties have new names in MRv2, but the MRv1 names will work for all properties except `mapred.job.restart.recover`.

MRv1	YARN / MRv2	Comment
io.sort.mb	mapreduce.task.io.sort.mb	MRv1 name still works
io.sort.factor	mapreduce.task.io.sort.factor	MRv1 name still works
io.sort.spill.percent	mapreduce.task.io.sort.spill.percent	MRv1 name still works
mapred.map.tasks	mapreduce.job.maps	MRv1 name still works
mapred.reduce.tasks	mapreduce.job.reduces	MRv1 name still works
mapred.job.map.memory.mb	mapreduce.map.memory.mb	MRv1 name still works
mapred.job.reduce.memory.mb	mapreduce.reduce.memory.mb	MRv1 name still works
mapred.map.child.log.level	mapreduce.map.log.level	MRv1 name still works
mapred.reduce.child.log.level	mapreduce.reduce.log.level	MRv1 name still works
mapred.inmem.merge.threshold	mapreduce.reduce.shuffle.merge.inmem.threshold	MRv1 name still works
mapred.job.shuffle.merge.percent	mapreduce.reduce.shuffle.merge.percent	MRv1 name still works
mapred.job.shuffle.input.buffer.percent	mapreduce.reduce.shuffle.input.buffer.percent	MRv1 name still works
mapred.job.reduce.input.buffer.percent	mapreduce.reduce.input.buffer.percent	MRv1 name still works
mapred.map.tasks.speculative.execution	mapreduce.map.speculative	Old one still works
mapred.reduce.tasks.speculative.execution	mapreduce.reduce.speculative	MRv1 name still works
mapred.min.split.size	mapreduce.input.fileinputformat.split.minsize	MRv1 name still works
keep.failed.task.files	mapreduce.task.files.preserve.failedtasks	MRv1 name still works
mapred.output.compress	mapreduce.output.fileoutputformat.compress	MRv1 name still works
mapred.map.output.compression.codec	mapreduce.map.output.compress.codec	MRv1 name still works
mapred.compress.map.output	mapreduce.map.output.compress	MRv1 name still works
mapred.output.compression.type	mapreduce.output.fileoutputformat.compress.type	MRv1 name still works
mapred.userlog.limit.kb	mapreduce.task.userlog.limit.kb	MRv1 name still works
jobclient.output.filter	mapreduce.client.output.filter	MRv1 name still works
jobclient.completion.poll.interval	mapreduce.client.completion.pollinterval	MRv1 name still works
jobclient.progress.monitor.poll.interval	mapreduce.client.progressmonitor.pollinterval	MRv1 name still works
mapred.task.profile	mapreduce.task.profile	MRv1 name still works
mapred.task.profile.maps	mapreduce.task.profile.maps	MRv1 name still works
mapred.task.profile.reduces	mapreduce.task.profile.reduces	MRv1 name still works
mapred.line.input.format.linespermap	mapreduce.input.lineinputformat.linespermap	MRv1 name still works
mapred.skip.attempts.to.start.skipping	mapreduce.task.skip.start.attempts	MRv1 name still works

MRv1	YARN / MRv2	Comment
mapred.skip.map.auto.incr.proc.count	mapreduce.map.skip.proc.count.autoincr	MRv1 name still works
mapred.skip.reduce.auto.incr.proc.count	mapreduce.reduce.skip.proc.count.autoincr	MRv1 name still works
mapred.skip.out.dir	mapreduce.job.skip.outdir	MRv1 name still works
mapred.skip.map.max.skip.records	mapreduce.map.skip.maxrecords	MRv1 name still works
mapred.skip.reduce.max.skip.groups	mapreduce.reduce.skip.maxgroups	MRv1 name still works
job.end.retry.attempts	mapreduce.job.end-notification.retry.attempts	MRv1 name still works
job.end.retry.interval	mapreduce.job.end-notification.retry.interval	MRv1 name still works
job.end.notification.url	mapreduce.job.end-notification.url	MRv1 name still works
mapred.merge.recordsBeforeProgress	mapreduce.task.merge.progress.records	MRv1 name still works
mapred.job.queue.name	mapreduce.job.queue.name	MRv1 name still works
mapred.reduce.slowstart.completed.maps	mapreduce.job.reduce.slowstart.completedmaps	MRv1 name still works
mapred.map.max.attempts	mapreduce.map.maxattempts	MRv1 name still works
mapred.reduce.max.attempts	mapreduce.reduce.maxattempts	MRv1 name still works
mapred.reduce.parallel.copies	mapreduce.reduce.shuffle.parallelcopies	MRv1 name still works
mapred.task.timeout	mapreduce.task.timeout	MRv1 name still works
mapred.max.tracker.failures	mapreduce.job.maxtaskfailures.per.tracker	MRv1 name still works
mapred.job.restart.recover	mapreduce.am.max-attempts	
mapred.combine.recordsBeforeProgress	mapreduce.task.combine.progress.records	MRv1 name should still work - see MAPREDUCE-5130

Miscellaneous Properties

MRv1	YARN / MRv2
mapred.heartbeats.in.second	yarn.resourcemanager.nodemangers.heartbeat-interval-ms
mapred.userlog.retain.hours	yarn.log-aggregation.retain-seconds

MRv1 Properties that have no MRv2 Equivalents

MRv1	Comment
mapreduce.tasktracker.group	
mapred.child.ulimit	
mapred.tasktracker.dns.interface	
mapred.tasktracker.dns.nameserver	
mapred.tasktracker.instrumentation	NodeManager does not accept instrumentation
mapred.job.reuse.jvm.num.tasks	JVM reuse no longer supported
mapreduce.job.jvm.numtasks	JVM reuse no longer supported
mapred.task.tracker.report.address	No need for this, as containers do not use IPC with NodeManagers, and ApplicationMaster ports are chosen at runtime

MRv1	Comment
<code>mapreduce.task.tmp.dir</code>	No longer configurable. Now always <code>tmp/</code> (under container's local dir)
<code>mapred.child.tmp</code>	No longer configurable. Now always <code>tmp/</code> (under container's local dir)
<code>mapred.temp.dir</code>	
<code>mapred.jobtracker.instrumentation</code>	ResourceManager does not accept instrumentation
<code>mapred.jobtracker.plugins</code>	ResourceManager does not accept plugins
<code>mapred.task.cache.level</code>	
<code>mapred.queue.names</code>	These go in the scheduler-specific configuration files
<code>mapred.system.dir</code>	
<code>mapreduce.tasktracker.cache.local.numberdirectories</code>	
<code>mapreduce.reduce.input.limit</code>	
<code>io.sort.record.percent</code>	Tuned automatically (MAPREDUCE-64)
<code>mapred.cluster.map.memory.mb</code>	Not necessary; MRv2 uses resources instead of slots
<code>mapred.cluster.reduce.memory.mb</code>	Not necessary; MRv2 uses resources instead of slots
<code>mapred.max.tracker.blacklists</code>	
<code>mapred.jobtracker.maxtasks.per.job</code>	Related configurations go in scheduler-specific configuration files
<code>mapred.jobtracker.taskScheduler.maxRunningTasksPerJob</code>	Related configurations go in scheduler-specific configuration files
<code>io.map.index.skip</code>	
<code>mapred.user.jobconf.limit</code>	
<code>mapred.local.dir.minspacestart</code>	
<code>mapred.local.dir.minspacekill</code>	
<code>hadoop.rpc.socket.factory.class.JobSubmissionProtocol</code>	
<code>mapreduce.tasktracker.outofband.heartbeat</code>	Always on
<code>mapred.jobtracker.job.history.block.size</code>	

Tuning YARN

This topic applies to YARN clusters only, and describes how to tune and optimize YARN for your cluster.

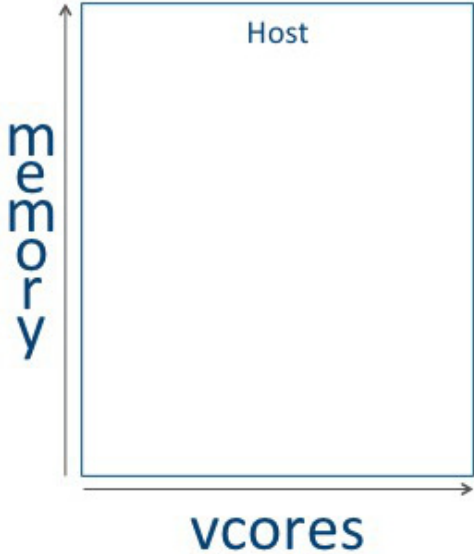


Note: Download the Cloudera [YARN tuning spreadsheet](#) to help calculate YARN configurations. For a short video overview, see [Tuning YARN Applications](#).

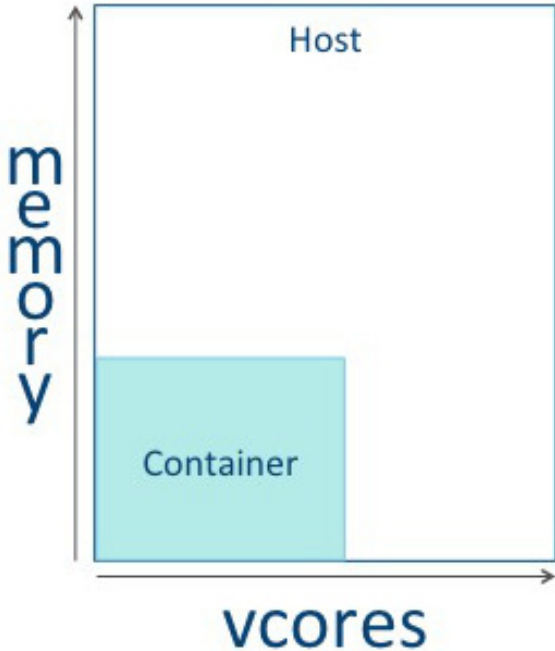
Overview

This overview provides an abstract description of a YARN cluster and the goals of YARN tuning.

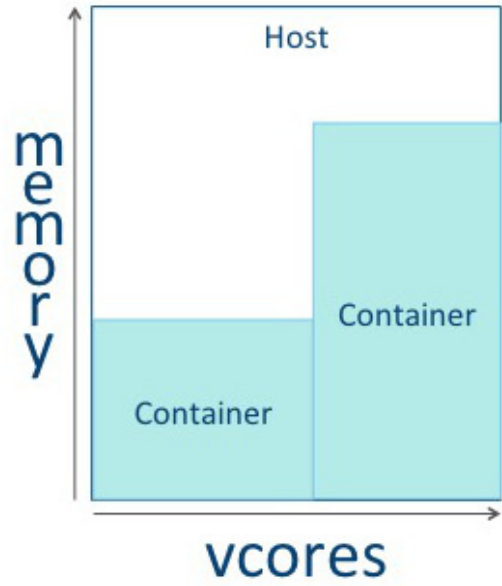
A YARN cluster is composed of host machines. Hosts provide memory and CPU resources. A *vc*ore, or virtual core, is a usage share of a host CPU.



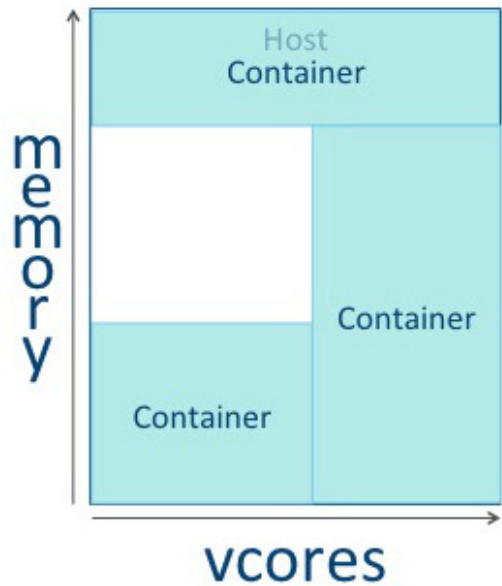
Tuning YARN consists primarily of optimally defining *containers* on your worker hosts. You can think of a container as a rectangular graph consisting of memory and vcores. Containers perform tasks.



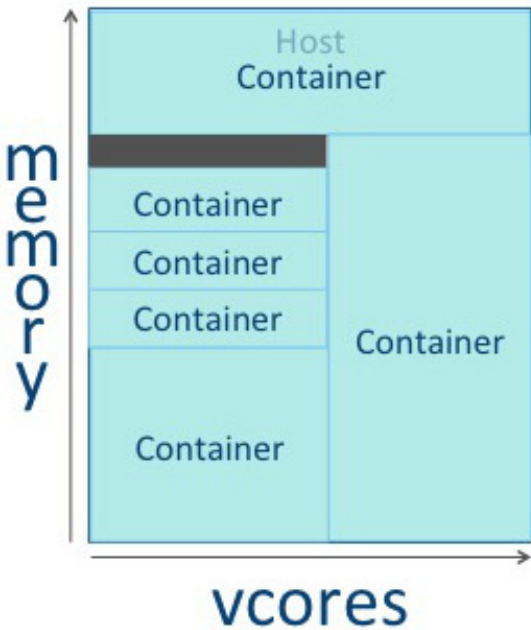
Some tasks use a great deal of memory, with minimal processing on a large volume of data.



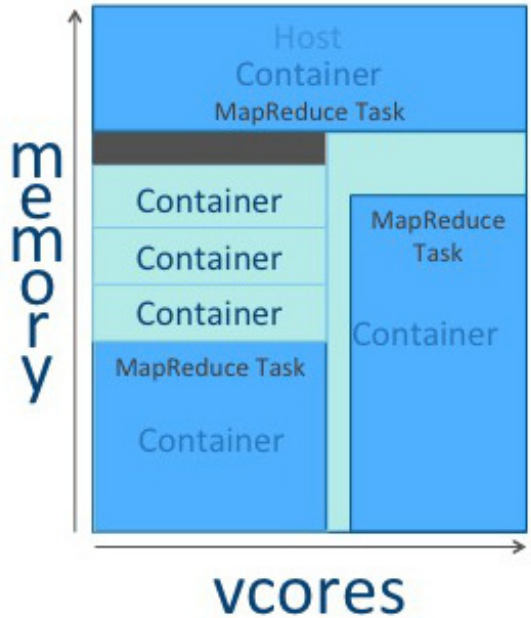
Other tasks require a great deal of processing power, but use less memory. For example, a Monte Carlo Simulation that evaluates many possible "what if?" scenarios uses a great deal of processing power on a relatively small dataset.



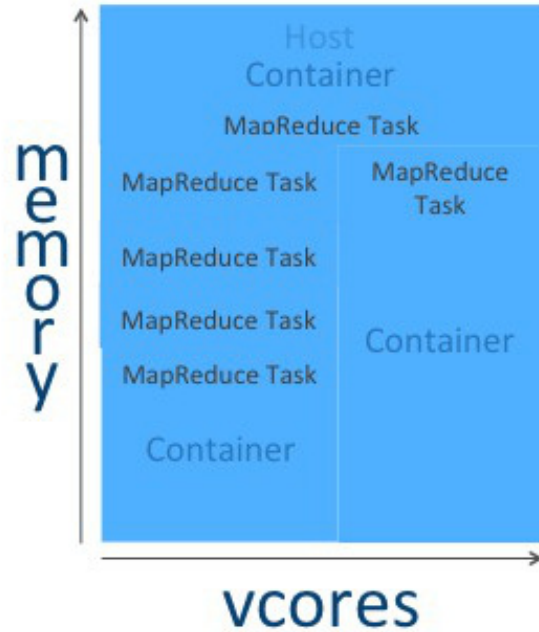
The YARN Resource Manager allocates memory and vcores to use all available resources in the most efficient way possible. Ideally, few or no resources are left idle.



An *application* is a YARN client program consisting of one or more tasks. Typically, a task uses all of the available resources in the container. A task cannot consume more than its designated allocation, ensuring that it cannot use all of the host CPU cycles or exceed its memory allotment.



Tune your YARN hosts to optimize the use of vcores and memory by configuring your containers to use all available resources, beyond those required for overhead and other services.



There are three phases to YARN tuning. The phases correspond to the tabs in the [YARN tuning spreadsheet](#).

1. Cluster configuration, where you configure your hosts.
2. YARN configuration, where you quantify memory and vcores.
3. MapReduce configuration, where you allocate minimum and maximum resources for specific map and reduce tasks.

YARN and MapReduce have many configurable properties. For a complete list, see [Cloudera Manager Configuration Properties](#). The YARN tuning spreadsheet lists the essential subset of these properties that are most likely to improve performance for common MapReduce applications.

Cluster Configuration

In the Cluster Configuration tab, you define the worker host configuration and cluster size for your YARN implementation.

Step 1: Worker Host Configuration

Step 1 is to define the configuration for a single worker host computer in your cluster.

STEP 1: Worker Host Configuration

Enter your likely machine configuration in the input boxes below. If you are uncertain what machines you plan on buying, put in some minimum values that will suit what you expect to buy. Last updated early 2016.

Host Components	Quantity	Description
RAM	256	Gigabytes
CPU	48	8 CPUs: 6 cores, 3.5 GHz, 15MB cache
HDD (Hard Disk Drive)	36	12x3TB SATA III Hard Drives in JBOD Configuration
Ethernet	2	1 Gigabit Ethernet

As with any system, the more memory and CPU resources available, the faster the cluster can process large amounts of data. A machine with 8 CPUs, each with 6 cores, provides 48 vcores per host.

3 TB hard drives in a 2-unit server installation with 12 available slots in JBOD (Just a Bunch Of Disks) configuration is a reasonable balance of performance and pricing at the time the spreadsheet was created. The cost of storage decreases over time, so you might consider 4 TB disks. Larger disks are expensive and not required for all use cases.

Two 1-Gigabit Ethernet ports provide sufficient throughput at the time the spreadsheet was published, but 10-Gigabit Ethernet ports are an option where price is of less concern than speed.

Step 2: Worker Host Planning

Step 2 is to allocate resources on each worker machine.

STEP 2: Worker Host Planning

Now that you have your base Host configuration from Step 1, use the table below to allocate resources, mainly CPU and memory, to the various software components that run on the host.

Service	Category	CPU (cores)	Memory (MB)	CM Static Service %	Notes
Operating System	Overhead	1	8192		N/A Most operating systems use 4-8GB minimum.
Cloudera Manager agent	Overhead	1	1024		N/A Allocate 1GB for Cloudera Manager agents, which track resource usage on a host.
Other services	Overhead	0	0		N/A Enter the required cores or memory for services not listed above.
HDFS DataNode	CDH	1	1024		4 Allocate 1GB for the HDFS DataNode.
Impala daemon	CDH	0	0		0 (Optional Service) Suggestion: Allocate at least 16GB memory when using Impala.
Hbase RegionServer	CDH	0	0		0 (Optional Service) Suggestion: Allocate no more than 12-16GB memory when using HBase Region Servers.
Solr Server	CDH	0	0		0 (Optional Service) Suggestion: Minimum 1GB for Solr server. More will be necessary depending on index sizes.
YARN NodeManager	CDH	1	1024		N/A Allocate 1GB for the YARN NodeManager.
Available Resources		44	250880		
Physical Cores to Vcores Multiplier		4			Set this ratio based on the expected number of concurrent threads per core. Use 1 for CPU intensive tasks up to 4 for standard I/O bound tasks.
YARN Available Vcores		176			This value will be used in STEP 4 for YARN Configuration
YARN Available Memory			250880		This value will be used in STEP 4 for YARN Configuration

Start with at least 8 GB for your operating system, and 1 GB for Cloudera Manager. If services outside of CDH require additional resources, add those numbers under Other Services.

The HDFS DataNode uses a minimum of 1 core and about 1 GB of memory. The same requirements apply to the YARN NodeManager.

The spreadsheet lists three optional services. For Impala, allocate at least 16 GB for the daemon. HBase RegionServer requires 12-16 GB of memory. Solr Server requires a minimum of 1 GB of memory.

Any remaining resources are available for YARN applications (Spark and MapReduce). In this example, 44 CPU cores are available. Set the multiplier for vcores you want on each physical core to calculate the total available vcores.

Step 3: Cluster Size

Having defined the specifications for each host in your cluster, enter the number of worker hosts needed to support your business case. To see the benefits of parallel computing, set the number of hosts to a minimum of 10.

STEP 3: Cluster Size

Enter the number of nodes you have (or expect to have) in the cluster

	Quantity
Number of Worker Hosts in the cluster	10

YARN Configuration

On the YARN Configuration tab, you verify your available resources and set minimum and maximum limits for each container.

Steps 4 and 5: Verify Settings

Step 4 pulls forward the memory and vcore numbers from step 2. Step 5 shows the total memory and vcores for the cluster.

STEP 4: YARN Configuration on Cluster

These are the first set of configuration values for your cluster. You can set these values in YARN->Configuration in Cloudera Manager.

YARN Configuration Property	Value	
yarn.nodemanager.resource.cpu-vcores	176	Copied from STEP 2 "Available Resources"
yarn.nodemanager.resource.memory-mb	250880	Copied from STEP 2 "Available Resources"

STEP 5: Verify YARN Settings on Cluster

Go to the Resource Manager Web UI (usually <http://<ResourceManagerIP>:8088/> and verify the "Memory Total" and "Vcores Total" matches the values above. If your machine has no bad nodes, then the numbers should match exactly.

Resource Manager Property to Check	Value	Note
Expected Value for "Vcores Total"	1760	Calculated from STEP 2 "YARN Available Vcores" and STEP 3
Expected Value for "Memory Total" (in GB)	2450	Calculated from STEP 2 "YARN Available Memory" and STEP 3

Step 6: Verify Container Settings on Cluster

In step 6, you can change the four values that impact the size of your containers.

The minimum number of vcores should be 1. When additional vcores are required, adding 1 at a time should result in the most efficient allocation. Set the maximum number of vcore reservations for a container to ensure that no single task consumes all available resources.

Set the minimum and maximum reservations for memory. The increment should be the smallest amount that can impact performance. Here, the minimum is approximately 1 GB, the maximum is approximately 8 GB, and the increment is 512 MB.

STEP 6: Verify Container Settings on Cluster

In order to have YARN jobs run cleanly, you need to configure the container properties.

YARN Container Configuration Property (Vcores)	Value	Description
yarn.scheduler.minimum-allocation-vcores	1	Minimum vcore reservation for a container
yarn.scheduler.maximum-allocation-vcores	32	Maximum vcore reservation for a container
yarn.scheduler.increment-allocation-vcores	1	Vcore allocations must be a multiple of this value
YARN Container Configuration Property (Memory)	Value	
yarn.scheduler.minimum-allocation-mb	1024	Minimum memory reservation for a container
yarn.scheduler.maximum-allocation-mb	8192	Maximum memory reservation for a container
yarn.scheduler.increment-allocation-mb	512	Memory allocations must be a multiple of this value

Step 6A: Cluster Container Capacity

Step 6A lets you validate the minimum and maximum number of containers in your cluster, based on the numbers you entered.

Step 6A: Cluster Container Capacity

This section will tell you the capacity of your cluster (in terms of containers).

Cluster Container Estimates	Value
Largest number of containers, based on memory configuration	2450
Smallest number of containers, based on memory configuration	306
Largest number of containers, based on vcore configuration	1760
Smallest number of containers, based on vcore configuration	55

Step 6B: Container Sanity Checking

Step 6B lets you see at a glance whether you have over-allocated resources.

STEP 6B: Container Sanity Checking

This section will do some basic checking of your container parameters in STEP 6 against the hosts.

Sanity Check	Check Status	Description
Vcore Max >= Vcore Min	GOOD	yarn.scheduler.maximum-allocation-vcores must be greater than or equal to yarn.scheduler.minimum-allocation-vcores
Memory Max >= Memory Min	GOOD	yarn.scheduler.maximum-allocation-mb must be greater than or equal to yarn.scheduler.minimum-allocation-mb
VCoreMin <= HostsVCores	GOOD	yarn.scheduler.minimum-allocation-vcores must be less than or equal to the yarn.nodemanager.resource.cpu-vcores

MapReduce Configuration

On the MapReduce Configuration tab, you can plan for increased task-specific memory capacity.

Step 7: MapReduce Configuration

You can increase the memory allocation for the ApplicationMaster, map tasks, and reduce tasks. The minimum vcore allocation for any task is always 1. The Spill/Sort memory allocation of 256 should be sufficient, and should be (rarely) increased if you determine that frequent spills to disk are hurting job performance.

STEP 7: MapReduce Configuration

Property	Property Type	Component	Value	Description
yarn.app.mapreduce.am.resource.cpu-vcores	Config	Application Master	1	AM container vcore reservation
yarn.app.mapreduce.am.resource.mb	Config	Application Master	1024	AM container memory reservation
mapreduce.map.cpu.vcores	Config	Map Task	1	Map task vcore reservation
mapreduce.map.memory.mb	Config	Map Task	1024	Map task memory reservation
mapreduce.reduce.cpu.vcores	Config	Reduce Task	1	Reduce task vcore reservation
mapreduce.reduce.memory.mb	Config	Reduce Task	1024	Reduce task memory reservation
mapreduce.task.io.sort.mb	Config	Spill/Sort (Map Task)	256	Spill/Sort memory reservation

Step 7A: MapReduce Sanity Checking

Step 7A lets you verify at a glance that all of your minimum and maximum resource allocations are within the parameters you set.

STEP 7A: MapReduce Sanity Checking

Sanity check MapReduce settings against container minimum/maximum properties.

Application Master Sanity Checks	Value	Description
yarn.app.mapreduce.am.resource.cpu-vcores >= container min	GOOD	Make sure ApplicationMaster vcore request fits within container limits
yarn.app.mapreduce.am.resource.cpu-vcores <= container max	GOOD	Ditto
yarn.app.mapreduce.am.resource.mb >= container min	GOOD	Make sure ApplicationMaster memory request fits within container limits
yarn.app.mapreduce.am.resource.mb <= container max	GOOD	Ditto
Map Task Sanity Checks	Value	Description
mapreduce.map.cpu.vcores >= container min	GOOD	Make sure Map Task vcore request fits within container limits
mapreduce.map.cpu.vcores <= container max	GOOD	Ditto
mapreduce.map.cpu.memory.mb >= container min	GOOD	Make sure Map Task memory request fits within container limits
mapreduce.map.cpu.memory.mb <= container max	GOOD	Ditto
Reduce Task Sanity Checks	Value	Description
mapreduce.reduce.cpu.vcores >= container min	GOOD	Make sure Reduce Task vcore request fits within container limits
mapreduce.reduce.cpu.vcores <= container max	GOOD	Ditto
mapreduce.reduce.cpu.memory.mb >= container min	GOOD	Make sure Reduce Task memory request fits within container limits
mapreduce.reduce.cpu.memory.mb <= container max	GOOD	Ditto

Configuring Your Cluster In Cloudera Manager

When you are satisfied with the cluster configuration estimates, use the values in the spreadsheet to set the corresponding properties in Cloudera Manager. For more information, see [Modifying Configuration Properties](#)

Table 23: Cloudera Manager Property Correspondence

Step	YARN/MapReduce Property	Cloudera Manager Equivalent
4	yarn.nodemanager.resource.cpu-vcores	Container Virtual CPU Cores
4	yarn.nodemanager.resource.memory-mb	Container Memory
6	yarn.scheduler.minimum-allocation-vcores	Container Virtual CPU Cores Minimum
6	yarn.scheduler.maximum-allocation-vcores	Container Virtual CPU Cores Maximum
6	yarn.scheduler.increment-allocation-vcores	Container Virtual CPU Cores Increment
6	yarn.scheduler.minimum-allocation-mb	Container Memory Minimum
6	yarn.scheduler.maximum-allocation-mb	Container Memory Maximum
6	yarn.scheduler.increment-allocation-mb	Container Memory Increment
7	yarn.app.mapreduce.am.resource.cpu-vcores	ApplicationMaster Virtual CPU Cores
7	yarn.app.mapreduce.am.resource.mb	ApplicationMaster Memory
7	mapreduce.map.cpu.vcores	Map Task CPU Virtual Cores
7	mapreduce.map.memory.mb	Map Task Memory
7	mapreduce.reduce.cpu.vcores	Reduce Task CPU Virtual Cores
7	mapreduce.reduce.memory.mb	Reduce Task Memory
7	mapreduce.task.io.sort.mb	I/O Sort Memory

Deploying CDH 5 on a Cluster



Note: Do the tasks in this section after installing the latest version of CDH; see [Installing the Latest CDH 5 Release](#) on page 155.

To deploy CDH 5 on a cluster, do the following:

1. [Configure Network Hosts](#)
2. [Configure HDFS](#)
3. Deploy [YARN with MapReduce v2 \(YARN\)](#) or [MapReduce v1 \(MRv1\)](#)

See also:

- [Configuring the Daemons to Start on Boot](#) on page 215
- [Optimizing Performance in CDH](#)
- [Configuring Centralized Cache Management in HDFS](#)
- [Managing HDFS Snapshots](#)
- [Configuring an NFSv3 Gateway](#)

Configuring Network Names



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x. If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

To ensure that the members of the cluster can communicate with each other, do the following on every system.

**Important:**

CDH requires IPv4. IPv6 is not supported.

1. Set the hostname of each system to a unique name (not `localhost`). For example:

```
$ sudo hostname myhost-1
```



Note: This is a temporary measure only. The hostname set by `hostname` does not survive across reboots.

2. Make sure the `/etc/hosts` file on each system contains the IP addresses and fully-qualified domain names (FQDN) of all the members of the cluster.

**Important:**

- The canonical name of each host in `/etc/hosts` **must** be the FQDN (for example `myhost-1.mynet.myco.com`), not the unqualified hostname (for example `myhost-1`). The canonical name is the first entry after the IP address.
- Do not use aliases, either in `/etc/hosts` or in configuring DNS.

If you are using DNS, storing this information in `/etc/hosts` is not required, but it is good practice.

3. Make sure the `/etc/sysconfig/network` file on each system contains the hostname you have just set (or verified) for that system, for example `myhost-1`.
4. Check that this system is consistently identified to the network:
 - a. Run `uname -a` and check that the hostname matches the output of the `hostname` command.
 - b. Run `/sbin/ifconfig` and note the value of `inet addr` in the `eth0` entry, for example:

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:A4:E8:97
          inet addr:172.29.82.176  Bcast:172.29.87.255  Mask:255.255.248.0
...

```

- c. Run `host -v -t A `hostname`` and make sure that `hostname` matches the output of the `hostname` command, and has the same IP address as reported by `ifconfig` for `eth0`; for example:

```
$ host -v -t A `hostname`
Trying "myhost.mynet.myco.com"
...
;; ANSWER SECTION:
myhost.mynet.myco.com. 60 IN A 172.29.82.176
```

5. For MRv1: make sure `conf/core-site.xml` and `conf/mapred-site.xml`, respectively, have the **hostnames** – not the IP addresses – of the NameNode and the JobTracker. These can be FQDNs (for example `myhost-1.mynet.myco.com`), or unqualified hostnames (for example `myhost-1`). See [Customizing Configuration Files](#) and [Deploying MapReduce v1 \(MRv1\) on a Cluster](#).
6. For YARN: make sure `conf/core-site.xml` and `conf/yarn-site.xml`, respectively, have the **hostnames** – not the IP addresses – of the NameNode, the ResourceManager, and the ResourceManager Scheduler. See [Customizing Configuration Files](#) and [Deploying MapReduce v2 \(YARN\) on a Cluster](#).
7. Make sure that components that depend on a client-server relationship – Oozie, HBase, ZooKeeper – are configured according to the instructions on their installation pages:
 - [Oozie Installation](#)
 - [HBase Installation](#)

- [ZooKeeper Installation](#)

Deploying HDFS on a Cluster

**Important:**

For instructions for configuring High Availability (HA) for the NameNode, see [HDFS High Availability](#). For instructions on using HDFS Access Control Lists (ACLs), see [Enabling HDFS Extended ACLs](#).

Proceed as follows to deploy HDFS on a cluster. Do this for all clusters, whether you are deploying MRv1 or YARN:

**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

1. [Copy the Hadoop configuration](#)
2. [Customize configuration files](#)
3. [Configure Local Storage Directories](#)
4. [Configure DataNodes to tolerate local storage directory failure](#)
5. [Format the NameNode](#)
6. [Configure a remote NameNode storage directory](#)
7. [Configure the Secondary NameNode \(if used\)](#)
8. [Optionally enable Trash](#)
9. [Optionally configure DataNode storage balancing](#)
10. [Optionally enable WebHDFS](#)
11. [Optionally configure LZ0](#)
12. [Start HDFS](#) on page 202
13. [Deploy MRv1 or YARN and start services](#)

**Note: Running Services**

When starting, stopping and restarting CDH components, always use the `service (8)` command rather than running scripts in `/etc/init.d` 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 scripts in `/etc/init.d`, 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).)

Copying the Hadoop Configuration and Setting Alternatives

To customize the Hadoop configuration:

1. Copy the default configuration to your custom directory:

```
$ sudo cp -r /etc/hadoop/conf.empty /etc/hadoop/conf.my_cluster
```

You can call this configuration anything you like; in this example, it's called `my_cluster`.

**Important:**

When performing the configuration tasks in this section, and when you go on to deploy MRv1 or YARN, edit the configuration files in this custom directory. Do not create your custom configuration in the default directory `/etc/hadoop/conf.empty`.

- CDH uses the `alternatives` setting to determine which Hadoop configuration to use. Set `alternatives` to point to your custom directory, as follows.

To manually set the configuration on Red Hat-compatible systems:

```
$ sudo alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.my_cluster
50
$ sudo alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

To manually set the configuration on Ubuntu and SLES systems:

```
$ sudo update-alternatives --install /etc/hadoop/conf hadoop-conf
/etc/hadoop/conf.my_cluster 50
$ sudo update-alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

This tells CDH to use the configuration in `/etc/hadoop/conf.my_cluster`.

You can display the current `alternatives` setting as follows.

To display the current setting on Red Hat-compatible systems:

```
sudo alternatives --display hadoop-conf
```

To display the current setting on Ubuntu, Debian, and SLES systems:

```
sudo update-alternatives --display hadoop-conf
```

You should see output such as the following:

```
hadoop-conf - status is auto.
link currently points to /etc/hadoop/conf.my_cluster
/etc/hadoop/conf.my_cluster - priority 50
/etc/hadoop/conf.empty - priority 10
Current 'best' version is /etc/hadoop/conf.my_cluster.
```

Because the configuration in `/etc/hadoop/conf.my_cluster` has the highest priority (50), that is the one CDH will use. For more information on `alternatives`, see the `update-alternatives(8)` man page on Ubuntu and SLES systems or the `alternatives(8)` man page On Red Hat-compatible systems.

Customizing Configuration Files

The following tables show the most important properties that you must configure for your cluster.

**Note:**

For information on other important configuration properties, and the configuration files, see the [Apache Cluster Setup](#) page.

Property	Configuration File	Description
<code>fs.defaultFS</code>	<code>core-site.xml</code>	Note: <code>fs.default.name</code> is deprecated. Specifies the NameNode and the default file system, in the form <code>hdfs://<namenode</code>

Property	Configuration File	Description
		<p><code>host>:<namenode port>/</code>. The default value is <code>file:///</code>. The default file system is used to resolve relative paths; for example, if <code>fs.default.name</code> or <code>fs.defaultFS</code> is set to <code>hdfs://mynamenode/</code>, the relative URI <code>/mydir/myfile</code> resolves to <code>hdfs://mynamenode/mydir/myfile</code>. Note: for the cluster to function correctly, the <code><namenode></code> part of the string must be the hostname (for example <code>mynamenode</code>), or the HA-enabled logical URI, not the IP address.</p>
<code>dfs.permissions.superusergroup</code>	<code>hdfs-site.xml</code>	Specifies the UNIX group containing users that will be treated as superusers by HDFS. You can stick with the value of 'hadoop' or pick your own group depending on the security policies at your site.

Sample Configuration

core-site.xml:

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://namenode-host.company.com:8020</value>
</property>
```

hdfs-site.xml:

```
<property>
  <name>dfs.permissions.superusergroup</name>
  <value>hadoop</value>
</property>
```

Configuring Local Storage Directories

You need to specify, create, and assign the correct permissions to the local directories where you want the HDFS daemons to store data. You specify the directories by configuring the following two properties in the `hdfs-site.xml` file.

Property	Configuration File Location	Description
<code>dfs.name.dir</code> or <code>dfs.namenode.name.dir</code>	<code>hdfs-site.xml</code> on the NameNode	This property specifies the URIs of the directories where the NameNode stores its metadata and edit logs. Cloudera recommends that you specify at least two directories. One of these should be located on an NFS mount point, unless you will be using a HDFS HA configuration .

Property	Configuration File Location	Description
dfs.data.dir Or dfs.datanode.data.dir	hdfs-site.xml on each DataNode	This property specifies the URIs of the directories where the DataNode stores blocks. Cloudera recommends that you configure the disks on the DataNode in a JBOD configuration, mounted at /data/1/ through /data/N, and configure dfs.data.dir or dfs.datanode.data.dir to specify file:///data/1/dfs/dn through file:///data/N/dfs/dn/.

**Note:**

dfs.data.dir and dfs.name.dir are deprecated; you should use dfs.datanode.data.dir and dfs.namenode.name.dir instead, though dfs.data.dir and dfs.name.dir will still work.

Sample configuration:

hdfs-site.xml on the NameNode:

```
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///data/1/dfs/nn,file:///nfsmount/dfs/nn</value>
</property>
```

hdfs-site.xml on each DataNode:

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///data/1/dfs/dn,file:///data/2/dfs/dn,file:///data/3/dfs/dn,file:///data/4/dfs/dn</value>
</property>
```

After specifying these directories as shown above, you must create the directories and assign the correct file permissions to them on each node in your cluster.

In the following instructions, local path examples are used to represent Hadoop parameters. Change the path examples to match your configuration.

Local directories:

- The dfs.name.dir or dfs.namenode.name.dir parameter is represented by the /data/1/dfs/nn and /nfsmount/dfs/nn path examples.
- The dfs.data.dir or dfs.datanode.data.dir parameter is represented by the /data/1/dfs/dn, /data/2/dfs/dn, /data/3/dfs/dn, and /data/4/dfs/dn examples.

To configure local storage directories for use by HDFS:

1. On a NameNode host: create the dfs.name.dir or dfs.namenode.name.dir local directories:

```
$ sudo mkdir -p /data/1/dfs/nn /nfsmount/dfs/nn
```

**Important:**

If you are using [High Availability \(HA\)](#), you should **not** configure these directories on an NFS mount; configure them on local storage.

2. On all DataNode hosts: create the `dfs.data.dir` or `dfs.datanode.data.dir` local directories:

```
$ sudo mkdir -p /data/1/dfs/dn /data/2/dfs/dn /data/3/dfs/dn /data/4/dfs/dn
```

3. Configure the owner of the `dfs.name.dir` or `dfs.namenode.name.dir` directory, and of the `dfs.data.dir` or `dfs.datanode.data.dir` directory, to be the `hdfs` user:

```
$ sudo chown -R hdfs:hdfs /data/1/dfs/nn /nfsmount/dfs/nn /data/1/dfs/dn /data/2/dfs/dn /data/3/dfs/dn /data/4/dfs/dn
```



Note:

For a list of the users created when you install CDH, see [Hadoop Users in Cloudera Manager and CDH](#).

Here is a summary of the correct owner and permissions of the local directories:

Directory	Owner	Permissions (see Footnote 1)
<code>dfs.name.dir</code> OR <code>dfs.namenode.name.dir</code>	<code>hdfs:hdfs</code>	<code>drwx-----</code>
<code>dfs.data.dir</code> OR <code>dfs.datanode.data.dir</code>	<code>hdfs:hdfs</code>	<code>drwx-----</code>

Footnote: 1 The Hadoop daemons automatically set the correct permissions for you on `dfs.data.dir` or `dfs.datanode.data.dir`. But in the case of `dfs.name.dir` or `dfs.namenode.name.dir`, permissions are currently incorrectly set to the file-system default, usually `drwxr-xr-x` (755). Use the `chmod` command to reset permissions for these `dfs.name.dir` or `dfs.namenode.name.dir` directories to `drwx-----` (700); for example:

```
$ sudo chmod 700 /data/1/dfs/nn /nfsmount/dfs/nn
```

or

```
$ sudo chmod go-rx /data/1/dfs/nn /nfsmount/dfs/nn
```



Note:

If you specified nonexistent directories for the `dfs.data.dir` or `dfs.datanode.data.dir` property in the `hdfs-site.xml` file, CDH 5 will shut down. (In previous releases, CDH silently ignored nonexistent directories for `dfs.data.dir`.)

Configuring DataNodes to Tolerate Local Storage Directory Failure

By default, the failure of a single `dfs.data.dir` or `dfs.datanode.data.dir` will cause the HDFS DataNode process to shut down, which results in the NameNode scheduling additional replicas for each block that is present on the DataNode. This causes needless replications of blocks that reside on disks that have not failed.

To prevent this, you can configure DataNodes to tolerate the failure of `dfs.data.dir` or `dfs.datanode.data.dir` directories; use the `dfs.datanode.failed.volumes.tolerated` parameter in `hdfs-site.xml`. For example, if the value for this parameter is 3, the DataNode will only shut down after four or more data directories have failed. This value is respected on DataNode startup; in this example the DataNode will start up as long as no more than three directories have failed.

**Note:**

It is important that `dfs.datanode.failed.volumes.tolerated` not be configured to tolerate too many directory failures, as the DataNode will perform poorly if it has few functioning data directories.

Formatting the NameNode

Before starting the NameNode for the first time you need to format the file system.

**Important:**

- Make sure you format the NameNode as user `hdfs`.
- If you are re-formatting the NameNode, keep in mind that this invalidates the DataNode storage locations, so you should remove the data under those locations after the NameNode is formatted.

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

**Note:**

If [Kerberos is enabled](#), do not use commands in the form `sudo -u <user> hadoop <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>`

You'll get a confirmation prompt; for example:

```
Re-format filesystem in /data/namedir ? (Y or N)
```



Note: Respond with an **upper-case Y**; if you use lower case, the process will abort.

Configuring a Remote NameNode Storage Directory

You should configure the NameNode to write to multiple storage directories, including one remote NFS mount. To keep NameNode processes from hanging when the NFS server is unavailable, configure the NFS mount as a `soft` mount (so that I/O requests that time out fail rather than hang), and set other options as follows:

```
tcp,soft,intr,timeo=10,retrans=10
```

These options configure a soft mount over TCP; transactions will be retried ten times (`retrans=10`) at 1-second intervals (`timeo=10`) before being deemed to have failed.

Example:

```
mount -t nfs -o tcp,soft,intr,timeo=10,retrans=10, <server>:<export> <mount_point>
```

where `<server>` is the remote host, `<export>` is the exported file system, and `<mount_point>` is the local mount point.

**Note:**

Cloudera recommends similar settings for shared HA mounts, as in the example that follows.

Example for HA:

```
mount -t nfs -o tcp,soft,intr,timeo=50,retrans=12, <server>:<export> <mount_point>
```

Note that in the HA case `timeo` should be set to 50 (five seconds), rather than 10 (1 second), and `retrans` should be set to 12, giving an overall timeout of 60 seconds.

For more information, see the man pages for `mount` and `nfs`.

Configuring Remote Directory Recovery

You can enable the `dfs.namenode.name.dir.restore` option so that the NameNode will attempt to recover a previously failed NameNode storage directory on the next checkpoint. This is useful for restoring a remote storage directory mount that has failed because of a network outage or intermittent NFS failure.

Configuring the Secondary NameNode



Important:

The Secondary NameNode does not provide failover or High Availability (HA). If you intend to configure [HA for the NameNode](#), skip this section: do not install or configure the Secondary NameNode (the Standby NameNode performs checkpointing). After completing the [HA software configuration](#), follow the installation instructions under [Deploying HDFS High Availability](#).

In non-HA deployments, configure a Secondary NameNode that will periodically merge the EditLog with the FSImage, creating a new FSImage which incorporates the changes which were in the EditLog. This reduces the amount of disk space consumed by the EditLog on the NameNode, and also reduces the restart time for the Primary NameNode.

A standard Hadoop cluster (not a Hadoop Federation or HA configuration), can have only one Primary NameNode plus one Secondary NameNode. On production systems, the Secondary NameNode should run on a different machine from the Primary NameNode to improve scalability (because the Secondary NameNode does not compete with the NameNode for memory and other resources to create the system snapshot) and durability (because the copy of the metadata is on a separate machine that is available if the NameNode hardware fails).

Configuring the Secondary NameNode on a Separate Machine

To configure the Secondary NameNode on a separate machine from the NameNode, proceed as follows.

1. Add the name of the machine that will run the Secondary NameNode to the `masters` file.
2. Add the following property to the `hdfs-site.xml` file:

```
<property>
  <name>dfs.namenode.http-address</name>
  <value><namenode.host.address>:50070</value>
  <description>
    The address and the base port on which the dfs NameNode Web UI will listen.
  </description>
</property>
```

**Note:**

- `dfs.http.address` is deprecated; use `dfs.namenode.http-address`.
- In most cases, you should set `dfs.namenode.http-address` to a routable IP address with port 50070. However, in some cases such as Amazon EC2, when the NameNode should bind to multiple local addresses, you may want to set `dfs.namenode.http-address` to `0.0.0.0:50070` *on the NameNode machine only*, and set it to a real, routable address on the Secondary NameNode machine. The different addresses are needed in this case because HDFS uses `dfs.namenode.http-address` for two different purposes: it defines both the address the NameNode binds to, and the address the Secondary NameNode connects to for checkpointing. Using `0.0.0.0` on the NameNode allows the NameNode to bind to all its local addresses, while using the externally-routable address on the Secondary NameNode provides the Secondary NameNode with a real address to connect to.

For more information, see [Multi-host SecondaryNameNode Configuration](#).

More about the Secondary NameNode

- The NameNode stores the HDFS metadata information in RAM to speed up interactive lookups and modifications of the metadata.
- For reliability, this information is flushed to disk periodically. To ensure that these writes are not a speed bottleneck, only the list of modifications is written to disk, not a full snapshot of the current filesystem. The list of modifications is appended to a file called `edits`.
- Over time, the `edits` log file can grow quite large and consume large amounts of disk space.
- When the NameNode is restarted, it takes the HDFS system state from the `fsimage` file, then applies the contents of the `edits` log to construct an accurate system state that can be loaded into the NameNode's RAM. If you restart a large cluster that has run for a long period with no Secondary NameNode, the `edits` log may be quite large, and so it can take some time to reconstruct the system state to be loaded into RAM.

When the Secondary NameNode is configured, it periodically constructs a checkpoint by compacting the information in the `edits` log and merging it with the most recent `fsimage` file; it then clears the `edits` log. So, when the NameNode restarts, it can use the latest checkpoint and apply the contents of the smaller `edits` log. The interval between checkpoints is determined by the checkpoint period (`dfs.namenode.checkpoint.period`) or the number of edit transactions (`dfs.namenode.checkpoint.txns`). The default checkpoint period is one hour, and the default number of edit transactions before a checkpoint is 1,000,000. The SecondaryNameNode will checkpoint in an hour if there have not been 1,000,000 edit transactions within the hour; it will checkpoint after 1,000,000 transactions have been committed if they were committed in under one hour.

Secondary NameNode Parameters

The behavior of the Secondary NameNode is controlled by the following parameters in `hdfs-site.xml`.

- `dfs.namenode.checkpoint.check.period`
- `dfs.namenode.checkpoint.txns`
- `dfs.namenode.checkpoint.dir`
- `dfs.namenode.checkpoint.edits.dir`
- `dfs.namenode.num.checkpoints.retained`

See <https://archive.cloudera.com/cdh5/cdh/5/hadoop/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml> for details.

Enabling Trash

**Important:**

The trash feature is disabled by default. Cloudera recommends that you enable it on all production clusters.

The Hadoop trash feature helps prevent accidental deletion of files and directories. If trash is enabled and a file or directory is deleted using the Hadoop shell, the file is moved to the `.Trash` directory in the user's home directory instead of being deleted. Deleted files are initially moved to the `Current` sub-directory of the `.Trash` directory, and their original path is preserved. If trash checkpointing is enabled, the `Current` directory is periodically renamed using a timestamp. Files in `.Trash` are permanently removed after a user-configurable time delay. Files and directories in the trash can be restored simply by moving them to a location outside the `.Trash` directory.



Note:

The trash feature works by default only for files and directories deleted using the Hadoop shell. Files or directories deleted programmatically using other interfaces (WebHDFS or the Java APIs, for example) are not moved to trash, even if trash is enabled, unless the program has implemented a call to the trash functionality. (Hue, for example, implements trash as of CDH 4.4.)

Users can bypass trash when deleting files using the shell by specifying the `-skipTrash` option to the `hadoop fs -rm -r` command. This can be useful when it is necessary to delete files that are too large for the user's quota.

Trash is configured with the following properties in the `core-site.xml` file:

CDH Parameter	Value	Description
<code>fs.trash.interval</code>	<i>minutes or 0</i>	The number of minutes after which a trash checkpoint directory is deleted. This option can be configured both on the server and the client. <ul style="list-style-type: none"> • If trash is enabled on the server configuration, then the value configured on the server is used and the client configuration is ignored. • If trash is disabled in the server configuration, then the client side configuration is checked. • If the value of this property is zero (the default), then the trash feature is disabled.
<code>fs.trash.checkpoint.interval</code>	<i>minutes or 0</i>	The number of minutes between trash checkpoints. Every time the checkpointer runs on the NameNode, it creates a new checkpoint of the "Current" directory and removes checkpoints older than <code>fs.trash.interval</code> minutes. This value should be smaller than or equal to <code>fs.trash.interval</code> . This option is configured on the server. If configured to zero (the default), then the value is set to the value of <code>fs.trash.interval</code> .

For example, to enable trash so that files deleted using the Hadoop shell are not deleted for 24 hours, set the value of the `fs.trash.interval` property in the server's `core-site.xml` file to a value of 1440.



Note:

The period during which a file remains in the trash starts when the file is moved to the trash, not when the file is last modified.

Configuring Storage-Balancing for the DataNodes

You can configure HDFS to distribute writes on each DataNode in a manner that balances out available storage among that DataNode's disk volumes.

By default a DataNode writes new block replicas to disk volumes solely on a round-robin basis. You can configure a volume-choosing policy that causes the DataNode to take into account how much space is available on each volume when deciding where to place a new replica.

You can configure

- how much DataNode volumes are allowed to differ in terms of bytes of free disk space before they are considered imbalanced, *and*
- what percentage of new block allocations will be sent to volumes with more available disk space than others.

To configure storage balancing, set the following properties in `hdfs-site.xml`.



Note: Keep in mind that if usage is markedly imbalanced among a given DataNode's storage volumes when you enable storage balancing, throughput on that DataNode will be affected initially, as writes are disproportionately directed to the under-utilized volumes.

Property	Value	Description
<code>dfs.datanode.fsdataset.volume.choosing.policy</code>	<code>org.apache.hadoop.hdfs.server.datanode.fsdataset.AvailableSpaceVolumeChoosingPolicy</code>	Enables storage balancing among the DataNode's volumes.
<code>dfs.datanode.available-space-volume-choosing-policy.balanced-space-threshold</code>	10737418240 (default)	The amount by which volumes are allowed to differ from each other in terms of bytes of free disk space before they are considered imbalanced. The default is 10737418240 (10 GB). If the free space on each volume is within this range of the other volumes, the volumes will be considered balanced and block assignments will be done on a pure round-robin basis.
<code>dfs.datanode.available-space-volume-choosing-policy.balanced-space-preference-fraction</code>	0.75 (default)	What proportion of new block allocations will be sent to volumes with more available disk space than others. The allowable range is 0.0-1.0, but set it in the range 0.5 - 1.0 (that is, 50-100%), since there should be no reason to prefer that volumes with less available disk space receive more block allocations.

Enabling WebHDFS



Note: To configure HttpFs instead, see [HttpFS Installation](#) on page 298.

If you want to use WebHDFS, you must first enable it.

To enable WebHDFS:

Set the following property in `hdfs-site.xml`:

```
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
```

To enable numeric usernames in WebHDFS:

By default, WebHDFS supports the following username pattern:

```
^[A-Za-z_][A-Za-z0-9._-]*[$]?$
```

You can override the default username pattern by setting the `dfs.webhdfs.user.provider.user.pattern` property in `hdfs-site.xml`. For example, to allow numerical usernames, the property can be set as follows:

```
<property>
  <name>dfs.webhdfs.user.provider.user.pattern</name>
  <value>^[A-Za-z0-9_][A-Za-z0-9._-]*[$]?$</value>
</property>
```



Important: The username pattern should be compliant with the requirements of the operating system in use. Hence, Cloudera recommends you use the default pattern and avoid modifying the `dfs.webhdfs.user.provider.user.pattern` property when possible.



Note:

- To use WebHDFS in a secure cluster, you must set additional properties to configure secure WebHDFS. For instructions, see the [Cloudera Security](#) guide.
- When you use WebHDFS in a [high-availability \(HA\)](#) configuration, you must supply the value of `dfs.nameservices` in the WebHDFS URI, rather than the address of a particular NameNode; for example:

```
hdfs dfs -ls webhdfs://nameservice1/, not
```

```
hdfs dfs -ls webhdfs://server1.myent.myco.com:20101/
```

Configuring LZO

If you have [installed LZO](#), configure it as follows.

To configure LZO:

Set the following property in `core-site.xml`.



Note:

If you copy and paste the *value* string, make sure you remove the line-breaks and carriage returns, which are included below because of page-width constraints.

```
<property>
  <name>io.compression.codecs</name>
  <value>org.apache.hadoop.io.compress.DefaultCodec,org.apache.hadoop.io.compress.GzipCodec,
  org.apache.hadoop.io.compress.BZip2Codec,com.hadoop.compression.lzo.LzoCodec,
  com.hadoop.compression.lzo.LzopCodec,org.apache.hadoop.io.compress.SnappyCodec</value>
</property>
```

For more information about LZO, see [Using LZO Compression](#).

Start HDFS

To deploy HDFS now, proceed as follows.

1. [Deploy the configuration](#).
2. [Start HDFS](#).
3. [Create the /tmp directory](#).

Deploy the configuration

To deploy your configuration to your entire cluster:

1. Push your custom directory (for example `/etc/hadoop/conf.my_cluster`) to each node in your cluster; for example:

```
$ scp -r /etc/hadoop/conf.my_cluster
myuser@myCDHnode-<n>.mycompany.com: /etc/hadoop/conf.my_cluster
```

2. Manually set alternatives on each node to point to that directory, as follows.

To manually set the configuration on Red Hat-compatible systems:

```
$ sudo alternatives --verbose --install /etc/hadoop/conf hadoop-conf
/etc/hadoop/conf.my_cluster 50
$ sudo alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

To manually set the configuration on Ubuntu and SLES systems:

```
$ sudo update-alternatives --install /etc/hadoop/conf hadoop-conf
/etc/hadoop/conf.my_cluster 50
$ sudo update-alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

For more information on alternatives, see the `update-alternatives(8)` man page on Ubuntu and SLES systems or the `alternatives(8)` man page on Red Hat-compatible systems.

Start HDFS

Start HDFS on each node in the cluster, as follows:

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

**Note:**

This starts all the CDH services installed on the node. This is normally what you want, but you can start services individually if you prefer.

Create the `/tmp` directory

**Important:**

If you do not create `/tmp` properly, with the right permissions as shown below, you may have problems with CDH components later. Specifically, if you do not create `/tmp` yourself, another process may create it automatically with restrictive permissions that will prevent your other applications from using it.

Create the `/tmp` directory after HDFS is up and running, and set its permissions to 1777 (`drwxrwxrwt`), as follows:

```
$ sudo -u hdfs hadoop fs -mkdir /tmp
$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp
```

**Note:**

If [Kerberos is enabled](#), do not use commands in the form `sudo -u <user> hadoop <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>`

Deploy YARN or MRv1

To to deploy MRv1 or YARN, and start HDFS services if you have not [already done so](#), see

- [Deploying MapReduce v2 \(YARN\) on a Cluster](#) on page 204 or
- [Deploying MapReduce v1 \(MRv1\) on a Cluster](#) on page 210

Deploying MapReduce v2 (YARN) on a Cluster



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

This section describes configuration tasks for YARN clusters only, and is specifically tailored for administrators who have [installed YARN from packages](#).



Important:

Do the following tasks after you have [configured and deployed HDFS](#):

1. [Configure properties for YARN clusters](#)
2. [Configure YARN daemons](#)
3. [Configure the History Server](#)
4. [Configure the Staging Directory](#)
5. [Deploy your custom configuration to your entire cluster](#)
6. [Start HDFS](#)
7. [Create the HDFS /tmp directory](#)
8. [Create the History Directory and Set Permissions](#)
9. [Create Log Directories](#)
10. [Verify the HDFS File Structure](#)
11. [Start YARN and the MapReduce JobHistory Server](#)
12. [Create a home directory for each MapReduce user](#)
13. [Configure the Hadoop daemons to start at boot time](#)



Important: Running Services

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

About MapReduce v2 (YARN)

The default installation in CDH 5 is MapReduce 2.x (MRv2) built on the YARN framework. In this document we usually refer to this new version as **YARN**. 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 slave 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 [Selecting Appropriate JAR files for your Jobs](#) on page 172.



Important:

Make sure you are not trying to run MRv1 and YARN on the same set of nodes at the same time. This is not recommended, especially in a cluster that is not managed by Cloudera Manager; it will degrade performance and may result in an unstable cluster deployment.

- If you have [installed YARN from packages](#), follow the instructions below to deploy it. (To deploy MRv1 instead, see [Deploying MapReduce v1 \(MRv1\) on a Cluster](#).)
- If you have installed CDH 5 from tarballs, the default deployment is YARN. Keep in mind that the instructions on this page are tailored for a deployment following installation from packages.

Step 1: Configure Properties for YARN Clusters



Note:

Edit these files in the custom directory you created when you [copied the Hadoop configuration](#). When you have finished, you will push this configuration to all the nodes in the cluster; see [Step 5](#).

Property	Configuration File	Description
mapreduce.framework.name	mapred-site.xml	If you plan on running YARN, you must set this property to the value of <code>yarn</code> .

Sample Configuration:

mapred-site.xml:

```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

Step 2: Configure YARN daemons

Configure the following services: ResourceManager (on a dedicated host) and NodeManager (on every host where you plan to run MapReduce v2 jobs).

The following table shows the most important properties that you must configure for your cluster in `yarn-site.xml`

Property	Recommended value	Description
yarn.nodemanager.aux-services	mapreduce_shuffle	Shuffle service that needs to be set for Map Reduce applications.
yarn.resourcemanager.hostname	resourcemanager.company.com	The following properties will be set to their default ports on this host: yarn.resourcemanager.address, yarn.resourcemanager.admin.address, yarn.resourcemanager.scheduler.address, yarn.resourcemanager.resource-tracker.address, yarn.resourcemanager.webapp.address

Property	Recommended value	Description
yarn.application.classpath	\$HADOOP_CONF_DIR, \$HADOOP_COMMON_HOME/*, \$HADOOP_COMMON_HOME/lib/*, \$HADOOP_HDFS_HOME/*, \$HADOOP_HDFS_HOME/lib/*, \$HADOOP_MAPRED_HOME/*, \$HADOOP_MAPRED_HOME/lib/*, \$HADOOP_YARN_HOME/*, \$HADOOP_YARN_HOME/lib/*	Classpath for typical applications.
yarn.log.aggregation-enable	true	

Next, you need to specify, create, and assign the correct permissions to the local directories where you want the YARN daemons to store data.

You specify the directories by configuring the following two properties in the `yarn-site.xml` file on all cluster nodes:

Property	Description
yarn.nodemanager.local-dirs	Specifies the URIs of the directories where the NodeManager stores its localized files. All of the files required for running a particular YARN application will be put here for the duration of the application run. Cloudera recommends that this property specify a directory on each of the JBOD mount points; for example, <code>file:///data/1/yarn/local</code> through <code>/data/N/yarn/local</code> .
yarn.nodemanager.log-dirs	Specifies the URIs of the directories where the NodeManager stores container log files. Cloudera recommends that this property specify a directory on each of the JBOD mount points; for example, <code>file:///data/1/yarn/logs</code> through <code>file:///data/N/yarn/logs</code> .
yarn.nodemanager.remote-app-log-dir	Specifies the URI of the directory where logs are aggregated. Set the value to <i>either</i> <code>hdfs://namenode-host.company.com:8020/var/log/hadoop-yarn/apps</code> , using the fully-qualified domain name of your NameNode host, <i>or</i> <code>hdfs://var/log/hadoop-yarn/apps</code> . See also Step 9 .

Here is an example configuration:

yarn-site.xml:

```
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>resourcemanager.company.com</value>
</property>
<property>
  <description>Classpath for typical applications.</description>
  <name>yarn.application.classpath</name>
  <value>
    $HADOOP_CONF_DIR,
    $HADOOP_COMMON_HOME/*,$HADOOP_COMMON_HOME/lib/*,
    $HADOOP_HDFS_HOME/*,$HADOOP_HDFS_HOME/lib/*,
```

```

        $HADOOP_MAPRED_HOME/*,$HADOOP_MAPRED_HOME/lib/*,
        $HADOOP_YARN_HOME/*,$HADOOP_YARN_HOME/lib/*
    </value>
</property>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.local-dirs</name>
<value>file:///data/1/yarn/local,file:///data/2/yarn/local,file:///data/3/yarn/local</value>
</property>
<property>
  <name>yarn.nodemanager.log-dirs</name>
<value>file:///data/1/yarn/logs,file:///data/2/yarn/logs,file:///data/3/yarn/logs</value>
</property>
<property>
  <name>yarn.log.aggregation-enable</name>
  <value>>true</value>
</property>
<property>
  <description>Where to aggregate logs</description>
  <name>yarn.nodemanager.remote-app-log-dir</name>
  <value>hdfs://<namenode-host.company.com>:8020/var/log/hadoop-yarn/apps</value>
</property>

```

After specifying these directories in the `yarn-site.xml` file, you must create the directories and assign the correct file permissions to them on each node in your cluster.

In the following instructions, local path examples are used to represent Hadoop parameters. Change the path examples to match your configuration.

To configure local storage directories for use by YARN:

1. Create the `yarn.nodemanager.local-dirs` local directories:

```
$ sudo mkdir -p /data/1/yarn/local /data/2/yarn/local /data/3/yarn/local
/data/4/yarn/local
```

2. Create the `yarn.nodemanager.log-dirs` local directories:

```
$ sudo mkdir -p /data/1/yarn/logs /data/2/yarn/logs /data/3/yarn/logs /data/4/yarn/logs
```

3. Configure the owner of the `yarn.nodemanager.local-dirs` directory to be the `yarn` user:

```
$ sudo chown -R yarn:yarn /data/1/yarn/local /data/2/yarn/local /data/3/yarn/local
/data/4/yarn/local
```

4. Configure the owner of the `yarn.nodemanager.log-dirs` directory to be the `yarn` user:

```
$ sudo chown -R yarn:yarn /data/1/yarn/logs /data/2/yarn/logs /data/3/yarn/logs
/data/4/yarn/logs
```

Here is a summary of the correct owner and permissions of the local directories:

Directory	Owner	Permissions
<code>yarn.nodemanager.local-dirs</code>	<code>yarn:yarn</code>	<code>drwxr-xr-x</code>
<code>yarn.nodemanager.log-dirs</code>	<code>yarn:yarn</code>	<code>drwxr-xr-x</code>

Step 3: Configure the History Server

If you have decided to run YARN on your cluster instead of MRv1, you should also run the MapReduce JobHistory Server. The following table shows the most important properties that you must configure in `mapred-site.xml`.

Property	Recommended value	Description
<code>mapreduce.jobhistory.address</code>	<code>historyserver.company.com:10020</code>	The address of the JobHistory Server <code>host:port</code>
<code>mapreduce.jobhistory.webapp.address</code>	<code>historyserver.company.com:19888</code>	The address of the JobHistory Server web application <code>host:port</code>

In addition, make sure proxying is enabled for the `mapred` user; configure the following properties in `core-site.xml`:

Property	Recommended value	Description
<code>hadoop.proxyuser.mapred.groups</code>	*	Allows the <code>mapred</code> user to move files belonging to users in these groups
<code>hadoop.proxyuser.mapred.hosts</code>	*	Allows the <code>mapred</code> user to move files belonging on these hosts

Step 4: Configure the Staging Directory

YARN requires a staging directory for temporary files created by running jobs. By default it creates `/tmp/hadoop-yarn/staging` with restrictive permissions that may prevent your users from running jobs. To forestall this, you should configure and create the staging directory yourself; in the example that follows we use `/user`:

1. Configure `yarn.app.mapreduce.am.staging-dir` in `mapred-site.xml`:

```
<property>
  <name>yarn.app.mapreduce.am.staging-dir</name>
  <value>/user</value>
</property>
```

2. Once HDFS is up and running, you will create this directory and a `history` subdirectory under it (see [Step 8](#)).

Alternatively, you can do the following:

1. Configure `mapreduce.jobhistory.intermediate-done-dir` and `mapreduce.jobhistory.done-dir` in `mapred-site.xml`.
2. Create these two directories.
3. Set permissions on `mapreduce.jobhistory.intermediate-done-dir` to `1777`.
4. Set permissions on `mapreduce.jobhistory.done-dir` to `750`.

If you configure `mapreduce.jobhistory.intermediate-done-dir` and `mapreduce.jobhistory.done-dir` as above, you can skip [Step 8](#).

Step 5: If Necessary, Deploy your Custom Configuration to your Entire Cluster

[Deploy the configuration](#) on page 203 if you have not already done so.

Step 6: If Necessary, Start HDFS on Every Node in the Cluster

[Start HDFS](#) on page 202 if you have not already done so.

Step 7: If Necessary, Create the HDFS `/tmp` Directory

[Create the `/tmp` directory](#) on page 203 if you have not already done so.

**Important:**

If you do not create `/tmp` properly, with the right permissions as shown below, you may have problems with CDH components later. Specifically, if you do not create `/tmp` yourself, another process may create it automatically with restrictive permissions that will prevent your other applications from using it.

Step 8: Create the history Directory and Set Permissions and Owner

This is a subdirectory of the staging directory you configured in [Step 4](#). In this example we're using `/user/history`. Create it and set permissions as follows:

```
sudo -u hdfs hadoop fs -mkdir -p /user/history
sudo -u hdfs hadoop fs -chmod -R 1777 /user/history
sudo -u hdfs hadoop fs -chown mapred:hadoop /user/history
```

Step 9: Create Log Directories**Note:**

See also [Step 2](#).

Create the `/var/log/hadoop-yarn` directory and set ownership:

```
sudo -u hdfs hadoop fs -mkdir -p /var/log/hadoop-yarn
sudo -u hdfs hadoop fs -chown yarn:mapred /var/log/hadoop-yarn
```

**Note:**

You need to create this directory because it is the parent of `/var/log/hadoop-yarn/apps` which is explicitly configured in `yarn-site.xml`.

Step 10: Verify the HDFS File Structure:

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

You should see:

```
drwxrwxrwt - hdfs supergroup          0 2012-04-19 14:31 /tmp
drwxr-xr-x - hdfs supergroup          0 2012-05-31 10:26 /user
drwxrwxrwt - mapred hadoop            0 2012-04-19 14:31 /user/history
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 11: Start YARN and the MapReduce JobHistory Server

To start YARN, start the Resource Manager and Node Manager services:

**Note:**

Make sure you always start Resource Manager before starting Node Manager services.

On the Resource Manager system:

```
$ sudo service hadoop-yarn-resourcemanager start
```

On each NodeManager system (typically the same ones where DataNode service runs):

```
$ sudo service hadoop-yarn-nodemanager start
```

To start the MapReduce JobHistory Server

On the MapReduce JobHistory Server system:

```
$ sudo service hadoop-mapreduce-historyserver start
```

Step 12: Create a Home Directory for each MapReduce User

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
```

Step 13: Configure the Hadoop Daemons to Start at Boot Time

See [Configuring the Hadoop Daemons to Start at Boot Time](#).

Deploying MapReduce v1 (MRv1) on a Cluster



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

This section describes configuration and startup tasks for MRv1 clusters only.



Important: Make sure you are not trying to run MRv1 and YARN on the same set of nodes at the same time. This is not recommended; it will degrade performance and may result in an unstable cluster deployment.

- Follow the instructions on this page to deploy MapReduce v1 (MRv1).
- If you have [installed YARN](#) and want to deploy it instead of MRv1, follow [these instructions](#) instead of the ones below.
- If you have installed CDH 5 from tarballs, the default deployment is YARN.



Important: Do these tasks after you have [configured and deployed HDFS](#):

1. [Configure properties for MRv1 clusters](#)
2. [Configure local storage directories for use by MRv1 daemons](#)
3. [Configure a health check script for DataNode processes](#)
4. [Configure JobTracker Recovery](#)
5. [If necessary, deploy the configuration](#)
6. [If necessary, start HDFS](#)

7. [Create the HDFS /tmp directory](#)
8. [Create MapReduce /var directories](#)
9. [Verify the HDFS File Structure](#)
10. [Create and configure the `mapred.system.dir` directory in HDFS](#)
11. [Start MapReduce](#)
12. [Create a Home Directory for each MapReduce User](#)
13. [Set the `HADOOP_MAPRED_HOME` environment variable](#)
14. [Configure the Hadoop daemons to start at boot time](#)



Important: Running Services

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

Step 1: Configuring Properties for MRv1 Clusters



Note: Edit these files in the custom directory you created when you [copied the Hadoop configuration](#).



Note: For instructions on configuring a highly available JobTracker, see [MapReduce \(MRv1\) JobTracker High Availability](#); you need to configure `mapred.job.tracker` differently in that case, and you must not use the port number.

Property	Configuration File	Description
<code>mapred.job.tracker</code>	<code>conf/mapred-site.xml</code>	If you plan to run your cluster with MRv1 daemons you need to specify the hostname and (optionally) port of the JobTracker's RPC server, in the form <code><host>:<port></code> . See Ports Used by Components of CDH 5 on page 25 for the default port. If the value is set to <code>local</code> , the default, the JobTracker runs on demand when you run a MapReduce job; do not try to start the JobTracker yourself in this case. Note: if you specify the host (rather than using <code>local</code>) this must be the hostname (for example <code>mynamenode</code>) not the IP address.

Sample configuration:

mapred-site.xml:

```
<property>
  <name>mapred.job.tracker</name>
  <value>jobtracker-host.company.com:8021</value>
</property>
```

Step 2: Configure Local Storage Directories for Use by MRv1 Daemons

For MRv1, you need to configure an additional property in the `mapred-site.xml` file.

Property	Configuration File Location	Description
<code>mapred.local.dir</code>	<code>mapred-site.xml</code> on each TaskTracker	This property specifies the directories where the TaskTracker will store temporary data and intermediate map output files while running MapReduce jobs. Cloudera recommends that this property specifies a directory on each of the JBOD mount points; for example, <code>/data/1/mapred/local</code> through <code>/data/N/mapred/local</code> .

Sample configuration:

mapred-site.xml on each TaskTracker:

```
<property>
  <name>mapred.local.dir</name>
  <value>/data/1/mapred/local,/data/2/mapred/local,/data/3/mapred/local</value>
</property>
```

After specifying these directories in the `mapred-site.xml` file, you must create the directories and assign the correct file permissions to them on each node in your cluster.

To configure local storage directories for use by MapReduce:

In the following instructions, local path examples are used to represent Hadoop parameters. The `mapred.local.dir` parameter is represented by the `/data/1/mapred/local`, `/data/2/mapred/local`, `/data/3/mapred/local`, and `/data/4/mapred/local` path examples. Change the path examples to match your configuration.

1. Create the `mapred.local.dir` local directories:

```
$ sudo mkdir -p /data/1/mapred/local /data/2/mapred/local /data/3/mapred/local
/data/4/mapred/local
```

2. Configure the owner of the `mapred.local.dir` directory to be the `mapred` user:

```
$ sudo chown -R mapred:hadoop /data/1/mapred/local /data/2/mapred/local
/data/3/mapred/local /data/4/mapred/local
```

The correct owner and permissions of these local directories are:

Owner	Permissions
<code>mapred:hadoop</code>	<code>drwxr-xr-x</code>

Step 3: Configure a Health Check Script for DataNode Processes

In CDH releases before CDH 4, the failure of a single `mapred.local.dir` caused the MapReduce TaskTracker process to shut down, resulting in the machine not being available to execute tasks. In CDH 5, as in CDH 4, the TaskTracker process will continue to execute tasks as long as it has a single functioning `mapred.local.dir` available. No configuration change is necessary to enable this behavior.

Because a TaskTracker that has few functioning local directories will not perform well, Cloudera recommends configuring a health script that checks if the DataNode process is running (if configured as described under [Configuring DataNodes](#)

to [Tolerate Local Storage Directory Failure](#), the DataNode will shut down after the configured number of directory failures). Here is an example health script that exits if the DataNode process is not running:

```
#!/bin/bash
if ! jps | grep -q DataNode ; then
  echo ERROR: datanode not up
fi
```

In practice, the `dfs.data.dir` and `mapred.local.dir` are often configured on the same set of disks, so a disk failure will result in the failure of both a `dfs.data.dir` and `mapred.local.dir`.

See the section titled "Configuring the Node Health Check Script" in [the Apache cluster setup documentation](#) for further details.

Step 4: Configure JobTracker Recovery

JobTracker recovery means that jobs that are running when JobTracker fails (for example, because of a system crash or hardware failure) are re-run when the JobTracker is restarted. Any jobs that were running at the time of the failure will be re-run from the beginning automatically.

A recovered job will have the following properties:

- It will have the same job ID as when it was submitted.
- It will run under the same user as the original job.
- It will write to the same output directory as the original job, overwriting any previous output.
- It will show as RUNNING on the JobTracker web page after you restart the JobTracker.

Enabling JobTracker Recovery

By default JobTracker recovery is off, but you can enable it by setting the property `mapreduce.jobtracker.restart.recover` to `true` in `mapred-site.xml`.

Step 5: If Necessary, Deploy your Custom Configuration to your Entire Cluster

[Deploy the configuration](#) on page 203 if you have not already done so.

Step 6: If Necessary, Start HDFS on Every Node in the Cluster

[Start HDFS](#) on page 202 if you have not already done so .

Step 7: If Necessary, Create the HDFS /tmp Directory

[Create the /tmp directory](#) on page 203 if you have not already done so.



Important:

If you do not create `/tmp` properly, with the right permissions as shown below, you may have problems with CDH components later. Specifically, if you don't create `/tmp` yourself, another process may create it automatically with restrictive permissions that will prevent your other applications from using it.

Step 8: Create MapReduce /var directories

```
sudo -u hdfs hadoop fs -mkdir -p /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
sudo -u hdfs hadoop fs -chmod 1777 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
sudo -u hdfs hadoop fs -chown -R mapred /var/lib/hadoop-hdfs/cache/mapred
```

Step 9: Verify the HDFS File Structure

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

You should see:

```
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 10: Create and Configure the `mapred.system.dir` Directory in HDFS

After you start HDFS and create `/tmp`, but before you start the JobTracker (see the [next step](#)), you must also create the HDFS directory specified by the `mapred.system.dir` parameter (by default `${hadoop.tmp.dir}/mapred/system` and configure it to be owned by the `mapred` user.

To create the directory in its default location:

```
$ sudo -u hdfs hadoop fs -mkdir /tmp/mapred/system
$ sudo -u hdfs hadoop fs -chown mapred:hadoop /tmp/mapred/system
```



Important:

If you create the `mapred.system.dir` directory in a different location, specify that path in the `conf/mapred-site.xml` file.

When starting up, MapReduce sets the permissions for the `mapred.system.dir` directory to `drwx-----`, assuming the user `mapred` owns that directory.

Step 11: Start MapReduce

To start MapReduce, start the TaskTracker and JobTracker services

On each TaskTracker system:

```
$ sudo service hadoop-0.20-mapreduce-tasktracker start
```

On the JobTracker system:

```
$ sudo service hadoop-0.20-mapreduce-jobtracker start
```

Step 12: Create a Home Directory for each MapReduce User

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
```

Step 13: Set `HADOOP_MAPRED_HOME`

For each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop in an MRv1 installation, set the `HADOOP_MAPRED_HOME` environment variable as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

Step 14: Configure the Hadoop Daemons to Start at Boot Time

See [Configuring the Daemons to Start on Boot](#) on page 215

Configuring the Daemons to Start on Boot



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).



Important:

Make sure you are not trying to run MRv1 and YARN on the same set of nodes at the same time. This is not recommended; it will degrade your performance and may result in an unstable MapReduce cluster deployment.

To start the Hadoop daemons at boot time and on restarts, enable their `init` scripts on the systems on which the services will run, using the `chkconfig` tool. See [Configuring init to Start Core Hadoop System Services](#).

Non-core services can also be started at boot time; after you install the non-core components, see [Configuring init to Start Non-core Hadoop System Services](#) for instructions.

Installing CDH 5 Components

In a new installation, you should install and deploy CDH before proceeding to install the components listed below. See [Installing the Latest CDH 5 Release](#) on page 155 and [Deploying CDH 5 on a Cluster](#) on page 190.



Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

CDH 5 Components

Use the following sections to install or upgrade CDH 5 components:

- [Crunch Installation](#) on page 216
- [Flume Installation](#) on page 217
- [HBase Installation](#) on page 227
- [HCatalog Installation](#) on page 254
- [Hive Installation](#) on page 272
- [HttpFS Installation](#) on page 298
- [Hue Installation](#) on page 301
- [Impala Installation](#) on page 259
- [KMS Installation](#) on page 329
- [Mahout Installation](#) on page 330
- [Oozie Installation](#) on page 332
- [Pig Installation](#) on page 346
- [Search Installation](#) on page 349
- [Sentry Installation](#) on page 361
- [Snappy Installation](#) on page 362
- [Spark Installation](#) on page 364

Installing Cloudera Manager and CDH

- [Sqoop 1 Installation](#) on page 372
- [Sqoop 2 Installation](#) on page 376
- [Whirr Installation](#) on page 381
- [ZooKeeper Installation](#)

See also the instructions for [installing or updating LZO](#).

Crunch Installation

The Apache Crunch™ project develops and supports Java APIs that simplify the process of creating data pipelines on top of Apache Hadoop. The Crunch APIs are modeled after FlumeJava (PDF), which is the library that Google uses for building data pipelines on top of their own implementation of MapReduce.

The Apache Crunch Java library provides a framework for writing, testing, and running MapReduce pipelines. Its goal is to make pipelines that are composed of many user-defined functions simple to write, easy to test, and efficient to run. Running on top of Hadoop MapReduce and Apache Spark, the Apache Crunch library is a simple Java API for tasks like joining and data aggregation that are tedious to implement on plain MapReduce. The APIs are especially useful when processing data that does not fit naturally into relational model, such as time series, serialized object formats like protocol buffers or Avro records, and HBase rows and columns. For Scala users, there is the Scrunch API, which is built on top of the Java APIs and includes a REPL (read-eval-print loop) for creating MapReduce pipelines.

The following sections describe how to install Crunch:

- [Crunch Prerequisites](#) on page 216
- [Crunch Packaging](#) on page 216
- [Installing and Upgrading Crunch](#) on page 216
- [Crunch Documentation](#) on page 217

Crunch Prerequisites

- An [operating system supported by CDH 5](#)
- [Oracle JDK](#)

Crunch Packaging

The packaging options for installing Crunch are:

- RPM packages
- Debian packages

There are two Crunch packages:

- `crunch`: provides all the functionality of crunch allowing users to create data pipelines over execution engines like MapReduce, Spark, and so on.
- `crunch-doc`: the documentation package.



Note: Crunch is also available as a parcel, included with the CDH 5 parcel. If you install CDH 5 with Cloudera Manager, Crunch will be installed automatically.

Installing and Upgrading Crunch

To install the Crunch packages:



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install or upgrade Crunch on a Red Hat system:

```
$ sudo yum install crunch
```

To install or upgrade Crunch on a SLES system:

```
$ sudo zypper install crunch
```

To install or upgrade Crunch on an Ubuntu or Debian system:

```
$ sudo apt-get install crunch
```

To use the Crunch documentation:

The Crunch docs are bundled in a `crunch-doc` package that should be installed separately.

```
$ sudo apt-get install crunch-doc
```

The contents of this package are saved under `/usr/share/doc/crunch*`.

After a package installation, the Crunch jars can be found in `/usr/lib/crunch`.

If you installed CDH 5 through Cloudera Manager, the CDH 5 parcel includes Crunch and the jars are installed automatically as part of the CDH 5 installation. By default the jars will be found in `/opt/cloudera/parcels/CDH/lib/crunch`.

Crunch Documentation

For more information about Crunch, see the following documentation:

- [Getting Started with Crunch](#)
- [Apache Crunch User Guide](#)

Flume Installation

Apache Flume is a distributed, reliable, and available system for efficiently collecting, aggregating and moving large amounts of log data from many different sources to a centralized data store.

**Note:**

To install Flume using Cloudera Manager, see [The Flume Service](#).

Upgrading Flume

Use the instructions that follow to upgrade Flume.

**Important: Running Services**

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

Upgrading Flume from an Earlier CDH 5 release

These instructions assume that you are upgrading Flume as part of an upgrade to the latest CDH 5 release, and have already performed the steps in [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

To upgrade Flume from an earlier CDH 5 release, install the new version of Flume using one of the methods described below: [Installing the Flume RPM or Debian Packages](#) on page 218 or [Installing the Flume Tarball](#) on page 218.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Flume Packaging

There are currently three packaging options available for installing Flume:

- Tarball (.tar.gz)
- RPM packages
- Debian packages

Installing the Flume Tarball

The Flume tarball is a self-contained package containing everything needed to use Flume on a Unix-like system. To install Flume from the tarball, you unpack it in the appropriate directory.



Note:

The tarball does not come with any scripts suitable for running Flume as a service or daemon. This makes the tarball distribution appropriate for *ad hoc* installations and preliminary testing, but a more complete installation is provided by the binary RPM and Debian packages.

To install the Flume tarball on Linux-based systems:

1. Run the following commands, replacing the `(component_version)` with the current version numbers for Flume and CDH.

```
$ cd /usr/local/lib
$ sudo tar -zxvf <path_to_flume-ng-(Flume_version)-cdh(CDH_version).tar.gz>
$ sudo mv flume-ng-(Flume_version)-cdh(CDH_version) flume-ng
```

For example,

```
$ cd /usr/local/lib
$ sudo tar -zxvf <path_to_flume-ng-1.4.0-cdh5.0.0.tar.gz>
$ sudo mv flume-ng-1.4.0-cdh5.0.0 flume-ng
```

2. To complete the configuration of a tarball installation, you must set your `PATH` variable to include the `bin/` subdirectory of the directory where you installed Flume. For example:

```
$ export PATH=/usr/local/lib/flume-ng/bin:$PATH
```

Installing the Flume RPM or Debian Packages

Installing the Flume RPM and Debian packages is more convenient than installing the Flume tarball because the packages:

- Handle dependencies

- Provide for easy upgrades
- Automatically install resources to conventional locations
- Handle daemon startup and shutdown.

The Flume RPM and Debian packages consist of three packages:

- `flume-ng` — Everything you need to run Flume
- `flume-ng-agent` — Handles starting and stopping the Flume agent as a service
- `flume-ng-doc` — Flume documentation

All Flume installations require the common code provided by `flume-ng`.



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install Flume on Ubuntu and other Debian systems:

```
$ sudo apt-get install flume-ng
```

To install Flume On Red Hat-compatible systems:

```
$ sudo yum install flume-ng
```

To install Flume on SLES systems:

```
$ sudo zypper install flume-ng
```

You may also want to enable automatic start-up on boot. To do this, install the Flume agent.

To install the Flume agent so Flume starts automatically on boot on Ubuntu and other Debian systems:

```
$ sudo apt-get install flume-ng-agent
```

To install the Flume agent so Flume starts automatically on boot on Red Hat-compatible systems:

```
$ sudo yum install flume-ng-agent
```

To install the Flume agent so Flume starts automatically on boot on SLES systems:

```
$ sudo zypper install flume-ng-agent
```

To install the documentation:

To install the documentation on Ubuntu and other Debian systems:

```
$ sudo apt-get install flume-ng-doc
```

To install the documentation on Red Hat-compatible systems:

```
$ sudo yum install flume-ng-doc
```

To install the documentation on SLES systems:

```
$ sudo zypper install flume-ng-doc
```

Flume Configuration

Flume 1.x provides a template configuration file for `flume.conf` called `conf/flume-conf.properties.template` and a template for `flume-env.sh` called `conf/flume-env.sh.template`.

1. Copy the Flume template property file `conf/flume-conf.properties.template` to `conf/flume.conf`, then edit it as appropriate.

```
$ sudo cp conf/flume-conf.properties.template conf/flume.conf
```

This is where you define your sources, sinks, and channels, and the flow within an agent. By default, the properties file is configured to work out of the box using a sequence generator source, a logger sink, and a memory channel. For information on configuring agent flows in Flume 1.x, as well as more details about the [supported sources, sinks and channels](#), see the documents listed under [Viewing the Flume Documentation](#).

2. Optionally, copy the template `flume-env.sh` file `conf/flume-env.sh.template` to `conf/flume-env.sh`.

```
$ sudo cp conf/flume-env.sh.template conf/flume-env.sh
```

The `flume-ng` executable looks for a file named `flume-env.sh` in the `conf` directory, and sources it if it finds it. Some use cases for using `flume-env.sh` are to specify a bigger heap size for the flume agent, or to specify debugging or profiling options via `JAVA_OPTS` when developing your own custom Flume NG components such as sources and sinks. If you do not make any changes to this file, then you need not perform the copy as it is effectively empty by default.

Verifying the Flume Installation

At this point, you should have everything necessary to run Flume, and the `flume-ng` command should be in your `$PATH`. You can test this by running:

```
$ flume-ng help
```

You should see something similar to this:

```
Usage: /usr/bin/flume-ng <command> [options]...

commands:
  help           display this help text
  agent         run a Flume agent
  avro-client    run an avro Flume client
  version       show Flume version info

global options:
  --conf,-c <conf>    use configs in <conf> directory
  --classpath,-C <cp> append to the classpath
  --dryrun,-d         do not actually start Flume, just print the command
  --Dproperty=value  sets a JDK system property value

agent options:
  --conf-file,-f <file> specify a config file (required)
  --name,-n <name>     the name of this agent (required)
  --help,-h           display help text

avro-client options:
  --rpcProps,-P <file> RPC client properties file with server connection params
  --host,-H <host>    hostname to which events will be sent (required)
  --port,-p <port>    port of the avro source (required)
  --dirname <dir>     directory to stream to avro source
  --filename,-F <file> text file to stream to avro source [default: std input]
  --headerFile,-R <file> headerFile containing headers as key/value pairs on each new
line
  --help,-h           display help text

Either --rpcProps or both --host and --port must be specified.
```

Note that if `<conf>` directory is specified, then it is always included first in the classpath.

**Note:**

If Flume is not found and you installed Flume from a tarball, make sure that `$FLUME_HOME/bin` is in your `$PATH`.

Running Flume

If Flume is installed using an RPM or Debian package, or managed by Cloudera Manager, you can use the following commands to start, stop, and restart the Flume agent using `init` scripts:

```
$ sudo service flume-ng-agent <start | stop | restart>
```

You can also run the agent in the foreground directly by using the `flume-ng agent` command:

```
$ /usr/bin/flume-ng agent -c <config-dir> -f <config-file> -n <agent-name>
```

For example:

```
$ /usr/bin/flume-ng agent -c /etc/flume-ng/conf -f /etc/flume-ng/conf/flume.conf -n agent
```

Files Installed by the Flume RPM and Debian Packages

Resource	Location	Notes
Configuration Directory	<code>/etc/flume-ng/conf</code>	
Configuration File	<code>/etc/flume-ng/conf/flume.conf</code>	This configuration will be picked-up by the flume agent startup script.
Template of User Customizable Configuration File	<code>/etc/flume-ng/conf/flume-conf.properties.template</code>	Contains a sample config. To use this configuration you should copy this file onto <code>/etc/flume-ng/conf/flume.conf</code> and then modify as appropriate
Template of User Customizable environment file	<code>/etc/flume-ng/conf/flume-env.sh.template</code>	If you want modify this file, copy it first and modify the copy
Daemon Log Directory	<code>/var/log/flume-ng</code>	Contains log files generated by flume agent
Default Flume Home	<code>/usr/lib/flume-ng</code>	Provided by RPMS and DEBS
Flume Agent startup script	<code>/etc/init.d/flume-ng-agent</code>	Provided by RPMS and DEBS
Recommended tar.gz Flume Home	<code>/usr/local/lib/flume-ng</code>	Recommended but installation dependent
Flume Wrapper Script	<code>/usr/bin/flume-ng</code>	Called by the Flume Agent startup script
Flume Agent configuration file	<code>/etc/default/flume-ng-agent</code>	Allows you to specify non-default values for the agent name and for the configuration file location

Supported Sources, Sinks, and Channels

The following tables list the only currently-supported sources, sinks, and channels. For more information, including information on developing custom components, see the documents listed under [Viewing the Flume Documentation](#).

Sources

Type	Description	Implementation Class
avro	Avro Netty RPC event source. Listens on Avro port and receives events from external Avro streams.	AvroSource
netcat	Netcat style TCP event source. Listens on a given port and turns each line of text into an event.	NetcatSource
seq	Monotonically incrementing sequence generator event source	SequenceGeneratorSource
exec	Execute a long-lived Unix process and read from stdout.	ExecSource
syslogtcp	Reads syslog data and generates flume events. Creates a new event for a string of characters separated by carriage return (\n).	SyslogTcpSource
syslogudp	Reads syslog data and generates flume events. Treats an entire message as a single event.	SyslogUDPSource
org.apache.flume.source.avroLegacy. AvroLegacySource	Allows the Flume 1.x agent to receive events from Flume 0.9.4 agents over avro rpc.	AvroLegacySource
org.apache.flume.source.thriftLegacy. ThriftLegacySource	Allows the Flume 1.x agent to receive events from Flume 0.9.4 agents over thrift rpc.	ThriftLegacySource
org.apache.flume.source.StressSource	Mainly for testing purposes. Not meant for production use. Serves as a continuous source of events where each event has the same payload.	StressSource
org.apache.flume.source.scribe. ScribeSource	Scribe event source. Listens on Scribe port and receives events from Scribe.	ScribeSource
multiport_syslogtcp	Multi-port capable version of the SyslogTcpSource.	MultiportSyslogTCPSource
spooldir	Used for ingesting data by placing files to be ingested into a "spooling" directory on disk.	SpoolDirectorySource
http	Accepts Flume events by HTTP POST and GET. GET should be used for experimentation only.	HTTPSource

Type	Description	Implementation Class
org.apache.flume.source.jms.JMSSource	Reads messages from a JMS destination such as a queue or topic.	JMSSource
org.apache.flume.agent.embedded.EmbeddedSource	Used only by the Flume embedded agent. See Flume Developer Guide for more details.	EmbeddedSource
Other (custom)	You need to specify the fully-qualified name of the custom source, and provide that class (and its dependent code) in Flume's classpath. You can do this by creating a JAR file to hold the custom code, and placing the JAR in Flume's <code>lib</code> directory.	—

Sinks

Type	Description	Implementation Class
logger	Log events at INFO level using configured logging subsystem (log4j by default)	LoggerSink
avro	Sink that invokes a pre-defined Avro protocol method for all events it receives (when paired with an avro source, forms tiered collection)	AvroSink
hdfs	Writes all events received to HDFS (with support for rolling, bucketing, HDFS-200 append, and more)	HDFSEventSink
file_roll	Writes all events received to one or more files.	RollingFileSink
org.apache.flume.hbase.HBaseSink	A simple sink that reads events from a channel and writes them synchronously to HBase. The <code>AsyncHBaseSink</code> is recommended. See Importing Data Into HBase .	HBaseSink
org.apache.flume.sink.hbase.AsyncHBaseSink	A simple sink that reads events from a channel and writes them asynchronously to HBase. This is the recommended HBase sink, but note that it does not support Kerberos. See Importing Data Into HBase .	AsyncHBaseSink
org.apache.flume.sink.solr.morphine.MorphineSolrSink	Extracts and transforms data from Flume events, and loads it into Apache Solr servers. See the section on <code>MorphineSolrSink</code> in the Flume User Guide listed under Viewing the Flume Documentation on page 227.	MorphineSolrSink

Type	Description	Implementation Class
Other (custom)	You need to specify the fully-qualified name of the custom sink, and provide that class (and its dependent code) in Flume's classpath. You can do this by creating a JAR file to hold the custom code, and placing the JAR in Flume's <code>lib</code> directory.	—

Channels

Type	Description	Implementation Class
memory	In-memory, fast, non-durable event transport	MemoryChannel
jdbc	JDBC-based, durable event transport (Derby-based)	JDBCChannel
file	File-based, durable event transport	FileChannel
Other (custom)	You need to specify the fully-qualified name of the custom channel, and provide that class (and its dependent code) in Flume's classpath. You can do this by creating a JAR file to hold the custom code, and placing the JAR in Flume's <code>lib</code> directory.	—

Providing for Disk Space Usage

It's important to provide plenty of disk space for any Flume File Channel. The largest consumers of disk space in the File Channel are the data logs. You can configure the File Channel to write these logs to multiple data directories. The following space will be consumed by default in each data directory:

- Current log file (up to 2 GB)
- Last log file (up to 2 GB)
- Pending delete log file (up to 2 GB)

Events in the queue could cause many more log files to be written, each of them up 2 GB in size by default.

You can configure both the maximum log file size (`MaxFileSize`) and the directories the logs will be written to (`DataDirs`) when you configure the File Channel; see the File Channel section of the [Flume User Guide](#) for details.

Using an On-disk Encrypted File Channel

Flume supports on-disk encryption of data on the local disk. To implement this:

- Generate an encryption key to use for the Flume Encrypted File Channel
- Configure on-disk encryption by setting parameters in the `flume.conf` file

**Important:**

Flume on-disk encryption operates with a maximum strength of 128-bit AES encryption unless the JCE unlimited encryption cryptography policy files are installed. Please see this Oracle document for information about enabling strong cryptography with JDK 1.6:

<http://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html>

Consult your security organization for guidance on the acceptable strength of your encryption keys. Cloudera has tested with AES-128, AES-192, and AES-256.

Generating Encryption Keys

Use the `keytool` program included in Oracle JDK 1.6 to create the AES encryption keys for use with Flume.

The command to generate a 128-bit key that uses the same password as the key store password is:

```
keytool -genseckey -alias key-1 -keyalg AES -keysize 128 -validity 9000 \
-keystore test.keystore -storetype jceks \
-storepass keyStorePassword
```

The command to generate a 128-bit key that uses a different password from that used by the key store is:

```
keytool -genseckey -alias key-0 -keypass keyPassword -keyalg AES \
-keysize 128 -validity 9000 -keystore test.keystore \
-storetype jceks -storepass keyStorePassword
```

The key store and password files can be stored anywhere on the file system; both files should have `flume` as the owner and `0600` permissions.

Please note that `-keysize` controls the strength of the AES encryption key, in bits; 128, 192, and 256 are the allowed values.

Configuration

Flume on-disk encryption is enabled by setting parameters in the `/etc/flume-ng/conf/flume.conf` file.

Basic Configuration

The first example is a basic configuration with an alias called `key-0` that uses the same password as the key store:

```
agent.channels.ch-0.type = file
agent.channels.ch-0.capacity = 10000
agent.channels.ch-0.encryption.cipherProvider = AESCTRNOPADDING
agent.channels.ch-0.encryption.activeKey = key-0
agent.channels.ch-0.encryption.keyProvider = JCEKSFILE
agent.channels.ch-0.encryption.keyProvider.keyStoreFile = /path/to/my.keystore
agent.channels.ch-0.encryption.keyProvider.keyStorePasswordFile =
/path/to/my.keystore.password
agent.channels.ch-0.encryption.keyProvider.keys = key-0
```

In the next example, `key-0` uses its own password which may be different from the key store password:

```
agent.channels.ch-0.type = file
agent.channels.ch-0.capacity = 10000
agent.channels.ch-0.encryption.cipherProvider = AESCTRNOPADDING
agent.channels.ch-0.encryption.activeKey = key-0
agent.channels.ch-0.encryption.keyProvider = JCEKSFILE
agent.channels.ch-0.encryption.keyProvider.keyStoreFile = /path/to/my.keystore
agent.channels.ch-0.encryption.keyProvider.keyStorePasswordFile =
/path/to/my.keystore.password
agent.channels.ch-0.encryption.keyProvider.keys = key-0
agent.channels.ch-0.encryption.keyProvider.keys.key-0.passwordFile =
/path/to/key-0.password
```

Changing Encryption Keys Over Time

To modify the key, modify the configuration as shown below. This example shows how to change the configuration to use key-1 instead of key-0:

```
agent.channels.ch-0.type = file
agent.channels.ch-0.capacity = 10000
agent.channels.ch-0.encryption.cipherProvider = AESCTRNOPADDING
agent.channels.ch-0.encryption.activeKey = key-1
agent.channels.ch-0.encryption.keyProvider = JCEKSFILE
agent.channels.ch-0.encryption.keyProvider.keyStoreFile = /path/to/my.keystore
agent.channels.ch-0.encryption.keyProvider.keyStorePasswordFile =
/path/to/my.keystore.password
agent.channels.ch-0.encryption.keyProvider.keys = key-0 key-1
```

The same scenario except that key-0 and key-1 have their own passwords is shown here:

```
agent.channels.ch-0.type = file
agent.channels.ch-0.capacity = 10000
agent.channels.ch-0.encryption.cipherProvider = AESCTRNOPADDING
agent.channels.ch-0.encryption.activeKey = key-1
agent.channels.ch-0.encryption.keyProvider = JCEKSFILE
agent.channels.ch-0.encryption.keyProvider.keyStoreFile = /path/to/my.keystore
agent.channels.ch-0.encryption.keyProvider.keyStorePasswordFile =
/path/to/my.keystore.password
agent.channels.ch-0.encryption.keyProvider.keys = key-0 key-1
agent.channels.ch-0.encryption.keyProvider.keys.key-0.passwordFile =
/path/to/key-0.password
agent.channels.ch-0.encryption.keyProvider.keys.key-1.passwordFile =
/path/to/key-1.password
```

Troubleshooting

If the unlimited strength JCE policy files are not installed, an error similar to the following is printed in the flume.log:

```
07 Sep 2012 23:22:42,232 ERROR [lifecycleSupervisor-1-0]
(org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider.getCipher:137) - Unable
to load key using transformation: AES/CTR/NoPadding; Warning: Maximum allowed key length
= 128 with the available JCE security policy files. Have you installed the JCE unlimited
strength jurisdiction policy files?
java.security.InvalidKeyException: Illegal key size
at javax.crypto.Cipher.a(DashoA13*..)
at javax.crypto.Cipher.a(DashoA13*..)
at javax.crypto.Cipher.a(DashoA13*..)
at javax.crypto.Cipher.init(DashoA13*..)
at javax.crypto.Cipher.init(DashoA13*..)
at
org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider.getCipher(AESCTRNoPaddingProvider.java:120)
at
org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider.access$200(AESCTRNoPaddingProvider.java:35)
at
org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider$AESCTRNoPaddingDecryptor.<init>(AESCTRNoPaddingProvider.java:94)
at
org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider$AESCTRNoPaddingDecryptor.<init>(AESCTRNoPaddingProvider.java:91)
at
org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider$DecryptorBuilder.build(AESCTRNoPaddingProvider.java:66)
at
org.apache.flume.channel.file.encryption.AESCTRNoPaddingProvider$DecryptorBuilder.build(AESCTRNoPaddingProvider.java:62)
at
org.apache.flume.channel.file.encryption.CipherProviderFactory.getDecrypter(CipherProviderFactory.java:47)
at org.apache.flume.channel.file.LogFileV3$SequentialReader.<init>(LogFileV3.java:257)
at
org.apache.flume.channel.file.LogFileFactory.getSequentialReader(LogFileFactory.java:110)
at org.apache.flume.channel.file.ReplayHandler.replayLog(ReplayHandler.java:258)
at org.apache.flume.channel.file.Log.replay(Log.java:339)
at org.apache.flume.channel.file.FileChannel.start(FileChannel.java:260)
at
org.apache.flume.lifecycle.LifecycleSupervisor$MonitorRunnable.run(LifecycleSupervisor.java:236)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:441)
at java.util.concurrent.FutureTask$Sync.innerRunAndReset(FutureTask.java:317)
at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:150)
```

```

at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$101(ScheduledThreadPoolExecutor.java:98)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.runPeriodic(ScheduledThreadPoolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:204)
at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
at java.lang.Thread.run(Thread.java:662)

```

Viewing the Flume Documentation

For additional Flume documentation, see the [Flume User Guide](#) and the [Flume Developer Guide](#).

For additional information about Flume, see the [Apache Flume wiki](#).

HBase Installation

Apache HBase provides large-scale tabular storage for Hadoop using the Hadoop Distributed File System (HDFS). Cloudera recommends installing HBase in a standalone mode before you try to run it on a whole cluster.



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).



Important: Running Services

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

Use the following sections to install, update, and configure HBase:

- [New Features and Changes for HBase in CDH 5](#)
- [Upgrading HBase](#)
- [Installing HBase](#)
- [Configuration Settings](#)
- [Starting HBase in Standalone Mode](#)
- [Configuring HBase in Pseudo-Distributed Mode](#)
- [Deploying HBase in a Cluster](#)
- [Using the Hbase Shell](#)
- [Using MapReduce with HBase](#)
- [Troubleshooting](#)
- [Apache HBase Documentation](#)
- [HBase Replication](#)
- [Managing HBase Snapshots](#)

Next Steps

After installing and configuring HBase, check out the following topics about using HBase:

- [Importing Data Into HBase](#)

- [Writing Data to HBase](#)
- [Reading Data from HBase](#)

New Features and Changes for HBase in CDH 5

CDH 5.0.x and 5.1.x each include major upgrades to HBase. Each of these upgrades provides exciting new features, as well as things to keep in mind when upgrading from a previous version.

For new features and changes introduced in older CDH 5 releases, skip to [CDH 5.1 HBase Changes](#) or [CDH 5.0.x HBase Changes](#).

CDH 5.3 HBase Changes

SlabCache Has Been Deprecated

SlabCache, which was marked as deprecated in CDH 5.2, has been removed in CDH 5.3. To configure the BlockCache, see [Configuring the HBase BlockCache](#) on page 241.

checkAndMutate(RowMutations) API

CDH 5.3 provides `checkAndMutate(RowMutations)`, in addition to existing support for atomic `checkAndPut` as well as `checkAndDelete` operations on individual rows ([HBASE-11796](#)).

CDH 5.2 HBase Changes

CDH 5.2 introduces HBase 0.98.6, which represents a minor upgrade to HBase. This upgrade introduces new features and moves some features which were previously marked as experimental to fully supported status. This overview provides information about the most important features, how to use them, and where to find out more information. Cloudera appreciates your feedback about these features.

JAVA_HOME must be set in your environment.

HBase now requires JAVA_HOME to be set in your environment. If it is not set, HBase will fail to start and an error will be logged. If you use Cloudera Manager, this is set automatically. If you use CDH without Cloudera Manager, JAVA_HOME should be set up as part of the overall installation. See [Java Development Kit Installation](#) on page 35 for instructions on setting JAVA_HOME, as well as other JDK-specific instructions.

The default value for `hbase.hstore.flusher.count` has increased from 1 to 2.

The default value for `hbase.hstore.flusher.count` has been increased from one thread to two. This new configuration can improve performance when writing to HBase under some workloads. However, for high IO workloads two flusher threads can create additional contention when writing to HDFS. If after upgrading to CDH 5.2. you see an increase in flush times or performance degradation, lowering this value to 1 is recommended. Use the RegionServer's advanced configuration snippet for `hbase-site.xml` if you use Cloudera Manager, or edit the file directly otherwise.

The default value for `hbase.hregion.memstore.block.multiplier` has increased from 2 to 4.

The default value for `hbase.hregion.memstore.block.multiplier` has increased from 2 to 4, in order to improve both throughput and latency. If you experience performance degradation due to this change, change the value setting to 2, using the RegionServer's advanced configuration snippet for `hbase-site.xml` if you use Cloudera Manager, or by editing the file directly otherwise.

SlabCache is deprecated, and BucketCache is now the default block cache.

CDH 5.1 provided full support for the BucketCache block cache. CDH 5.2 deprecates usage of SlabCache in favor of BucketCache. To configure BucketCache, see [BucketCache Block Cache](#) on page 231

Changed Syntax of `user_permissions` Shell Command

The pattern-matching behavior for the `user_permissions` HBase Shell command has changed. Previously, either of the following two commands would return permissions of all known users in HBase:

```
hbase> user_permissions '*'
```

```
hbase> user_permissions '.*'
```

The first variant is no longer supported. The second variant is the only supported operation and also supports passing in other Java regular expressions.

New Properties for IPC Configuration

If the Hadoop configuration is read after the HBase configuration, Hadoop's settings can override HBase's settings if the names of the settings are the same. To avoid the risk of override, HBase has renamed the following settings (by prepending 'hbase.') so that you can set them independent of your setting for Hadoop. If you do not use the HBase-specific variants, the Hadoop settings will be used. If you have not experienced issues with your configuration, there is no need to change it.

Hadoop Configuration Property	New HBase Configuration Property
<code>ipc.server.listen.queue.size</code>	<code>hbase.ipc.server.listen.queue.size</code>
<code>ipc.server.max.callqueue.size</code>	<code>hbase.ipc.server.max.callqueue.size</code>
<code>ipc.server.max.callqueue.length</code>	<code>hbase.ipc.server.max.callqueue.length</code>
<code>ipc.server.read.threadpool.size</code>	<code>hbase.ipc.server.read.threadpool.size</code>
<code>ipc.server.tcpkeepalive</code>	<code>hbase.ipc.server.tcpkeepalive</code>
<code>ipc.server.tcpnodelay</code>	<code>hbase.ipc.server.tcpnodelay</code>
<code>ipc.client.call.purge.timeout</code>	<code>hbase.ipc.client.call.purge.timeout</code>
<code>ipc.client.connection.maxidletime</code>	<code>hbase.ipc.client.connection.maxidletime</code>
<code>ipc.client.idlethreshold</code>	<code>hbase.ipc.client.idlethreshold</code>
<code>ipc.client.kill.max</code>	<code>hbase.ipc.client.kill.max</code>

Snapshot Manifest Configuration

Snapshot manifests were previously a feature included in HBase in CDH 5 but not in Apache HBase. They are now included in Apache HBase 0.98.6. To use snapshot manifests, you now need to set `hbase.snapshot.format.version` to 2 in `hbase-site.xml`. This is the default for HBase in CDH 5.2, but the default for Apache HBase 0.98.6 is 1. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. The new snapshot code can read both version 1 and 2. However, if you use version 2, you will not be able to read these snapshots on HBase versions prior to 0.98.

Not using manifests (setting `hbase.snapshot.format.version` to 1) can cause excess load on the NameNode and impact performance.

Tags

Tags, which allow metadata to be stored in HFiles alongside cell data, are a feature of HFile version 3, are needed for per-cell access controls and visibility labels. Tags were previously considered an experimental feature but are now fully supported.

Per-Cell Access Controls

Per-cell access controls were introduced as an experimental feature in CDH 5.1 and are fully supported in CDH 5.2. You must use HFile version 3 in order to use per-cell access controls. For more information about access controls, see [Per-Cell Access Controls](#) on page 234.

Experimental Features



Warning: These features are still considered experimental. Experimental features are not supported and Cloudera does not recommend using them in production environments or with important data.

Visibility Labels

You can now specify a list of visibility labels, such as CONFIDENTIAL, TOPSECRET, or PUBLIC, at the cell level. You can associate users with these labels to enforce visibility of HBase data. These labels can be grouped into complex expressions using logical operators &, |, and ! (AND, OR, NOT). A given user is associated with a set of visibility labels, and the policy for associating the labels is pluggable. A coprocessor, `org.apache.hadoop.hbase.security.visibility.DefaultScanLabelGenerator`, checks for visibility labels on cells that would be returned by a Get or Scan and drops the cells that a user is not authorized to see, before returning the results. The same coprocessor saves visibility labels as tags, in the HFiles alongside the cell data, when a Put operation includes visibility labels. You can specify custom implementations of `ScanLabelGenerator` by setting the property `hbase.regionserver.scan.visibility.label.generator.class` to a comma-separated list of classes in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise.

No labels are configured by default. You can add a label to the system using either the `VisibilityClient#addLabels()` API or the `add_label` shell command. Similar APIs and shell commands are provided for deleting labels and assigning them to users. Only a user with superuser access (the `hbase.superuser` access level) can perform these operations.

To assign a visibility label to a cell, you can label the cell using the API method `Mutation#setCellVisibility(new CellVisibility(<labelExp>))`. An API is provided for managing visibility labels, and you can also perform many of the operations using HBase Shell.

Previously, visibility labels could not contain the symbols &, |, !, (and), but this is no longer the case.

For more information about visibility labels, see the [Visibility Labels](#) section of the *Apache HBase Reference Guide*.

If you use visibility labels along with access controls, you must ensure that the Access Controller is loaded before the Visibility Controller in the list of coprocessors. This is the default configuration. See [HBASE-11275](#).

Visibility labels are an **experimental** feature introduced in CDH 5.1, and still experimental in CDH 5.2.

Transparent Server-Side Encryption

Transparent server-side encryption can now be enabled for both HFiles and write-ahead logs (WALs), to protect their contents at rest. To configure transparent encryption, first create an encryption key, then configure the appropriate settings in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. See the [Transparent Encryption](#) section in the *Apache HBase Reference Guide* for more information.

Transparent server-side encryption is an **experimental** feature introduced in CDH 5.1, and still experimental in CDH 5.2.

Stripe Compaction

Stripe compaction is a compaction scheme that segregates the data inside a region by row key, creating "stripes" of data which are visible within the region but transparent to normal operations. This striping improves read performance in common scenarios and greatly reduces variability by avoiding large or inefficient compactions.

Configuration guidelines and more information are available at [Stripe Compaction](#).

To configure stripe compaction for a single table from within the HBase shell, use the following syntax.

```
alter <table>, CONFIGURATION => {<setting> => <value>}
Example: alter 'orders', CONFIGURATION => {'hbase.store.stripe.fixed.count' => 10}
```

To configure stripe compaction for a column family from within the HBase shell, use the following syntax.

```
alter <table>, {NAME => <column family>, CONFIGURATION => {<setting => <value>}}
Example: alter 'logs', {NAME => 'blobs', CONFIGURATION =>
{'hbase.store.stripe.fixed.count' => 10}}
```

Stripe compaction is an **experimental** feature in CDH 5.1, and still experimental in CDH 5.2.

Distributed Log Replay

After a RegionServer fails, its failed region is assigned to another RegionServer, which is marked as "recovering" in ZooKeeper. A SplitLogWorker directly replays edits from the WAL of the failed RegionServer to the region at its new location. When a region is in "recovering" state, it can accept writes but no reads (including Append and Increment), region splits or merges. Distributed Log Replay extends the distributed log splitting framework. It works by directly replaying WAL edits to another RegionServer instead of creating `recovered.edits` files.

Distributed log replay provides the following advantages over using the current distributed log splitting functionality on its own.

- It eliminates the overhead of writing and reading a large number of `recovered.edits` files. It is not unusual for thousands of `recovered.edits` files to be created and written concurrently during a RegionServer recovery. Many small random writes can degrade overall system performance.
- It allows writes even when a region is in recovering state. It only takes seconds for a recovering region to accept writes again.

To enable distributed log replay, set `hbase.master.distributed.log.replay` to `true` in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. You must also enable HFile version 3. Distributed log replay is unsafe for rolling upgrades.

Distributed log replay is an **experimental** feature in CDH 5.1, and still experimental in CDH 5.2.

CDH 5.1 HBase Changes

CDH 5.1 introduces HBase 0.98, which represents a major upgrade to HBase. This upgrade introduces several new features, including a section of features which are considered experimental and should not be used in a production environment. This overview provides information about the most important features, how to use them, and where to find out more information. Cloudera appreciates your feedback about these features.

In addition to HBase 0.98, Cloudera has pulled in changes from [HBASE-10883](#), [HBASE-10964](#), [HBASE-10823](#), [HBASE-10916](#), and [HBASE-11275](#). Implications of these changes are detailed below and in the Release Notes.

BucketCache Block Cache

A new offheap BlockCache implementation, BucketCache, was introduced as an experimental feature in CDH 5 Beta 1, and is now fully supported in CDH 5.1. BucketCache can be used in either of the following two configurations:

- As a CombinedBlockCache with both onheap and offheap caches.
- As an L2 cache for the default onheap LruBlockCache

BucketCache requires less garbage-collection than SlabCache, which is the other offheap cache implementation in HBase. It also has many optional configuration settings for fine-tuning. All available settings are documented in the [API documentation for CombinedBlockCache](#). Following is a simple example configuration.

1. First, edit `hbase-env.sh` and set `-XX:MaxDirectMemorySize` to the total size of the desired onheap plus offheap, in this case, 5 GB (but expressed as 5G). To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise.

```
-XX:MaxDirectMemorySize=5G
```

2. Next, add the following configuration to `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. This configuration uses 80% of the `-XX:MaxDirectMemorySize` (4 GB) for offheap, and the remainder (1 GB) for onheap.

```
<property>
  <name>hbase.bucketcache.ioengine</name>
  <value>offheap</value>
</property>
<property>
  <name>hbase.bucketcache.percentage.in.combinedcache</name>
  <value>0.8</value>
</property>
<property>
  <name>hbase.bucketcache.size</name>
  <value>5120</value>
</property>
```

3. Restart or rolling restart your cluster for the configuration to take effect.

Access Control for EXEC Permissions

A new access control level has been added to check whether a given user has EXEC permission. This can be specified at the level of the cluster, table, row, or cell.

To use EXEC permissions, perform the following procedure.

- Install the AccessController coprocessor either as a system coprocessor or on a table as a table coprocessor.
- Set the `hbase.security.exec.permission.checks` configuration setting in `hbase-site.xml` to `true`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise..

For more information on setting and revoking security permissions, see the [Access Control](#) section of the *Apache HBase Reference Guide*.

Reverse Scan API

A reverse scan API has been introduced. This allows you to scan a table in reverse. Previously, if you wanted to be able to access your data in either direction, you needed to store the data in two separate tables, each ordered differently. This feature was implemented in [HBASE-4811](#).

To use the reverse scan feature, use the new `Scan.setReversed(boolean reversed)` API. If you specify a `startRow` and `stopRow`, to scan in reverse, the `startRow` needs to be lexicographically after the `stopRow`. See the [Scan](#) API documentation for more information.

MapReduce Over Snapshots

You can now run a MapReduce job over a snapshot from HBase, rather than being limited to live data. This provides the ability to separate your client-side work load from your live cluster if you need to run resource-intensive MapReduce jobs and can tolerate using potentially-stale data. You can either run the MapReduce job on the snapshot within HBase, or export the snapshot and run the MapReduce job against the exported file.

Running a MapReduce job on an exported file outside of the scope of HBase relies on the permissions of the underlying filesystem and server, and bypasses ACLs, visibility labels, and encryption that may otherwise be provided by your HBase cluster.

A new API, `TableSnapshotInputFormat`, is provided. For more information, see [TableSnapshotInputFormat](#).

MapReduce over snapshots was introduced in CDH 5.0.

Stateless Streaming Scanner over REST

A new stateless streaming scanner is available over the REST API. Using this scanner, clients do not need to restart a scan if the REST server experiences a transient failure. All query parameters are specified during the REST request. Query parameters include `startrow`, `endrow`, `columns`, `starttime`, `endtime`, `maxversions`, `batchtime`, and `limit`. Following are a few examples of using the stateless streaming scanner.

Scan the entire table, return the results in JSON.

```
curl -H "Accept: application/json" https://localhost:8080/ExampleScanner/*
```

Scan the entire table, return the results in XML.

```
curl -H "Content-Type: text/xml" https://localhost:8080/ExampleScanner/*
```

Scan only the first row.

```
curl -H "Content-Type: text/xml" \
https://localhost:8080/ExampleScanner/*?limit=1
```

Scan only specific columns.

```
curl -H "Content-Type: text/xml" \
https://localhost:8080/ExampleScanner/*?columns=a:1,b:1
```

Scan for rows between starttime and endtime.

```
curl -H "Content-Type: text/xml" \
https://localhost:8080/ExampleScanner/*?starttime=1389900769772\
&endtime=1389900800000
```

Scan for a given row prefix.

```
curl -H "Content-Type: text/xml" https://localhost:8080/ExampleScanner/test*
```

For full details about the stateless streaming scanner, see the [API documentation](#) for this feature.

Delete Methods of Put Class Now Use Constructor Timestamps

The `Delete()` methods of the `Put` class of the HBase Client API previously ignored the constructor's timestamp, and used the value of `HConstants.LATEST_TIMESTAMP`. This behavior was different from the behavior of the `add()` methods. The `Delete()` methods now use the timestamp from the constructor, creating consistency in behavior across the `Put` class. See [HBASE-10964](#).

Experimental Features

Warning: These features are still considered experimental. Experimental features are not supported and Cloudera does not recommend using them in production environments or with important data.

Visibility Labels

You can now specify a list of visibility labels, such as `CONFIDENTIAL`, `TOPSECRET`, or `PUBLIC`, at the cell level. You can associate users with these labels to enforce visibility of HBase data. These labels can be grouped into complex expressions using logical operators `&`, `|`, and `!` (AND, OR, NOT). A given user is associated with a set of visibility labels, and the policy for associating the labels is pluggable. A coprocessor, `org.apache.hadoop.hbase.security.visibility.DefaultScanLabelGenerator`, checks for visibility labels on cells that would be returned by a `Get` or `Scan` and drops the cells that a user is not authorized to see, before returning the results. The same coprocessor saves visibility labels as tags, in the HFiles alongside the cell data, when a `Put` operation includes visibility labels. You can specify custom implementations of `ScanLabelGenerator` by setting the property `hbase.regionserver.scan.visibility.label.generator.class` to a comma-separated list of classes.

No labels are configured by default. You can add a label to the system using either the `VisibilityClient#addLabels()` API or the `add_label` shell command. Similar APIs and shell commands are provided for deleting labels and assigning them to users. Only a user with superuser access (the `hbase.superuser` access level) can perform these operations.

To assign a visibility label to a cell, you can label the cell using the API method `Mutation#setCellVisibility(new CellVisibility(<labelExp>))`;

Visibility labels and request authorizations cannot contain the symbols `&`, `|`, `!`, `(` and `)` because they are reserved for constructing visibility expressions. See [HBASE-10883](#).

For more information about visibility labels, see the [Visibility Labels](#) section of the *Apache HBase Reference Guide*.

If you use visibility labels along with access controls, you must ensure that the Access Controller is loaded before the Visibility Controller in the list of coprocessors. This is the default configuration. See [HBASE-11275](#).

In order to use per-cell access controls or visibility labels, you must use HFile version 3. To enable HFile version 3, add the following to `hbase-site.xml`, using an [advanced code snippet](#) if you use Cloudera Manager, or directly to the file if your deployment is unmanaged.. Changes will take effect after the next major compaction.

```
<property>
  <name>hfile.format.version</name>
  <value>3</value>
</property>
```

Visibility labels are an **experimental** feature introduced in CDH 5.1.

Per-Cell Access Controls

You can now specify access control levels at the per-cell level, as well as at the level of the cluster, table, or row.

A new parent class has been provided, which encompasses `Get`, `Scan`, and `Query`. This change also moves the `getFilter` and `setFilter` methods of `Get` and `Scan` to the common parent class. Client code may need to be recompiled to take advantage of per-cell ACLs. See the [Access Control](#) section of the *Apache HBase Reference Guide* for more information.

The ACLs for cells having timestamps in the future are not considered for authorizing the pending mutation operations. See [HBASE-10823](#).

If you use visibility labels along with access controls, you must ensure that the Access Controller is loaded before the Visibility Controller in the list of coprocessors. This is the default configuration.

In order to use per-cell access controls or visibility labels, you must use HFile version 3. To enable HFile version 3, add the following to `hbase-site.xml`, using an [advanced code snippet](#) if you use Cloudera Manager, or directly to the file if your deployment is unmanaged.. Changes will take effect after the next major compaction.

```
<property>
  <name>hfile.format.version</name>
  <value>3</value>
</property>
```

Per-cell access controls are an **experimental** feature introduced in CDH 5.1.

Transparent Server-Side Encryption

Transparent server-side encryption can now be enabled for both HFiles and write-ahead logs (WALs), to protect their contents at rest. To configure transparent encryption, first create an encryption key, then configure the appropriate settings in `hbase-site.xml`. See the [Transparent Encryption](#) section in the *Apache HBase Reference Guide* for more information.

Transparent server-side encryption is an **experimental** feature introduced in CDH 5.1.

Stripe Compaction

Stripe compaction is a compaction scheme that segregates the data inside a region by row key, creating "stripes" of data which are visible within the region but transparent to normal operations. This striping improves read performance in common scenarios and greatly reduces variability by avoiding large or inefficient compactions.

Configuration guidelines and more information are available at [Stripe Compaction](#).

To configure stripe compaction for a single table from within the HBase shell, use the following syntax.

```
alter <table>, CONFIGURATION => {<setting> => <value>}
Example: alter 'orders', CONFIGURATION => {'hbase.store.stripe.fixed.count' => 10}
```

To configure stripe compaction for a column family from within the HBase shell, use the following syntax.

```
alter <table>, {NAME => <column family>, CONFIGURATION => {<setting => <value>}}
Example: alter 'logs', {NAME => 'blobs', CONFIGURATION =>
{'hbase.store.stripe.fixed.count' => 10}}
```

Stripe compaction is an **experimental** feature in CDH 5.1.

Distributed Log Replay

After a RegionServer fails, its failed region is assigned to another RegionServer, which is marked as "recovering" in ZooKeeper. A SplitLogWorker directly replays edits from the WAL of the failed RegionServer to the region at its new location. When a region is in "recovering" state, it can accept writes but no reads (including Append and Increment), region splits or merges. Distributed Log Replay extends the distributed log splitting framework. It works by directly replaying WAL edits to another RegionServer instead of creating `recovered.edits` files.

Distributed log replay provides the following advantages over using the current distributed log splitting functionality on its own.

- It eliminates the overhead of writing and reading a large number of `recovered.edits` files. It is not unusual for thousands of `recovered.edits` files to be created and written concurrently during a RegionServer recovery. Many small random writes can degrade overall system performance.
- It allows writes even when a region is in recovering state. It only takes seconds for a recovering region to accept writes again.

To enable distributed log replay, set `hbase.master.distributed.log.replay` to `true`. You must also enable HFile version 3. Distributed log replay is unsafe for rolling upgrades.

Distributed log replay is an **experimental** feature in CDH 5.1.

CDH 5.0.x HBase Changes

HBase in CDH 5.0.x is based on the Apache HBase 0.96 release. When upgrading to CDH 5.0.x, keep the following in mind.

Wire Compatibility

HBase in CDH 5.0.x (HBase 0.96) is not wire compatible with CDH 4 (based on 0.92 and 0.94 releases). Consequently, rolling upgrades from CDH 4 to CDH 5 are not possible because existing CDH 4 HBase clients cannot make requests to CDH 5 servers and CDH 5 HBase clients cannot make requests to CDH 4 servers. Clients of the Thrift and REST proxy servers, however, retain wire compatibility between CDH 4 and CDH 5.

Upgrade is Not Reversible

The upgrade from CDH 4 HBase to CDH 5 HBase is irreversible and requires HBase to be shut down completely. Executing the upgrade script reorganizes existing HBase data stored on HDFS into new directory structures, converts HBase 0.90 HFile v1 files to the improved and optimized HBase 0.96 HFile v2 file format, and rewrites the `hbase.version` file. This upgrade also removes transient data stored in ZooKeeper during the conversion to the new data format.

These changes were made to reduce the impact in future major upgrades. Previously HBase used brittle custom data formats and this move shifts HBase's RPC and persistent data to a more evolvable Protocol Buffer data format.

API Changes

The HBase User API (Get, Put, Result, Scanner etc; see [Apache HBase API documentation](#)) has evolved and attempts have been made to make sure the HBase Clients are source code compatible and thus should recompile without needing any source code modifications. This cannot be guaranteed however, since with the conversion to Protocol Buffers (ProtoBufs), some relatively obscure APIs have been removed. Rudimentary efforts have also been made to preserve recompile compatibility with advanced APIs such as Filters and Coprocessors. These advanced APIs are still evolving and our guarantees for API compatibility are weaker here.

For information about changes to custom filters, see [Custom Filters](#).

As of 0.96, the User API has been marked and all attempts at compatibility in future versions will be made. A version of the javadoc that only contains the User API can be found [here](#).

HBase Metrics Changes

HBase provides a metrics framework based on JMX beans. Between HBase 0.94 and 0.96, the metrics framework underwent many changes. Some beans were added and removed, some metrics were moved from one bean to another, and some metrics were renamed or removed. Click [here](#) to download the CSV spreadsheet which provides a mapping.

Custom Filters

If you used custom filters written for HBase 0.94, you need to recompile those filters for HBase 0.96. The custom filter must be altered to fit with the newer interface that uses protocol buffers. Specifically two new methods, `toByteArray(...)` and `parseFrom(...)`, which are detailed in detailed in the [Filter API](#). These should be used instead of the old methods `write(...)` and `readFields(...)`, so that protocol buffer serialization is used. To see what changes were required to port one of HBase's own custom filters, see the [Git commit](#) that represented porting the `SingleColumnValueFilter` filter.

Checksums

In CDH 4, HBase relied on HDFS checksums to protect against data corruption. When you upgrade to CDH 5, HBase checksums are now turned on by default. With this configuration, HBase reads data and then verifies the checksums. Checksum verification inside HDFS will be switched off. If the HBase-checksum verification fails, then the HDFS checksums are used instead for verifying data that is being read from storage. Once you turn on HBase checksums, you will not be able to roll back to an earlier HBase version.

You should see a modest performance gain after setting `hbase.regionserver.checksum.verify` to true for data that is not already present in the RegionServer's block cache.

To enable or disable checksums, modify the following configuration properties in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise.

```
<property>
  <name>hbase.regionserver.checksum.verify</name>
  <value>true</value>
  <description>
    If set to true, HBase will read data and then verify checksums for
    hfile blocks. Checksum verification inside HDFS will be switched off.
    If the hbase-checksum verification fails, then it will switch back to
    using HDFS checksums.
  </description>
</property>
```

The default value for the `hbase.hstore.checksum.algorithm` property has also changed to CRC32. Previously, Cloudera advised setting it to NULL due to performance issues which are no longer a problem.

```
<property>
  <name>hbase.hstore.checksum.algorithm</name>
  <value>CRC32</value>
  <description>
    Name of an algorithm that is used to compute checksums. Possible values
    are NULL, CRC32, CRC32C.
  </description>
</property>
```

Upgrading HBase



Note: To see which version of HBase is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).



Important: Before you start, make sure you have read and understood the previous section, [New Features and Changes for HBase in CDH 5](#) on page 228, and check the [Known Issues in CDH 5](#) and [Incompatible Changes](#) for HBase.

Coprocessors and Custom JARs

When upgrading HBase from one major version to another (such as upgrading from CDH 4 to CDH 5), you must recompile coprocessors and custom JARs *after* the upgrade.

Never rely on HBase directory layout on disk.

The HBase directory layout is an implementation detail and is subject to change. Do not rely on the directory layout for client or administration functionality. Instead, access HBase using the supported APIs.

Upgrading HBase from a Lower CDH 5 Release



Important: Rolling upgrade is not supported between a CDH 5 Beta release and a CDH 5 GA release. Cloudera recommends using Cloudera Manager if you need to do rolling upgrades.

To upgrade HBase from a lower CDH 5 release, proceed as follows.

The instructions that follow assume that you are upgrading HBase as part of an upgrade to the latest CDH 5 release, and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

Step 1: Perform a Graceful Cluster Shutdown



Note: Upgrading using rolling restart is not supported.

To shut HBase down gracefully:

1. Stop the Thrift server and clients, then stop the cluster.

- a. Stop the Thrift server and clients:

```
sudo service hbase-thrift stop
```

- b. Stop the cluster by shutting down the master and the region servers:

- Use the following command on the master node:

```
sudo service hbase-master stop
```

- Use the following command on each node hosting a region server:

```
sudo service hbase-regionserver stop
```

2. Stop the ZooKeeper Server:

```
$ sudo service zookeeper-server stop
```

Step 2: Install the new version of HBase



Note: You may want to take this opportunity to upgrade ZooKeeper, but you do not *have* to upgrade Zookeeper before upgrading HBase; the new version of HBase will run with the older version of Zookeeper. For instructions on upgrading ZooKeeper, see [Upgrading ZooKeeper from an Earlier CDH 5 Release](#) on page 386.

To install the new version of HBase, follow directions in the next section, [Installing HBase](#) on page 238.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Installing HBase

To install HBase On Red Hat-compatible systems:

```
$ sudo yum install hbase
```

To install HBase on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase
```

To install HBase on SLES systems:

```
$ sudo zypper install hbase
```



Note: See also [Starting HBase in Standalone Mode](#) on page 245, [Configuring HBase in Pseudo-Distributed Mode](#) on page 247, and [Deploying HBase on a Cluster](#) on page 250 for more information on configuring HBase for different modes.

To list the installed files on Ubuntu and Debian systems:

```
$ dpkg -L hbase
```

To list the installed files on Red Hat and SLES systems:

```
$ rpm -ql hbase
```

You can see that the HBase package has been configured to conform to the Linux Filesystem Hierarchy Standard. (To learn more, run `man hier`).

You are now ready to enable the server daemons you want to use with Hadoop. You can also enable Java-based client access by adding the JAR files in `/usr/lib/hbase/` and `/usr/lib/hbase/lib/` to your Java class path.

Configuration Settings for HBase

This section contains information on configuring the Linux host and HDFS for HBase.

Using DNS with HBase

HBase uses the local hostname to report its IP address. Both forward and reverse DNS resolving should work. If your server has multiple interfaces, HBase uses the interface that the primary hostname resolves to. If this is insufficient, you can set `hbase.regionserver.dns.interface` in the `hbase-site.xml` file to indicate the primary interface. To work properly, this setting requires that your cluster configuration is consistent and every host has the same network interface configuration. As an alternative, you can set `hbase.regionserver.dns.nameserver` in the `hbase-site.xml` file to use a different DNS name server than the system-wide default.

Using the Network Time Protocol (NTP) with HBase

The clocks on cluster members must be synchronized for your cluster to function correctly. Some skew is tolerable, but excessive skew could generate odd behaviors. Run NTP or another clock synchronization mechanism on your cluster. If you experience problems querying data or unusual cluster operations, verify the system time. For more information about NTP, see the [NTP website](#).

Setting User Limits for HBase

Because HBase is a database, it opens many files at the same time. The default setting of 1024 for the maximum number of open files on most Unix-like systems is insufficient. Any significant amount of loading will result in failures and cause error message such as `java.io.IOException...(Too many open files)` to be logged in the HBase or HDFS log files. For more information about this issue, see the [Apache HBase Book](#). You may also notice errors such as:

```
2010-04-06 03:04:37,542 INFO org.apache.hadoop.hdfs.DFSClient: Exception
increaseBlockOutputStream java.io.EOFException
2010-04-06 03:04:37,542 INFO org.apache.hadoop.hdfs.DFSClient: Abandoning block
blk_-6935524980745310745_1391901
```

Another setting you should configure is the number of processes a user is permitted to start. The default number of processes is typically 1024. Consider raising this value if you experience `OutOfMemoryException` errors.

Configuring ulimit for HBase

Cloudera recommends increasing the maximum number of file handles to more than 10,000. Note that increasing the file handles for the user who is running the HBase process is an operating system configuration, not an HBase configuration. Also, a common mistake is to increase the number of file handles for a particular user but, for whatever reason, HBase will be running as a different user. HBase prints the `ulimit` it is using on the first line in the logs. Make sure that it is correct.

To change the maximum number of open files for a given user, use the `ulimit -n` command while logged in as that user. To set the maximum number of processes a user may start, use the `ulimit -u` command. The `ulimit` command can be used to set many other limits besides the number of open files. Refer to the online documentation for your operating system, or the output of the `man ulimit` command, for more information. To make the changes persistent, you can add the command to the user's Bash initialization file (typically `~/.bash_profile` or `~/.bashrc`). Alternatively, you can configure the settings in the Pluggable Authentication Module (PAM) configuration files if your operating system uses PAM and includes the `pam_limits.so` shared library.

Configuring ulimit using Pluggable Authentication Modules

If you are using `ulimit`, you must make the following configuration changes:

1. In the `/etc/security/limits.conf` file, add the following lines, adjusting the values as appropriate. This assumes that your HDFS user is called `hdfs` and your HBase user is called `hbase`.

```
hdfs -      nofile 32768
hdfs -      nproc  2048
hbase -     nofile 32768
hbase -     nproc  2048
```

**Note:**

- Only the `root` user can edit this file.
- If this change does not take effect, check other configuration files in the `/etc/security/limits.d/` directory for lines containing the `hdfs` or `hbase` user and the `nofile` value. Such entries may be overriding the entries in `/etc/security/limits.conf`.

To apply the changes in `/etc/security/limits.conf` on Ubuntu and Debian systems, add the following line in the `/etc/pam.d/common-session` file:

```
session required pam_limits.so
```

For more information on the `ulimit` command or per-user operating system limits, refer to the documentation for your operating system.

Using `dfs.datanode.max.transfer.threads` with HBase

A Hadoop HDFS DataNode has an upper bound on the number of files that it can serve at any one time. The upper bound is controlled by the `dfs.datanode.max.transfer.threads` property (the property is spelled in the code exactly as shown here). Before loading, make sure you have configured the value for `dfs.datanode.max.transfer.threads` in the `conf/hdfs-site.xml` file (by default found in `/etc/hadoop/conf/hdfs-site.xml`) to at least 4096 as shown below:

```
<property>
  <name>dfs.datanode.max.transfer.threads</name>
  <value>4096</value>
</property>
```

Restart HDFS after changing the value for `dfs.datanode.max.transfer.threads`. If the value is not set to an appropriate value, strange failures can occur and an error message about exceeding the number of transfer threads will be added to the DataNode logs. Other error messages about missing blocks are also logged, such as:

```
06/12/14 20:10:31 INFO hdfs.DFSCClient: Could not obtain block
blk_XXXXXXXXXXXXXXXXXXXXXXXXX_YYYYYYYY from any node:
java.io.IOException: No live nodes contain current block. Will get new block locations
from namenode and retry...
```



Note: The property `dfs.datanode.max.transfer.threads` is a HDFS 2 property which replaces the deprecated property `dfs.datanode.max.xcievers`.

Configuring BucketCache in HBase

The default `BlockCache` implementation in HBase is `CombinedBlockCache`, and the default off-heap `BlockCache` is `BucketCache`. `SlabCache` is now deprecated. See [Configuring the HBase BlockCache](#) on page 241 for information about configuring the `BlockCache` using Cloudera Manager or the command line.

Configuring Encryption in HBase

It is possible to encrypt the HBase root directory within HDFS, using [HDFS Data At Rest Encryption](#). This provides an additional layer of protection in case the HDFS filesystem is compromised.

If you use this feature in combination with bulk-loading of HFiles, you must configure `hbase.bulkload.staging.dir` to point to a location within the same encryption zone as the HBase root directory. Otherwise, you may encounter errors such as:

```
org.apache.hadoop.ipc.RemoteException(java.io.IOException):
/tmp/output/f/5237a8430561409bb641507f0c531448 can't be moved into an encryption zone.
```


You can also choose to only encrypt specific column families, which encrypts individual HFiles while leaving others unencrypted, using [HBase Transparent Encryption at Rest](#). This provides a balance of data security and performance.

Checksums in CDH 5

The default values for checksums have changed during the history of CDH 5. For information about configuring checksums, see [New Features and Changes for HBase in CDH 5](#) on page 228.

Configuring the HBase BlockCache

HBase provides both on-heap and off-heap BlockCache implementations.

- **On-heap:** The default on-heap BlockCache implementation is `LruBlockCache`. You can optionally use `BucketCache` on-heap as well as off-heap.
- **Combined:** If your operations use more data than will fit into the heap, you can use the `BucketCache` as a L2 cache for the on-heap `LruBlockCache`. This implementation is referred to as `CombinedBlockCache`.

Contents of the BlockCache

In order to configure the BlockCache, you need to understand its contents.

- **Your data:** Each time a Get or Scan operation occurs, the result is added to the BlockCache if it was not already there.
- **Row keys:** When a value is loaded into the cache, its row key is also cached. This is one reason to make your row keys as small as possible. A larger row key takes up more space in the cache.
- **hbase:meta:** The `hbase:meta` catalog table keeps track of which RegionServer is serving which regions. It can consume several megabytes of cache if you have a large number of regions, and has `in-memory` access priority, which means HBase attempts to keep it in the cache as long as possible.
- **Indexes of HFiles:** HBase stores its data in HDFS in a format called *HFile*. These HFiles contain indexes which allow HBase to seek for data within them without needing to open the entire HFile. The size of an index is a factor of the block size, the size of your row keys, and the amount of data you are storing. For big data sets, the size can exceed 1 GB per RegionServer, although the entire index is unlikely to be in the cache at the same time. If you use the `BucketCache`, indexes are always cached on-heap.
- **Bloom filters:** If you use Bloom filters, they are stored in the BlockCache.

All of the sizes of these objects are highly dependent on your usage patterns and the characteristics of your data. For this reason, the HBase Web UI and Cloudera Manager each expose several metrics to help you size and tune the BlockCache.

Choosing a BlockCache Configuration

The HBase team has published the [results of exhaustive BlockCache testing](#), which revealed the following guidelines.

- If the data set fits completely in cache, the default configuration, which uses the on-heap `LruBlockCache`, is the best choice. If the eviction rate is low, garbage collection is 50% less that of the `CombinedBlockCache`, and throughput is at least 20% higher.
- Otherwise, if your cache is experiencing a consistently high eviction rate, use `CombinedBlockCache`, which causes 30-50% of the garbage collection of `LruBlockCache` when the eviction rate is high.
- `CombinedBlockCache` using *file mode* on solid-state disks has a better garbage-collection profile but lower throughput than `CombinedBlockCache` using *off-heap mode*.

Bypassing the BlockCache

If the data needed for an operation does not all fit in memory, using the BlockCache can be counter-productive, because data that you are still using may be evicted, or even if other data is evicted, excess garbage collection can adversely effect performance. For this type of operation, you may decide to bypass the BlockCache. To bypass the BlockCache for a given Scan or Get, use the `setCacheBlocks(false)` method.

In addition, you can prevent a specific column family's contents from being cached, by setting its `BLOCKCACHE` configuration to `false`. Use the following syntax in HBase Shell:

```
hbase> alter 'myTable', CONFIGURATION => {NAME => 'myCF', BLOCKCACHE => 'false'}
```

About the `LruBlockCache`

The `LruBlockCache` implementation resides entirely within the Java heap. Fetching from `LruBlockCache` will always be faster than `BucketCache`, because no disk seeks are required. However, `LruBlockCache` is more impacted by garbage-collection and performance can be less predictable over time than `BucketCache`.

`LruBlockCache` contains three levels of block priority to allow for scan-resistance and in-memory column families:

- **Single access priority:** The first time a block is loaded from HDFS, that block is given single access priority, which means that it will be part of the first group to be considered during evictions. Scanned blocks are more likely to be evicted than blocks that are used more frequently.
- **Multi access priority:** If a block in the single access priority group is accessed again, that block is assigned multi access priority, which moves it to the second group considered during evictions, and is therefore less likely to be evicted.
- **In-memory access priority:** If the block belongs to a column family which is configured with the `in-memory` configuration option, its priority is changed to in memory access priority, regardless of its access pattern. This group is the last group considered during evictions, but is not guaranteed not to be evicted. Catalog tables are configured with in-memory access priority.

To configure a column family for in-memory access, use the following syntax in HBase Shell:

```
hbase> alter 'myTable', 'myCF', CONFIGURATION => {IN_MEMORY => 'true'}
```

To use the Java API to configure a column family for in-memory access, use the `HColumnDescriptor.setInMemory(true)` method.

Configuring the `LruBlockCache`

When you use the `LruBlockCache`, the blocks needed to satisfy each read are cached, evicting older cached objects if the `LruBlockCache` is full. The size cached objects for a given read may be significantly larger than the actual result of the read. For instance, if HBase needs to scan through 20 HFile blocks to return a 100 byte result, and the HFile blocksize is 100 KB, the read will add $20 * 100$ KB to the `LruBlockCache`.

Because the `LruBlockCache` resides entirely within the Java heap, the amount of which is available to HBase and what percentage of the heap is available to the `LruBlockCache` strongly impact performance. By default, the amount of HBase's heap reserved for the `LruBlockCache` (`hfile.block.cache.size`) is `.25`, or 25%. To determine the amount of memory available to HBase, use the following formula. The `0.99` factor allows 1% of heap to be available for evicting items from the cache.

```
number of RegionServers * heap size * hfile.block.cache.size * 0.99
```

To tune the size of the `LruBlockCache`, you can add `RegionServers` to increase it, or you can tune `hfile.block.cache.size` to reduce it. Reducing it will cause cache evictions to happen more often.

About the `BucketCache`

The `BucketCache` stores cached objects in different "buckets", based upon the sizes of the objects. By default, the buckets are all the same size, controlled by the configuration property `hfile.bucketcache.size`. However, you can configure multiple bucket sizes if your data fits into different, predictable size categories, using the `hfile.bucketcache.sizes` property instead, which takes a comma-separated list of sizes as its value. Evictions are managed independently per bucket.

The physical location of the `BucketCache` storage can be either in memory (off-heap) or in a file stored in a fast disk.

- **Off-heap:** This is the default configuration, where the `BucketCache` is used as an L2 cache for the on-heap `LruBlockCache`.
- **File-based:** You can use the file-based storage mode to store the `BucketCache` on an SSD or FusionIO device, giving the `LruBlockCache` a faster L2 cache to work with.

Configuring the `BucketCache`

This table summarizes the important configuration properties for the `BucketCache`. To configure the `BucketCache`, see [Configuring the BucketCache Using Cloudera Manager](#) on page 244 or [Configuring the BucketCache Using the Command Line](#) on page 245

Table 24: BucketCache Configuration Properties

Property	Default	Description
<code>hbase.bucketcache.combinedcache.enabled</code>	true	When <code>BucketCache</code> is enabled, use it as a L2 cache for <code>LruBlockCache</code> . If set to true, indexes and Bloom filters are kept in the <code>LruBlockCache</code> and the data blocks are kept in the <code>BucketCache</code>
<code>hbase.bucketcache.ioengine</code>	none (<code>BucketCache</code> is disabled by default)	Where to store the contents of the <code>BucketCache</code> . One of: <code>onheap</code> , <code>offheap</code> , or <code>file:/path/to/file</code> .
<code>hfile.block.cache.size</code>	0.4	A float between 0.0 and 1.0. This factor multiplied by the Java heap size is the size of the L1 cache.
<code>hfile.bucketcache.size</code>	0	Either the size of the <code>BucketCache</code> (if expressed as an integer) or the percentage of the total heap to use for the <code>BucketCache</code> , if expressed as a float between 0.0 and 1.0. See hbase.bucketcache.percentage.in.combinedcache . A simplified configuration is planned for HBase 1.0.
<code>hbase.bucketcache.percentage.in.combinedcache</code>	none	In HBase 0.98, this property controls the percentage of the <code>CombinedBlockCache</code> which will be used by the <code>BucketCache</code> . The <code>LruBlockCache L1</code> size is calculated to be $(1 - \text{hbase.bucketcache.percentage.in.combinedcache}) * \text{size-of-bucketcache}$ and the <code>BucketCache</code> size is $\text{hbase.bucketcache.percentage.in.combinedcache} * \text{size-of-bucket-cache}$. where <code>size-of-bucket-cache</code> itself is either the value of the configuration <code>hbase.bucketcache.size</code> if it is specified in megabytes, or <code>hbase.bucketcache.size * -XX:MaxDirectMemorySize</code> if <code>hbase.bucketcache.size</code> is between 0 and 1.0.

Property	Default	Description
		A simplified configuration is planned for HBase 1.0.
<code>hbase.bucketcache.bucket.sizes</code>	4, 8, 16, 32, 40, 48, 56, 64, 96, 128, 192, 256, 384, 512 KB	A comma-separated list of sizes for buckets for the <code>BucketCache</code> if you prefer to use multiple sizes. The sizes should be multiples of the default blocksize, ordered from smallest to largest. The sizes you use will depend on your data patterns. This parameter is experimental.
<code>-XX:MaxDirectMemorySize</code>	<code>MaxDirectMemorySize = BucketCache + 1</code>	A JVM option to configure the maximum amount of direct memory available to the JVM. You do not need to manually configure this parameter. It is automatically calculated and configured based on the following formula: <code>MaxDirectMemorySize = BucketCache + 1</code>

Configuring the BucketCache Using Cloudera Manager

1. In the Cloudera Manager UI, go to **Clusters > Services > HBase**. Go to **Configuration**.
2. Search for the setting **Java Heap Size of HBase RegionServer** in Bytes and set a value higher than the desired size of your `BucketCache`. For instance, if you want a `BlockCache` size of 4 GB, you might set the heap size to 5 GB. This accommodates a 4 GB `BucketCache` with 1 GB left over.
3. Edit the parameter `HBASE_OPTS` in the **HBase Service Advanced Configuration Snippet for `hbase-env.sh`** and add the JVM option `-XX:MaxDirectMemorySize=<size>G`, replacing `<size>` with a value large enough to contain your heap and off-heap `BucketCache`, expressed as a number of gigabytes.
4. Add the following settings to the **HBase Service Advanced Configuration Snippet for `hbase-site.xml`**, using values appropriate to your situation. See [Table 24: BucketCache Configuration Properties](#) on page 243. This example configures the L1 cache to use 20% of the heap (5 GB x 20% is 1 GB), and configures the `BucketCache` to use 4 GB. Because `hbase.bucketcache.combinedcache.enabled` defaults to `true`, this configuration uses the `CombinedBlockCache`.

```
<property>
  <name>hbase.bucketcache.ioengine</name>
  <value>offheap</value>
</property>
<property>
  <name>hfile.block.cache.size</name>
  <value>0.2</value>
</property>
<property>
  <name>hbase.bucketcache.size</name>
  <value>4196</value>
</property>
```

5. Restart or rolling restart your `RegionServers` for the changes to take effect.

Configuring the BucketCache Using the Command Line

**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.3.x . If you use an earlier version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

1. First, configure the off-heap cache size for each RegionServer by editing its hbase-env.sh file and adding a line like the following:

```
HBASE_OFFHEAPSIZE=5G
```

This value sets the total heap size for a given RegionServer.

2. Next, configure the properties in [Table 24: BucketCache Configuration Properties](#) on page 243 as appropriate. The following example configures the L1 cache to use 20% of the heap (5 GB x 20 % is 1 GB), and configures the BucketCache to use 4 GB. Because `hbase.bucketcache.combinedcache.enabled` defaults to `true`, this configuration uses the `CombinedBlockCache`.

```
<property>
  <name>hbase.bucketcache.ioengine</name>
  <value>offheap</value>
</property>
<property>
  <name>hfile.block.cache.size</name>
  <value>0.2</value>
</property>
<property>
  <name>hbase.bucketcache.size</name>
  <value>4196</value>
</property>
```

3. Restart each RegionServer for the changes to take effect.

Monitoring the BlockCache

Cloudera Manager provides metrics to monitor the performance of the `BlockCache`, to assist you in tuning your configuration. See [HBase Metrics](#).

You can view further detail and graphs using the RegionServer UI. To access the RegionServer UI in Cloudera Manager, go to the Cloudera Manager page for the host, click the **RegionServer** process, and click **HBase RegionServer Web UI**.

If you do not use Cloudera Manager, access the `BlockCache` reports at

`http://regionServer_host:22102/rs-status#memoryStats`, replacing `regionServer_host` with the hostname or IP address of your RegionServer.

Starting HBase in Standalone Mode

**Note:**

You can skip this section if you are already running HBase in distributed or pseudo-distributed mode.

By default, HBase ships configured for *standalone mode*. In this mode of operation, a single JVM hosts the HBase Master, an HBase Region Server, and a ZooKeeper quorum peer. HBase stores your data in a location on the local filesystem, rather than using HDFS. Standalone mode is only appropriate for initial testing.



Important:

If you have configured [High Availability for the NameNode \(HA\)](#), you cannot deploy HBase in standalone mode without modifying the default configuration, because both the standalone HBase process and ZooKeeper (required by HA) will try to bind to port 2181. You can configure a different port for ZooKeeper, but in most cases it makes more sense to deploy HBase in distributed mode in an HA cluster.

In order to run HBase in standalone mode, you must install the HBase Master package.

Installing the HBase Master

To install the HBase Master on Red Hat-compatible systems:

```
$ sudo yum install hbase-master
```

To install the HBase Master on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-master
```

To install the HBase Master on SLES systems:

```
$ sudo zypper install hbase-master
```

Starting the HBase Master

- On Red Hat and SLES systems (using `.rpm` packages) you can now start the HBase Master by using the included service script:

```
$ sudo service hbase-master start
```

- On Ubuntu systems (using Debian packages) the HBase Master starts when the HBase package is installed.

To verify that the standalone installation is operational, visit `http://localhost:60010`. The list of RegionServers at the bottom of the page should include one entry for your local machine.



Note:

Although you have only started the master process, in *standalone* mode this same process is also internally running a region server and a ZooKeeper peer. In the next section, you will break out these components into separate JVMs.

If you see this message when you start the HBase standalone master:

```
Starting Hadoop HBase master daemon: starting master, logging to
/usr/lib/hbase/logs/hbase-hbase-master/cloudera-vm.out
Couldnt start ZK at requested address of 2181, instead got: 2182. Aborting. Why? Because
clients (eg shell) wont be able to find this ZK quorum
hbase-master.
```

you will need to stop the `hadoop-zookeeper-server` (or `zookeeper-server`) or uninstall the `hadoop-zookeeper-server` (or `zookeeper`) package.

See also [Accessing HBase by using the HBase Shell](#) on page 251, [Using MapReduce with HBase](#) on page 252 and [Troubleshooting HBase](#) on page 252.

*Installing and Starting the HBase Thrift Server***To install Thrift on Red Hat-compatible systems:**

```
$ sudo yum install hbase-thrift
```

To install Thrift on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-thrift
```

To install Thrift on SLES systems:

```
$ sudo zypper install hbase-thrift
```

You can now use the `service` command to start the Thrift server:

```
$ sudo service hbase-thrift start
```

*Installing and Configuring HBase REST***To install HBase REST on Red Hat-compatible systems:**

```
$ sudo yum install hbase-rest
```

To install HBase REST on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-rest
```

To install HBase REST on SLES systems:

```
$ sudo zypper install hbase-rest
```

You can use the `service` command to run an `init.d` script, `/etc/init.d/hbase-rest`, to start the REST server; for example:

```
$ sudo service hbase-rest start
```

The script starts the server by default on port 8080. This is a commonly used port and so may conflict with other applications running on the same host.

If you need change the port for the REST server, configure it in `hbase-site.xml`, for example:

```
<property>
  <name>hbase.rest.port</name>
  <value>60050</value>
</property>
```

**Note:**

You can use `HBASE_REST_OPTS` in `hbase-env.sh` to pass other settings (such as heap size and GC parameters) to the REST server JVM.

Configuring HBase in Pseudo-Distributed Mode

Note: You can skip this section if you are already running HBase in distributed mode, or if you intend to use only standalone mode.

Pseudo-distributed mode differs from *standalone* mode in that each of the component processes run in a separate JVM. It differs from *distributed mode* in that each of the separate processes run on the same server, rather than multiple servers in a cluster. This section also assumes you wish to store your HBase data in HDFS rather than on the local filesystem.



Note: Before you start

- This section assumes you have already installed the [HBase master](#) and gone through the [standalone](#) configuration steps.
- If the HBase master is already running in standalone mode, stop it as follows before continuing with pseudo-distributed configuration:
- To stop the CDH 4 version: `sudo service hadoop-hbase-master stop`, *or*
- To stop the CDH 5 version if that version is already running: `sudo service hbase-master stop`

Modifying the HBase Configuration

To enable pseudo-distributed mode, you must first make some configuration changes. Open `/etc/hbase/conf/hbase-site.xml` in your editor of choice, and insert the following XML properties between the `<configuration>` and `</configuration>` tags. The `hbase.cluster.distributed` property directs HBase to start each process in a separate JVM. The `hbase.rootdir` property directs HBase to store its data in an HDFS filesystem, rather than the local filesystem. Be sure to replace `myhost` with the hostname of your HDFS NameNode (as specified by `fs.default.name` or `fs.defaultFS` in your `conf/core-site.xml` file); you may also need to change the port number from the default (8020).

```
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://myhost:8020/hbase</value>
</property>
```

Creating the `/hbase` Directory in HDFS

Before starting the HBase Master, you need to create the `/hbase` directory in HDFS. The HBase master runs as `hbase:hbase` so it does not have the required permissions to create a top level directory.

To create the `/hbase` directory in HDFS:

```
$ sudo -u hdfs hadoop fs -mkdir /hbase
$ sudo -u hdfs hadoop fs -chown hbase /hbase
```



Note: If [Kerberos is enabled](#), do not use commands in the form `sudo -u <user> hadoop <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>`

Enabling Servers for Pseudo-distributed Operation

After you have configured HBase, you must enable the various servers that make up a distributed HBase cluster. HBase uses three required types of servers:

- [Installing and Starting ZooKeeper Server](#)
- [Starting the HBase Master](#)
- [Starting an HBase RegionServer](#)

Installing and Starting ZooKeeper Server

HBase uses ZooKeeper Server as a highly available, central location for cluster management. For example, it allows clients to locate the servers, and ensures that only one master is active at a time. For a small cluster, running a ZooKeeper node collocated with the NameNode is recommended. For larger clusters, contact Cloudera Support for configuration help.

Install and start the ZooKeeper Server in standalone mode by running the commands shown in the [Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server](#) on page 387

Starting the HBase Master

After ZooKeeper is running, you can start the HBase master in standalone mode.

```
$ sudo service hbase-master start
```

Starting an HBase RegionServer

The RegionServer is the HBase process that actually hosts data and processes requests. The RegionServer typically runs on all HBase nodes except for the node running the HBase master node.

To enable the HBase RegionServer On Red Hat-compatible systems:

```
$ sudo yum install hbase-regionserver
```

To enable the HBase RegionServer on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-regionserver
```

To enable the HBase RegionServer on SLES systems:

```
$ sudo zypper install hbase-regionserver
```

To start the RegionServer:

```
$ sudo service hbase-regionserver start
```

Verifying the Pseudo-Distributed Operation

After you have started ZooKeeper, the Master, and a RegionServer, the pseudo-distributed cluster should be up and running. You can verify that each of the daemons is running using the `jps` tool from the Oracle JDK, which you can obtain from [here](#). If you are running a pseudo-distributed HDFS installation and a pseudo-distributed HBase installation on one machine, `jps` will show the following output:

```
$ sudo jps
32694 Jps
30674 HRegionServer
29496 HMaster
28781 DataNode
28422 NameNode
30348 QuorumPeerMain
```

You should also be able to navigate to `http://localhost:60010` and verify that the local RegionServer has registered with the Master.

Installing and Starting the HBase Thrift Server

The HBase Thrift Server is an alternative gateway for accessing the HBase server. Thrift mirrors most of the HBase client APIs while enabling popular programming languages to interact with HBase. The Thrift Server is multiplatform and more performant than REST in many situations. Thrift can be run collocated along with the RegionServers, but

should not be collocated with the NameNode or the JobTracker. For more information about Thrift, visit <http://thrift.apache.org/>.

To enable the HBase Thrift Server On RHEL-compatible systems:

```
$ sudo yum install hbase-thrift
```

To enable the HBase Thrift Server on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-thrift
```

To enable the HBase Thrift Server on SLES systems:

```
$ sudo zypper install hbase-thrift
```

To start the Thrift server:

```
$ sudo service hbase-thrift start
```

See also [Accessing HBase by using the HBase Shell](#) on page 251, [Using MapReduce with HBase](#) on page 252 and [Troubleshooting HBase](#) on page 252.

Deploying HBase on a Cluster

After you have HBase running in pseudo-distributed mode, the same configuration can be extended to running on a distributed cluster.



Note: Before you start

This section assumes that you have already installed the [HBase Master](#) and the [HBase Region Server](#) and gone through the steps for [standalone](#) and [pseudo-distributed](#) configuration. You are now about to distribute the processes across multiple hosts; see [Choosing Where to Deploy the Processes](#) on page 250.

Choosing Where to Deploy the Processes

For small clusters, Cloudera recommends designating one node in your cluster as the HBase Master node. On this node, you will typically run the HBase Master and a ZooKeeper quorum peer. These master processes may be collocated with the Hadoop NameNode and JobTracker for small clusters.

Designate the remaining nodes as RegionServer nodes. On each node, Cloudera recommends running a Region Server, which may be collocated with a Hadoop TaskTracker (MRv1) and a DataNode. When co-locating with TaskTrackers, be sure that the resources of the machine are not oversubscribed – it's safest to start with a small number of MapReduce slots and work up slowly.

Configuring for Distributed Operation

After you have decided which machines will run each process, you can edit the configuration so that the nodes can locate each other. In order to do so, you should make sure that the configuration files are synchronized across the cluster. Cloudera strongly recommends the use of a configuration management system to synchronize the configuration files, though you can use a simpler solution such as `rsync` to get started quickly.

The only configuration change necessary to move from pseudo-distributed operation to fully-distributed operation is the addition of the ZooKeeper Quorum address in `hbase-site.xml`. Insert the following XML property to configure the nodes with the address of the node where the ZooKeeper quorum peer is running:

```
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>mymasternode</value>
</property>
```

The `hbase.zookeeper.quorum` property is a comma-separated list of hosts on which ZooKeeper servers are running. If one of the ZooKeeper servers is down, HBase will use another from the list. By default, the ZooKeeper service is bound to port 2181. To change the port, add the `hbase.zookeeper.property.clientPort` property to `hbase-site.xml` and set the value to the port you want ZooKeeper to use. For more information, see [this chapter](#) of the Apache HBase Reference Guide.

To start the cluster, start the services in the following order:

1. The ZooKeeper Quorum Peer
2. The HBase Master
3. Each of the HBase RegionServers

After the cluster is fully started, you can view the HBase Master web interface on port 60010 and verify that each of the RegionServer nodes has registered properly with the master.

See also [Accessing HBase by using the HBase Shell](#) on page 251, [Using MapReduce with HBase](#) on page 252 and [Troubleshooting HBase](#) on page 252. For instructions on improving the performance of local reads, see [Optimizing Performance in CDH](#).

Accessing HBase by using the HBase Shell

After you have started HBase, you can access the database in an interactive way by using the HBase Shell, which is a command interpreter for HBase which is written in Ruby. Always run HBase administrative commands such as the HBase Shell, `hbck`, or `bulk-load` commands as the HBase user (typically `hbase`).

```
$ hbase shell
```

Setting Virtual Machine Options for HBase Shell

HBase in CDH 5.2 and newer allows you to set variables for the virtual machine running HBase Shell, by using the `HBASE_SHELL_OPTS` environment variable. This example sets several options in the virtual machine.

```
$ HBASE_SHELL_OPTS="-verbose:gc -XX:+PrintGCApplicationStoppedTime -XX:+PrintGCDateStamps
-XX:+PrintGCDetails -Xloggc:$HBASE_HOME/logs/gc-hbase.log" ./bin/hbase shell
```

Scripting with HBase Shell

A new feature of HBase Shell in CDH 5.2 is non-interactive mode. This mode allows you to use HBase Shell in scripts, and allow the script to access the exit status of the HBase Shell commands. To invoke non-interactive mode, use the `-n` or `--non-interactive` switch. This small example script shows how to use HBase Shell in a Bash script.

```
#!/bin/bash
echo 'list' | hbase shell -n
status=$?
if [ $status -ne 0 ]; then
    echo "The command may have failed."
fi
```

Successful HBase Shell commands return an exit status of 0. However, an exit status other than 0 does not necessarily indicate a failure, but should be interpreted as unknown. For example, a command may succeed, but while waiting for the response, the client may lose connectivity. In that case, the client has no way to know the outcome of the command. In the case of a non-zero exit status, your script should check to be sure the command actually failed before taking further action.

You can also write Ruby scripts for use with HBase Shell. Example Ruby scripts are included in the `hbase-examples/src/main/ruby/` directory.

HBase Online Merge

CDH 5 supports online merging of regions. HBase splits big regions automatically but does not support merging small regions automatically. To complete an online merge of two regions of a table, you need to use the HBase shell to issue

the online merge command. By default, both regions to be merged should be neighbors, that is, one end key of a region should be the start key of the other region. Even though you can "force" merge any two regions of the same table, this is not recommended as it could create overlaps.

The Master and RegionServer both participate in online merges. When the request to merge is sent to the Master, the Master moves the regions to be merged to the same RegionServer, usually the one where the region with the higher load resides. The Master then requests the RegionServer to merge the two regions. The RegionServer processes this request locally. Once the two regions are merged, the new region will be online and available for server requests while the old regions will be taken offline.

For merging two consecutive regions use the following command:

```
hbase> merge_region 'ENCODED_REGIONNAME', 'ENCODED_REGIONNAME'
```

For merging regions that are not adjacent, passing `true` as the third parameter will force the merge.

```
hbase> merge_region 'ENCODED_REGIONNAME', 'ENCODED_REGIONNAME', true
```

Using MapReduce with HBase

To run MapReduce jobs that use HBase, you need to add the HBase and Zookeeper JAR files to the Hadoop Java classpath. You can do this by adding the following statement to each job:

```
TableMapReduceUtil.addDependencyJars(job);
```

This distributes the JAR files to the cluster along with your job and adds them to the job's classpath, so that you do not need to edit the MapReduce configuration.

You can find more information about `addDependencyJars` in the documentation listed under [Viewing the HBase Documentation](#) on page 254.

When getting an `Configuration` object for a HBase MapReduce job, instantiate it using the `HBaseConfiguration.create()` method.

Troubleshooting HBase

The Cloudera HBase packages have been configured to place logs in `/var/log/hbase`. Cloudera recommends tailing the `.log` files in this directory when you start HBase to check for any error messages or failures.

Table Creation Fails after Installing LZO

If you install LZO after starting the Region Server, you will not be able to create a table with LZO compression until you re-start the Region Server.

Why this happens

When the Region Server starts, it runs `CompressionTest` and caches the results. When you try to create a table with a given form of compression, it refers to those results. You have installed LZO since starting the Region Server, so the cached results, which pre-date LZO, cause the create to fail.

What to do

Restart the Region Server. Now table creation with LZO will succeed.

Thrift Server Crashes after Receiving Invalid Data

The Thrift server may crash if it receives a large amount of invalid data, due to a buffer overrun.

Why this happens

The Thrift server allocates memory to check the validity of data it receives. If it receives a large amount of invalid data, it may need to allocate more memory than is available. This is due to a limitation in the Thrift library itself.

What to do

To prevent the possibility of crashes due to buffer overruns, use the framed and compact transport protocols. These protocols are disabled by default, because they may require changes to your client code. The two options to add to your `hbase-site.xml` are `hbase.regionserver.thrift.framed` and `hbase.regionserver.thrift.compact`. Set each of these to `true`, as in the XML below. You can also specify the maximum frame size, using the `hbase.regionserver.thrift.framed.max_frame_size_in_mb` option.

```
<property>
  <name>hbase.regionserver.thrift.framed</name>
  <value>true</value>
</property>
<property>
  <name>hbase.regionserver.thrift.framed.max_frame_size_in_mb</name>
  <value>2</value>
</property>
<property>
  <name>hbase.regionserver.thrift.compact</name>
  <value>true</value>
</property>
```

HBase is using more disk space than expected.

HBase StoreFiles (also called HFiles) store HBase row data on disk. HBase stores other information on disk, such as write-ahead logs (WALs), snapshots, data that would otherwise be deleted but would be needed to restore from a stored snapshot.



Warning: The following information is provided to help you troubleshoot high disk usage only. Do not edit or remove any of this data outside the scope of the HBase APIs or HBase Shell, or your data is very likely to become corrupted.

Table 25: HBase Disk Usage

Location	Purpose	Troubleshooting Notes
<code>/hbase/.snapshots</code>	Contains one subdirectory per snapshot.	To list snapshots, use the HBase Shell command <code>list_snapshots</code> . To remove a snapshot, use <code>delete_snapshot</code> .
<code>/hbase/.archive</code>	Contains data that would otherwise have been deleted (either because it was explicitly deleted or expired due to TTL or version limits on the table) but that is required to restore from an existing snapshot.	To free up space being taken up by excessive archives, delete the snapshots that refer to them. Snapshots never expire so data referred to by them is kept until the snapshot is removed. Do not remove anything from <code>/hbase/.archive</code> manually, or you will corrupt your snapshots.
<code>/hbase/.logs</code>	Contains HBase WAL files that are required to recover regions in the event of a RegionServer failure.	WALs are removed when their contents are verified to have been written to StoreFiles. Do not remove them manually. If the size of any subdirectory of <code>/hbase/.logs/</code> is growing, examine the HBase server logs to find the root cause for why WALs are not being processed correctly.
<code>/hbase/logs/.oldWALs</code>	Contains HBase WAL files that have already been written to disk. A HBase	To tune the length of time a WAL stays in the <code>.oldWALs</code> before it is removed,

Location	Purpose	Troubleshooting Notes
	maintenance thread removes them periodically based on a TTL.	configure the <code>hbase.master.logcleaner.ttl</code> property, which defaults to 60000 milliseconds, or 1 hour.
<code>/hbase/.logs/.corrupt</code>	Contains corrupted HBase WAL files.	Do not remove corrupt WALs manually. If the size of any subdirectory of <code>/hbase/.logs/</code> is growing, examine the HBase server logs to find the root cause for why WALs are not being processed correctly.

Viewing the HBase Documentation

For additional HBase documentation, see <https://archive.cloudera.com/cdh5/cdh/5/hbase/>.

HCatalog Installation

As of CDH 5, HCatalog is part of Apache Hive.

HCatalog provides table data access for CDH components such as Pig, Sqoop, and MapReduce. Table definitions are maintained in the Hive metastore, which HCatalog requires. HCatalog makes the same table information available to Hive, Pig, MapReduce, and REST clients. This page explains how to install and configure HCatalog for REST access and for MapReduce and Pig access. For Sqoop, see the [section on Sqoop-HCatalog integration](#) in the Sqoop User Guide.

Use the sections that follow to install, configure and use HCatalog:

- [Prerequisites](#)
- [Installing and Upgrading the HCatalog RPM or Debian Packages](#) on page 254
- [Host Configuration Changes](#)
- [Starting and Stopping the WebHCat REST Server](#)
- [Accessing Table Data with the Command-line API](#)
- [Accessing Table Data with MapReduce](#)
- [Accessing Table Data with Pig](#)
- [Accessing Table Data with REST](#)
- [Apache HCatalog Documentation](#)

You can use HCatalog to import data to HBase. See [Importing Data Into HBase](#).

For more information, see the [HCatalog documentation](#).

HCatalog Prerequisites

- An [operating system supported by CDH 5](#)
- [Oracle JDK](#)
- The Hive [metastore and its database](#). The Hive metastore must be running in [remote mode](#) (as a service).

Installing and Upgrading the HCatalog RPM or Debian Packages

Installing the HCatalog RPM or Debian packages is more convenient than installing the HCatalog tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations

HCatalog comprises the following packages:

- `hive-hcatalog` - HCatalog wrapper for accessing the Hive metastore, libraries for MapReduce and Pig, and a command-line program
- `hive-webhcat` - A REST API server for HCatalog
- `hive-webhcat-server` - Installs `hive-webhcat` and a server `init` script



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

Upgrading HCatalog from an Earlier CDH 5 Release



Important:

If you have installed the `hive-hcatalog-server` package in the past, you must remove it before you proceed; otherwise the upgrade will fail.

Follow instructions under [Installing the WebHCat REST Server](#) on page 255 and [Installing HCatalog for Use with Pig and MapReduce](#) on page 256.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

The upgrade is now complete.

Installing the WebHCat REST Server



Note:

It is not necessary to install WebHCat if you will not be using the REST API. Pig and MapReduce do not need it.

To install the WebHCat REST server on a Red Hat system:

```
$ sudo yum install hive-webhcat-server
```

To install the WebHCat REST server components on an Ubuntu or other Debian system:

```
$ sudo apt-get install hive-webhcat-server
```

To install the WebHCat REST server components on a SLES system:

```
$ sudo zypper install hive-webhcat-server
```

**Note:**

- You can change the default port 50111 by creating or editing the following file and restarting WebHCat:

```
/etc/webhcat/conf/webhcat-site.xml
```

The property to change is:

```
<configuration>
  <property>
    <name>templeton.port</name>
    <value>50111</value>
    <description>The HTTP port for the main server.</description>
  </property>
</configuration>
```

- To uninstall WebHCat you must remove two packages: `hive-webhcat-server` and `hive-webhcat`.

Installing HCatalog for Use with Pig and MapReduce

On hosts that will be used to launch Pig scripts or MapReduce applications using table information, install HCatalog as follows:

To install the HCatalog client components on a Red Hat system:

```
$ sudo yum install hive-hcatalog
```

To install the HCatalog client components on an Ubuntu or other Debian system:

```
$ sudo apt-get install hive-hcatalog
```

To install the HCatalog client components on a SLES system:

```
$ sudo zypper install hive-hcatalog
```

Configuration Change on Hosts Used with HCatalog

You must update `/etc/hive/conf/hive-site.xml` on all hosts where WebHCat will run, as well as all hosts where Pig or MapReduce will be used with HCatalog, so that Metastore clients know where to find the Metastore.

Add or edit the `hive.metastore.uris` property as follows:

```
<property>
  <name>hive.metastore.uris</name>
  <value>thrift://<hostname>:9083</value>
</property>
```

where `<hostname>` is the host where the HCatalog server components are running, for example `hive.examples.com`.

Starting and Stopping the WebHCat REST server

```
$ sudo service webhcat-server start
$ sudo service webhcat-server stop
```

Accessing Table Information with the HCatalog Command-line API

```
# Create a table
$ hcat -e "create table groups(name string,placeholder string,id int) row format delimited
fields terminated by ':' stored as textfile"
OK
```



```
# Get the schema for a table
$ hcat -e "desc groups"
OK
name string
placeholder string
id int

# Create another table
$ hcat -e "create table groupids(name string,id int)"
OK
```

See the [HCatalog documentation](#) for information on using the HCatalog command-line application.

Accessing Table Data with MapReduce

You can download an example of a MapReduce program that reads from the groups table (consisting of data from `/etc/group`), extracts the first and third columns, and inserts them into the `groupids` table. Proceed as follows.

1. Download the program from <https://github.com/cloudera/hcatalog-examples.git>.
2. Build the example JAR file:

```
$ cd hcatalog-examples
$ mvn package
```

3. Load data from the local file system into the groups table:

```
$ hive -e "load data local inpath '/etc/group' overwrite into table groups"
```

4. Set up the environment that is needed for copying the required JAR files to HDFS, for example:

```
$ export HCAT_HOME=/usr/lib/hive-hcatalog
$ export HIVE_HOME=/usr/lib/hive
$ HIVE_VERSION=0.11.0-cdh5.0.0
$ HCATJAR=$HCAT_HOME/share/hcatalog/hcatalog-core-$HIVE_VERSION.jar
$ HCATPIGJAR=$HCAT_HOME/share/hcatalog/hcatalog-pig-adapter-$HIVE_VERSION.jar
$ export HADOOP_CLASSPATH=$HCATJAR:$HCATPIGJAR:$HIVE_HOME/lib/hive-exec-$HIVE_VERSION.jar\
:$HIVE_HOME/lib/hive-metastore-$HIVE_VERSION.jar:$HIVE_HOME/lib/jdo-api-*.jar:$HIVE_HOME/lib/libfb303-*.jar\
:$HIVE_HOME/lib/libthrift-*.jar:$HIVE_HOME/lib/slf4j-api-*.jar:$HIVE_HOME/conf:/etc/hadoop/conf
$ LIBJARS=`echo $HADOOP_CLASSPATH | sed -e 's/:/,/g'`
$ export LIBJARS=$LIBJARS,$HIVE_HOME/lib/antlr-runtime-*.jar
```



Note: You can find current version numbers for CDH dependencies in CDH's root `pom.xml` file for the current release, for example [cdh-root-5.0.0.pom](#).)

5. Run the job:

```
$ hadoop jar target/UseHCat-1.0.jar com.cloudera.test.UseHCat -files $HCATJAR -libjars
$LIBJARS groups groupids
```

Accessing Table Data with Pig

When using table information from the Hive metastore with Pig, add the `-useHCatalog` option when invoking pig:

```
$ pig -useHCatalog test.pig
```

In the script, use `HCatLoader` to have table schema retrieved automatically:

```
A = LOAD 'groups' USING org.apache.hcatalog.pig.HCatLoader();
DESCRIBE A;
```

Output:

```
A: {name: chararray,placeholder: chararray,id: int}
```

Accessing Table Information with REST

Table information can be retrieved from any host that has HTTP access to the host where the WebHCat server is running. A Web browser or an HTTP client such as curl or wget can be used to verify the functionality.

The base URL for REST access to table information is `http://<SERVERHOST>:50111/templeton/v1/ddl`.

Examples of specific URLs:

```
http://<SERVERHOST>:50111/templeton/v1/ddl/database/?user.name=hive
http://<SERVERHOST>:50111/templeton/v1/ddl/database/default/table/?user.name=hive
http://<SERVERHOST>:50111/templeton/v1/ddl/database/default/table/groups?user.name=hive
```

Example output:

```
{"columns":[{"name":"name","type":"string"},{"name":"placeholder","type":"string"},{"name":"id","type":"int"}],"database":"default","table":"groupable"}
```

Supported REST Endpoints

The General and DDL endpoints are supported, for accessing Hive metadata. If you need submission capabilities for MapReduce, Hive, or Pig jobs, consider using Oozie, which is a more mature interface. See [Installing Oozie](#) on page 334.

Category	Resource Type	Description
General	:version (GET)	Return a list of supported response types.
	status (GET)	Return the WebHCat server status.
	version (GET)	Return a list of supported versions and the current version.
	version/hive (GET)	Return the Hive version being run.
	version/hadoop (GET)	Return the Hadoop version being run.
DDL	ddl (POST)	Perform an HCatalog DDL command.
	ddl/database (GET)	List HCatalog databases.
	ddl/database/:db (GET)	Describe an HCatalog database.
	ddl/database/:db (PUT)	Create an HCatalog database.
	ddl/database/:db (DELETE)	Delete (drop) an HCatalog database.
	ddl/database/:db/table (GET)	List the tables in an HCatalog database.
	ddl/database/:db/table/:table (GET)	Describe an HCatalog table.
	ddl/database/:db/table/:table (PUT)	Create a new HCatalog table.
	ddl/database/:db/table/:table (POST)	Rename an HCatalog table.
	ddl/database/:db/table/:table (DELETE)	Delete (drop) an HCatalog table.
	ddl/database/:db/table/:existingtable/like/:newtable (PUT)	Create a new HCatalog table like an existing one.

Category	Resource Type	Description
	ddl/database/:db/table/:table/partition (GET)	List all partitions in an HCatalog table.
	ddl/database/:db/table/:table/partition/:partition (GET)	Describe a single partition in an HCatalog table.
	ddl/database/:db/table/:table/partition/:partition (PUT)	Create a partition in an HCatalog table.
	ddl/database/:db/table/:table/partition/:partition (DELETE)	Delete (drop) a partition in an HCatalog table.
	ddl/database/:db/table/:table/column (GET)	List the columns in an HCatalog table.
	ddl/database/:db/table/:table/column/:column (GET)	Describe a single column in an HCatalog table.
	ddl/database/:db/table/:table/column/:column (PUT)	Create a column in an HCatalog table.
	ddl/database/:db/table/:table/property (GET)	List table properties.
	ddl/database/:db/table/:table/property/:property (GET)	Return the value of a single table property.
	ddl/database/:db/table/:table/property/:property (PUT)	Set a table property.

Viewing the HCatalog Documentation

See [Apache wiki page](#).

Impala Installation

Impala is an open-source add-on to the Cloudera Enterprise Core that returns rapid responses to queries.



Note:

Under CDH 5, Impala is included as part of the CDH installation and no separate steps are needed.

What is Included in an Impala Installation

Impala is made up of a set of components that can be installed on multiple nodes throughout your cluster. The key installation step for performance is to install the `impalad` daemon (which does most of the query processing work) on *all* data nodes in the cluster.

The Impala package installs these binaries:

- `impalad` - The Impala daemon. Plans and executes queries against HDFS and HBase data. [Run one impalad process](#) on each node in the cluster that has a data node.
- `statedored` - Name service that tracks location and status of all `impalad` instances in the cluster. [Run one instance of this daemon](#) on a node in your cluster. Most production deployments run this daemon on the namenode.
- `catalogd` - Metadata coordination service that broadcasts changes from Impala DDL and DML statements to all affected Impala nodes, so that new tables, newly loaded data, and so on are immediately visible to queries submitted through any Impala node. (Prior to Impala 1.2, you had to run the `REFRESH` or `INVALIDATE METADATA` statement on each node to synchronize changed metadata. Now those statements are only required if you perform

the DDL or DML through Hive.) [Run one instance of this daemon](#) on a node in your cluster, preferably on the same host as the `statedored` daemon.

- `impala-shell` - [Command-line interface](#) for issuing queries to the Impala daemon. You install this on one or more hosts anywhere on your network, not necessarily data nodes or even within the same cluster as Impala. It can connect remotely to any instance of the Impala daemon.

Before doing the installation, ensure that you have all necessary prerequisites. See [Impala Requirements](#) on page 260 for details.

Impala Requirements

To perform as expected, Impala depends on the availability of the software, hardware, and configurations described in the following sections.

Product Compatibility Matrix

The ultimate source of truth about compatibility between various versions of CDH, Cloudera Manager, and various CDH components is the [Product Compatibility Matrix for CDH and Cloudera Manager](#).

For Impala, see the [Impala compatibility matrix page](#).

Supported Operating Systems

The relevant supported operating systems and versions for Impala are the same as for the corresponding CDH 5 platforms. For details, see the *Supported Operating Systems* page for [CDH 5](#).

Hive Metastore and Related Configuration

Impala can interoperate with data stored in Hive, and uses the same infrastructure as Hive for tracking metadata about schema objects such as tables and columns. The following components are prerequisites for Impala:

- MySQL or PostgreSQL, to act as a metastore database for both Impala and Hive.



Note:

Installing and configuring a Hive metastore is an Impala requirement. Impala does not work without the metastore database. For the process of installing and configuring the metastore, see [Impala Installation](#) on page 259.

Always configure a **Hive metastore service** rather than connecting directly to the metastore database. The Hive metastore service is required to interoperate between possibly different levels of metastore APIs used by CDH and Impala, and avoids known issues with connecting directly to the metastore database. The Hive metastore service is set up for you by default if you install through Cloudera Manager 4.5 or higher.

A summary of the metastore installation process is as follows:

- Install a MySQL or PostgreSQL database. Start the database if it is not started after installation.
- Download the [MySQL connector](#) or the [PostgreSQL connector](#) and place it in the `/usr/share/java/` directory.
- Use the appropriate command line tool for your database to create the metastore database.
- Use the appropriate command line tool for your database to grant privileges for the metastore database to the `hive` user.
- Modify `hive-site.xml` to include information matching your particular database: its URL, user name, and password. You will copy the `hive-site.xml` file to the Impala Configuration Directory later in the Impala installation process.

- **Optional:** Hive. Although only the Hive metastore database is required for Impala to function, you might install Hive on some client machines to create and load data into tables that use certain file formats. See [How Impala Works with Hadoop File Formats](#) for details. Hive does not need to be installed on the same data nodes as Impala; it just needs access to the same metastore database.

Java Dependencies

Although Impala is primarily written in C++, it does use Java to communicate with various Hadoop components:

- The officially supported JVM for Impala is the Oracle JVM. Other JVMs might cause issues, typically resulting in a failure at `impalad` startup. In particular, the JamVM used by default on certain levels of Ubuntu systems can cause `impalad` to fail to start.
- Internally, the `impalad` daemon relies on the `JAVA_HOME` environment variable to locate the system Java libraries. Make sure the `impalad` service is not run from an environment with an incorrect setting for this variable.
- All Java dependencies are packaged in the `impala-dependencies.jar` file, which is located at `/usr/lib/impala/lib/`. These map to everything that is built under `fe/target/dependency`.

Networking Configuration Requirements

As part of ensuring best performance, Impala attempts to complete tasks on local data, as opposed to using network connections to work with remote data. To support this goal, Impala matches the **hostname** provided to each Impala daemon with the **IP address** of each DataNode by resolving the hostname flag to an IP address. For Impala to work with local data, use a single IP interface for the DataNode and the Impala daemon on each machine. Ensure that the Impala daemon's hostname flag resolves to the IP address of the DataNode. For single-homed machines, this is usually automatic, but for multi-homed machines, ensure that the Impala daemon's hostname resolves to the correct interface. Impala tries to detect the correct hostname at start-up, and prints the derived hostname at the start of the log in a message of the form:

```
Using hostname: impala-daemon-1.example.com
```

In the majority of cases, this automatic detection works correctly. If you need to explicitly set the hostname, do so by setting the `--hostname` flag.

Hardware Requirements

During join operations, portions of data from each joined table are loaded into memory. Data sets can be very large, so ensure your hardware has sufficient memory to accommodate the joins you anticipate completing.

While requirements vary according to data set size, the following is generally recommended:

- CPU - Impala version 2.x uses the SSE4.1 instruction set, which is included in newer processors.



Note: This required level of processor is higher than in Impala version 1.x. Be sure to check the hardware of the hosts in your cluster before upgrading to Impala 2.x or the equivalent versions of CDH (5.2.0 and higher).

- Memory - 128 GB or more recommended, ideally 256 GB or more. If the intermediate results during query processing on a particular node exceed the amount of memory available to Impala on that node, the query writes temporary work data to disk, which can lead to long query times. Note that because the work is parallelized, and intermediate results for aggregate queries are typically smaller than the original data, Impala can query and join tables that are much larger than the memory available on an individual node.
- Storage - DataNodes with 12 or more disks each. I/O speeds are often the limiting factor for disk performance with Impala. Ensure that you have sufficient disk space to store the data Impala will be querying.

User Account Requirements

Impala creates and uses a user and group named `impala`. Do not delete this account or group and do not modify the account's or group's permissions and rights. Ensure no existing systems obstruct the functioning of these accounts and groups. For example, if you have scripts that delete user accounts not in a white-list, add these accounts to the list of permitted accounts.

For the resource management feature to work (in combination with CDH 5 and the YARN and Llama components), the `impala` user must be a member of the `hdfs` group. This setup is performed automatically during a new install, but not when upgrading from earlier Impala releases to Impala 1.2. If you are upgrading a node to CDH 5 that already had Impala 1.1 or 1.0 installed, manually add the `impala` user to the `hdfs` group.

For correct file deletion during `DROP TABLE` operations, Impala must be able to move files to the HDFS trashcan. You might need to create an HDFS directory `/user/impala`, writable by the `impala` user, so that the trashcan can be created. Otherwise, data files might remain behind after a `DROP TABLE` statement.

Impala should not run as root. Best Impala performance is achieved using direct reads, but root is not permitted to use direct reads. Therefore, running Impala as root negatively affects performance.

By default, any user can connect to Impala and access all the associated databases and tables. You can enable authorization and authentication based on the Linux OS user who connects to the Impala server, and the associated groups for that user. [Overview of Impala Security](#) for details. These security features do not change the underlying file permission requirements; the `impala` user still needs to be able to access the data files.

Installing Impala without Cloudera Manager

Before installing Impala manually, make sure all applicable nodes have the appropriate hardware configuration, levels of operating system and CDH, and any other software prerequisites. See [Impala Requirements](#) on page 260 for details.

You can install Impala across many hosts or on one host:

- Installing Impala across multiple machines creates a distributed configuration. For best performance, install Impala on **all** DataNodes.
- Installing Impala on a single machine produces a pseudo-distributed cluster.

To install Impala on a host:

1. Install CDH as described in the Installation section of the [CDH 5 Installation Guide](#).
2. Install the Hive metastore somewhere in your cluster, as described in the Hive Installation topic in the [CDH 5 Installation Guide](#). As part of this process, you configure the Hive metastore to use an external database as a metastore. Impala uses this same database for its own table metadata. You can choose either a MySQL or PostgreSQL database as the metastore. The process for configuring each type of database is described in the CDH Installation Guide).

Cloudera recommends setting up a Hive metastore service rather than connecting directly to the metastore database; this configuration is required when running Impala under CDH 4.1. Make sure the `/etc/impala/conf/hive-site.xml` file contains the following setting, substituting the appropriate hostname for `metastore_server_host`:

```
<property>
<name>hive.metastore.uris</name>
<value>thrift://metastore_server_host:9083</value>
</property>
<property>
<name>hive.metastore.client.socket.timeout</name>
<value>3600</value>
<description>MetaStore Client socket timeout in seconds</description>
</property>
```

3. (Optional) If you installed the full Hive component on any host, you can verify that the metastore is configured properly by starting the Hive console and querying for the list of available tables. Once you confirm that the console starts, exit the console to continue the installation:

```
$ hive
Hive history file=/tmp/root/hive_job_log_root_201207272011_678722950.txt
hive> show tables;
table1
table2
hive> quit;
$
```

4. Confirm that your package management command is aware of the Impala repository settings, as described in [Impala Requirements](#) on page 260. (For CDH 4, this is a different repository than for CDH.) You might need to download a repo or list file into a system directory underneath `/etc`.
5. Use **one** of the following sets of commands to install the Impala package:

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum install impala # Binaries for daemons
$ sudo yum install impala-server # Service start/stop script
$ sudo yum install impala-state-store # Service start/stop script
$ sudo yum install impala-catalog # Service start/stop script
```

For SUSE systems:

```
$ sudo zypper install impala # Binaries for daemons
$ sudo zypper install impala-server # Service start/stop script
$ sudo zypper install impala-state-store # Service start/stop script
$ sudo zypper install impala-catalog # Service start/stop script
```

For Debian or Ubuntu systems:

```
$ sudo apt-get install impala # Binaries for daemons
$ sudo apt-get install impala-server # Service start/stop script
$ sudo apt-get install impala-state-store # Service start/stop script
$ sudo apt-get install impala-catalog # Service start/stop script
```



Note: Cloudera recommends that you not install Impala on any HDFS NameNode. Installing Impala on NameNodes provides no additional data locality, and executing queries with such a configuration might cause memory contention and negatively impact the HDFS NameNode.

- Copy the client `hive-site.xml`, `core-site.xml`, `hdfs-site.xml`, and `hbase-site.xml` configuration files to the Impala configuration directory, which defaults to `/etc/impala/conf`. Create this directory if it does not already exist.
- Use **one** of the following commands to install `impala-shell` on the machines from which you want to issue queries. You can install `impala-shell` on any supported machine that can connect to DataNodes that are running `impalad`.

For RHEL/CentOS systems:

```
$ sudo yum install impala-shell
```

For SUSE systems:

```
$ sudo zypper install impala-shell
```

For Debian/Ubuntu systems:

```
$ sudo apt-get install impala-shell
```

- Complete any required or recommended configuration, as described in [Post-Installation Configuration for Impala](#). Some of these configuration changes are mandatory. (They are applied automatically when you install using Cloudera Manager.)

Once installation and configuration are complete, see [Starting Impala](#) on page 267 for how to activate the software on the appropriate nodes in your cluster.

If this is your first time setting up and using Impala in this cluster, run through some of the exercises in [Impala Tutorials](#) to verify that you can do basic operations such as creating tables and querying them.

Upgrading Impala

Upgrading Impala involves stopping Impala services, using your operating system's package management tool to upgrade Impala to the latest version, and then restarting Impala services.

**Note:**

- Each version of CDH 5 has an associated version of Impala. When you upgrade from CDH 4 to CDH 5, you get whichever version of Impala comes with the associated level of CDH. Depending on the version of Impala you were running on CDH 4, this could install a lower level of Impala on CDH 5. For example, if you upgrade to CDH 5.0 from CDH 4 plus Impala 1.4, the CDH 5.0 installation comes with Impala 1.3. Always check the associated level of Impala before upgrading to a specific version of CDH 5. Where practical, upgrade from CDH 4 to the latest CDH 5, which also has the latest Impala.
- When you upgrade Impala, also upgrade Cloudera Manager if necessary:
 - Users running Impala on CDH 5 must upgrade to Cloudera Manager 5.0.0 or higher.
 - Users running Impala on CDH 4 must upgrade to Cloudera Manager 4.8 or higher. Cloudera Manager 4.8 includes management support for the Impala catalog service, and is the minimum Cloudera Manager version you can use.
 - Cloudera Manager is continually updated with configuration settings for features introduced in the latest Impala releases.
- If you are upgrading from CDH 5 beta to CDH 5.0 production, make sure you are using the appropriate CDH 5 repositories shown on the [CDH version and packaging](#) page, then follow the procedures throughout the rest of this section.
- Every time you upgrade to a new major or minor Impala release, see [Cloudera Impala Incompatible Changes](#) in the *Release Notes* for any changes needed in your source code, startup scripts, and so on.
- Also check [Cloudera Impala Known Issues](#) in the *Release Notes* for any issues or limitations that require workarounds.
- Due to a change to the implementation of logging in Impala 1.1.1 and higher, currently you should change the default setting for the `logbuflevel` property for the Impala service after installing through Cloudera Manager. In Cloudera Manager, on the log settings page for the Impala service, change the setting **Impala Daemon Log Buffer Level (logbuflevel)** from -1 to 0. You might change this setting to a value higher than 0, if you prefer to reduce the I/O overhead for logging, at the expense of possibly losing some lower-priority log messages in the event of a crash.
- For the resource management feature to work (in combination with CDH 5 and the YARN and Llama components), the `impala` user must be a member of the `hdfs` group. This setup is performed automatically during a new install, but not when upgrading from earlier Impala releases to Impala 1.2. If you are upgrading a node to CDH 5 that already had Impala 1.1 or 1.0 installed, manually add the `impala` user to the `hdfs` group.

Upgrading Impala through Cloudera Manager - Parcels

Parcels are an alternative binary distribution format available in Cloudera Manager 4.5 and higher.



Important: In CDH 5, there is not a separate Impala parcel; Impala is part of the main CDH 5 parcel. Each level of CDH 5 has a corresponding version of Impala, and you upgrade Impala by upgrading CDH. See the [CDH 5 upgrade instructions](#) and choose the instructions for parcels. The remainder of this section only covers parcel upgrades for Impala under CDH 4.

To upgrade Impala for CDH 4 in a Cloudera Managed environment, using parcels:

1. If you originally installed using packages and now are switching to parcels, remove all the Impala-related packages first. You can check which packages are installed using one of the following commands, depending on your operating system:

```
rpm -qa # RHEL, Oracle Linux, CentOS, Debian
dpkg --get-selections # Debian
```


and then remove the packages using one of the following commands:

```
sudo yum remove pkg_names # RHEL, Oracle Linux, CentOS
sudo zypper remove pkg_names # SLES
sudo apt-get purge pkg_names # Ubuntu, Debian
```

2. Connect to the Cloudera Manager Admin Console.
3. Go to the **Hosts > Parcels** tab. You should see a parcel with a newer version of Impala that you can upgrade to.
4. Click **Download**, then **Distribute**. (The button changes as each step completes.)
5. Click **Activate**.
6. When prompted, click **Restart** to restart the Impala service.

Upgrading Impala through Cloudera Manager - Packages

To upgrade Impala in a Cloudera Managed environment, using packages:

1. Connect to the Cloudera Manager Admin Console.
2. In the **Services** tab, click the **Impala** service.
3. Click **Actions** and click **Stop**.
4. Use **one** of the following sets of commands to update Impala on each Impala node in your cluster:

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala
$ sudo yum update hadoop-lzo-cdh4 # Optional; if this package is already installed.
```

For SUSE systems:

```
$ sudo zypper update impala
$ sudo zypper update hadoop-lzo-cdh4 # Optional; if this package is already installed
```

For Debian or Ubuntu systems:

```
$ sudo apt-get install impala
$ sudo apt-get install hadoop-lzo-cdh4 # Optional; if this package is already installed
```

5. Use **one** of the following sets of commands to update Impala shell on each node on which it is installed:

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala-shell
```

For SUSE systems:

```
$ sudo zypper update impala-shell
```

For Debian or Ubuntu systems:

```
$ sudo apt-get install impala-shell
```

6. Connect to the Cloudera Manager Admin Console.
7. In the **Services** tab, click the Impala service.
8. Click **Actions** and click **Start**.

Upgrading Impala without Cloudera Manager

To upgrade Impala on a cluster not managed by Cloudera Manager, run these Linux commands on the appropriate hosts in your cluster:

1. Stop Impala services.

a. Stop `impalad` on each Impala node in your cluster:

```
$ sudo service impala-server stop
```

b. Stop any instances of the state store in your cluster:

```
$ sudo service impala-state-store stop
```

c. Stop any instances of the catalog service in your cluster:

```
$ sudo service impala-catalog stop
```

2. Check if there are new recommended or required configuration settings to put into place in the configuration files, typically under `/etc/impala/conf`. See [Post-Installation Configuration for Impala](#) for settings related to performance and scalability.

3. Use **one of the following sets of commands to update Impala on each Impala node in your cluster:**

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala-server
$ sudo yum update hadoop-lzo-cdh4 # Optional; if this package is already installed
$ sudo yum update impala-catalog # New in Impala 1.2; do yum install when upgrading from 1.1.
```

For SUSE systems:

```
$ sudo zypper update impala-server
$ sudo zypper update hadoop-lzo-cdh4 # Optional; if this package is already installed
$ sudo zypper update impala-catalog # New in Impala 1.2; do zypper install when upgrading from 1.1.
```

For Debian or Ubuntu systems:

```
$ sudo apt-get install impala-server
$ sudo apt-get install hadoop-lzo-cdh4 # Optional; if this package is already installed
$ sudo apt-get install impala-catalog # New in Impala 1.2.
```

4. Use **one of the following sets of commands to update Impala shell on each node on which it is installed:**

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala-shell
```

For SUSE systems:

```
$ sudo zypper update impala-shell
```

For Debian or Ubuntu systems:

```
$ sudo apt-get install impala-shell
```

5. Depending on which release of Impala you are upgrading from, you might find that the symbolic links `/etc/impala/conf` and `/usr/lib/impala/sbin` are missing. If so, see [Cloudera Impala Known Issues](#) for the procedure to work around this problem.

6. Restart Impala services:

- a. Restart the Impala state store service on the desired nodes in your cluster. Expect to see a process named `statedstore` if the service started successfully.

```
$ sudo service impala-state-store start
$ ps ax | grep [s]tatedstore
6819 ?      Sl      0:07 /usr/lib/impala/sbin/statedstore -log_dir=/var/log/impala
-state_store_port=24000
```

Restart the state store service *before* the Impala server service to avoid “Not connected” errors when you run `impala-shell`.

- b. Restart the Impala catalog service on whichever host it runs on in your cluster. Expect to see a process named `catalogd` if the service started successfully.

```
$ sudo service impala-catalog restart
$ ps ax | grep [c]atalogd
6068 ?      Sl      4:06 /usr/lib/impala/sbin/catalogd
```

- c. Restart the Impala daemon service on each node in your cluster. Expect to see a process named `impalad` if the service started successfully.

```
$ sudo service impala-server start
$ ps ax | grep [i]mpalad
7936 ?      Sl      0:12 /usr/lib/impala/sbin/impalad -log_dir=/var/log/impala
-state_store_port=24000 -use_statestore
-state_store_host=127.0.0.1 -be_port=22000
```

**Note:**

If the services did not start successfully (even though the `sudo service` command might display [OK]), check for errors in the Impala log file, typically in `/var/log/impala`.

Starting Impala

To activate Impala if it is installed but not yet started:

1. Set any necessary configuration options for the Impala services. See [Modifying Impala Startup Options](#) on page 268 for details.
2. Start one instance of the Impala statestore. The statestore helps Impala to distribute work efficiently, and to continue running in the event of availability problems for other Impala nodes. If the statestore becomes unavailable, Impala continues to function.
3. Start one instance of the Impala catalog service.
4. Start the main Impala service on one or more DataNodes, ideally on all DataNodes to maximize local processing and avoid network traffic due to remote reads.

Once Impala is running, you can conduct interactive experiments using the instructions in [Impala Tutorials](#) and try [Using the Impala Shell \(impala-shell Command\)](#).

Starting Impala through Cloudera Manager

If you installed Impala with Cloudera Manager, use Cloudera Manager to start and stop services. The Cloudera Manager GUI is a convenient way to check that all services are running, to set configuration options using form fields in a browser, and to spot potential issues such as low disk space before they become serious. Cloudera Manager automatically starts all the Impala-related services as a group, in the correct order. See [the Cloudera Manager Documentation](#) for details.

**Note:**

Currently, Impala UDFs and UDAs are not persisted in the metastore database. Information about these functions is held in the memory of the `catalogd` daemon. You must reload them by running the `CREATE FUNCTION` statements again each time you restart the `catalogd` daemon.

Starting Impala from the Command Line

To start the Impala state store and Impala from the command line or a script, you can either use the `service` command or you can start the daemons directly through the `impalad`, `statestored`, and `catalogd` executables.

Start the Impala statestore and then start `impalad` instances. You can modify the values the service initialization scripts use when starting the statestore and Impala by editing `/etc/default/impala`.

Start the statestore service using a command similar to the following:

```
$ sudo service impala-state-store start
```

Start the catalog service using a command similar to the following:

```
$ sudo service impala-catalog start
```

Start the Impala service on each data node using a command similar to the following:

```
$ sudo service impala-server start
```

**Note:**

Currently, Impala UDFs and UDAs are not persisted in the metastore database. Information about these functions is held in the memory of the `catalogd` daemon. You must reload them by running the `CREATE FUNCTION` statements again each time you restart the `catalogd` daemon.

If any of the services fail to start, review:

- [Reviewing Impala Logs](#)
- [Troubleshooting Impala](#)

Modifying Impala Startup Options

The configuration options for the Impala-related daemons let you choose which hosts and ports to use for the services that run on a single host, specify directories for logging, control resource usage and security, and specify other aspects of the Impala software.

Configuring Impala Startup Options through Cloudera Manager

If you manage your cluster through Cloudera Manager, configure the settings for all the Impala-related daemons by navigating to this page: **Services > Impala > Configuration > View and Edit**. See the Cloudera Manager documentation for [instructions about how to configure Impala through Cloudera Manager](#).

If the Cloudera Manager interface does not yet have a form field for a newly added option, or if you need to use special options for debugging and troubleshooting, the **Advanced** option page for each daemon includes one or more fields where you can enter option names directly. In Cloudera Manager 4, these fields are labelled **Safety Valve**; in Cloudera Manager 5, they are called **Advanced Configuration Snippet**. There is also a free-form field for query options, on the top-level **Impala Daemon** options page.

Configuring Impala Startup Options through the Command Line

When you run Impala in a non-Cloudera Manager environment, the Impala server, statestore, and catalog services start up using values provided in a defaults file, `/etc/default/impala`.

This file includes information about many resources used by Impala. Most of the defaults included in this file should be effective in most cases. For example, typically you would not change the definition of the `CLASSPATH` variable, but you would always set the address used by the statestore server. Some of the content you might modify includes:

```
IMPALA_STATE_STORE_HOST=127.0.0.1
IMPALA_STATE_STORE_PORT=24000
IMPALA_BACKEND_PORT=22000
IMPALA_LOG_DIR=/var/log/impala
IMPALA_CATALOG_SERVICE_HOST=...
IMPALA_STATE_STORE_HOST=...

export IMPALA_STATE_STORE_ARGS=${IMPALA_STATE_STORE_ARGS:- \
  -log_dir=${IMPALA_LOG_DIR} -state_store_port=${IMPALA_STATE_STORE_PORT}}
IMPALA_SERVER_ARGS=" \
  -log_dir=${IMPALA_LOG_DIR} \
  -catalog_service_host=${IMPALA_CATALOG_SERVICE_HOST} \
  -state_store_port=${IMPALA_STATE_STORE_PORT} \
  -use_statestore \
  -state_store_host=${IMPALA_STATE_STORE_HOST} \
  -be_port=${IMPALA_BACKEND_PORT}"
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-false}
```

To use alternate values, edit the defaults file, then restart all the Impala-related services so that the changes take effect. Restart the Impala server using the following commands:

```
$ sudo service impala-server restart
Stopping Impala Server:           [ OK ]
Starting Impala Server:           [ OK ]
```

Restart the Impala statestore using the following commands:

```
$ sudo service impala-state-store restart
Stopping Impala State Store Server: [ OK ]
Starting Impala State Store Server: [ OK ]
```

Restart the Impala catalog service using the following commands:

```
$ sudo service impala-catalog restart
Stopping Impala Catalog Server:    [ OK ]
Starting Impala Catalog Server:    [ OK ]
```

Some common settings to change include:

- **Statestore address.** Cloudera recommends the statestore be on a separate host not running the `impalad` daemon. In that recommended configuration, the `impalad` daemon cannot refer to the statestore server using the loopback address. If the statestore is hosted on a machine with an IP address of `192.168.0.27`, change:

```
IMPALA_STATE_STORE_HOST=127.0.0.1
```

to:

```
IMPALA_STATE_STORE_HOST=192.168.0.27
```

- **Catalog server address (including both the hostname and the port number).** Update the value of the `IMPALA_CATALOG_SERVICE_HOST` variable. Cloudera recommends the catalog server be on the same host as the statestore. In that recommended configuration, the `impalad` daemon cannot refer to the catalog server using the loopback address. If the catalog service is hosted on a machine with an IP address of `192.168.0.27`, add the following line:

```
IMPALA_CATALOG_SERVICE_HOST=192.168.0.27:26000
```

The `/etc/default/impala` defaults file currently does not define an `IMPALA_CATALOG_ARGS` environment variable, but if you add one it will be recognized by the service startup/shutdown script. Add a definition for this

variable to `/etc/default/impala` and add the option `-catalog_service_host=hostname`. If the port is different than the default 26000, also add the option `-catalog_service_port=port`.

- Memory limits. You can limit the amount of memory available to Impala. For example, to allow Impala to use no more than 70% of system memory, change:

```
export IMPALA_SERVER_ARGS=${IMPALA_SERVER_ARGS:- \
  -log_dir=${IMPALA_LOG_DIR} \
  -state_store_port=${IMPALA_STATE_STORE_PORT} \
  -use_statestore -state_store_host=${IMPALA_STATE_STORE_HOST} \
  -be_port=${IMPALA_BACKEND_PORT}}
```

to:

```
export IMPALA_SERVER_ARGS=${IMPALA_SERVER_ARGS:- \
  -log_dir=${IMPALA_LOG_DIR} -state_store_port=${IMPALA_STATE_STORE_PORT} \
  -use_statestore -state_store_host=${IMPALA_STATE_STORE_HOST} \
  -be_port=${IMPALA_BACKEND_PORT} -mem_limit=70%}
```

You can specify the memory limit using absolute notation such as 500m or 2G, or as a percentage of physical memory such as 60%.



Note: Queries that exceed the specified memory limit are aborted. Percentage limits are based on the physical memory of the machine and do not consider cgroups.

- Core dump enablement. To enable core dumps, change:

```
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-false}
```

to:

```
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-true}
```



Note: The location of core dump files may vary according to your operating system configuration. Other security settings may prevent Impala from writing core dumps even when this option is enabled.

- Authorization using the open source Sentry plugin. Specify the `-server_name` and `-authorization_policy_file` options as part of the `IMPALA_SERVER_ARGS` and `IMPALA_STATE_STORE_ARGS` settings to enable the core Impala support for authentication. See [Starting the impalad Daemon with Sentry Authorization Enabled](#) for details.
- Auditing for successful or blocked Impala queries, another aspect of security. Specify the `-audit_event_log_dir=directory_path` option and optionally the `-max_audit_event_log_file_size=number_of_queries` and `-abort_on_failed_audit_event` options as part of the `IMPALA_SERVER_ARGS` settings, for each Impala node, to enable and customize auditing. See [Auditing Impala Operations](#) for details.
- Password protection for the Impala web UI, which listens on port 25000 by default. This feature involves adding some or all of the `--webserver_password_file`, `--webserver_authentication_domain`, and `--webserver_certificate_file` options to the `IMPALA_SERVER_ARGS` and `IMPALA_STATE_STORE_ARGS` settings. See [Security Guidelines for Impala](#) for details.
- Another setting you might add to `IMPALA_SERVER_ARGS` is a comma-separated list of query options and values:

```
-default_query_options='option=value,option=value,...'
```

These options control the behavior of queries performed by this `impalad` instance. The option values you specify here override the default values for [Impala query options](#), as shown by the `SET` statement in `impala-shell`.

- Options for resource management, in conjunction with the YARN and Llama components. These options include `-enable_rm`, `-llama_host`, `-llama_port`, `-llama_callback_port`, and `-cgroup_hierarchy_path`. Additional options to help fine-tune the resource estimates are `--rm_always_use_defaults`, `--rm_default_memory=size`, and `--rm_default_cpu_cores`. For details about these options, see [impalad Startup Options for Resource Management](#). See [Integrated Resource Management with YARN](#) for information about resource management in general, and [The Llama Daemon](#) for information about the Llama daemon.
- During troubleshooting, Cloudera Support might direct you to change other values, particularly for `IMPALA_SERVER_ARGS`, to work around issues or gather debugging information.

The following startup options for `impalad` enable resource management and customize its parameters for your cluster configuration:

- `-enable_rm`: Whether to enable resource management or not, either `true` or `false`. The default is `false`. None of the other resource management options have any effect unless `-enable_rm` is turned on.
- `-llama_host`: Hostname or IP address of the Llama service that Impala should connect to. The default is `127.0.0.1`.
- `-llama_port`: Port of the Llama service that Impala should connect to. The default is `15000`.
- `-llama_callback_port`: Port that Impala should start its Llama callback service on. Llama reports when resources are granted or preempted through that service.
- `-cgroup_hierarchy_path`: Path where YARN and Llama will create cgroups for granted resources. Impala assumes that the cgroup for an allocated container is created in the path `'cgroup_hierarchy_path + container_id'`.
- `-rm_always_use_defaults`: If this Boolean option is enabled, Impala ignores computed estimates and always obtains the default memory and CPU allocation from Llama at the start of the query. These default estimates are approximately 2 CPUs and 4 GB of memory, possibly varying slightly depending on cluster size, workload, and so on. Cloudera recommends enabling `-rm_always_use_defaults` whenever resource management is used, and relying on these default values (that is, leaving out the two following options).
- `-rm_default_memory=size`: Optionally sets the default estimate for memory usage for each query. You can use suffixes such as `M` and `G` for megabytes and gigabytes, the same as with the [MEM_LIMIT](#) query option. Only has an effect when `-rm_always_use_defaults` is also enabled.
- `-rm_default_cpu_cores`: Optionally sets the default estimate for number of virtual CPU cores for each query. Only has an effect when `-rm_always_use_defaults` is also enabled.



Note:

These startup options for the `impalad` daemon are different from the command-line options for the `impala-shell` command. For the `impala-shell` options, see [impala-shell Configuration Options](#).

Checking the Values of Impala Configuration Options

You can check the current runtime value of all these settings through the Impala web interface, available by default at `http://impala_hostname:25000/varz` for the `impalad` daemon, `http://impala_hostname:25010/varz` for the `statedored` daemon, or `http://impala_hostname:25020/varz` for the `catalogd` daemon. In the Cloudera Manager interface, you can see the link to the appropriate **service_name Web UI** page when you look at the status page for a specific daemon on a specific host.

Startup Options for `impalad` Daemon

The `impalad` daemon implements the main Impala service, which performs query processing and reads and writes the data files.

Startup Options for `statedored` Daemon

The `statedored` daemon implements the Impala statestore service, which monitors the availability of Impala services across the cluster, and handles situations such as nodes becoming unavailable or becoming available again.

Startup Options for catalogd Daemon

The `catalogd` daemon implements the Impala catalog service, which broadcasts metadata changes to all the Impala nodes when Impala creates a table, inserts data, or performs other kinds of DDL and DML operations.

By default, the metadata loading and caching on startup happens asynchronously, so Impala can begin accepting requests promptly. To enable the original behavior, where Impala waited until all metadata was loaded before accepting any requests, set the `catalogd` configuration option `--load_catalog_in_background=false`.

Hive Installation

**Note: Install Cloudera Repository**

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

**Note: Running Services**

When starting, stopping and restarting CDH components, always use the `service (8)` command rather than running scripts in `/etc/init.d` 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 scripts in `/etc/init.d`, 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).)



Warning: HiveServer1 is deprecated in CDH 5.3, and will be removed in a future release of CDH. Users of HiveServer1 should upgrade to [HiveServer2](#) as soon as possible.

Using Hive data in HBase is a common task. See [Importing Data Into HBase](#).

Use the following sections to install, update, and configure Hive.

About Hive

Apache Hive is a powerful data warehousing application built on top of Hadoop; it enables you to access your data using Hive QL, a language that is similar to SQL.

**Note:**

As of CDH 5, Hive includes HCatalog, but you still need to install HCatalog separately if you want to use it; see [HCatalog Installation](#) on page 254.

Install Hive on your client machine(s) from which you submit jobs; you do not need to install it on the nodes in your Hadoop cluster.

HiveServer2

[HiveServer2](#) is an improved version of HiveServer that supports a Thrift API tailored for JDBC and ODBC clients, Kerberos authentication, and multi-client concurrency. The CLI for HiveServer2 is [Beeline](#).



Warning: Because of concurrency and security issues, HiveServer1 and the Hive CLI are deprecated in CDH 5 and will be removed in a future release. Cloudera recommends you migrate to [Beeline](#) and [HiveServer2](#) as soon as possible. The Hive CLI is not needed if you are using Beeline with HiveServer2.

Upgrading Hive

Upgrade Hive on all the hosts on which it is running: servers and clients.



Note: To see which version of Hive is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Checklist to Help Ensure Smooth Upgrades

The following best practices for configuring and maintaining Hive will help ensure that upgrades go smoothly.

- Configure periodic backups of the [metastore database](#). Use `mysqldump`, or the equivalent for your vendor if you are not using MySQL.
- Make sure `datanucleus.autoCreateSchema` is set to `false` (in all types of database) and `datanucleus.fixedDatastore` is set to `true` (for MySQL and Oracle) in *all* `hive-site.xml` files. See the [configuration instructions](#) for more information about setting the properties in `hive-site.xml`.
- Insulate the metastore database from users by running the metastore service in [Remote mode](#). If you do not follow this recommendation, make sure you remove `DROP`, `ALTER`, and `CREATE` privileges from the Hive user configured in `hive-site.xml`. See [Configuring the Hive Metastore](#) on page 277 for complete instructions for each type of supported database.



Warning:

Make sure you have read and understood all [incompatible changes](#) before you upgrade Hive.

Upgrading Hive from an Earlier Version of CDH 5

The instructions that follow assume that you are upgrading Hive as part of a CDH 5 upgrade, and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).



Important:

- If you are currently running Hive under MRv1, check for the following property and value in `/etc/mapred/conf/mapred-site.xml`:

```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

Remove this property before you proceed; otherwise Hive queries spawned from MapReduce jobs will fail with a null pointer exception (NPE).

- If you have installed the `hive-hcatalog-server` package in the past, you must remove it before you proceed; otherwise the upgrade will fail.
- CDH 5.2 and later clients cannot communicate with CDH 5.1 and earlier servers. This means that you must upgrade the server before the clients.

To upgrade Hive from an earlier version of CDH 5, proceed as follows.

Step 1: Stop all Hive Processes and Daemons



Warning:

You **must** make sure no Hive processes are running. If Hive processes are running during the upgrade, the new version will not work correctly.

1. Stop any HiveServer processes that are running:

```
$ sudo service hive-server stop
```

2. Stop any HiveServer2 processes that are running:

```
$ sudo service hive-server2 stop
```

3. Stop the metastore:

```
$ sudo service hive-metastore stop
```

Step 2: Install the new Hive version on all hosts (Hive servers and clients)

See [Installing Hive](#) on page 276

Step 3: Verify that the Hive Metastore is Properly Configured

See [Configuring the Hive Metastore](#) on page 277 for detailed instructions.

Step 4: Upgrade the Metastore Schema



Important:

- Cloudera recommends that you make a backup copy of your metastore database before running the `schematool` or the upgrade scripts. You might need this backup copy if there are problems during the upgrade or if you need to downgrade to a previous version.
- You *must* upgrade the metastore schema to the version corresponding to the new version of Hive before starting Hive after the upgrade. Failure to do so may result in metastore corruption.

To upgrade the Hive metastore schema, you can use either the Hive `schematool` or use the schema upgrade scripts that are provided with the Hive package. Cloudera recommends that you use the `schematool`.

Using Hive `schematool` (Recommended):

The Hive distribution includes a command-line tool for Hive metastore schema manipulation called `schematool`. This tool can be used to initialize the metastore schema for the current Hive version. It can also upgrade the schema from an older version to the current one. You must add properties to the `hive-site.xml` before you can use it. See [Using the Hive Schema Tool](#) on page 294 for information about how to set the tool up and for usage examples. To upgrade the schema, use the `upgradeSchemaFrom` option to specify the version of the schema you are currently using. For example, if you are upgrading a MySQL metastore schema from Hive 0.13.1, use the following syntax:

```
$ schematool -dbType mysql -passWord <db_user_pswd> -upgradeSchemaFrom
0.13.1 -userName <db_user_name>
Metastore connection URL:
jdbc:mysql://<cluster_address>:3306/<user_name>?useUnicode=true&characterEncoding=UTF-8
Metastore Connection Driver : com.mysql.jdbc.Driver
Metastore connection User: <user_name>
Starting upgrade metastore schema from version 0.13.1 to <new_version>
Upgrade script upgrade-0.13.1-to-<new_version>.mysql.sql
Completed pre-0-upgrade-0.13.1-to-<new_version>.mysql.sql
Completed upgrade-0.13.1-to-<new_version>.mysql.sql
schemaTool completed
```



Note: The `upgradeSchemaFrom` option requires the Hive version and not the CDH version. See [CDH Packaging and Tarball Information](#) for information about which Hive version ships with each CDH release.

Using Schema Upgrade Scripts:

Navigate to the directory where the schema upgrade scripts are located:

- If you installed CDH with parcels, the scripts are in the following location:

```
/opt/cloudera/parcels/CDH/lib/hive/scripts/metastore/upgrade/<database_name>
```

- If you installed CDH with packages, the scripts are in the following location:

```
/usr/lib/hive/scripts/metastore/upgrade/<database_name>
```

For example, if your Hive metastore is MySQL and you installed CDH with packages, navigate to `/usr/lib/hive/scripts/metastore/upgrade/mysql`.

Run the appropriate schema upgrade scripts in order. Start with the script for your database type and Hive version, and run all subsequent scripts.

For example, if you are currently running Hive 0.13.1 with MySQL and upgrading to Hive 1.1.0, start with the script for 0.13.0 to 0.14.0 for MySQL, and then run the script for Hive 0.14.0 to 1.1.0.

For more information about using the scripts to upgrade the schema, see the README in the directory with the scripts.

Step 5: Start the Metastore, HiveServer2, and Beeline

See:

- [Starting the Metastore](#) on page 292
- [Starting, Stopping, and Using HiveServer2](#) on page 292

The upgrade is now complete.

Troubleshooting: if you failed to upgrade the metastore

If you failed to upgrade the metastore as instructed above, proceed as follows.

1. Identify the problem.

The symptoms are as follows:

- Hive stops accepting queries.
- In a cluster managed by Cloudera Manager, the Hive Metastore canary fails.
- An error such as the following appears in the Hive Metastore Server logs:

```
Hive Schema version 0.13.0 does not match metastore's schema version 0.12.0 Metastore is not upgraded or corrupt.
```

2. Resolve the problem.

If the problem you are having matches the symptoms just described, do the following:

1. Stop all Hive services; for example:

```
$ sudo service hive-server2 stop
$ sudo service hive-metastore stop
```

2. Run the Hive schematool, as instructed [here](#).

Make sure the value you use for the `-upgradeSchemaFrom` option matches the version you are *currently running* (not the new version). For example, if the error message in the log is

```
Hive Schema version 0.13.0 does not match metastore's schema version 0.12.0 Metastore
is not upgraded or corrupt.
```

then the value of `-upgradeSchemaFrom` must be `0.12.0`.

3. Restart the Hive services you stopped.

Installing Hive

Install the appropriate Hive packages using the appropriate command for your distribution.

OS	Command
RHEL-compatible	<code>\$ sudo yum install <pkg1> <pkg2> ...</code>
SLES	<code>\$ sudo zypper install <pkg1> <pkg2> ...</code>
Ubuntu or Debian	<code>\$ sudo apt-get install <pkg1> <pkg2> ...</code>

The packages are:

- `hive` – base package that provides the complete language and runtime
- `hive-metastore` – provides scripts for running the metastore as a standalone service (optional)
- `hive-server2` – provides scripts for running HiveServer2
- `hive-hbase` - optional; install this package if you want to [use Hive with HBase](#).

Heap Size and Garbage Collection for Hive Components

Hive Component Memory Recommendations

HiveServer2 and the Hive metastore require sufficient memory in order to run correctly. The default heap size of 256 MB for each component is inadequate for production workloads. Consider the following guidelines for sizing the heap for each component, based upon your cluster size.

Number of Concurrent Connections	HiveServer2 Heap Size Minimum Recommendation	Hive Metastore Heap Size Minimum Recommendation
Up to 40 concurrent connections (Cloudera recommends splitting HiveServer2 into multiple instances and load balancing once you start allocating >12 GB to HiveServer2. ¹)	12 GB	12 GB
Up to 20 concurrent connections	6 GB	10 GB
Up to 10 concurrent connections	4 GB	8 GB
Single connection	2 GB	4 GB

¹ The objective is to size to reduce impact of Java garbage collection on active processing by the service.



Important: These numbers are general guidance only, and may be affected by factors such as number of columns, partitions, complex joins, and client activity among other things. It is important to review and refine through testing based on your anticipated deployment to arrive at best values for your environment.

In addition, the Beehive CLI should use a heap size of at least 2 GB.

The `permGenSize` should be set to 512M for all.

Configuring Heap Size and Garbage Collection for Hive Components

To configure the heap size for HiveServer2 and Hive metastore, set the `-Xmx` parameter in the `HADOOP_OPTS` variable to the desired maximum heap size in the `hive-env.sh` advanced configuration snippet if you use Cloudera Manager or otherwise edit `/etc/hive/hive-env.sh`.

To configure the heap size for the Beehive CLI, set the `HADOOP_HEAPSIZE` environment variable in the `hive-env.sh` advanced configuration snippet if you use Cloudera Manager or otherwise edit `/etc/hive/hive-env.sh` before starting the Beehive CLI.

The following example shows a configuration with the following settings:

- HiveServer2 uses 12 GB heap
- Hive metastore uses 12 GB heap
- Hive clients use 2 GB heap

The settings to change are in bold. All of these lines are commented out (prefixed with a `#` character) by default. Uncomment the lines by removing the `#` character.

```
if [ "$SERVICE" = "cli" ]; then
  if [ -z "$DEBUG" ]; then
    export HADOOP_OPTS="$HADOOP_OPTS -XX:NewRatio=12 -Xmx12288m -Xms10m
-XX:MaxHeapFreeRatio=40 -XX:MinHeapFreeRatio=15 -XX:+UseParNewGC -XX:-UseGCOverheadLimit"
  else
    export HADOOP_OPTS="$HADOOP_OPTS -XX:NewRatio=12 -Xmx12288m -Xms10m
-XX:MaxHeapFreeRatio=40 -XX:MinHeapFreeRatio=15 -XX:-UseGCOverheadLimit"
  fi
fi

export HADOOP_HEAPSIZE=2048
```

You can choose whether to use the Concurrent Collector or the New Parallel Collector for garbage collection, by passing `-XX:+UseParNewGC` or `-XX:+UseConcMarkSweepGC` in the `HADOOP_OPTS` lines above, and you can tune the garbage collection overhead limit by setting `-XX:-UseGCOverheadLimit`. To enable the garbage collection overhead limit, remove the setting or change it to `-XX:+UseGCOverheadLimit`.

Configuration for WebHCat

If you want to use WebHCat, you need to set the `PYTHON_CMD` variable in `/etc/default/hive-webhcat-server` after installing Hive; for example:

```
export PYTHON_CMD=/usr/bin/python
```

Configuring the Hive Metastore

The Hive metastore service stores the metadata for Hive tables and partitions in a relational database, and provides clients (including Hive) access to this information using the metastore service API. This page explains the deployment options and provides instructions for setting up a database in a recommended configuration.

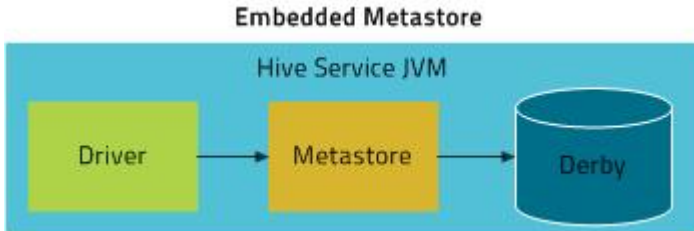
Metastore Deployment Modes



Note: On this page, **HiveServer** refers to HiveServer1 or HiveServer2, whichever you are using.

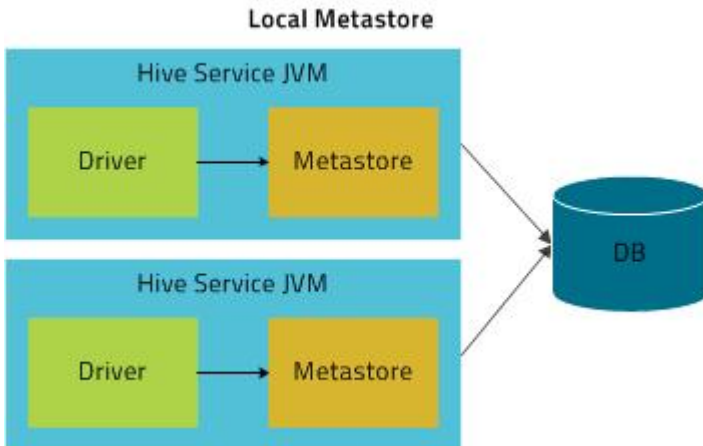
Embedded Mode

Cloudera recommends using this mode for experimental purposes only.



Embedded mode is the default metastore deployment mode for CDH. In this mode, the metastore uses a Derby database, and both the database and the metastore service run embedded in the main HiveServer process. Both are started for you when you start the HiveServer process. This mode requires the least amount of effort to configure, but it can support only one active user at a time and is not certified for production use.

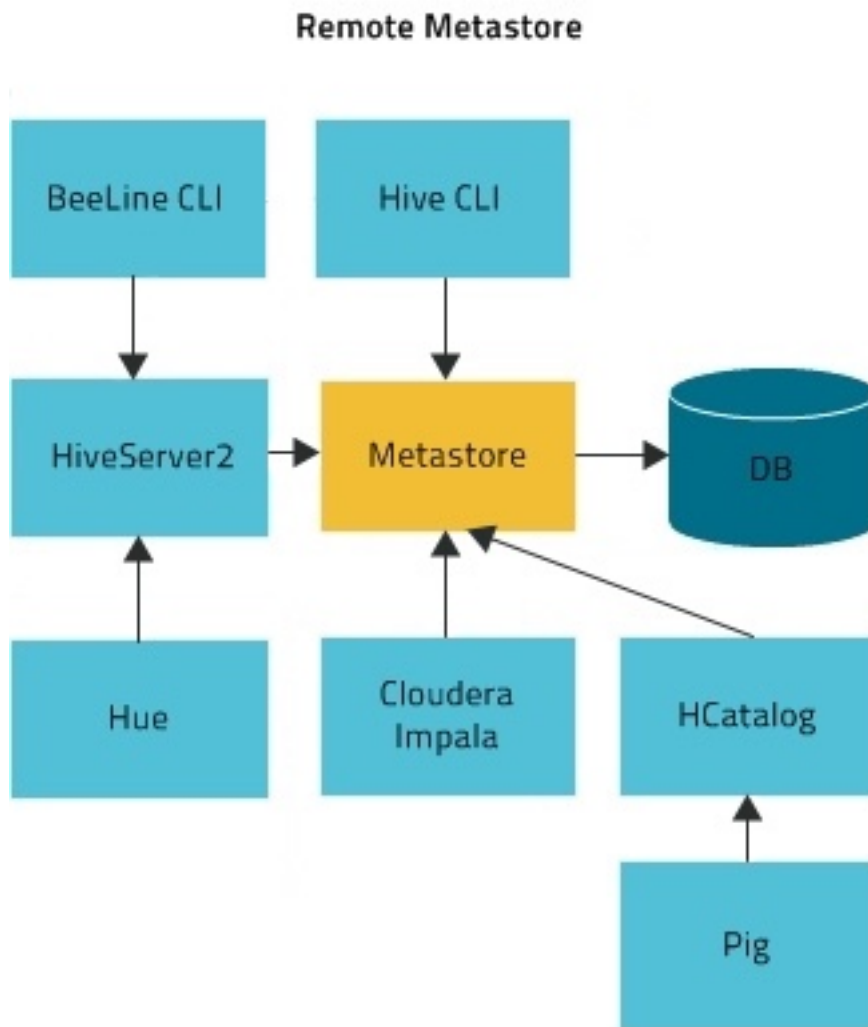
Local Mode



In Local mode, the Hive metastore service runs in the same process as the main HiveServer process, but the metastore database runs in a separate process, and can be on a separate host. The embedded metastore service communicates with the metastore database over JDBC.

Remote Mode

Cloudera recommends that you use this mode.



In Remote mode, the Hive metastore service runs in its own JVM process; HiveServer2, HCatalog, Cloudera Impala™, and other processes communicate with it using the Thrift network API (configured using the `hive.metastore.uris` property). The metastore service communicates with the metastore database over JDBC (configured using the `javax.jdo.option.ConnectionURL` property). The database, the HiveServer process, and the metastore service can all be on the same host, but running the HiveServer process on a separate host provides better availability and scalability.

The main advantage of Remote mode over Local mode is that Remote mode does not require the administrator to share JDBC login information for the metastore database with each Hive user. [HCatalog](#) requires this mode.

Supported Metastore Databases

See [Supported Databases](#) on page 19 for up-to-date information on supported databases. Cloudera strongly encourages you to use MySQL because it is the most popular with the rest of the Hive user community, and so receives more testing than the other options.

Metastore Memory Requirements

Number of Concurrent Connections	HiveServer2 Heap Size Minimum Recommendation	Hive Metastore Heap Size Minimum Recommendation
Up to 40 concurrent connections (Cloudera recommends splitting HiveServer2 into multiple instances)	12 GB	12 GB

Number of Concurrent Connections	HiveServer2 Heap Size Minimum Recommendation	Hive Metastore Heap Size Minimum Recommendation
and load balancing once you start allocating >12 GB to HiveServer2. ²		
Up to 20 concurrent connections	6 GB	10 GB
Up to 10 concurrent connections	4 GB	8 GB
Single connection	2 GB	4 GB

For information on configuring heap for Hive MetaStore, as well as HiveServer2 and Hive clients, see [Heap Size and Garbage Collection for Hive Components](#) on page 276.

Configuring the Metastore Database

This section describes how to configure Hive to use a remote database, with examples for [MySQL](#) and [PostgreSQL](#), and [Oracle](#).

The configuration properties for the Hive metastore are documented in the [Hive Metastore Administration documentation](#) on the Apache wiki.



Note: For information about additional configuration that may be needed in a secure cluster, see [Hive Authentication](#).

Configuring a Remote MySQL Database for the Hive Metastore

Cloudera recommends you configure a database for the metastore on one or more remote servers (that is, on a host or hosts separate from the HiveServer1 or HiveServer2 process). MySQL is the most popular database to use. Proceed as follows.

1. Install and start MySQL if you have not already done so

To install MySQL on a Red Hat system:

```
$ sudo yum install mysql-server
```

To install MySQL on a SLES system:

```
$ sudo zypper install mysql
$ sudo zypper install libmysqlclient_r15
```

To install MySQL on a Debian/Ubuntu system:

```
$ sudo apt-get install mysql-server
```

After using the command to install MySQL, you may need to respond to prompts to confirm that you do want to complete the installation. After installation completes, start the `mysql` daemon.

On Red Hat systems

```
$ sudo service mysqld start
```

On SLES and Debian/Ubuntu systems

```
$ sudo service mysql start
```

2. Configure the MySQL service and connector

² The objective is to size to reduce impact of Java garbage collection on active processing by the service.

Before you can run the Hive metastore with a remote MySQL database, you must configure a connector to the remote MySQL database, set up the initial database schema, and configure the MySQL user account for the Hive user.

To install the MySQL connector on a RHEL 6 system:

On the Hive Metastore server host, install `mysql-connector-java` and symbolically link the file into the `/usr/lib/hive/lib/` directory.

```
$ sudo yum install mysql-connector-java
$ ln -s /usr/share/java/mysql-connector-java.jar
  /usr/lib/hive/lib/mysql-connector-java.jar
```

To install the MySQL connector on a RHEL 5 system:

Download the MySQL JDBC driver from <http://www.mysql.com/downloads/connector/j/5.1.html>. You will need to sign up for an account if you do not already have one, and log in, before you can download it. Then copy it to the `/usr/lib/hive/lib/` directory. For example:

```
$ sudo cp mysql-connector-java-version/mysql-connector-java-version-bin.jar
  /usr/lib/hive/lib/
```



Note: At the time of publication, *version* was 5.1.31, but the version may have changed by the time you read this. If you are using MySQL version 5.6, you must use version 5.1.26 or higher of the driver.

To install the MySQL connector on a SLES system:

On the Hive Metastore server host, install `mysql-connector-java` and symbolically link the file into the `/usr/lib/hive/lib/` directory.

```
$ sudo zypper install mysql-connector-java
$ ln -s /usr/share/java/mysql-connector-java.jar
  /usr/lib/hive/lib/mysql-connector-java.jar
```

To install the MySQL connector on a Debian/Ubuntu system:

On the Hive Metastore server host, install `mysql-connector-java` and symbolically link the file into the `/usr/lib/hive/lib/` directory.

```
$ sudo apt-get install libmysql-java
$ ln -s /usr/share/java/libmysql-java.jar /usr/lib/hive/lib/libmysql-java.jar
```

Configure MySQL to use a strong password and to start at boot. Note that in the following procedure, your current root password is blank. Press the Enter key when you're prompted for the root password.

To set the MySQL root password:

```
$ sudo /usr/bin/mysql_secure_installation
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
```

```
Reload privilege tables now? [Y/n] Y
All done!
```

To make sure the MySQL server starts at boot:

- On Red Hat systems:

```
$ sudo /sbin/chkconfig mysqld on
$ sudo /sbin/chkconfig --list mysqld
mysqld          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

- On SLES systems:

```
$ sudo chkconfig --add mysql
```

- On Debian/Ubuntu systems:

```
$ sudo chkconfig mysql on
```

3. Create the database and user

The instructions in this section assume you are using [Remote mode](#), and that the MySQL database is installed on a separate host from the metastore service, which is running on a host named `metastorehost` in the example.



Note:

If the metastore service will run on the host where the database is installed, replace `'metastorehost'` in the `CREATE USER` example with `'localhost'`. Similarly, the value of `javax.jdo.option.ConnectionURL` in `/etc/hive/conf/hive-site.xml` (discussed in the next step) must be `jdbc:mysql://localhost/metastore`. For more information on adding MySQL users, see <http://dev.mysql.com/doc/refman/5.5/en/adding-users.html>.

Create the initial database schema. Cloudera recommends using the [Hive schema tool](#) to do this.

If for some reason you decide not to use the schema tool, you can use the `hive-schema-0.12.0.mysql.sql` file instead; that file is located in the `/usr/lib/hive/scripts/metastore/upgrade/mysql` directory. Proceed as follows if you decide to use `hive-schema-0.12.0.mysql.sql`.

Example using `hive-schema-0.12.0.mysql.sql`



Note:

Do this only if you are not using the Hive schema tool.

```
$ mysql -u root -p
Enter password:
mysql> CREATE DATABASE metastore;
mysql> USE metastore;
mysql> SOURCE /usr/lib/hive/scripts/metastore/upgrade/mysql/hive-schema-n.n.n.mysql.sql;
```

You also need a MySQL user account for Hive to use to access the metastore. It is very important to prevent this user account from creating or altering tables in the metastore database schema.



Important: To prevent users from inadvertently corrupting the metastore schema when they use lower or higher versions of Hive, set the `hive.metastore.schema.verification` property to `true` in `/usr/lib/hive/conf/hive-site.xml` on the metastore host.

Example

```
mysql> CREATE USER 'hive'@'metastorehost' IDENTIFIED BY 'mypassword';
...
mysql> REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'hive'@'metastorehost';
mysql> GRANT ALL PRIVILEGES ON metastore.* TO 'hive'@'metastorehost';
mysql> FLUSH PRIVILEGES;
mysql> quit;
```

4. Configure the metastore service to communicate with the MySQL database

This step shows the configuration properties you need to set in `hive-site.xml` (`/usr/lib/hive/conf/hive-site.xml`) to configure the metastore service to communicate with the MySQL database, and provides sample settings. Though you can use the same `hive-site.xml` on all hosts (client, metastore, HiveServer), `hive.metastore.uris` is the only property that **must** be configured on all of them; the others are used only on the metastore host.

Given a MySQL database running on `myhost` and the user account `hive` with the password `mypassword`, set the configuration as follows (overwriting any existing values).

**Note:**

The `hive.metastore.local` property is no longer supported as of Hive 0.10; setting `hive.metastore.uris` is sufficient to indicate that you are using a remote metastore.

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://myhost/metastore</value>
  <description>the URL of the MySQL database</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.jdbc.Driver</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hive</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>mypassword</value>
</property>

<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>>false</value>
</property>

<property>
  <name>datanucleus.fixedDatastore</name>
  <value>>true</value>
</property>

<property>
  <name>datanucleus.autoStartMechanism</name>
  <value>SchemaTable</value>
</property>

<property>
  <name>hive.metastore.uris</name>
  <value>thrift://<n.n.n.n>:9083</value>
  <description>IP address (or fully-qualified domain name) and port of the metastore
  host</description>
</property>
```

```
<property>  
<name>hive.metastore.schema.verification</name>  
<value>true</value>  
</property>
```

Configuring a Remote PostgreSQL Database for the Hive Metastore

Before you can run the Hive metastore with a remote PostgreSQL database, you must configure a connector to the remote PostgreSQL database, set up the initial database schema, and configure the PostgreSQL user account for the Hive user.

1. Install and start PostgreSQL if you have not already done so

To install PostgreSQL on a Red Hat system:

```
$ sudo yum install postgresql-server
```

To install PostgreSQL on a SLES system:

```
$ sudo zypper install postgresql-server
```

To install PostgreSQL on a Debian/Ubuntu system:

```
$ sudo apt-get install postgresql
```

After using the command to install PostgreSQL, you may need to respond to prompts to confirm that you do want to complete the installation. To finish installation on RHEL compatible systems, you need to initialize the database. Please note that this operation is not needed on Ubuntu and SLES systems as it's done automatically on first start:

To initialize database files on Red Hat compatible systems

```
$ sudo service postgresql initdb
```

To ensure that your PostgreSQL server will be accessible over the network, you need to do some additional configuration.

First you need to edit the `postgresql.conf` file. Set the `listen_addresses` property to `*`, to make sure that the PostgreSQL server starts listening on all your network interfaces. Also make sure that the `standard_conforming_strings` property is set to `off`.

You can check that you have the correct values as follows:

On Red-Hat-compatible systems:

```
$ sudo cat /var/lib/pgsql/data/postgresql.conf | grep -e listen -e  
standard_conforming_strings  
listen_addresses = '*'  
standard_conforming_strings = off
```

On SLES systems:

```
$ sudo cat /var/lib/pgsql/data/postgresql.conf | grep -e listen -e  
standard_conforming_strings  
listen_addresses = '*'  
standard_conforming_strings = off
```

On Ubuntu and Debian systems:

```
$ cat /etc/postgresql/9.1/main/postgresql.conf | grep -e listen -e  
standard_conforming_strings
```

```
listen_addresses = '*'
standard_conforming_strings = off
```

You also need to configure authentication for your network in `pg_hba.conf`. You need to make sure that the PostgreSQL user that you will create later in this procedure will have access to the server from a remote host. To do this, add a new line into `pg_hba.conf` that has the following information:

```
host    <database>    <user>    <network address>    <mask>
md5
```

The following example allows all users to connect from all hosts to all your databases:

```
host    all            all            0.0.0.0    0.0.0.0    md5
```



Note:

This configuration is applicable only for a network listener. Using this configuration won't open all your databases to the entire world; the user must still supply a password to authenticate himself, and privilege restrictions configured in PostgreSQL will still be applied.

After completing the installation and configuration, you can start the database server:

Start PostgreSQL Server

```
$ sudo service postgresql start
```

Use `chkconfig` utility to ensure that your PostgreSQL server will start at a boot time. For example:

```
chkconfig postgresql on
```

You can use the `chkconfig` utility to verify that PostgreSQL server will be started at boot time, for example:

```
chkconfig --list postgresql
```

2. Install the PostgreSQL JDBC driver

Before you can run the Hive metastore with a remote PostgreSQL database, you must configure a JDBC driver to the remote PostgreSQL database, set up the initial database schema, and configure the PostgreSQL user account for the Hive user.

To install the PostgreSQL JDBC Driver on a Red Hat 6 system:

On the Hive Metastore server host, install `postgresql-jdbc` package and create symbolic link to the `/usr/lib/hive/lib/` directory. For example:

```
$ sudo yum install postgresql-jdbc
$ ln -s /usr/share/java/postgresql-jdbc.jar /usr/lib/hive/lib/postgresql-jdbc.jar
```

To install the PostgreSQL connector on a Red Hat 5 system:

You need to manually download the PostgreSQL connector from <http://jdbc.postgresql.org/download.html> and move it to the `/usr/lib/hive/lib/` directory. For example:

```
$ wget http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar
$ mv postgresql-9.2-1002.jdbc4.jar /usr/lib/hive/lib/
```

**Note:**

You may need to use a different version if you have a different version of Postgres. You can check the version as follows:

```
$ sudo rpm -qa | grep postgres
```

To install the PostgreSQL JDBC Driver on a SLES system:

On the Hive Metastore server host, install `postgresql-jdbc` and symbolically link the file into the `/usr/lib/hive/lib/` directory.

```
$ sudo zypper install postgresql-jdbc
$ ln -s /usr/share/java/postgresql-jdbc.jar /usr/lib/hive/lib/postgresql-jdbc.jar
```

To install the PostgreSQL JDBC Driver on a Debian/Ubuntu system:

On the Hive Metastore server host, install `libpostgresql-jdbc-java` and symbolically link the file into the `/usr/lib/hive/lib/` directory.

```
$ sudo apt-get install libpostgresql-jdbc-java
$ ln -s /usr/share/java/postgresql-jdbc4.jar /usr/lib/hive/lib/postgresql-jdbc4.jar
```

3. Create the metastore database and user account

Proceed as in the following example, using the appropriate script in `/usr/lib/hive/scripts/metastore/upgrade/postgres/`:

```
$ sudo -u postgres psql
postgres=# CREATE USER hiveuser WITH PASSWORD 'mypassword';
postgres=# CREATE DATABASE metastore;
postgres=# \c metastore;
You are now connected to database 'metastore'.
postgres=# \i
/usr/lib/hive/scripts/metastore/upgrade/postgres/hive-schema-0.12.0.postgres.sql
SET
SET
...
```

Now you need to grant permission for all metastore tables to user `hiveuser`. PostgreSQL does not have statements to grant the permissions for all tables at once; you'll need to grant the permissions one table at a time. You could automate the task with the following SQL script:

**Note:**

If you are running these commands interactively and are still in the Postgres session initiated at the beginning of this step, you do not need to repeat `sudo -u postgres psql`.

```
bash# sudo -u postgres psql
metastore=# \c metastore
metastore=# \pset tuples_only on
metastore=# \o /tmp/grant-privs
metastore=# SELECT 'GRANT SELECT,INSERT,UPDATE,DELETE ON "' || schemaname || '".' ||
||tablename ||'" TO hiveuser ;'
metastore=# FROM pg_tables
metastore=# WHERE tableowner = CURRENT_USER and schemaname = 'public';
metastore=# \o
metastore=# \pset tuples_only off
metastore=# \i /tmp/grant-privs
```

You can verify the connection from the machine where you'll be running the metastore service as follows:

```
psql -h myhost -U hiveuser -d metastore
metastore=#
```

4. Configure the metastore service to communicate with the PostgreSQL database

This step shows the configuration properties you need to set in `hive-site.xml` (`/usr/lib/hive/conf/hive-site.xml`) to configure the metastore service to communicate with the PostgreSQL database. Though you can use the same `hive-site.xml` on all hosts (client, metastore, HiveServer), `hive.metastore.uris` is the only property that **must** be configured on all of them; the others are used only on the metastore host.

Given a PostgreSQL database running on host `myhost` under the user account `hive` with the password `mypassword`, you would set configuration properties as follows.



Note:

- The instructions in this section assume you are using [Remote mode](#), and that the PostgreSQL database is installed on a separate host from the metastore server.
- The `hive.metastore.local` property is no longer supported as of Hive 0.10; setting `hive.metastore.uris` is sufficient to indicate that you are using a remote metastore.

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:postgresql://myhost/metastore</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>org.postgresql.Driver</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hiveuser</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>mypassword</value>
</property>

<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>>false</value>
</property>

<property>
  <name>hive.metastore.uris</name>
  <value>thrift://<n.n.n.n>:9083</value>
  <description>IP address (or fully-qualified domain name) and port of the metastore
  host</description>
</property>

<property>
  <name>hive.metastore.schema.verification</name>
  <value>>true</value>
</property>
```

5. Test connectivity to the metastore

```
$ hive -e "show tables;"
```



Note: This will take a while the first time.

Configuring a Remote Oracle Database for the Hive Metastore

Before you can run the Hive metastore with a remote Oracle database, you must configure a connector to the remote Oracle database, set up the initial database schema, and configure the Oracle user account for the Hive user.

1. Install and start Oracle

The Oracle database is not part of any Linux distribution and must be purchased, downloaded and installed separately. You can use the [Express edition](#), which can be downloaded free from Oracle website.

2. Install the Oracle JDBC Driver

You must download the Oracle JDBC Driver from the Oracle website and put the JDBC JAR file into the `/usr/lib/hive/lib/` directory. For example, the version 6 JAR file is named `ojdbc6.jar`. The driver is available for download [here](#). For information about which Oracle Java versions are supported, see [CDH and Cloudera Manager Supported JDK Versions](#).



Note: These URLs were correct at the time of publication, but the Oracle site is restructured frequently.

```
$ sudo mv ojdbc<version_number>.jar /usr/lib/hive/lib/
```

3. Create the Metastore database and user account

Connect to your Oracle database as an administrator and create the user that will use the Hive metastore.

```
$ sqlplus "sys as sysdba"
SQL> create user hiveuser identified by mypassword;
SQL> grant connect to hiveuser;
SQL> grant all privileges to hiveuser;
```

Connect as the newly created `hiveuser` user and load the initial schema, as in the following example (use the appropriate script for the current release in `/usr/lib/hive/scripts/metastore/upgrade/oracle/`):

```
$ sqlplus hiveuser
SQL> @/usr/lib/hive/scripts/metastore/upgrade/oracle/hive-schema-0.12.0.oracle.sql
```

Connect back as an administrator and remove the power privileges from user `hiveuser`. Then grant limited access to all the tables:

```
$ sqlplus "sys as sysdba"
SQL> revoke all privileges from hiveuser;
SQL> BEGIN
 2     FOR R IN (SELECT owner, table_name FROM all_tables WHERE owner='HIVEUSER') LOOP
 3         EXECUTE IMMEDIATE 'grant SELECT,INSERT,UPDATE,DELETE on
'|R.owner||'.'||R.table_name||' to hiveuser';
 4     END LOOP;
 5 END;
 6
 7 /
```

4. Configure the Metastore Service to Communicate with the Oracle Database

This step shows the configuration properties you need to set in `hive-site.xml` (`/usr/lib/hive/conf/hive-site.xml`) to configure the metastore service to communicate with the Oracle database, and provides sample settings. Though you can use the same `hive-site.xml` on all hosts (client,

metastore, HiveServer), `hive.metastore.uris` is the only property that must be configured on all of them; the others are used only on the metastore host.

Example

Given an Oracle database running on `myhost` and the user account `hiveuser` with the password `mypassword`, set the configuration as follows (overwriting any existing values):

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:oracle:thin:@//myhost/xe</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>oracle.jdbc.OracleDriver</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hiveuser</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>mypassword</value>
</property>

<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>>false</value>
</property>

<property>
  <name>datanucleus.fixedDatastore</name>
  <value>>true</value>
</property>

<property>
  <name>hive.metastore.uris</name>
  <value>thrift://<n.n.n.n>:9083</value>
  <description>IP address (or fully-qualified domain name) and port of the metastore
  host</description>
</property>

<property>
  <name>hive.metastore.schema.verification</name>
  <value>>true</value>
</property>
```

Configuring HiveServer2

You must make the following configuration changes before using HiveServer2. Failure to do so may result in unpredictable behavior.



Warning: HiveServer1 is deprecated in CDH 5.3, and will be removed in a future release of CDH. Users of HiveServer1 should upgrade to [HiveServer2](#) as soon as possible.

HiveServer2 Memory Requirements

Number of Concurrent Connections	HiveServer2 Heap Size Minimum Recommendation	Hive Metastore Heap Size Minimum Recommendation
Up to 40 concurrent connections (Cloudera recommends splitting HiveServer2 into multiple instances)	12 GB	12 GB

Number of Concurrent Connections	HiveServer2 Heap Size Minimum Recommendation	Hive Metastore Heap Size Minimum Recommendation
and load balancing once you start allocating >12 GB to HiveServer2. ³		
Up to 20 concurrent connections	6 GB	10 GB
Up to 10 concurrent connections	4 GB	8 GB
Single connection	2 GB	4 GB



Important: These numbers are general guidance only, and may be affected by factors such as number of columns, partitions, complex joins, and client activity among other things. It is important to review and refine through testing based on your anticipated deployment to arrive at best values for your environment.

For information on configuring heap for HiveServer2, as well as Hive Metastore and Hive clients, see [Heap Size and Garbage Collection for Hive Components](#) on page 276.

Table Lock Manager (Required)

You must properly configure and enable Hive's Table Lock Manager. This requires installing ZooKeeper and setting up a ZooKeeper ensemble; see [ZooKeeper Installation](#).



Important: Failure to do this will prevent HiveServer2 from handling concurrent query requests and may result in data corruption.

Enable the lock manager by setting properties in `/etc/hive/conf/hive-site.xml` as follows (substitute your actual ZooKeeper node names for those in the example):

```
<property>
  <name>hive.support.concurrency</name>
  <description>Enable Hive's Table Lock Manager Service</description>
  <value>true</value>
</property>

<property>
  <name>hive.zookeeper.quorum</name>
  <description>Zookeeper quorum used by Hive's Table Lock Manager</description>
  <value>zk1.myco.com,zk2.myco.com,zk3.myco.com</value>
</property>
```



Important: Enabling the Table Lock Manager without specifying a list of valid Zookeeper quorum nodes will result in unpredictable behavior. Make sure that both properties are properly configured.

(The above settings are also needed if you are still using HiveServer1. HiveServer1 is deprecated; migrate to HiveServer2 as soon as possible.)

`hive.zookeeper.client.port`

If ZooKeeper is not using the default value for `ClientPort`, you need to set `hive.zookeeper.client.port` in `/etc/hive/conf/hive-site.xml` to the same value that ZooKeeper is using. Check `/etc/zookeeper/conf/zoo.cfg` to find the value for `ClientPort`. If `ClientPort` is set to any value other than

³ The objective is to size to reduce impact of Java garbage collection on active processing by the service.

2181 (the default), `sethive.zookeeper.client.port` to the same value. For example, if `ClientPort` is set to 2222, set `hive.zookeeper.client.port` to 2222 as well:

```
<property>
  <name>hive.zookeeper.client.port</name>
  <value>2222</value>
  <description>
    The port at which the clients will connect.
  </description>
</property>
```

JDBC driver

The connection URL format and the driver class are different for `HiveServer2` and `HiveServer1`:

HiveServer version	Connection URL	Driver Class
HiveServer2	<code>jdbc:hive2://<host>:<port></code>	<code>org.apache.hive.jdbc.HiveDriver</code>
HiveServer1	<code>jdbc:hive://<host>:<port></code>	<code>org.apache.hadoop.hive.jdbc.HiveDriver</code>

Authentication

`HiveServer2` can be [configured](#) to authenticate all connections; by default, it allows any client to connect. `HiveServer2` supports either [Kerberos](#) or [LDAP](#) authentication; configure this in the `hive.server2.authentication` property in the `hive-site.xml` file. You can also configure [Pluggable Authentication](#), which allows you to use a custom authentication provider for `HiveServer2`; and [HiveServer2 Impersonation](#), which allows users to execute queries and access HDFS files as the connected user rather than the super user who started the `HiveServer2` daemon. For more information, see [Hive Security Configuration](#).

Running HiveServer2 and HiveServer Concurrently



Important: Cloudera strongly recommends running `HiveServer2` instead of the original `HiveServer` (`HiveServer1`) package; `HiveServer1` is deprecated.

`HiveServer2` and `HiveServer1` can be run concurrently on the same system, sharing the same data sets. This allows you to run `HiveServer1` to support, for example, Perl or Python scripts that use the native `HiveServer1` Thrift bindings.

Both `HiveServer2` and `HiveServer1` bind to port 10000 by default, so at least one of them must be configured to use a different port. You can set the port for `HiveServer2` in `hive-site.xml` by means of the `hive.server2.thrift.port` property. For example:

```
<property>
  <name>hive.server2.thrift.port</name>
  <value>10001</value>
  <description>TCP port number to listen on, default 10000</description>
</property>
```

You can also specify the port (and the host IP address in the case of `HiveServer2`) by setting these environment variables:

HiveServer version	Port	Host Address
HiveServer2	<code>HIVE_SERVER2_THRIFT_PORT</code>	<code>HIVE_SERVER2_THRIFT_BIND_HOST</code>
HiveServer1	<code>HIVE_PORT</code>	< Host bindings cannot be specified >

Starting the Metastore

**Important:**

If you are running the metastore in [Remote mode](#), you **must** start the metastore before starting HiveServer2.

To run the metastore as a daemon, the command is:

```
$ sudo service hive-metastore start
```

File System Permissions

Your Hive data is stored in HDFS, normally under `/user/hive/warehouse`. The `/user/hive` and `/user/hive/warehouse` directories need to be created if they do not already exist. Make sure this location (or any path you specify as `hive.metastore.warehouse.dir` in your `hive-site.xml`) exists and is writable by the users whom you expect to be creating tables.



Important: If you are using Sentry, do not follow the instructions on this page. See [Before Enabling the Sentry Service](#) for information on how to set up the Hive warehouse directory permissions for use with Sentry.

In addition, each user submitting queries must have an HDFS home directory. `/tmp` (on the local file system) must be world-writable, as Hive makes extensive use of it.

[HiveServer2 Impersonation](#) allows users to execute queries and access HDFS files as the connected user.

If you do not enable impersonation, HiveServer2 by default executes all Hive tasks as the user ID that starts the Hive server; for clusters that use Kerberos authentication, this is the ID that maps to the [Kerberos principal](#) used with HiveServer2. Setting permissions to `1777`, as recommended above, allows this user access to the Hive warehouse directory.

You can change this default behavior by setting `hive.metastore.execute.setugi` to `true` *on both the server and client*. This setting causes the metastore server to use the client's user and group permissions.

Starting, Stopping, and Using HiveServer2

HiveServer2 is an improved version of HiveServer that supports Kerberos authentication and multi-client concurrency. Cloudera recommends HiveServer2.

**Warning:**

If you are running the metastore in [Remote mode](#), you must start the Hive metastore before you start HiveServer2. HiveServer2 tries to communicate with the metastore as part of its initialization bootstrap. If it is unable to do this, it fails with an error.

To start HiveServer2:

```
$ sudo service hive-server2 start
```

To stop HiveServer2:

```
$ sudo service hive-server2 stop
```

To confirm that HiveServer2 is working, start the `beeline` CLI and use it to execute a `SHOW TABLES` query on the HiveServer2 process:

```
$ /usr/lib/hive/bin/beeline
beeline> !connect jdbc:hive2://localhost:10000 username password
org.apache.hive.jdbc.HiveDriver
0: jdbc:hive2://localhost:10000> SHOW TABLES;
show tables;
+-----+
| tab_name |
+-----+
+-----+
No rows selected (0.238 seconds)
0: jdbc:hive2://localhost:10000>
```

Using the Beeline CLI

Beeline is the CLI (command-line interface) developed specifically to interact with HiveServer2. It is based on the [SQLLine CLI](#) written by Marc Prud'hommeaux.



Note:

Cloudera does not currently support using the Thrift HTTP protocol to connect Beeline to HiveServer2 (meaning that you cannot set `hive.server2.transport.mode=http`). Use the Thrift TCP protocol.

Use the following commands to start `beeline` and connect to a running HiveServer2 process. In this example the HiveServer2 process is running on `localhost` at port 10000:

```
$ beeline
beeline> !connect jdbc:hive2://localhost:10000 username password
org.apache.hive.jdbc.HiveDriver
0: jdbc:hive2://localhost:10000>
```



Note:

If you are using HiveServer2 on a cluster that does *not* have Kerberos security enabled, then the password is arbitrary in the command for starting Beeline.

If you are using HiveServer2 on a cluster that does have Kerberos security enabled, see [HiveServer2 Security Configuration](#).

As of CDH 5.2, there are still some Hive CLI features that are *not* available with Beeline. For example:

- Beeline does not show query logs like the Hive CLI
- When adding JARs to HiveServer2 with Beeline, the JARs must be on the HiveServer2 host.

At present the best source for documentation on Beeline is the original [SQLLine documentation](#).

Starting HiveServer1 and the Hive Console



Important:

Because of concurrency and security issues, HiveServer1 is deprecated in CDH 5 and will be removed in a future release. Cloudera recommends you migrate to [Beeline](#) and HiveServer2 as soon as possible. The Hive Console is not needed if you are using Beeline with HiveServer2.

To start HiveServer1:

```
$ sudo service hiveserver start
```

See also [Running HiveServer2 and HiveServer Concurrently](#) on page 291.

To start the Hive console:

```
$ hive
hive>
```

To confirm that Hive is working, issue the `show tables;` command to list the Hive tables; be sure to use a semi-colon after the command:

```
hive> show tables;
OK
Time taken: 10.345 seconds
```

Using Hive with HBase

To allow Hive scripts to use HBase, proceed as follows.

1. [Install](#) the `hive-hbase` package.
2. Add the following statements to the top of each script. Replace the `<Guava_version>` string with the current version numbers for Guava. (You can find current version numbers for CDH dependencies such as Guava in CDH's root `pom.xml` file for the current release, for example [cdh-root-5.0.0.pom](#).)

```
ADD JAR /usr/lib/hive/lib/zookeeper.jar;
ADD JAR /usr/lib/hive/lib/hive-hbase-handler.jar
ADD JAR /usr/lib/hive/lib/guava-<Guava_version>.jar;
ADD JAR /usr/lib/hive/lib/hbase-client.jar;
ADD JAR /usr/lib/hive/lib/hbase-common.jar;
ADD JAR /usr/lib/hive/lib/hbase-hadoop-compat.jar;
ADD JAR /usr/lib/hive/lib/hbase-hadoop2-compat.jar;
ADD JAR /usr/lib/hive/lib/hbase-protocol.jar;
ADD JAR /usr/lib/hive/lib/hbase-server.jar;
ADD JAR /usr/lib/hive/lib/htrace-core.jar;
```

Using the Hive Schema Tool

Schema Version Verification

Hive now records the schema version in the metastore database and verifies that the metastore schema version is compatible with the Hive binaries that are going to access the metastore. The Hive properties to implicitly create or alter the existing schema are disabled by default. Hence, Hive will not attempt to change the metastore schema implicitly. When you execute a Hive query against an old schema, it will fail to access the metastore displaying an error message as follows:

```
$ build/dist/bin/hive -e "show tables"
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask.
java.lang.RuntimeException: Unable to instantiate
org.apache.hadoop.hive.metastore.HiveMetaStoreClient
```

The error log will contain an entry similar to the following:

```
...
Caused by: MetaException(message:Version information not found in metastore. )
    at org.apache.hadoop.hive.metastore.ObjectStore.checkSchema(ObjectStore.java:5638)
...
```

To suppress the schema check and allow the metastore to implicitly modify the schema, you need to set the `hive.metastore.schema.validation` configuration property to `false` in `hive-site.xml`.

Using schematool

Use the Hive `schematool` to initialize the metastore schema for the current Hive version or to upgrade the schema from an older version. The tool tries to find the current schema from the metastore if it is available there.

The `schematool` determines the SQL scripts that are required to initialize or upgrade the schema and then executes those scripts against the backend database. The metastore database connection information such as JDBC URL, JDBC

driver and database credentials are extracted from the Hive configuration. You can provide alternate database credentials if needed.

The following options are available as part of the `schematool` package.

```
$ schematool -help
usage: schemaTool
  -dbType <databaseType>      Metastore database type
  -dryRun                       List SQL scripts (no execute)

  -help                         Print this message
  -info                         Show config and schema details
  -initSchema                   Schema initialization
  -initSchemaTo <initTo>       Schema initialization to a version
  -passWord <password>         Override config file password
  -upgradeSchema                Schema upgrade
  -upgradeSchemaFrom <upgradeFrom> Schema upgrade from a version
  -userName <user>             Override config file user name
  -verbose                       Only print SQL statements
```

The `dbType` option should always be specified and can be one of the following:

```
derby|mysql|postgres|oracle
```

Usage Examples

- Initialize your metastore to the current schema for a new Hive setup using the `initSchema` option.

```
$ schematool -dbType derby -initSchema
Metastore connection URL:      jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:     APP
Starting metastore schema initialization to <new_version>
Initialization script hive-schema-<new_version>.derby.sql
Initialization script completed
schemaTool completed
```

- Get schema information using the `info` option.

```
$ schematool -dbType derby -info
Metastore connection URL:      jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:     APP
Hive distribution version:     <new_version>
Required schema version:      <new_version>
Metastore schema version:     <new_version>
schemaTool completed
```

- If you attempt to get schema information from older metastores that did not store version information, the tool will report an error as follows.

```
$ schematool -dbType derby -info
Metastore connection URL:      jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:     APP
Hive distribution version:     <new_version>
Required schema version:      <new_version>
org.apache.hadoop.hive.metastore.HiveMetaException: Failed to get schema version.
*** schemaTool failed ***
```

- You can upgrade schema from a CDH 4 release by specifying the `upgradeSchemaFrom` option.

```
$ schematool -dbType derby -upgradeSchemaFrom 0.10.0
Metastore connection URL:      jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :  org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:     APP
Starting upgrade metastore schema from version 0.10.0 to <new_version>
```

Installing Cloudera Manager and CDH

```
Upgrade script upgrade-0.10.0-to-<new_version>.derby.sql
Completed upgrade-0.10.0-to-<new_version>.derby.sql
Upgrade script upgrade-0.11.0-to-<new_version>.derby.sql
Completed upgrade-0.11.0-to-<new_version>.derby.sql
schemaTool completed
```

The Hive versions of the older CDH releases are:

CDH Releases	Hive Version
CDH 3	0.7.0
CDH 4.0	0.8.0
CDH 4.1	0.9.0
CDH 4.2 and higher	0.10.0

- If you want to find out all the required scripts for a schema upgrade, use the `dryRun` option.

```
$ build/dist/bin/schematool -dbType derby -upgradeSchemaFrom 0.7.0 -dryRun
13/09/27 17:06:31 WARN conf.Configuration: hive.server2.enable.impersonation is
deprecated. Instead, use hive.server2.enable.doAs
Metastore connection URL:          jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :     org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:        APP
Starting upgrade metastore schema from version 0.7.0 to <new_version>
Upgrade script upgrade-0.7.0-to-0.8.0.derby.sql
Upgrade script upgrade-0.8.0-to-0.9.0.derby.sql
Upgrade script upgrade-0.9.0-to-0.10.0.derby.sql
Upgrade script upgrade-0.10.0-to-0.11.0.derby.sql
Upgrade script upgrade-0.11.0-to-<new_version>.derby.sql
schemaTool completed
```

Installing the Hive JDBC on Clients

If you want to install only the JDBC on your Hive clients, proceed as follows.



Note:

The CDH 5.2 Hive JDBC driver is not wire-compatible with the CDH 5.1 version of HiveServer2. Make sure you upgrade Hive clients and all other Hive hosts in tandem: the server first, and then the clients.

1. Install the package (it is included in CDH packaging). Use one of the following commands, depending on the target operating system:

- On Red-Hat-compatible systems:

```
$ sudo yum install hive-jdbc
```

- On SLES systems:

```
$ sudo zypper install hive-jdbc
```

- On Ubuntu or Debian systems:

```
$ sudo apt-get install hive-jdbc
```

2. Add `/usr/lib/hive/lib/*.jar` and `/usr/lib/hadoop/*.jar` to your classpath.

You are now ready to run your JDBC client. For more information see the [Hive Client](#) document.

Setting HADOOP_MAPRED_HOME

- For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop in a YARN installation, make sure that the `HADOOP_MAPRED_HOME` environment variable is set correctly, as follows:

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

- For each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop in an MRv1 installation, set the `HADOOP_MAPRED_HOME` environment variable as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

Configuring the Metastore to use HDFS High Availability

See [Upgrading the Hive Metastore to use HDFS HA](#).

Troubleshooting Hive

This section provides guidance on problems you may encounter while installing, upgrading, or running Hive.

Too Many Small Partitions

It can be tempting to partition your data into many small partitions to try to increase speed and concurrency. However, Hive functions best when data is partitioned into larger partitions. For example, consider partitioning a 100 TB table into 10,000 partitions, each 10 GB in size. In addition, do not use more than 10,000 partitions per table. Having too many small partitions puts significant strain on the Hive MetaStore and does not improve performance.

Hive Queries Fail with "Too many counters" Error

Explanation

Hive operations use various counters while executing MapReduce jobs. These per-operator counters are enabled by the configuration setting `hive.task.progress`. This is disabled by default; if it is enabled, Hive may create a large number of counters (4 counters per operator, plus another 20).



Note:

If dynamic partitioning is enabled, Hive implicitly enables the counters during data load.

By default, CDH restricts the number of MapReduce counters to 120. Hive queries that require more counters will fail with the "Too many counters" error.

What To Do

If you run into this error, set `mapreduce.job.counters.max` in `mapred-site.xml` to a higher value.

Viewing the Hive Documentation

For additional Hive documentation, see [the Apache Hive wiki](#).

To view the Cloudera video tutorial about using Hive, see [Introduction to Apache Hive](#).

HttpFS Installation



Important: Running Services

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

Use the following sections to install and configure HttpFS:

About HttpFS

Apache Hadoop HttpFS is a service that provides HTTP access to HDFS.

HttpFS has a REST HTTP API supporting all HDFS filesystem operations (both read and write).

Common HttpFS use cases are:

- Read and write data in HDFS using HTTP utilities (such as `curl` or `wget`) and HTTP libraries from languages other than Java (such as Perl).
- Transfer data between HDFS clusters running different versions of Hadoop (overcoming RPC versioning issues), for example using Hadoop DistCp.
- Read and write data in HDFS in a cluster behind a firewall. (The HttpFS server acts as a gateway and is the only system that is allowed to send and receive data through the firewall).

HttpFS supports Hadoop pseudo-authentication, HTTP SPNEGO Kerberos, and additional authentication mechanisms using a plugin API. HttpFS also supports Hadoop proxy user functionality.

The `webhdfs` client file system implementation can access HttpFS using the Hadoop filesystem command (`hadoop fs`), by using Hadoop DistCp, and from Java applications using the Hadoop file system Java API.

The HttpFS HTTP REST API is interoperable with the WebHDFS REST HTTP API.

For more information about HttpFS, see [Hadoop HDFS over HTTP](#).

HttpFS Packaging

There are two packaging options for installing HttpFS:

- The `hadoop-httpfs` RPM package
- The `hadoop-httpfs` Debian package

You can also download a Hadoop tarball, which includes HttpFS, from [here](#).

HttpFS Prerequisites

Prerequisites for installing HttpFS are:

- An [operating system supported by CDH 5](#)
- Java: see [Java Development Kit Installation](#) for details



Note:

To see which version of HttpFS is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#). CDH 5 Hadoop works with the CDH 5 version of HttpFS.

Installing HttpFS

HttpFS is distributed in the `hadoop-httpfs` package. To install it, use your preferred package manager application. Install the package on the system that will run the HttpFS server.



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install the HttpFS package on a Red Hat-compatible system:

```
$ sudo yum install hadoop-httpfs
```

To install the HttpFS server package on a SLES system:

```
$ sudo zypper install hadoop-httpfs
```

To install the HttpFS package on an Ubuntu or Debian system:

```
$ sudo apt-get install hadoop-httpfs
```



Note:

Installing the `httpfs` package creates an `httpfs` service configured to start HttpFS at system startup time.

You are now ready to configure HttpFS. See the [next section](#).

Configuring HttpFS

When you install HttpFS from an RPM or Debian package, HttpFS creates all configuration, documentation, and runtime files in the standard Unix directories, as follows.

Type of File	Where Installed
Binaries	<code>/usr/lib/hadoop-httpfs/</code>
Configuration	<code>/etc/hadoop-httpfs/conf/</code>
Documentation	<i>for SLES:</i> <code>/usr/share/doc/packages/hadoop-httpfs/</code>
	<i>for other platforms:</i> <code>/usr/share/doc/hadoop-httpfs/</code>

Type of File	Where Installed
Data	<code>/var/lib/hadoop-httpfs/</code>
Logs	<code>/var/log/hadoop-httpfs/</code>
temp	<code>/var/tmp/hadoop-httpfs/</code>
PID file	<code>/var/run/hadoop-httpfs/</code>

Configuring the HDFS HttpFS Will Use

HttpFS reads the HDFS configuration from the `core-site.xml` and `hdfs-site.xml` files in `/etc/hadoop/conf/`. If necessary edit those files to configure the HDFS HttpFS will use.

Configuring the HttpFS Proxy User

Edit `core-site.xml` and define the Linux user that will run the HttpFS server as a Hadoop proxy user. For example:

```
<property>
<name>hadoop.proxyuser.httpfs.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.httpfs.groups</name>
<value>*</value>
</property>
```

Then restart Hadoop to make the proxy user configuration active.

Configuring HttpFS with Kerberos Security

To configure HttpFS with Kerberos Security, see [HttpFS Authentication](#).

Starting the HttpFS Server

After you have completed all of the required configuration steps, you can start HttpFS:

```
$ sudo service hadoop-httpfs start
```

If you see the message `Server httpfs started!`, `status NORMAL` in the `httpfs.log` log file, the system has started successfully.



Note:

By default, HttpFS server runs on port 14000 and its URL is `http://<HTTPFS_HOSTNAME>:14000/webhdfs/v1`.

Stopping the HttpFS Server

To stop the HttpFS server:

```
$ sudo service hadoop-httpfs stop
```

Using the HttpFS Server with curl

You can use a tool such as `curl` to access HDFS using HttpFS. For example, to obtain the home directory of the user `babu`, use a command such as this:

```
$ curl "http://localhost:14000/webhdfs/v1?op=gethomedirectory&user.name=babu"
```

You should see output such as this:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie:
hadoop.auth="u=babu&p=babu&t=simple&e=1332977755010&s=JVfT4T785K4jeeLNWXK68rc/0xI=" ;
Version=1; Path=/
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 28 Mar 2012 13:35:55 GMT

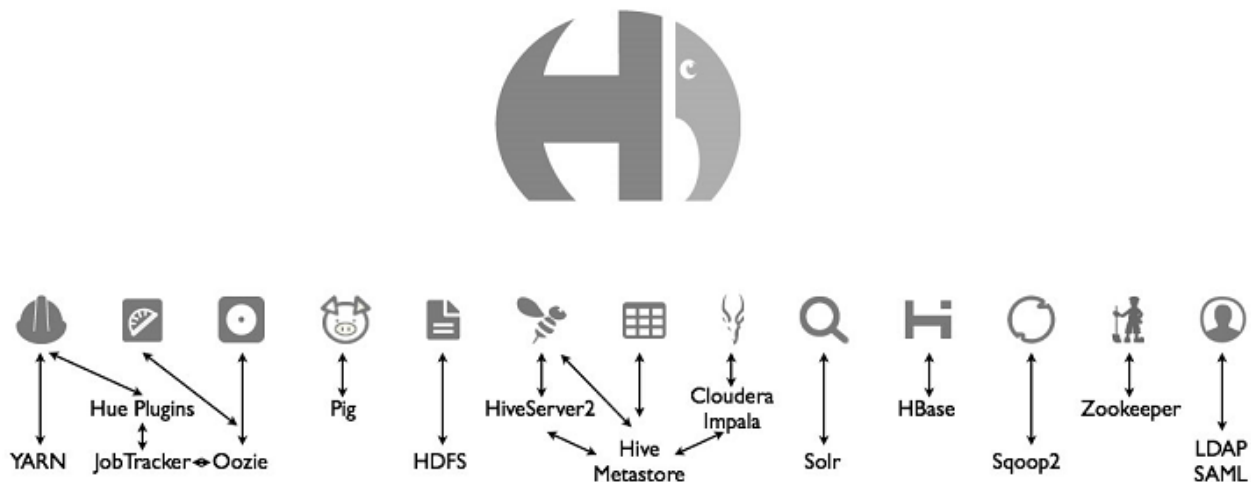
{"Path": "\/user\/babu"}
```

See the [WebHDFS REST API web page](#) for complete documentation of the API.

Hue Installation

Hue is a suite of applications that provide web-based access to CDH components and a platform for building [custom applications](#).

The following figure illustrates how Hue works. Hue Server is a "container" web application that sits in between your CDH installation and the browser. It hosts the Hue applications and communicates with various servers that interface with CDH components.



The Hue Server uses a [database](#) to manage session, authentication, and Hue application data. For example, the Job Designer application stores job designs in the database.

In a CDH cluster, the Hue Server runs on a special node. For optimal performance, this should be one of the nodes within your cluster, though it can be a remote node as long as there are no overly restrictive firewalls. For small clusters of less than 10 nodes, you can use your existing master node as the Hue Server. In a pseudo-distributed installation, the Hue Server runs on the same machine as the rest of your CDH services.



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).



Note: Running Services

When starting, stopping and restarting CDH components, always use the `service (8)` command rather than running scripts in `/etc/init.d` 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 scripts in `/etc/init.d`, 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).)

Follow the instructions in the following sections to upgrade, install, configure, and administer Hue.

- [Supported Browsers](#)
- [Upgrading Hue](#) on page 302
- [Installing Hue](#) on page 304
- [Configuring CDH Components for Hue](#) on page 306
- [Hue Configuration](#) on page 310
- [Administering Hue](#) on page 319
- [Hue User Guide](#)

Supported Browsers for Hue

The Hue UI is supported on the following browsers:

- Windows: Chrome, Firefox 17+, Internet Explorer 9+, Safari 5+
- Linux: Chrome, Firefox 17+
- Mac: Chrome, Firefox 17+, Safari 5+

Upgrading Hue



Note:

To see which version of Hue is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Upgrading Hue from CDH 4 to CDH 5

If you have already removed Hue as part of your upgrade to CDH 5, skip to [Installing and Configuring Hue](#).

Step 1: Stop the Hue Server

See [Starting and Stopping the Hue Server](#) on page 320.

Step 2: Uninstall the Old Version of Hue

- On RHEL systems:

```
$ sudo yum remove hue-common
```

- On SLES systems:

```
$ sudo zypper remove hue-common
```

- On Ubuntu or Debian systems:

```
sudo apt-get remove hue-common
```

Step 3: Install Hue 3.x

Follow the instructions under [Installing Hue](#).

If Using MySQL as Hue Backend: You may face issues after the upgrade if the default engine for MySQL doesn't match the engine used by the Hue tables. To confirm the match:

1. Open the `my.cnf` file for MySQL, search for `"default-storage-engine"` and note its value.
2. Connect to MySQL and run the following commands:

```
use hue;
show create table auth_user;
```

3. Search for the `"ENGINE="` line and confirm that its value matches the one for the `"default-storage-engine"` above.

If the default engines do not match, Hue will display a warning on its start-up page ([http://\\$HUE_HOST:\\$HUE_PORT/about](http://$HUE_HOST:$HUE_PORT/about)). Work with your database administrator to convert the current Hue MySQL tables to the engine in use by MySQL, as noted by the `"default-storage-engine"` property.

**Important: Configuration files**

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Step 4: Start the Hue Server

See [Starting and Stopping the Hue Server](#) on page 320.

Upgrading Hue from an Earlier CDH 5 Release

You can upgrade Hue either as part of an overall upgrade to the latest CDH 5 release (see [Upgrading from an Earlier CDH 5 Release to the Latest Release](#)) or independently. To upgrade Hue from an earlier CDH 5 release to the latest CDH 5 release, proceed as follows.

Step 1: Stop the Hue Server

See [Starting and Stopping the Hue Server](#) on page 320.

**Warning:**

You **must** stop Hue. If Hue is running during the upgrade, the new version will not work correctly.

Step 2: Install the New Version of Hue

Follow the instructions under [Installing Hue](#) on page 304.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Step 3: Start the Hue Server

See [Starting and Stopping the Hue Server](#) on page 320.

Installing Hue

This section describes Hue installation and configuration on a cluster. The steps in this section apply whether you are installing on a single machine in pseudo-distributed mode, or on a cluster.

Install Python 2.6 or 2.7

CDH 5 Hue will only work with the default Python version of the operating system on which it is being installed. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue. However, RHEL 5 and CentOS 5 users will have to download Python 2.6 from the EPEL repository as described below.

To install packages from the EPEL repository, download the appropriate repository rpm packages to your machine and then install Python using `yum`. For example, use the following commands for RHEL 5 or CentOS 5:

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

Installing the Hue Packages



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper/YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

You must install the `hue-common` package on the machine where you will run the Hue Server. In addition, if you will be using Hue with MRv1, you must install the `hue-plugins` package on the system where you are running the JobTracker. (In pseudo-distributed mode, these will all be the same system.)

The `hue` meta-package installs the `hue-common` package and all the Hue applications; you also need to install `hue-server`, which contains the Hue start and stop scripts.



Note: If you do not know which system your JobTracker is on, install the `hue-plugins` package on every node in the cluster.

On RHEL systems:

- On the Hue Server machine, install the hue package:

```
$ sudo yum install hue
```

- For MRv1: on the system that hosts the JobTracker, if different from the Hue server machine, install the hue-plugins package:

```
$ sudo yum install hue-plugins
```

On SLES systems:

- On the Hue Server machine, install the hue package:

```
$ sudo zypper install hue
```

- For MRv1: on the system that hosts the JobTracker, if different from the Hue server machine, install the hue-plugins package:

```
$ sudo zypper install hue-plugins
```

On Ubuntu or Debian systems:

- On the Hue Server machine, install the hue package:

```
$ sudo apt-get install hue
```

- For MRv1: on the system that hosts the JobTracker, if different from the Hue server machine, install the hue-plugins package:

```
$ sudo apt-get install hue-plugins
```



Important: For all operating systems, restart the Hue service once installation is complete. See [Starting and Stopping the Hue Server](#) on page 320.

Hue Dependencies

The following table shows the components that are dependencies for the different Hue applications:

Component	Dependent Applications
HDFS	Core, File Browser
MapReduce	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
YARN	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
Oozie	Job Designer, Oozie Editor/Dashboard
Hive	Hive Editor, Metastore Tables
Impala	Impala Editor, Metastore Tables
HBase	HBase Browser
Pig	Pig Editor, Oozie
Search	Solr Search
Spark	Spark
Sentry	Hadoop Security

Component	Dependent Applications
Sqoop	Oozie
Sqoop 2	Sqoop Transfer
ZooKeeper	ZooKeeper

Configuring CDH Components for Hue

To enable communication between the Hue Server and CDH components, you must make minor changes to your CDH installation by adding the properties described in this section to your CDH configuration files in `/etc/hadoop-0.20/conf/` or `/etc/hadoop/conf/`. If you are installing on a cluster, make the following configuration changes to your existing CDH installation on **each node** in your cluster.

WebHDFS or HttpFS Configuration

Hue can use either of the following to access HDFS data:

- **WebHDFS** provides high-speed data transfer with good locality because clients talk directly to the DataNodes inside the Hadoop cluster.
- **HttpFS** is a proxy service appropriate for integration with external systems that are not behind the cluster's firewall.

Both WebHDFS and HttpFS use the HTTP REST API so they are fully interoperable, but Hue must be configured to use one or the other. For HDFS HA deployments, you must use HttpFS.

To configure Hue to use either WebHDFS or HttpFS, do the following steps:

1. For WebHDFS only:

- Add the following property in `hdfs-site.xml` to enable WebHDFS in the NameNode and DataNodes:

```
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
```

- Restart your HDFS cluster.

2. Configure Hue as a proxy user for all other users and groups, meaning it may submit a request on behalf of any other user:

WebHDFS: Add to `core-site.xml`:

```
<!-- Hue WebHDFS proxy user setting -->
<property>
  <name>hadoop.proxyuser.hue.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.hue.groups</name>
  <value>*</value>
</property>
```

HttpFS: Verify that `/etc/hadoop-httpfs/conf/httpfs-site.xml` has the following configuration:

```
<!-- Hue HttpFS proxy user setting -->
<property>
  <name>httpfs.proxyuser.hue.hosts</name>
  <value>*</value>
</property>
<property>
  <name>httpfs.proxyuser.hue.groups</name>
  <value>*</value>
</property>
```

If the configuration is not present, add it to `/etc/hadoop-httpfs/conf/httpfs-site.xml` and restart the HttpFS daemon.

3. Verify that `core-site.xml` has the following configuration:

```
<property>
<name>hadoop.proxyuser.httpfs.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.httpfs.groups</name>
<value>*</value>
</property>
```

If the configuration is not present, add it to `/etc/hadoop/conf/core-site.xml` and restart Hadoop.

4. With root privileges, update `hadoop.hdfs_clusters.default.webhdfs_url` in `hue.ini` to point to the address of either WebHDFS or HttpFS.

```
[hadoop]
[[hdfs_clusters]]
[[[default]]]
# Use WebHdfs/HttpFs as the communication mechanism.
```

WebHDFS:

```
...
webhdfs_url=http://FQDN:50070/webhdfs/v1/
```

HttpFS:

```
...
webhdfs_url=http://FQDN:14000/webhdfs/v1/
```



Note: If the `webhdfs_url` is uncommented and explicitly set to the empty value, Hue falls back to using the Thrift plugin used in Hue 1.x. This is not recommended.

MRv1 Configuration

Hue communicates with the JobTracker using the Hue plugin, which is a `.jar` file that should be placed in your MapReduce `lib` directory.



Important: The `hue-plugins` package installs the Hue plugins in your MapReduce `lib` directory, `/usr/lib/hadoop/lib`. If you are not using the package-based installation procedure, perform the following steps to install the Hue plugins.

If your JobTracker and Hue Server are located on the same host, copy the file over. If you are currently using CDH 4, your MapReduce library directory might be in `/usr/lib/hadoop/lib`.

```
$ cd /usr/lib/hue
$ cp desktop/libs/hadoop/java-lib/hue-plugins-*.jar /usr/lib/hadoop-0.20-mapreduce/lib
```

If your JobTracker runs on a different host, `scp` the Hue plugins `.jar` file to the JobTracker host.

Add the following properties to `mapred-site.xml`:

```
<property>
  <name>jobtracker.thrift.address</name>
  <value>0.0.0.0:9290</value>
</property>
<property>
  <name>mapred.jobtracker.plugins</name>
```

```
<value>org.apache.hadoop.thriftfs.ThriftJobTrackerPlugin</value>
<description>Comma-separated list of jobtracker plug-ins to be activated.</description>
</property>
```

You can confirm that the plugins are running correctly by tailing the daemon logs:

```
$ tail --lines=500 /var/log/hadoop-0.20-mapreduce/hadoop*jobtracker*.log | grep
ThriftPlugin
2009-09-28 16:30:44,337 INFO org.apache.hadoop.thriftfs.ThriftPluginServer: Starting
Thrift server
2009-09-28 16:30:44,419 INFO org.apache.hadoop.thriftfs.ThriftPluginServer:
Thrift server listening on 0.0.0.0:9290
```



Note: If you enable ACLs in the JobTracker, you must add users to the JobTracker `mapred.queue.default.acl-administer-jobs` property in order to allow Hue to display jobs in the Job Browser application. For example, to give the hue user access to the JobTracker, you would add the following property:

```
<property>
  <name>mapred.queue.default.acl-administer-jobs</name>
  <value>hue</value>
</property>
```

Repeat this for every user that requires access to the job details displayed by the JobTracker.

If you have any mapred queues besides "default", you must add a property for each queue:

```
<property>
<name>mapred.queue.default.acl-administer-jobs</name>
<value>hue</value>
</property>
<property>
<name>mapred.queue.queue1.acl-administer-jobs</name>
<value>hue</value>
</property>
<property>
<name>mapred.queue.queue2.acl-administer-jobs</name>
<value>hue</value>
</property>
```

Oozie Configuration

In order to run DistCp, Streaming, Pig, Sqoop, and Hive jobs in Job Designer or the Oozie Editor/Dashboard application, you must make sure the Oozie shared libraries are installed for the correct version of MapReduce (MRv1 or YARN). See [Installing the Oozie ShareLib in Hadoop HDFS](#) for instructions.

To configure Hue as a default proxy user, add the following properties to `/etc/oozie/conf/oozie-site.xml`:

```
<!-- Default proxyuser configuration for Hue -->
<property>
  <name>oozie.service.ProxyUserService.proxyuser.hue.hosts</name>
  <value>*</value>
</property>
<property>
  <name>oozie.service.ProxyUserService.proxyuser.hue.groups</name>
  <value>*</value>
</property>
```

Search Configuration

See [Search Configuration](#) on page 314 for details on how to configure the Search application for Hue.

HBase Configuration

See [HBase Configuration](#) on page 314 for details on how to configure the HBase Browser application.



Note: HBase Browser requires Thrift Server 1 to be running.

Hive Configuration

The Beeswax daemon has been replaced by HiveServer2. Hue should therefore point to a running HiveServer2. This change involved the following major updates to the `[beeswax]` section of the Hue configuration file, `hue.ini`.

```
[beeswax]
# Host where Hive server Thrift daemon is running.
# If Kerberos security is enabled, use fully-qualified domain name (FQDN).
## hive_server_host=<FQDN of HiveServer2>

# Port where HiveServer2 Thrift server runs on.
## hive_server_port=10000
```

Existing Hive Installation

In the Hue configuration file `hue.ini`, modify `hive_conf_dir` to point to the directory containing `hive-site.xml`.

No Existing Hive Installation

Familiarize yourself with the configuration options in `hive-site.xml`. See [Hive Installation](#). Having a `hive-site.xml` is optional but often useful, particularly on setting up a metastore. You can locate it using the `hive_conf_dir` configuration variable.

Permissions

See [File System Permissions](#) in the Hive Installation section.

Other Hadoop Settings

HADOOP_CLASSPATH

If you are setting `$HADOOP_CLASSPATH` in your `hadoop-env.sh`, be sure to set it in such a way that user-specified options are preserved. For example:

Correct:

```
# HADOOP_CLASSPATH=<your_additions>:$HADOOP_CLASSPATH
```

Incorrect:

```
# HADOOP_CLASSPATH=<your_additions>
```

This enables certain components of Hue to add to Hadoop's classpath using the environment variable.

hadoop.tmp.dir

If your users are likely to be submitting jobs both using Hue and from the same machine via the command line interface, they will be doing so as the `hue` user when they are using Hue and via their own user account when they are using the command line. This leads to some contention on the directory specified by `hadoop.tmp.dir`, which defaults to `/tmp/hadoop- $\{user.name\}$` . Specifically, `hadoop.tmp.dir` is used to unpack JARs in `/usr/lib/hadoop`. One work around to this is to set `hadoop.tmp.dir` to `/tmp/hadoop- $\{user.name\}$ - $\{hue.suffix\}$` in the `core-site.xml` file:

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/tmp/hadoop- $\{user.name\}$ - $\{hue.suffix\}$ </value>
</property>
```

Unfortunately, when the `hue.suffix` variable is unset, you'll end up with directories named `/tmp/hadoop-user.name- $\{$ hue.suffix $\}$` in `/tmp`. Despite that, Hue will still work.

Hue Configuration

This section describes configuration you perform in the Hue configuration file `hue.ini`. The location of the Hue configuration file varies depending on how Hue is installed. The location of the Hue configuration folder is displayed when you view the Hue configuration.



Note: Only the root user can edit the Hue configuration file.

Viewing the Hue Configuration



Note: You must be a Hue superuser to view the Hue configuration.

When you log in to Hue, the start-up page displays information about any misconfiguration detected.

To view the Hue configuration, do one of the following:

- Visit `http://myserver:port` and click the **Configuration** tab.
- Visit `http://myserver:port/dump_config`.

Hue Server Configuration

This section describes Hue Server settings.

Specifying the Hue Server HTTP Address

These configuration properties are under the `[desktop]` section in the Hue configuration file.

Hue uses the CherryPy web server. You can use the following options to change the IP address and port that the web server listens on. The default setting is port 8888 on all configured IP addresses.

```
# Webserver listens on this address and port
http_host=0.0.0.0
http_port=8888
```

Specifying the Secret Key

For security, you should specify the secret key that is used for secure hashing in the session store:

1. Open the Hue configuration file.
2. In the `[desktop]` section, set the `secret_key` property to a long series of random characters (30 to 60 characters is recommended). For example,

```
secret_key=qpbdxoewsqlkhztybvfidtvekfusgdlofbcfghaswuicmqp
```



Note: If you do not specify a secret key, your session cookies will not be secure. Hue will run but it will also display error messages telling you to set the secret key.

Authentication

By default, the first user who logs in to Hue can choose any username and password and automatically becomes an administrator. This user can create other user and administrator accounts. Hue users should correspond to the Linux users who will use Hue; make sure you use the same name as the Linux username.

By default, user information is stored in the Hue database. However, the authentication system is pluggable. You can configure authentication to use an LDAP directory (Active Directory or OpenLDAP) to perform the authentication, or you can import users and groups from an LDAP directory. See [Configuring an LDAP Server for User Admin](#) on page 314.

For more information, see the [Hue SDK Documentation](#).

Configuring the Hue Server for SSL

You can optionally configure Hue to serve over HTTPS. As of CDH 5, pyOpenSSL is now part of the Hue build and does not need to be installed manually. To configure SSL, perform the following steps from the root of your Hue installation path:

1. Configure Hue to use your private key by adding the following options to the Hue configuration file:

```
ssl_certificate=/path/to/certificate
ssl_private_key=/path/to/key
```



Note: Hue can only support a private key without a passphrase.

2. On a production system, you should have an appropriate key signed by a well-known Certificate Authority. If you're just testing, you can create a self-signed key using the `openssl` command that may be installed on your system:

```
# Create a key
$ openssl genrsa 1024 > host.key
# Create a self-signed certificate
$ openssl req -new -x509 -nodes -sha1 -key host.key > host.cert
```



Note: Uploading files using the Hue File Browser over HTTPS requires using a proper SSL Certificate. Self-signed certificates do not work.

Authentication Backend Options for Hue

The table below gives a list of authentication backends Hue can be configured with including the recent [SAML backend](#) that enables single sign-on authentication. The `backend` configuration property is available in the `[[auth]]` section under `[desktop]`.

backend	<code>django.contrib.auth.backends.ModelBackend</code>	This is the default authentication backend used by Django .
	<code>desktop.auth.backend.AllowAllBackend</code>	This backend does not require a password for users to log in. All users are automatically authenticated and the username is set to what is provided.
	<code>desktop.auth.backend.AllowFirstUserDjangoBackend</code>	This is the default Hue backend. It creates the first user that logs in as the super user. After this, it relies on Django and the user manager to authenticate users.
	<code>desktop.auth.backend.LdapBackend</code>	Authenticates users against an LDAP service.
	<code>desktop.auth.backend.PamBackend</code>	Authenticates users with PAM (pluggable authentication module). The authentication mode depends on the PAM module used.

<code>desktop.auth.backend.SpnegoDjangoBackend</code>	SPNEGO is an authentication mechanism negotiation protocol. Authentication can be delegated to an authentication server, such as a Kerberos KDC, depending on the mechanism negotiated.
<code>desktop.auth.backend.RemoteUserDjangoBackend</code>	Authenticating remote users with the Django backend.
<code>desktop.auth.backend.OAuthBackend</code>	Delegates authentication to a third-party OAuth server.
<code>libsaml.backend.SAML2Backend</code>	Secure Assertion Markup Language (SAML) single sign-on (SSO) backend. Delegates authentication to the configured Identity Provider. See Configuring Hue for SAML for more details.



Note: All backends that delegate authentication to a third-party authentication server eventually import users into the Hue database. While the metadata is stored in the database, user authentication will still take place outside Hue.

Beeswax Configuration

In the `[beeswax]` section of the configuration file, you can optionally specify the following:

<code>hive_server_host</code>	The fully-qualified domain name or IP address of the host running HiveServer2.
<code>hive_server_port</code>	The port of the HiveServer2 Thrift server. Default: 10000.
<code>hive_conf_dir</code>	The directory containing <code>hive-site.xml</code> , the HiveServer2 configuration file.


Cloudera Impala Query UI Configuration

In the `[impala]` section of the configuration file, you can optionally specify the following:

<code>server_host</code>	The hostname or IP address of the Impala Server. Default: localhost.
<code>server_port</code>	The port of the Impalad Server. Default: 21050
<code>impersonation_enabled</code>	Turn on/off impersonation mechanism when talking to Impala. Default: False

DB Query Configuration

The DB Query app can have any number of databases configured in the `[[databases]]` section under `[librdbms]`. A database is known by its section name (`sqlite`, `mysql`, `postgresql`, and `oracle` as in the list below). For details on supported databases and versions, see [Supported Databases](#) on page 19.

Database Type	Configuration Properties
SQLite: [[[sqlite]]]	<pre># Name to show in the UI. ## nice_name=SQLite # For SQLite, name defines the path to the database. ## name=/tmp/sqlite.db # Database backend to use. ## engine=sqlite</pre>
MySQL, Oracle or PostgreSQL: [[[mysql]]] <div style="border: 1px solid green; padding: 5px; margin-top: 10px;">  Note: Replace with oracle or postgresql as required. </div>	<pre># Name to show in the UI. ## nice_name="My SQL DB" # For MySQL and PostgreSQL, name is the name of the database. # For Oracle, Name is instance of the Oracle server. # For express edition # this is 'xe' by default. ## name=mysql # Database backend to use. This can be: # 1. mysql # 2. postgresql # 3. oracle ## engine=mysql # IP or hostname of the database to connect to. ## host=localhost # Port the database server is listening to. Defaults are: # 1. MySQL: 3306 # 2. PostgreSQL: 5432 # 3. Oracle Express Edition: 1521 ## port=3306 # Username to authenticate with when connecting to the database. ## user=example # Password matching the username to authenticate with when connecting to the database. ## password=example</pre>

Pig Editor Configuration

In the [pig] section of the configuration file, you can optionally specify the following:

remote_data_dir	Location on HDFS where the Pig examples are stored.
-----------------	---

Sqoop Configuration

In the [sqoop] section of the configuration file, you can optionally specify the following:

server_url	The URL of the sqoop2 server.
------------	-------------------------------

Job Browser Configuration

By default, any user can see submitted job information for all users. You can restrict viewing of submitted job information by optionally setting the following property under the [jobbrowser] section in the Hue configuration file:

share_jobs	Indicate that jobs should be shared with all users. If set to false, they will be visible only to the owner and administrators.
------------	---

Job Designer

In the [jobsub] section of the configuration file, you can optionally specify the following:

remote_data_dir	Location in HDFS where the Job Designer examples and templates are stored.
-----------------	--

Oozie Editor/Dashboard Configuration

By default, any user can see all workflows, coordinators, and bundles. You can restrict viewing of workflows, coordinators, and bundles by optionally specifying the following property under the [oozie] section of the Hue configuration file:

oozie_jobs_count	Maximum number of Oozie workflows or coordinators or bundles to retrieve in one API call.
remote_data_dir	The location in HDFS where Oozie workflows are stored.

Also see [Liboozie Configuration](#) on page 318

Search Configuration

In the [search] section of the configuration file, you can optionally specify the following:

security_enabled	Indicate whether Solr requires clients to perform Kerberos authentication.
empty_query	Query sent when no term is entered. Default: * : *
solr_url	URL of the Solr server.

HBase Configuration

In the [hbase] section of the configuration file, you can optionally specify the following:

truncate_limit	Hard limit of rows or columns per row fetched before truncating. Default: 500
hbase_clusters	Comma-separated list of HBase Thrift servers for clusters in the format of "(name host:port)". Default: (Cluster localhost:9090)

User Admin Configuration

In the [useradmin] section of the configuration file, you can optionally specify the following:

default_user_group	The name of the group to which a manually created user is automatically assigned. Default: default.
--------------------	--

Configuring an LDAP Server for User Admin

User Admin can interact with an LDAP server, such as Active Directory, in one of two ways:

- You can import user and group information from your current Active Directory infrastructure using the LDAP Import feature in the User Admin application. User authentication is then performed by User Admin based on the

imported user and password information. You can then manage the imported users, along with any users you create directly in User Admin. See [Enabling Import of Users and Groups from an LDAP Directory](#) on page 315.

- You can configure User Admin to use an LDAP server as the authentication back end, which means users logging in to Hue will authenticate to the LDAP server, rather than against a username and password kept in User Admin. In this scenario, your users must all reside in the LDAP directory. See [Enabling the LDAP Server for User Authentication](#) on page 316 for further information.

Enabling Import of Users and Groups from an LDAP Directory

User Admin can import users and groups from an Active Directory using the Lightweight Directory Authentication Protocol (LDAP). In order to use this feature, you must configure User Admin with a set of LDAP settings in the Hue configuration file.



Note: If you import users from LDAP, you must set passwords for them manually; password information is not imported.

1. In the Hue configuration file, configure the following properties in the `[[ldap]]` section:

Property	Description	Example
<code>base_dn</code>	The search base for finding users and groups.	<code>base_dn="DC=mycompany,DC=com"</code>
<code>nt_domain</code>	The NT domain to connect to (only for use with Active Directory).	<code>nt_domain=mycompany.com</code>
<code>ldap_url</code>	URL of the LDAP server.	<code>ldap_url=ldap://auth.mycompany.com</code>
<code>ldap_cert</code>	Path to certificate for authentication over TLS (optional).	<code>ldap_cert=/mycertsdir/myTLScert</code>
<code>bind_dn</code>	Distinguished name of the user to bind as – not necessary if the LDAP server supports anonymous searches.	<code>bind_dn="CN=ServiceAccount,DC=mycompany,DC=com"</code>
<code>bind_password</code>	Password of the bind user – not necessary if the LDAP server supports anonymous searches.	<code>bind_password=P@ssw0rd</code>

2. Configure the following properties in the `[[users]]` section:

Property	Description	Example
<code>user_filter</code>	Base filter for searching for users.	<code>user_filter="objectclass=*"</code>
<code>user_name_attr</code>	The username attribute in the LDAP schema.	<code>user_name_attr=sAMAccountName</code>

3. Configure the following properties in the `[[groups]]` section:


Property	Description	Example
<code>group_filter</code>	Base filter for searching for groups.	<code>group_filter="objectclass=*"</code>
<code>group_name_attr</code>	The username attribute in the LDAP schema.	<code>group_name_attr=cn</code>



Note: If you provide a TLS certificate, it must be signed by a Certificate Authority that is trusted by the LDAP server.

Enabling the LDAP Server for User Authentication

You can configure User Admin to use an LDAP server as the authentication back end, which means users logging in to Hue will authenticate to the LDAP server, rather than against usernames and passwords managed by User Admin.

 **Important:**

Be aware that when you enable the LDAP back end for user authentication, user authentication by User Admin will be disabled. This means there will be no superuser accounts to log into Hue unless you take one of the following actions:


- Import one or more superuser accounts from Active Directory and assign them superuser permission.
- If you have already enabled the LDAP authentication back end, log into Hue using the LDAP back end, which will create a LDAP user. Then disable the LDAP authentication back end and use User Admin to give the superuser permission to the new LDAP user.

After assigning the superuser permission, enable the LDAP authentication back end.

1. In the Hue configuration file, configure the following properties in the `[[ldap]]` section:

Property	Description	Example
<code>ldap_url</code>	URL of the LDAP server, prefixed by <code>ldap://</code> or <code>ldaps://</code>	<code>ldap_url=ldap://auth.mycompany.com</code>
<code>search_bind_authentication</code>	Search bind authentication is now the default instead of direct bind. To revert to direct bind, the value of this property should be set to <code>false</code> . When using search bind semantics, Hue will ignore the following <code>nt_domain</code> and <code>ldap_username_pattern</code> properties.	<code>search_bind_authentication=false</code>
<code>nt_domain</code>	The NT domain over which the user connects (not strictly necessary if using <code>ldap_username_pattern</code>).	<code>nt_domain=mycompany.com</code>
<code>ldap_username_pattern</code>	Pattern for searching for usernames – Use <code><username></code> for the username parameter. For use when using <code>LdapBackend</code> for Hue authentication	<code>ldap_username_pattern="uid=<username>,ou=People,dc=mycompany,dc=com"</code>

2. If you are using TLS or secure ports, add the following property to specify the path to a TLS certificate file:

Property	Description	Example
<code>ldap_cert</code>	Path to certificate for authentication over TLS. <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> Note: If you provide a TLS certificate, it must be signed by a Certificate Authority that is trusted by the LDAP server.</div>	<code>ldap_cert=/mycertsdir/myTLScert</code>

3. In the `[[auth]]` sub-section inside `[desktop]` change the following:

<code>backend</code>	Change the setting of <code>backend</code> from <code>backend=desktop.auth.backend.AllowFirstUserDjangoBackend</code>
----------------------	--

	to backend=desktop.auth.backend.LdapBackend
--	--

Hadoop Configuration

The following configuration variables are under the `[hadoop]` section in the Hue configuration file.

HDFS Cluster Configuration

Hue currently supports only one HDFS cluster, which you define under the `[hdfs_clusters]` sub-section. The following properties are supported:

<code>[[[default]]]</code>	The section containing the default settings.
<code>fs_defaultfs</code>	The equivalent of <code>fs.defaultFS</code> (also referred to as <code>fs.default.name</code>) in a Hadoop configuration.
<code>webhdfs_url</code>	The HttpFS URL. The default value is the HTTP port on the NameNode.

YARN (MRv2) and MapReduce (MRv1) Cluster Configuration

Job Browser can display both MRv1 and MRv2 jobs, but must be configured to display one type at a time by specifying either `[yarn_clusters]` or `[mapred_clusters]` sections in the Hue configuration file.

The following YARN cluster properties are defined under the under the `[yarn_clusters]` sub-section:

<code>[[[default]]]</code>	The section containing the default settings.
<code>resourcemanager_host</code>	The fully-qualified domain name of the host running the ResourceManager.
<code>resourcemanager_port</code>	The port for the ResourceManager IPC service.
<code>submit_to</code>	If your Oozie is configured to use a YARN cluster, then set this to true. Indicate that Hue should submit jobs to this YARN cluster.
<code>proxy_api_url</code>	URL of the ProxyServer API. Default: <code>http://localhost:8088</code>
<code>history_server_api_url</code>	URL of the HistoryServer API Default: <code>http://localhost:19888</code>

The following MapReduce cluster properties are defined under the `[mapred_clusters]` sub-section:

<code>[[[default]]]</code>	The section containing the default settings.
<code>jobtracker_host</code>	The fully-qualified domain name of the host running the JobTracker.
<code>jobtracker_port</code>	The port for the JobTracker IPC service.
<code>submit_to</code>	If your Oozie is configured with to use a 0.20 MapReduce service, then set this to true. Indicate that Hue should submit jobs to this MapReduce cluster.

**Note: High Availability (MRv1):**

Add High Availability (HA) support for your MRv1 cluster by specifying a failover JobTracker. You can do this by configuring the following property under the `[[ha]]` sub-section for MRv1.

```
# Enter the host on which you are running the failover JobTracker
# jobtracker_host=<localhost-ha>
```

High Availability (YARN):

Add the following `[[ha]]` section under the `[hadoop] > [[yarn_clusters]]` sub-section in `hue.ini` with configuration properties for a second Resource Manager. As long as you have the `logical_name` property specified as below, jobs submitted to Oozie will work. The Job Browser, however, will *not* work with HA in this case.

```
[[ha]]
resourcemanager_host=<second_resource_manager_host_FQDN>
resourcemanager_api_url=http://<second_resource_manager_host_URL>
proxy_api_url=<second_resource_manager_proxy_URL>
history_server_api_url=<history_server_API_URL>
resourcemanager_port=<port_for_RM_IPC>
security_enabled=false
submit_to=true
logical_name=XXXX
```

Liboozie Configuration

In the `[liboozie]` section of the configuration file, you can optionally specify the following:

<code>security_enabled</code>	Indicate whether Oozie requires clients to perform Kerberos authentication.
<code>remote_deployment_dir</code>	The location in HDFS where the workflows and coordinators are deployed when submitted by a non-owner.
<code>oozie_url</code>	The URL of the Oozie server.

Sentry Configuration

In the `[libsentry]` section of the configuration file, specify the following:

<code>hostname</code>	Hostname or IP of server. Default: localhost
<code>port</code>	The port where the Sentry service is running. Default: 8038
<code>sentry_conf_dir</code>	Sentry configuration directory, where <code>sentry-site.xml</code> is located. Default: <code>/etc/sentry/conf</code>

Hue will also automatically pick up the HiveServer2 server name from Hive's `sentry-site.xml` file at `/etc/hive/conf`.

If you have enabled Kerberos for the Sentry service, allow Hue to connect to the service by adding the `hue` user to the following property in the `/etc/sentry/conf/sentry-store-site.xml` file.

```
<property>
  <name>sentry.service.allow.connect</name>
  <value>impala,hive,solr,hue</value>
</property>
```

ZooKeeper Configuration

**Warning:**

CDH does not support using Zookeeper with Hue.

In the [zookeeper] section of the configuration file, you can specify the following:

host_ports	Comma-separated list of ZooKeeper servers in the format "host:port". Example: localhost:2181,localhost:2182,localhost:2183
rest_url	The URL of the REST Contrib service (required for znode browsing). Default: http://localhost:9998

Setting up REST Service for ZooKeeper

ZooKeeper Browser requires the [ZooKeeper REST](#) service to be running. Follow the instructions below to set this up.

Step 1: Git and build the ZooKeeper repository

```
git clone https://github.com/apache/zookeeper
cd zookeeper
ant
Buildfile: /home/hue/Development/zookeeper/build.xml

init:
[mkdir] Created dir: /home/hue/Development/zookeeper/build/classes
[mkdir] Created dir: /home/hue/Development/zookeeper/build/lib
[mkdir] Created dir: /home/hue/Development/zookeeper/build/package/lib
[mkdir] Created dir: /home/hue/Development/zookeeper/build/test/lib
...
```

Step 2: Start the REST service

```
cd src/contrib/rest
nohup ant run&
```

Step 3: Update ZooKeeper configuration properties (if required)

If ZooKeeper and the REST service are not on the same machine as Hue, update the [Hue configuration file](#) and specify the correct hostnames and ports as shown in the sample configuration below:

```
[zookeeper]
...
[[clusters]]
...
[[[default]]]
    # Zookeeper ensemble. Comma separated list of Host/Port.
    # e.g. localhost:2181,localhost:2182,localhost:2183
    ## host_ports=localhost:2181

    # The URL of the REST contrib service
    ## rest_url=http://localhost:9998
```

You should now be able to successfully run the ZooKeeper Browser app.

Administering Hue

The following sections contain details about managing and operating a Hue installation:

- [Starting and Stopping the Hue Server](#) on page 320
- [Configuring Your Firewall for Hue](#) on page 320
- [Anonymous Usage Data Collection](#) on page 320

- [Managing Hue Processes](#) on page 320
- [Viewing Hue Logs](#) on page 321

Hue Superusers and Users

Hue's User Admin application provides two levels of user privileges: superusers and users.

- Superusers — The first user who logs into Hue after its installation becomes the first superuser. Superusers have permissions to perform administrative functions such as:
 - Add and delete users
 - Add and delete groups
 - Assign permissions to groups
 - Change a user into a superuser
 - Import users and groups from an LDAP server
- Users — can change their name, email address and password. They can log in to Hue and run Hue applications, subject to the permissions provided to the Hue groups to which they belong.

Starting and Stopping the Hue Server

The `hue-server` package includes service scripts to start and stop the Hue Server.

To start the Hue Server:

```
$ sudo service hue start
```

To restart the Hue Server:

```
$ sudo service hue restart
```

To stop the Hue Server:

```
$ sudo service hue stop
```

Configuring Your Firewall for Hue

Hue currently requires that the machines within your cluster can connect to each other freely over TCP. The machines outside your cluster must be able to open TCP port 8888 on the Hue Server (or the configured Hue web HTTP port) to interact with the system.

Anonymous Usage Data Collection

Hue tracks anonymized pages and application versions to gather information about application usage levels. The data collected does not include any hostnames or IDs.

For Hue 2.5.0 and higher, you can restrict this data collection by setting the `collect_usage` property to `false` in the `[desktop]` section in the Hue configuration file, `hue.ini`.

```
[desktop]
...
# Help improve Hue with anonymous usage analytics.
# Use Google Analytics to see how many times an application or specific section of an
application is used, nothing more.
## collect_usage=false
```

If you are using an earlier version of Hue, disable this data collection by navigating to Step 3 of Hue's Quick Start Wizard. Under **Anonymous usage analytics**, uncheck the **Check to enable usage analytics** checkbox.

Managing Hue Processes

A script called `supervisor` manages all Hue processes. The supervisor is a watchdog process; its only purpose is to spawn and monitor other processes. A standard Hue installation starts and monitors the `runcpserver` process.

- `runcpserver` – a web server that provides the core web functionality of Hue

If you have installed other applications into your Hue instance, you may see other daemons running under the supervisor as well.

You can see the supervised processes running in the output of `ps -f -u hue`.

Note that the supervisor automatically restarts these processes if they fail for any reason. If the processes fail repeatedly within a short time, the supervisor itself shuts down.

Viewing Hue Logs

Hue logs are stored in `/var/log/hue`. In the Hue UI, select **About Hue > Server Logs**. You can also view these logs at `http://myserver:port/logs`.

Hue generates `.log` and `.out` files for each supervised process. The `.log` files write log information with `log4j`. The `.out` files write standard output (stdout) and standard error (stderr) streams.

The following Hue logs are available.

Log Name	Description
<code>access.log</code>	Filtered list of all successful attempts to access the Hue Web UI
<code>audit.log</code>	Audit log visible in Cloudera Navigator
<code>collectstatic.log</code>	Static files that support the Hue Web UI (images, JavaScript files, .css, and so on)
<code>error.log</code>	Filtered list of all nontrivial errors
<code>kt_renewer.log</code>	Kerberos ticket renews
<code>metrics_hue_server.log</code>	Usage data for monitoring in Cloudera Manager
<code>migrate.log</code>	Database and table migrations
<code>runcpserver.log</code>	Hue (CherryPy) web server info
<code>syncdb.log</code>	Database and table creations

Hue Database

The Hue server requires an SQL database to store small amounts of data, including user account information as well as history of job submissions and Hive queries. The Hue server supports a lightweight embedded database and several types of external databases. If you elect to configure Hue to use an external database, upgrades may require more manual steps.

Prerequisites

Before using an external database, with Hue, install all of the support libraries required by your operating system. See [Development Preferences](#) in the Hue documentation for the full list.

Embedded Database

By default, Hue is configured to use the embedded database SQLite for this purpose, and should require no configuration or management by the administrator.

Inspecting the Embedded Hue Database

The default SQLite database used by Hue is located in `/var/lib/hue/desktop.db`. You can inspect this database from the command line using the `sqlite3` program. For example:

```
# sqlite3 /var/lib/hue/desktop.db
SQLite version 3.6.22
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select username from auth_user;
```

```
admin
test
sample
sqlite>
```



Important: It is strongly recommended that you avoid making any modifications to the database directly using `sqlite3`, though `sqlite3` is useful for management or troubleshooting.

Backing up the Embedded Hue Database

If you use the default embedded SQLite database, copy the `desktop.db` file to another node for backup. Cloudera recommends that you backup regularly, and also that you backup before upgrading to a new version of Hue.

External Database

Although SQLite is the default database, some advanced users may prefer to have Hue access an external database. Hue supports MySQL, PostgreSQL, and Oracle. See [Supported Databases](#) on page 19 for the supported versions.



Note: In the instructions that follow, dumping the database and editing the JSON objects is only necessary if you have data in SQLite that you need to migrate. If you do not need to migrate data from SQLite, you can skip those steps.

Configuring the Hue Server to Store Data in MySQL



Important: Cloudera requires you to use InnoDB, *not* MyISAM, as your MySQL engine.

1. Shut down the Hue server if it is running.
2. Dump the existing database data to a text file. Note that using the `.json` extension is required.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue dumpdata > <some-temporary-file>.json
```

3. Open `<some-temporary-file>.json` and remove all JSON objects with `useradmin.userprofile` in the `model` field. Here are some examples of JSON objects that should be deleted.

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "creation_method": "HUE",
    "user": 1,
    "home_directory": "/user/alice"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "creation_method": "HUE",
    "user": 1100714,
    "home_directory": "/user/bob"
  }
},
.....
```

4. Start the Hue server.
5. Install the MySQL client developer package.

OS	Command
RHEL	\$ sudo yum install mysql-devel
SLES	\$ sudo zypper install mysql-devel
Ubuntu or Debian	\$ sudo apt-get install libmysqlclient-dev

6. Install the MySQL connector.

OS	Command
RHEL	\$ sudo yum install mysql-connector-java
SLES	\$ sudo zypper install mysql-connector-java
Ubuntu or Debian	\$ sudo apt-get install libmysql-java

7. Install and start MySQL.

OS	Command
RHEL	\$ sudo yum install mysql-server
SLES	\$ sudo zypper install mysql \$ sudo zypper install libmysqlclient_r15
Ubuntu or Debian	\$ sudo apt-get install mysql-server

8. Change the `/etc/my.cnf` file as follows:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
bind-address=<ip-address>
default-storage-engine=InnoDB
sql_mode=STRICT_ALL_TABLES
```

9. Start the `mysql` daemon.

OS	Command
RHEL	\$ sudo service mysqld start
SLES and Ubuntu or Debian	\$ sudo service mysql start

10. Configure MySQL to use a strong password. In the following procedure, your current `root` password is blank. Press the **Enter** key when you're prompted for the root password.

```
$ sudo /usr/bin/mysql_secure_installation
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
```

```
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
All done!
```

11 Configure MySQL to start at boot.

OS	Command
RHEL	\$ sudo /sbin/chkconfig mysqld on \$ sudo /sbin/chkconfig --list mysqld mysqld 0:off 1:off 2:on 3:on 4:on 5:on 6:off
SLES	\$ sudo chkconfig --add mysql
Ubuntu or Debian	\$ sudo chkconfig mysql on

12 Create the Hue database and grant privileges to a hue user to manage the database.

```
mysql> create database hue;
Query OK, 1 row affected (0.01 sec)
mysql> grant all on hue.* to 'hue'@'localhost' identified by '<secretpassword>';
Query OK, 0 rows affected (0.00 sec)
```

13 Open the Hue configuration file in a text editor.

14 Edit the Hue configuration file hue.ini. Directly below the [[database]] section under the [desktop] line, add the following options (and modify accordingly for your setup):

```
host=localhost
port=3306
engine=mysql
user=hue
password=<secretpassword>
name=hue
```

15 As the hue user, load the existing data and create the necessary database tables using syncdb and migrate commands. When running these commands, Hue will try to access a logs directory, located at /opt/cloudera/parcels/CDH/lib/hue/logs, which might be missing. If that is the case, first create the logs directory and give the hue user and group ownership of the directory.

```
$ sudo mkdir /opt/cloudera/parcels/CDH/lib/hue/logs
$ sudo chown hue:hue /opt/cloudera/parcels/CDH/lib/hue/logs
$ sudo -u hue <HUE_HOME>/build/env/bin/hue syncdb --noinput
$ sudo -u hue <HUE_HOME>/build/env/bin/hue migrate
$ mysql -u hue -p <secretpassword>
mysql > SHOW CREATE TABLE auth_permission;
```

16 (InnoDB only) Drop the foreign key.

```
mysql > ALTER TABLE auth_permission DROP FOREIGN KEY content_type_id_refs_id_XXXXXX;
```

17 Delete the rows in the django_content_type table.

```
mysql > DELETE FROM hue.django_content_type;
```

18 Load the data.

```
$ <HUE_HOME>/build/env/bin/hue loaddata <some-temporary-file>.json
```

19 (InnoDB only) Add the foreign key.

```
$ mysql -u hue -p <secretpassword>
mysql > ALTER TABLE auth_permission ADD FOREIGN KEY (`content_type_id`) REFERENCES
`django_content_type` (`id`);
```

Configuring the Hue Server to Store Data in PostgreSQL



Warning: Hue requires PostgreSQL 8.4 or higher.

1. Shut down the Hue server if it is running.
2. Dump the existing database data to a text file. Note that using the `.json` extension is required.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue dumpdata > <some-temporary-file>.json
```

3. Open `<some-temporary-file>.json` and remove all JSON objects with `useradmin.userprofile` in the `model` field. Here are some examples of JSON objects that should be deleted.

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "creation_method": "HUE",
    "user": 1,
    "home_directory": "/user/alice"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "creation_method": "HUE",
    "user": 1100714,
    "home_directory": "/user/bob"
  }
},
.....
```

4. Install required packages.

OS	Command
RHEL	<code>\$ sudo yum install postgresql-devel gcc python-devel</code>
SLES	<code>\$ sudo zypper install postgresql-devel gcc python-devel</code>
Ubuntu or Debian	<code>\$ sudo apt-get install postgresql-devel gcc python-devel</code>

5. Install the module that provides the connector to PostgreSQL.

```
sudo -u hue <HUE_HOME>/build/env/bin/pip install setuptools
sudo -u hue <HUE_HOME>/build/env/bin/pip install psycopg2
```

6. Install the PostgreSQL server.

OS	Command
RHEL	<code>\$ sudo yum install postgresql-server</code>
SLES	<code>\$ sudo zypper install postgresql-server</code>

OS	Command
Ubuntu or Debian	\$ sudo apt-get install postgresql

7. Initialize the data directories:

```
$ service postgresql initdb
```

8. Configure client authentication.

- a. Edit `/var/lib/pgsql/data/pg_hba.conf`.
- b. Set the authentication methods for local to `trust` and for host to `password` and add the following line at the end.

```
host hue hue 0.0.0.0/0 md5
```

9. Start the PostgreSQL server.

```
$ su - postgres
# /usr/bin/postgres -D /var/lib/pgsql/data > logfile 2>&1 &
```

10 Configure PostgreSQL to listen on all network interfaces.

Edit `/var/lib/pgsql/data/postgresql.conf` and set `listen_addresses`:

```
listen_addresses = '0.0.0.0' # Listen on all addresses
```

11 Create the hue database and grant privileges to a hue user to manage the database.

```
# psql -U postgres
postgres=# create database hue;
postgres=# \c hue;
You are now connected to database 'hue'.
postgres=# create user hue with password '<secretpassword>';
postgres=# grant all privileges on database hue to hue;
postgres=# \q
```

12 Restart the PostgreSQL server.

```
$ sudo service postgresql restart
```

13 Verify connectivity.

```
psql -h localhost -U hue -d hue
Password for user hue: <secretpassword>
```

14 Configure the PostgreSQL server to start at boot.

OS	Command
RHEL	\$ sudo /sbin/chkconfig postgresql on \$ sudo /sbin/chkconfig --list postgresql postgresql 0:off 1:off 2:on 3:on 4:on 5:on 6:off
SLES	\$ sudo chkconfig --add postgresql
Ubuntu or Debian	\$ sudo chkconfig postgresql on

15 Open the Hue configuration file in a text editor.

- 16** Edit the Hue configuration file `hue.ini`. Directly below the `[[database]]` section under the `[desktop]` line, add the following options (and modify accordingly for your setup):

```
host=localhost
port=5432
engine=postgresql_psycpg2
user=hue
password=<secretpassword>
name=hue
```

- 17** As the `hue` user, configure Hue to load the existing data and create the necessary database tables. You will need to run both the `migrate` and `syncdb` commands. When running these commands, Hue will try to access a `logs` directory, located at `/opt/cloudera/parcels/CDH/lib/hue/logs`, which might be missing. If that is the case, first create the `logs` directory and give the `hue` user and group ownership of the directory.

```
$ sudo mkdir /opt/cloudera/parcels/CDH/lib/hue/logs
$ sudo chown hue:hue /opt/cloudera/parcels/CDH/lib/hue/logs
$ sudo -u hue <HUE_HOME>/build/env/bin/hue syncdb --noinput
$ sudo -u hue <HUE_HOME>/build/env/bin/hue migrate
```

- 18** Determine the foreign key ID.

```
bash# su - postgres
$ psql -h localhost -U hue -d hue
postgres=# \d auth_permission;
```

- 19** Drop the foreign key that you retrieved in the previous step.

```
postgres=# ALTER TABLE auth_permission DROP CONSTRAINT content_type_id_refs_id_<XXXXXX>;
```

- 20** Delete the rows in the `django_content_type` table.

```
postgres=# TRUNCATE django_content_type CASCADE;
```

- 21** Load the data.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue loaddata <some-temporary-file>.json
```

- 22** Add back the foreign key you dropped.

```
bash# su - postgres
$ psql -h localhost -U hue -d hue
postgres=# ALTER TABLE auth_permission ADD CONSTRAINT content_type_id_refs_id_<XXXXXX>
FOREIGN KEY (content_type_id) REFERENCES django_content_type(id) DEFERRABLE INITIALLY
DEFERRED;
```

Configuring the Hue Server to Store Data in Oracle

1. Ensure Python 2.6 or higher is installed on the server Hue is running on.
2. Download the Oracle client libraries at [Instant Client for Linux x86-64](#) Version 11.1.0.7.0, Basic and SDK (with headers) zip files to the same directory.
3. Unzip the zip files.
4. Set environment variables to reference the libraries.

```
$ export ORACLE_HOME=<download directory>
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME
```

5. Create a symbolic link for the shared object:

```
$ cd $ORACLE_HOME
$ ln -sf libclntsh.so.11.1 libclntsh.so
```

6. Get a data dump by executing:

```
$ <HUE_HOME>/build/env/bin/hue dumpdata > <some-temporary-file>.json --indent 2
```

7. Edit the Hue configuration file `hue.ini`. Directly below the `[[database]]` section under the `[desktop]` line, add the following options (and modify accordingly for your setup):

```
host=localhost
port=1521
engine=oracle
user=hue
password=<secretpassword>
name=<SID of the Oracle database, for example, 'XE'>
```

To use the Oracle service name instead of the SID, use the following configuration instead:

```
port=0
engine=oracle
user=hue
password=password
name=oracle.example.com:1521/orcl.example.com
```

The directive `port=0` allows Hue to use a service name. The `name` string is the connect string, including hostname, port, and service name.

To add support for a multithreaded environment, set the `threaded` option to `true` under the `[desktop]>[[database]]` section.

```
options={'threaded':true}
```

8. Grant required permissions to the Hue user in Oracle:

```
grant alter any index to hue;
grant alter any table to hue;
grant alter database link to hue;
grant create any index to hue;
grant create any sequence to hue;
grant create database link to hue;
grant create session to hue;
grant create table to hue;
grant drop any sequence to hue;
grant select any dictionary to hue;
grant drop any table to hue;
grant create procedure to hue;
grant create trigger to hue;
```

9. As the `hue` user, configure Hue to load the existing data and create the necessary database tables. You will need to run both the `syncdb` and `migrate` commands. When running these commands, Hue will try to access a `logs` directory, located at `/opt/cloudera/parcels/CDH/lib/hue/logs`, which might be missing. If that is the case, first create the `logs` directory and give the `hue` user and group ownership of the directory.

```
$ sudo mkdir /opt/cloudera/parcels/CDH/lib/hue/logs
$ sudo chown hue:hue /opt/cloudera/parcels/CDH/lib/hue/logs
$ sudo -u hue <HUE_HOME>/build/env/bin/hue syncdb --noinput
$ sudo -u hue <HUE_HOME>/build/env/bin/hue migrate
```


10 Ensure that you are still connected to Oracle as the `hue` user and delete all data from the Oracle tables:

```
SELECT 'DELETE FROM ' || '.' || table_name || ';' FROM user_tables;
```

11 Run the statements generated in the preceding step.

12 Load the data.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue loaddata <some-temporary-file>.json
```

Viewing the Hue User Guide

For additional information about Hue, see the [Hue User Guide](#).

KMS Installation

Hadoop Key Management Service (KMS) is a cryptographic key management server based on Hadoop's **KeyProvider** API. It provides a client which is a KeyProvider implementation that interacts with the KMS using the HTTP REST API. Both the KMS and its client support HTTP SPNEGO Kerberos authentication and SSL-secured communication. The KMS is a Java-based web application which runs using a pre-configured Tomcat server bundled with the Hadoop distribution.

Installing and Upgrading KMS



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install or upgrade KMS on a RHEL-compatible system:

```
$ sudo yum install hadoop-kms hadoop-kms-server
```

To install or upgrade KMS on a SLES system:

```
$ sudo zypper install hadoop-kms hadoop-kms-server
```

To install or upgrade KMS on an Ubuntu or Debian system:

```
$ sudo apt-get install hadoop-kms hadoop-kms-server
```

Troubleshooting: upgrading `hadoop-kms` from 5.2.x and 5.3.x releases on SLES

The problem described in this section affects SLES upgrades from 5.2.x releases earlier than 5.2.4, and from 5.3.x releases earlier than 5.3.2.

Problem

The problem occurs when you try to upgrade the `hadoop-kms` package, for example:

```
Installing: hadoop-kms-2.5.0+cdh5.3.2+801-1.cdh5.3.2.p0.224.sles11 [error]
12:54:19 Installation of hadoop-kms-2.5.0+cdh5.3.2+801-1.cdh5.3.2.p0.224.sles11 failed:
12:54:19 (with --nodeps --force) Error: Subprocess failed. Error: RPM failed: warning:
/var/cache/zypp/packages/cdh/RPMS/x86_64/hadoop-kms-2.5.0+cdh5.3.2+801-1.cdh5.3.2.p0.224.sles11.x86_64.rpm:
Header V4 DSA signature: NOKEY, key ID e8f86acd
12:54:19 error: %postun(hadoop-kms-2.5.0+cdh5.3.1+791-1.cdh5.3.1.p0.17.sles11.x86_64)
scriptlet failed, exit status 1
12:54:19
```

**Note:**

- The `hadoop-kms` package is not installed automatically with CDH, so you will encounter this error only if you are explicitly upgrading an existing version of KMS.
- The examples in this section show an upgrade from CDH 5.3.x; the 5.2.x case looks very similar.

What to Do

If you see an error similar to the one in the example above, proceed as follows:

1. Abort, or ignore the error (it doesn't matter which):

```
Abort, retry, ignore? [a/r/i] (a): i
```

2. Perform cleanup.

a. `# rpm -qa hadoop-kms`

You will see two versions of `hadoop-kms`; for example:

```
hadoop-kms-2.5.0+cdh5.3.1+791-1.cdh5.3.1.p0.17.sles11
hadoop-kms-2.5.0+cdh5.3.2+801-1.cdh5.3.2.p0.224.sles11
```

- b. Remove the older version, in this example

`hadoop-kms-2.5.0+cdh5.3.1+791-1.cdh5.3.1.p0.17.sles11:`

```
# rpm -e --noscripts hadoop-kms-2.5.0+cdh5.3.1+791-1.cdh5.3.1.p0.17.sles11
```

3. Verify that the older version of the package has been removed:

```
# rpm -qa hadoop-kms
```

Now you should see only the newer package:

```
hadoop-kms-2.5.0+cdh5.3.2+801-1.cdh5.3.2.p0.224.sles11
```

Mahout Installation

[Apache Mahout](#) is a machine-learning tool. By enabling you to build machine-learning libraries that are scalable to "reasonably large" datasets, it aims to make building intelligent applications easier and faster.

**Note:**

To see which version of Mahout is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

The main use cases for Mahout are:

- **Recommendation mining**, which tries to identify things users will like on the basis of their past behavior (for example shopping or online-content recommendations)
- **Clustering**, which groups similar items (for example, documents on similar topics)
- **Classification**, which learns from existing categories what members of each category have in common, and on that basis tries to categorize new items
- **Frequent item-set mining**, which takes a set of item-groups (such as terms in a query session, or shopping-cart content) and identifies items that usually appear together

**Important:**

If you have not already done so, install the Cloudera `yum`, `zypper/YaST` or `apt` repository before using the instructions below to install Mahout. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155.

Upgrading Mahout

**Note:**

To see which version of Mahout is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Upgrading Mahout from an Earlier CDH 5 Release to the Latest CDH 5 Release

To upgrade Mahout to the latest release, simply install the new version; see [Installing Mahout](#) on page 331.

**Important: Configuration files**

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Installing Mahout

You can install Mahout from an RPM or Debian package, or from a [tarball](#).

**Note:**

To see which version of Mahout is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Installing from packages is more convenient than installing the tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations

These instructions assume that you will install from packages if possible.

**Note: Install Cloudera Repository**

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper/YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install Mahout on a RHEL system:

```
$ sudo yum install mahout
```

To install Mahout on a SLES system:

```
$ sudo zypper install mahout
```

To install Mahout on an Ubuntu or Debian system:

```
$ sudo apt-get install mahout
```

To access Mahout documentation:

The Mahout docs are bundled in a `mahout-doc` package that should be installed separately.

```
$ sudo apt-get install mahout-doc
```

The contents of this package are saved under `/usr/share/doc/mahout*`.

The Mahout Executable

The Mahout executable is installed in `/usr/bin/mahout`. Use this executable to run your analysis.

Getting Started with Mahout

To get started with Mahout, you can follow the instructions in this [Apache Mahout Quickstart](#).

Viewing the Mahout Documentation

For more information about Mahout, see mahout.apache.org.

Oozie Installation

About Oozie

Apache Oozie Workflow Scheduler for Hadoop is a workflow and coordination service for managing Apache Hadoop jobs:

- Oozie Workflow jobs are Directed Acyclical Graphs (DAGs) of *actions*; *actions* are typically Hadoop jobs (MapReduce, Streaming, Pipes, Pig, Hive, Sqoop, etc).
- Oozie Coordinator jobs trigger recurrent Workflow jobs based on time (frequency) and data availability.
- Oozie Bundle jobs are sets of Coordinator jobs managed as a single job.

Oozie is an extensible, scalable and data-aware service that you can use to orchestrate dependencies among jobs running on Hadoop.

- To find out more about Oozie, see <https://archive.cloudera.com/cdh5/cdh/5/oozie/>.
- To install or upgrade Oozie, follow the directions on this page.



Important: Running Services

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

Oozie Packaging

There are two packaging options for installing Oozie:

- Separate RPM packages for the Oozie server (`oozie`) and client (`oozie-client`)

- Separate Debian packages for the Oozie server (`oozie`) and client (`oozie-client`)

You can also [download an Oozie tarball](#).

Oozie Prerequisites

- Prerequisites for installing Oozie server:
 - An [operating system supported by CDH 5](#)
 - [Oracle JDK](#)
 - A [supported database](#) if you are not planning to use the default (Derby).
- Prerequisites for installing Oozie client:
 - [Oracle JDK](#)



Note:

- To see which version of Oozie is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Upgrading Oozie

Follow these instructions to upgrade Oozie to CDH 5 from RPM or Debian Packages.

Upgrading Oozie from an Earlier CDH 5 Release

The steps that follow assume you are upgrading Oozie as part of an overall upgrade to the latest CDH 5 release and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

To upgrade Oozie to the latest CDH 5 release, proceed as follows.

Step 1: Back Up the Configuration

Back up the Oozie configuration files in `/etc/oozie` and the Oozie database.

For convenience you may want to save Oozie configuration files in your home directory; you will need them after installing the new version of Oozie.

Step 2: Stop the Oozie Server.

To stop the Oozie Server:

```
sudo service oozie stop
```

Step 3: Install Oozie

Follow the procedure under [Installing Oozie](#) on page 334 and then proceed to [Configuring Oozie after Upgrading from an Earlier CDH 5 Release](#) on page 335.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Installing Oozie

Oozie is distributed as two separate packages; a client package (`oozie-client`) and a server package (`oozie`). Depending on what you are planning to install, choose the appropriate packages and install them using your preferred package manager application.



Note:

The Oozie server package, `oozie`, is preconfigured to work with MRv2 (YARN). To configure the Oozie server to work with MRv1, see [Configuring the Hadoop Version to Use](#).



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install the Oozie server package on an Ubuntu and other Debian system:

```
$ sudo apt-get install oozie
```

To install the Oozie client package on an Ubuntu and other Debian system:

```
$ sudo apt-get install oozie-client
```

To install the Oozie server package on a RHEL-compatible system:

```
$ sudo yum install oozie
```

To install the Oozie client package on a RHEL-compatible system:

```
$ sudo yum install oozie-client
```

To install the Oozie server package on a SLES system:

```
$ sudo zypper install oozie
```

To install the Oozie client package on a SLES system:

```
$ sudo zypper install oozie-client
```

**Note:**

Installing the `oozie` package creates an `oozie` service configured to start Oozie at system startup time.

You are now ready to configure Oozie. See the [next section](#).

Configuring Oozie

This section explains how to configure Oozie and it provides procedures for configuring the proper version of Oozie for new installations and after upgrades.

Configuring which Hadoop Version to Use

The Oozie client does not interact directly with Hadoop MapReduce, and so it does not require any MapReduce configuration.

The Oozie server can work with either MRv1 or YARN. *It cannot work with both simultaneously.*

You set the MapReduce version the Oozie server works with by means of the `alternatives` command (or `update-alternatives`, depending on your operating system). As well as distinguishing between YARN and MRv1, the commands differ depending on whether or not you are using [SSL](#).

- To use YARN (without SSL):

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.http
```

- To use YARN (with SSL):

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.https
```

- To use MRv1 (without SSL) :

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.http.mr1
```

- To use MRv1 (with SSL) :

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.https.mr1
```

**Important: If you are upgrading from a release earlier than CDH 5 Beta 2**

In earlier releases, the mechanism for setting the MapReduce version was the `CATALINA_BASE` variable in `/etc/oozie/conf/oozie-env.sh`. This does not work as of CDH 5 Beta 2, and in fact could cause problems. Check your `/etc/oozie/conf/oozie-env.sh` and make sure you have the new version. The new version contains the line:

```
export CATALINA_BASE=/var/lib/oozie/tomcat-deployment
```

Configuring Oozie after Upgrading from an Earlier CDH 5 Release

Note: If you are installing Oozie for the first time, skip this section and proceed with [Configuring Oozie after a New Installation](#) on page 338.

Step 1: Update Configuration Files

1. Edit the new Oozie CDH 5 `oozie-site.xml`, and set all customizable properties to the values you set in the previous `oozie-site.xml`.

2. If necessary do the same for the `oozie-log4j.properties`, `oozie-env.sh` and the `adminusers.txt` files.

Step 2: Upgrade the Database

**Important:**

- Do not proceed before you have edited the configuration files as instructed in [Step 1](#).
- Before running the database upgrade tool, copy or symbolically link the JDBC driver JAR for the database you are using into the `/var/lib/oozie/` directory.

Oozie CDH 5 provides a command-line tool to perform the database schema and data upgrade. The tool uses Oozie configuration files to connect to the database and perform the upgrade.

The database upgrade tool works in two modes: it can do the upgrade in the database or it can produce an SQL script that a database administrator can run manually. If you use the tool to perform the upgrade, you must do it as a database user who has permissions to run DDL operations in the Oozie database.

- **To run the Oozie database upgrade tool against the database:**

**Important:**

This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh upgrade -run
```

You will see output such as this (the output of the script may differ slightly depending on the database vendor):

```
Validate DB Connection
DONE
Check DB schema exists
DONE
Verify there are not active Workflow Jobs
DONE
Check OOZIE_SYS table does not exist
DONE
Get Oozie DB version
DONE
Upgrade SQL schema
DONE
Upgrading to db schema for Oozie 4.0.0-cdh5.0.0
Update db.version in OOZIE_SYS table to 3
DONE
Converting text columns to bytea for all tables
DONE
Get Oozie DB version
DONE

Oozie DB has been upgraded to Oozie version '4.0.0-cdh5.0.0'

The SQL commands have been written to: /tmp/ooziedb-8676029205446760413.sql
```

- **To create the upgrade script:**



Important: This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh upgrade -sqlfile SCRIPT
```


For example:

```
$ sudo -u oozie /usr/lib/bin/ooziedb.sh upgrade -sqlfile oozie-upgrade.sql
```

You should see output such as the following (the output of the script may differ slightly depending on the database vendor):

```
Validate DB Connection
DONE
Check DB schema exists
DONE
Verify there are not active Workflow Jobs
DONE
Check OOZIE_SYS table does not exist
DONE
Get Oozie DB version
DONE
Upgrade SQL schema
DONE
Upgrading to db schema for Oozie 4.0.0-cdh5.0.0
Update db.version in OOZIE_SYS table to 3
DONE
Converting text columns to bytea for all tables
DONE
Get Oozie DB version
DONE
```

The SQL commands have been written to: oozie-upgrade.sql

WARN: The SQL commands have NOT been executed, you must use the '-run' option



Important: If you used the `-sqlfile` option instead of `-run`, Oozie database schema has not been upgraded. You need to run the `oozie-upgrade` script against your database.

Step 3: Upgrade the Oozie Shared Library



Important: This step is required; the current version of Oozie does not work with shared libraries from an earlier version.

The Oozie installation bundles two shared libraries, one for MRv1 and one for YARN. Make sure you install the right one for the MapReduce version you are using:

- The shared library file for YARN is `oozie-sharelib-yarn.tar.gz`.
- The shared library file for MRv1 is `oozie-sharelib-mr1.tar.gz`.

To upgrade the shared library, proceed as follows.

1. Delete the Oozie shared libraries from HDFS. For example:

```
$ sudo -u oozie hadoop fs -rmr /user/oozie/share
```



Note:

- 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>`
- If the current shared libraries are in another location, make sure you use this other location when you run the above command(s).

2. install the Oozie CDH 5 shared libraries. For example:

```
$ sudo oozie-setup sharelib create -fs <FS_URI> -locallib /usr/lib/oozie/oozie-sharelib-yarn.tar.gz
```

where *FS_URI* is the HDFS URI of the filesystem that the shared library should be installed on (for example, `hdfs://<HOST>:<PORT>`).



Important:

If you are installing Oozie to work with MRv1, make sure you use `oozie-sharelib-mr1.tar.gz` instead.

Step 4: Start the Oozie Server

Now you can start Oozie:

```
$ sudo service oozie start
```

Check Oozie's `oozie.log` to verify that Oozie has started successfully.

Step 5: Upgrade the Oozie Client

Although older Oozie clients work with the new Oozie server, you need to install the new version of the Oozie client in order to use all the functionality of the Oozie server.

To upgrade the Oozie client, if you have not already done so, follow the steps under [Installing Oozie](#) on page 334.

Configuring Oozie after a New Installation



Note: Follow the instructions in this section if you are installing Oozie for the first time. If you are upgrading Oozie from an earlier CDH 5 release, skip this subsection and see: [Configuring Oozie after Upgrading from an Earlier CDH 5 Release](#) on page 335.

When you install Oozie from an RPM or Debian package, Oozie server creates all configuration, documentation, and runtime files in the standard Linux directories, as follows.

Type of File	Where Installed
binaries	<code>/usr/lib/oozie/</code>
configuration	<code>/etc/oozie/conf/</code>
documentation	for SLES: <code>/usr/share/doc/packages/oozie/</code> for other platforms: <code>/usr/share/doc/oozie/</code>
examples TAR.GZ	for SLES: <code>/usr/share/doc/packages/oozie/</code> for other platforms: <code>/usr/share/doc/oozie/</code>

Type of File	Where Installed
sharelib TAR.GZ	<code>/usr/lib/oozie/</code>
data	<code>/var/lib/oozie/</code>
logs	<code>/var/log/oozie/</code>
temp	<code>/var/tmp/oozie/</code>
PID file	<code>/var/run/oozie/</code>

Deciding Which Database to Use

Oozie has a built-in Derby database, but Cloudera recommends that you use a [PostgreSQL](#), [MySQL](#), or [Oracle](#) database instead, for the following reasons:

- Derby runs in embedded mode and it is not possible to monitor its health.
- It is not clear how to implement a live backup strategy for the embedded Derby database, though it may be possible.
- Under load, Cloudera has observed locks and rollbacks with the embedded Derby database which don't happen with server-based databases.

See [Supported Databases](#) on page 19 for tested database versions.

Configuring Oozie to Use PostgreSQL

Use the procedure that follows to configure Oozie to use PostgreSQL instead of Apache Derby.

Install PostgreSQL 8.4.x or 9.0.x.

Create the Oozie user and Oozie database.

For example, using the PostgreSQL `psql` command-line tool:

```
$ psql -U postgres
Password for user postgres: *****

postgres=# CREATE ROLE oozie LOGIN ENCRYPTED PASSWORD 'oozie'
NOSUPERUSER INHERIT CREATEDB NOCREATEROLE;
CREATE ROLE

postgres=# CREATE DATABASE "oozie" WITH OWNER = oozie
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'en_US.UTF8'
LC_CTYPE = 'en_US.UTF8'
CONNECTION LIMIT = -1;
CREATE DATABASE

postgres=# \q
```

Configure PostgreSQL to accept network connections for the oozie user.

1. Edit the `postgresql.conf` file and set the `listen_addresses` property to `*`, to make sure that the PostgreSQL server starts listening on all your network interfaces. Also make sure that the `standard_conforming_strings` property is set to `off`.

2. Edit the PostgreSQL `data/pg_hba.conf` file as follows:

```
host    oozie        oozie        0.0.0.0/0        md5
```

Reload the PostgreSQL configuration.

```
$ sudo -u postgres pg_ctl reload -s -D /opt/PostgreSQL/8.4/data
```

Configure Oozie to use PostgreSQL

Edit the `oozie-site.xml` file as follows:

```
...  
<property>  
  <name>oozie.service.JPAService.jdbc.driver</name>  
  <value>org.postgresql.Driver</value>  
</property>  
<property>  
  <name>oozie.service.JPAService.jdbc.url</name>  
  <value>jdbc:postgresql://localhost:5432/oozie</value>  
</property>  
<property>  
  <name>oozie.service.JPAService.jdbc.username</name>  
  <value>oozie</value>  
</property>  
<property>  
  <name>oozie.service.JPAService.jdbc.password</name>  
  <value>oozie</value>  
</property>  
...
```



Note:

In the JDBC URL property, replace `localhost` with the hostname where PostgreSQL is running. In the case of PostgreSQL, unlike MySQL or Oracle, there is no need to download and install the JDBC driver separately, as it is license-compatible with Oozie and bundled with it.

Configuring Oozie to Use MySQL

Use the procedure that follows to configure Oozie to use MySQL instead of Apache Derby.

Install and start MySQL 5.x

Create the Oozie database and Oozie MySQL user.

For example, using the MySQL `mysql` command-line tool:

```
$ mysql -u root -p  
Enter password:  
  
mysql> create database oozie default character set utf8;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> exit  
Bye
```

Configure Oozie to use MySQL.

Edit properties in the `oozie-site.xml` file as follows:

```

...
<property>
  <name>oozie.service.JPAAService.jdbc.driver</name>
  <value>com.mysql.jdbc.Driver</value>
</property>
<property>
  <name>oozie.service.JPAAService.jdbc.url</name>
  <value>jdbc:mysql://localhost:3306/oozie</value>
</property>
<property>
  <name>oozie.service.JPAAService.jdbc.username</name>
  <value>oozie</value>
</property>
<property>
  <name>oozie.service.JPAAService.jdbc.password</name>
  <value>oozie</value>
</property>
...

```



Note: In the JDBC URL property, replace `localhost` with the hostname where MySQL is running.

Add the MySQL JDBC Driver JAR to Oozie

Copy or symbolically link the MySQL JDBC driver JAR into one of the following directories:

- For installations that use *packages*: `/var/lib/oozie/`
- For installations that use *parcels*: `/opt/cloudera/parcels/CDH/lib/oozie/lib/`

directory.



Note: You must manually download the MySQL JDBC driver JAR file.

Configuring Oozie to use Oracle

Use the procedure that follows to configure Oozie to use Oracle 11g instead of Apache Derby.

Install and start Oracle 11g.

Use [Oracle's instructions](#).

Create the Oozie Oracle user and grant privileges.

The following example uses the Oracle `sqlplus` command-line tool, and shows the privileges Cloudera recommends.

```

$ sqlplus system@localhost

Enter password: *****

SQL> create user oozie identified by oozie default tablespace users temporary tablespace
temp;

User created.

SQL> grant alter any index to oozie;
grant alter any table to oozie;
grant alter database link to oozie;
grant create any index to oozie;
grant create any sequence to oozie;

```

```
grant create database link to oozie;  
grant create session to oozie;  
grant create table to oozie;  
grant drop any sequence to oozie;  
grant select any dictionary to oozie;  
grant drop any table to oozie;  
grant create procedure to oozie;  
grant create trigger to oozie;  
  
SQL> exit  
  
$
```



Important:

Do *not* make the following grant:

```
grant select any table;
```

Configure Oozie to use Oracle.

Edit the `oozie-site.xml` file as follows.

```
...  
<property>  
  <name>oozie.service.JPAService.jdbc.driver</name>  
  <value>oracle.jdbc.OracleDriver</value>  
</property>  
<property>  
  <name>oozie.service.JPAService.jdbc.url</name>  
  <value>jdbc:oracle:thin:@//myhost:1521/oozie</value>  
</property>  
<property>  
  <name>oozie.service.JPAService.jdbc.username</name>  
  <value>oozie</value>  
</property>  
<property>  
  <name>oozie.service.JPAService.jdbc.password</name>  
  <value>oozie</value>  
</property>  
...
```



Note: In the JDBC URL property, replace `myhost` with the hostname where Oracle is running and replace `oozie` with the TNS name of the Oracle database.

Add the Oracle JDBC driver JAR to Oozie.

Copy or symbolically link the Oracle JDBC driver JAR into the `/var/lib/oozie/` directory.



Note: You must manually download the Oracle JDBC driver JAR file.

Creating the Oozie Database Schema

After configuring Oozie database information and creating the corresponding database, create the Oozie database schema. Oozie provides a database tool for this purpose.



Note: The Oozie database tool uses Oozie configuration files to connect to the database to perform the schema creation; before you use the tool, make you have created a database and configured Oozie to work with it as described above.

The Oozie database tool works in 2 modes: it can create the database, or it can produce an SQL script that a database administrator can run to create the database manually. If you use the tool to create the database schema, you must have the permissions needed to execute DDL operations.

To run the Oozie database tool against the database



Important: This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh create -run
```

You should see output such as the following (the output of the script may differ slightly depending on the database vendor) :

```
Validate DB Connection.
DONE
Check DB schema does not exist
DONE
Check OOZIE_SYS table does not exist
DONE
Create SQL schema
DONE
DONE
Create OOZIE_SYS table
DONE

Oozie DB has been created for Oozie version '4.0.0-cdh5.0.0'

The SQL commands have been written to: /tmp/ooziedb-5737263881793872034.sql
```

To create the upgrade script



Important: This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

Run `/usr/lib/oozie/bin/ooziedb.sh create -sqlfile SCRIPT`. For example:

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh create -sqlfile oozie-create.sql
```

You should see output such as the following (the output of the script may differ slightly depending on the database vendor) :

```
Validate DB Connection.
DONE
Check DB schema does not exist
DONE
Check OOZIE_SYS table does not exist
DONE
Create SQL schema
DONE
DONE
Create OOZIE_SYS table
DONE

Oozie DB has been created for Oozie version '4.0.0-cdh5.0.0'
```

The SQL commands have been written to: `oozie-create.sql`

WARN: The SQL commands have NOT been executed, you must use the `'-run'` option



Important: If you used the `-sqlfile` option instead of `-run`, Oozie database schema has not been created. You must run the `oozie-create.sql` script against your database.

Enabling the Oozie Web Console

To enable the Oozie web console, download and add the ExtJS library to the Oozie server. *If you have not already done this*, proceed as follows.

Step 1: Download the Library

Download the ExtJS version 2.2 library from <https://archive.cloudera.com/gplextras/misc/ext-2.2.zip> and place it a convenient location.

Step 2: Install the Library

Extract the `ext-2.2.zip` file into `/var/lib/oozie`.

Configuring Oozie with Kerberos Security

To configure Oozie with Kerberos security, see [Oozie Authentication](#).

Installing the Oozie Shared Library in Hadoop HDFS

The Oozie installation bundles the Oozie shared library, which contains all of the necessary JARs to enable workflow jobs to run streaming, DistCp, Pig, Hive, and Sqoop actions.

The Oozie installation bundles two shared libraries, one for MRv1 and one for YARN. Make sure you install the right one for the MapReduce version you are using:

- The shared library file for MRv1 is `oozie-sharelib-mr1.tar.gz`.
- The shared library file for YARN is `oozie-sharelib-yarn.tar.gz`.



Important: If Hadoop is configured with Kerberos security enabled, you must first configure Oozie with Kerberos Authentication. For instructions, see [Oozie Security Configuration](#). Before running the commands in the following instructions, you must run the `sudo -u oozie kinit -k -t /etc/oozie/oozie.keytab` and `kinit -k hdfs` commands. Then, instead of using commands in the form `sudo -u user command`, use just `command`; for example, `$ hadoop fs -mkdir /user/oozie`

To install the Oozie shared library in Hadoop HDFS in the oozie user home directory

```
$ sudo -u hdfs hadoop fs -mkdir /user/oozie
$ sudo -u hdfs hadoop fs -chown oozie:oozie /user/oozie
$ sudo oozie-setup sharelib create -fs <FS_URI> -locallib
/usr/lib/oozie/oozie-sharelib-yarn.tar.gz
```

where `FS_URI` is the HDFS URI of the filesystem that the shared library should be installed on (for example, `hdfs://<HOST>:<PORT>`).



Important: If you are installing Oozie to work with MRv1 use `oozie-sharelib-mr1.tar.gz` instead.

Configuring Support for Oozie Uber JARs

An **uber JAR** is a JAR that contains other JARs with dependencies in a `lib/` folder inside the JAR. You can configure the cluster to handle uber JARs properly for the MapReduce action (as long as it does not include any streaming or pipes) by setting the following property in the `oozie-site.xml` file:

```
...
  <property>
    <name>oozie.action.mapreduce.uber.jar.enable</name>
    <value>true</value>
  </property>
...
```

When this property is set, users can use the `oozie.mapreduce.uber.jar` configuration property in their MapReduce workflows to notify Oozie that the specified JAR file is an uber JAR.

Configuring Oozie to Run against a Federated Cluster

To run Oozie against a federated HDFS cluster using ViewFS, configure the `oozie.service.HadoopAccessorService.supported.filesystems` property in `oozie-site.xml` as follows:

```
<property>
  <name>oozie.service.HadoopAccessorService.supported.filesystems</name>
  <value>hdfs,viewfs</value>
</property>
```

Starting, Stopping, and Accessing the Oozie Server

Starting the Oozie Server

After you have completed *all* of the required configuration steps, you can start Oozie:

```
$ sudo service oozie start
```

If you see the message `Oozie System ID [oozie-oozie] started in the oozie.log log file`, the system has started successfully.



Note:

By default, Oozie server runs on port 11000 and its URL is `http://<OOZIE_HOSTNAME>:11000/oozie`.

Stopping the Oozie Server

```
$ sudo service oozie stop
```

Accessing the Oozie Server with the Oozie Client

The Oozie client is a command-line utility that interacts with the Oozie server using the Oozie web-services API.

Use the `/usr/bin/oozie` script to run the Oozie client.

For example, if you want to invoke the client on the same machine where the Oozie server is running:

```
$ oozie admin -oozie http://localhost:11000/oozie -status
System mode: NORMAL
```

To make it convenient to use this utility, set the environment variable `OOZIE_URL` to point to the URL of the Oozie server. Then you can skip the `-oozie` option.

For example, if you want to invoke the client on the same machine where the Oozie server is running, set the `OOZIE_URL` to `http://localhost:11000/oozie`.

```
$ export OOZIE_URL=http://localhost:11000/oozie
$ oozie admin -version
Oozie server build version: 4.0.0-cdh5.0.0
```



Important:

If Oozie is configured with Kerberos Security enabled:

- You must have a Kerberos session running. For example, you can start a session by running the `kinit` command.
- **Do not** use `localhost` as in the above examples.

As with every service that uses Kerberos, Oozie has a Kerberos *principal* in the form `<SERVICE>/<HOSTNAME>@<REALM>`. In a Kerberos configuration, you **must** use the `<HOSTNAME>` value in the Kerberos principal to specify the Oozie server; for example, if the `<HOSTNAME>` in the principal is `myoozieserver.mydomain.com`, set `OOZIE_URL` as follows:

```
$ export OOZIE_URL=http://myoozieserver.mydomain.com:11000/oozie
```

If you use an alternate hostname or the IP address of the service, Oozie will not work properly.

Accessing the Oozie Server with a Browser

If you have enabled the Oozie web console by adding the ExtJS library, you can connect to the console at `http://<OOZIE_HOSTNAME>:11000/oozie`.



Note:

If the Oozie server is configured to use Kerberos HTTP SPNEGO Authentication, you must use a web browser that supports Kerberos HTTP SPNEGO (for example, Firefox or Internet Explorer).

Viewing the Oozie Documentation

For additional Oozie documentation, see <https://archive.cloudera.com/cdh5/cdh/5/oozie/>.

Pig Installation

Apache Pig enables you to analyze large amounts of data using Pig's query language called Pig Latin. Pig Latin queries run in a distributed way on a Hadoop cluster.

Use the following sections to install or upgrade Pig:

- [Upgrading Pig](#)
- [Installing Pig](#)
- [Using Pig with HBase](#)
- [Installing DataFu](#)
- [Apache Pig Documentation](#)

Upgrading Pig



Note:

To see which version of Pig is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [Release Notes](#).

Upgrading Pig from an Earlier CDH 5 release

The instructions that follow assume that you are upgrading Pig as part of a CDH 5 upgrade, and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

To upgrade Pig from an earlier CDH 5 release:

1. Exit the Grunt shell and make sure no Pig scripts are running.

2. Install the new version, following the instructions in the next section, [Installing Pig](#) on page 347.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from <file> to <file>.rpmsave. If you then re-install the package (probably to install a new version) the package manager creates a new <file> with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Installing Pig



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera yum, zypper/YaST or apt repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install Pig On RHEL-compatible systems:

```
$ sudo yum install pig
```

To install Pig on SLES systems:

```
$ sudo zypper install pig
```

To install Pig on Ubuntu and other Debian systems:

```
$ sudo apt-get install pig
```



Note:

Pig automatically uses the active Hadoop configuration (whether standalone, pseudo-distributed mode, or distributed). After installing the Pig package, you can start Pig.

To start Pig in interactive mode (YARN)



Important:

- For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop in a YARN installation, make sure that the `HADOOP_MAPRED_HOME` environment variable is set correctly, as follows:

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

- For each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop in an MRv1 installation, set the `HADOOP_MAPRED_HOME` environment variable as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

To start Pig, use the following command.

```
$ pig
```

To start Pig in interactive mode (MRv1)

Use the following command:

```
$ pig
```

You should see output similar to the following:

```
2012-02-08 23:39:41,819 [main] INFO org.apache.pig.Main - Logging error messages to:
/home/arvind/pig-0.11.0-cdh5b1/bin/pig_1328773181817.log
2012-02-08 23:39:41,994 [main] INFO
org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop
file system at: hdfs://localhost/
...
grunt>
```

Examples

To verify that the input and output directories from the [YARN](#) or [MRv1](#) example `grep` job exist, list an HDFS directory from the Grunt Shell:

```
grunt> ls
hdfs://localhost/user/joe/input <dir>
hdfs://localhost/user/joe/output <dir>
```

To run a `grep` example job using Pig for `grep` inputs:

```
grunt> A = LOAD 'input';
grunt> B = FILTER A BY $0 MATCHES '.*dfs[a-z.]+.*';
grunt> DUMP B;
```



Note:

To check the status of your job while it is running, look at the ResourceManager web console (YARN) or JobTracker web console (MRv1).

Using Pig with HBase

To allow Pig scripts to use HBase, add the following statement to the top of each script. Replace the `<component_version>` strings with the current HBase, ZooKeeper and CDH version numbers.

```
register /usr/lib/zookeeper/zookeeper-<ZooKeeper_version>-cdh<CDH_version>.jar
register /usr/lib/hbase/hbase-<HBase_version>-cdh<CDH_version>-security.jar
```

For example,

```
register /usr/lib/zookeeper/zookeeper-3.4.5-cdh5.0.0.jar
register /usr/lib/hbase/hbase-0.95.2-cdh5.0.0-security.jar
```

In addition, Pig needs to be able to access the `hbase-site.xml` file on the Hadoop client. Pig searches for the file within the `/etc/hbase/conf` directory on the client, or in Pig's `CLASSPATH` variable.

For more information about using Pig with HBase, see [Importing Data Into HBase](#).

Installing DataFu

DataFu is a collection of Apache Pig UDFs (User-Defined Functions) for statistical evaluation that were developed by LinkedIn and have now been open sourced under an Apache 2.0 license.

To use DataFu:

1. Install the DataFu package:

Operating system	Install command
Red-Hat-compatible	<code>sudo yum install pig-udf-datafu</code>
SLES	<code>sudo zypper install pig-udf-datafu</code>
Debian or Ubuntu	<code>sudo apt-get install pig-udf-datafu</code>

This puts the `datafu-0.0.4-cdh5.0.0.jar` file in `/usr/lib/pig`.

2. Register the JAR. Replace the `<component_version>` string with the current DataFu and CDH version numbers.

```
REGISTER /usr/lib/pig/datafu-<DataFu_version>-cdh<CDH_version>.jar
```

For example,

```
REGISTER /usr/lib/pig/datafu-0.0.4-cdh5.0.0.jar
```

A number of usage examples and other information are available at <https://github.com/linkedin/datafu>.

Viewing the Pig Documentation

For additional Pig documentation, see <https://archive.cloudera.com/cdh5/cdh/5/pig>.

Search Installation

This documentation describes how to install Cloudera Search powered by Solr. It also explains how to install and start supporting tools and services such as the ZooKeeper Server, MapReduce tools for use with Cloudera Search, and Flume Solr Sink.

After installing Cloudera Search as described in this document, you can configure and use Cloudera Search as described in the [Cloudera Search User Guide](#). The user guide includes the [Cloudera Search Tutorial](#), as well as topics that describe extracting, transforming, and loading data, establishing high availability, and troubleshooting.

Cloudera Search documentation includes:

- [CDH 5 Release Notes](#)
- [CDH Version and Packaging Information](#)
- [Cloudera Search User Guide](#)
- [Cloudera Search Frequently Asked Questions](#)

Preparing to Install Cloudera Search

Cloudera Search provides interactive search and scalable indexing. Before you begin installing Cloudera Search:

- Decide whether to install Cloudera Search using Cloudera Manager or using package management tools.
- Decide on which machines to install Cloudera Search and with which other services to collocate Search.
- Consider the sorts of tasks, workloads, and types of data you will be searching. This information can help guide your deployment process.



Important: Cloudera Search does not support `contrib` modules, such as `DataImportHandler`.

Choosing Where to Deploy the Cloudera Search Processes

You can collocate a Cloudera Search server (`solr-server` package) with a Hadoop TaskTracker (MRv1) and a DataNode. When collocating with TaskTrackers, be sure that the machine resources are not oversubscribed. Start with a small number of MapReduce slots and increase them gradually.

For instructions describing how and where to install `solr-mapreduce`, see [Installing MapReduce Tools for use with Cloudera Search](#). For information about the Search package, see the Using Cloudera Search section in the [Cloudera Search Tutorial](#).

Guidelines for Deploying Cloudera Search

Memory

CDH initially deploys Solr with a Java virtual machine (JVM) size of 1 GB. In the context of Search, 1 GB is a small value. Starting with this small value simplifies JVM deployment, but the value is insufficient for most actual use cases. Consider the following when determining an optimal JVM size for production usage:

- The more searchable material you have, the more memory you need. All things being equal, 10 TB of searchable data requires more memory than 1 TB of searchable data.
- What is indexed in the searchable material. Indexing all fields in a collection of logs, email messages, or Wikipedia entries requires more memory than indexing only the `Date Created` field.
- The level of performance required. If the system must be stable and respond quickly, more memory may help. If slow responses are acceptable, you may be able to use less memory.

To ensure an appropriate amount of memory, consider your requirements and experiment in your environment. In general:

- 4 GB is sufficient for some smaller loads or for evaluation.
- 12 GB is sufficient for some production environments.
- 48 GB is sufficient for most situations.

Deployment Requirements

The information in this topic should be considered as guidance instead of absolute requirements. Using a sample application to benchmark different use cases and data types and sizes can help you identify the most important performance factors.

To determine how best to deploy search in your environment, define use cases. The same Solr index can have very different hardware requirements, depending on queries performed. The most common variation in hardware requirement is memory. For example, the memory requirements for faceting vary depending on the number of unique

terms in the faceted field. Suppose you want to use faceting on a field that has ten unique values. In this case, only ten logical containers are required for counting. No matter how many documents are in the index, memory overhead is almost nonexistent.

Conversely, the same index could have unique timestamps for every entry, and you want to facet on that field with a `: -type` query. In this case, each index requires its own logical container. With this organization, if you had a large number of documents—500 million, for example—then faceting across 10 fields would increase the RAM requirements significantly.

For this reason, use cases and some characterizations of the data is required before you can estimate hardware requirements. Important parameters to consider are:

- Number of documents. For Cloudera Search, sharding is almost always required.
- Approximate word count for each potential field.
- What information is stored in the Solr index and what information is only for searching. Information stored in the index is returned with the search results.
- Foreign language support:
 - How many different languages appear in your data?
 - What percentage of documents are in each language?
 - Is language-specific search supported? This determines whether accent folding and storing the text in a single field is sufficient.
 - What language families will be searched? For example, you could combine all Western European languages into a single field, but combining English and Chinese into a single field is not practical. Even with more similar sets of languages, using a single field for different languages can be problematic. For example, sometimes accents alter the meaning of a word, and in such a case, accent folding loses important distinctions.
- Faceting requirements:
 - Be wary of faceting on fields that have many unique terms. For example, faceting on timestamps or free-text fields typically has a high cost. Faceting on a field with more than 10,000 unique values is typically not useful. Ensure that any such faceting requirement is necessary.
 - What types of facets are needed? You can facet on queries as well as field values. Faceting on queries is often useful for dates. For example, “in the last day” or “in the last week” can be valuable. Using Solr Date Math to facet on a bare “NOW” is almost always inefficient. Facet-by-query is not memory-intensive because the number of logical containers is limited by the number of queries specified, no matter how many unique values are in the underlying field. This can enable faceting on fields that contain information such as dates or times, while avoiding the problem described for faceting on fields with unique terms.
- Sorting requirements:
 - Sorting requires one integer for each document (`maxDoc`), which can take up significant memory. Additionally, sorting on strings requires storing each unique string value.
- Paging requirements. End users rarely look beyond the first few pages of search results. For use cases requiring *deep paging* (paging through a large number of results), using *cursors* can improve performance and resource utilization. For more information, see [Pagination of Results](#) on the Apache Solr wiki. Cursors are supported in CDH 5.2 and higher.
- Is an “advanced” search capability planned? If so, how will it be implemented? Significant design decisions depend on user motivation levels:
 - Can users be expected to learn about the system? “Advanced” screens could intimidate e-commerce users, but these screens may be most effective if users can be expected to learn them.
 - How long will your users wait for results? Data mining results in longer user wait times. You want to limit user wait times, but other design requirements can affect response times.
- How many simultaneous users must your system accommodate?
- Update requirements. An update in Solr refers both to adding new documents and changing existing documents:
 - Loading new documents:

Installing Cloudera Manager and CDH

- Bulk. Will the index be rebuilt from scratch in some cases, or will there only be an initial load?
- Incremental. At what rate will new documents enter the system?
- Updating documents. Can you characterize the expected number of modifications to existing documents?
- How much latency is acceptable between when a document is added to Solr and when it is available in Search results?
- Security requirements. Solr has no built-in security options, although Cloudera Search supports [authentication using Kerberos](#) and [authorization using Sentry](#). In Solr, document-level security is usually best accomplished by indexing authorization tokens with the document. The number of authorization tokens applied to a document is largely irrelevant; for example, thousands are reasonable but can be difficult to administer. The number of authorization tokens associated with a particular user should be no more than 100 in most cases. Security at this level is often enforced by appending an “fq” clause to the query, and adding thousands of tokens in an “fq” clause is expensive.
 - A *post filter*, also known as a *no-cache filter*, can help with access schemes that cannot use an “fq” clause. These are not cached and are applied only after all less-expensive filters are applied.
 - If grouping, faceting is not required to accurately reflect true document counts, so you can use some shortcuts. For example, ACL filtering is expensive in some systems, sometimes requiring database access. If completely accurate faceting is required, you must completely process the list to reflect accurate facets.
- Required query rate, usually measured in queries-per-second (QPS):
 - At a minimum, deploy machines with sufficient hardware resources to provide an acceptable response rate for a single user. You can create queries that burden the system so much that performance for even a small number of users is unacceptable. In this case, resharding is necessary.
 - If QPS is only somewhat slower than required and you do not want to reshard, you can improve performance by adding replicas to each shard.
 - As the number of shards in your deployment increases, so too does the likelihood that one of the shards will be unusually slow. In this case, the general QPS rate falls, although very slowly. This typically occurs as the number of shards reaches the hundreds.

Installing Cloudera Search

You can install Cloudera Search in one of two ways:

- Using the Cloudera Manager installer, as described in [Installing Search](#) on page 122. This technique is recommended for reliable and verifiable Search installation.
- Using the manual process described in [Installing Cloudera Search without Cloudera Manager](#) on page 352. This process requires you to configure access to the Cloudera Search repository and then install Search packages.



Note: Depending on which installation approach you use, Search is installed to different locations.

- Installing Search with Cloudera Manager using parcels results in changes under `/opt/cloudera/parcels`.
- Installing using packages, either manually or using Cloudera Manager, results in changes to various locations throughout the file system. Common locations for changes include `/usr/lib/`, `/etc/default/`, and `/usr/share/doc/`.

Installing Cloudera Search without Cloudera Manager

- Cloudera Search for CDH 5 is included with CDH 5. To install Cloudera Search for CDH 5 using packages, see [Installing the Latest CDH 5 Release](#) on page 155.



Note: This page describes how to install CDH using packages as well as how to install CDH using Cloudera Manager.

You can also elect to install Cloudera Search manually. For example, you might choose to install Cloudera Search manually if you have an existing installation to which you want to add Search.

To use CDH 5, which includes Cloudera Search:

- For general information about using repositories to install or upgrade Cloudera software, see Understanding Custom Installation Solutions in [Understanding Custom Installation Solutions](#).
- For instructions on installing or upgrading CDH, see [CDH 5 Installation](#) and the instructions for [Upgrading from CDH 4 to CDH 5](#).
- For CDH 5 repository locations and client `.repo` files, which include Cloudera Search, see [CDH Version and Packaging Information](#).

Cloudera Search provides the following packages:

Package Name	Description
solr	Solr
solr-server	Platform specific service script for starting, stopping, or restart Solr.
solr-doc	Cloudera Search documentation.
solr-mapreduce	Tools to index documents using MapReduce.
solr-crunch	Tools to index documents using Crunch.
search	Examples, Contrib, and Utility code and data.

Before You Begin Installing Cloudera Search Without Cloudera Manager

The installation instructions assume that the `sudo` command is configured on the hosts where you are installing Cloudera Search. If `sudo` is not configured, use the root user (`superuser`) to configure Cloudera Search.



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 the root directory (`/`) and removes environment variables except `LANG` and `TERM`. This creates a predictable environment in which to administer the service. If you use `/etc/init.d` scripts directly, any environment variables continue to be applied, potentially producing unexpected results. If you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

Install Cloudera's repository: before using the instructions in this guide to install or upgrade Cloudera Search from packages, install Cloudera's `yum`, `zypper/YaST` or `apt` repository, and install or upgrade CDH and make sure it is functioning correctly.

Installing Solr Packages

This topic describes how to complete a new installation of Solr packages. To upgrade an existing installation, see [Upgrading Cloudera Search](#) on page 359.

To install Cloudera Search on RHEL systems:

```
sudo yum install solr-server
```

To install Cloudera Search on Ubuntu and Debian systems:

```
$ sudo apt-get install solr-server
```

To install Cloudera Search on SLES systems:

```
$ sudo zypper install solr-server
```



Note: See also [Deploying Cloudera Search](#) on page 354.

To list the installed files on RHEL and SLES systems:

```
$ rpm -ql solr-server solr
```

To list the installed files on Ubuntu and Debian systems:

```
$ dpkg -L solr-server solr
```

You can see that the Cloudera Search packages are configured according to the Linux Filesystem Hierarchy Standard.

Next, enable the server daemons you want to use with Hadoop. You can also enable Java-based client access by adding the JAR files in `/usr/lib/solr/` and `/usr/lib/solr/lib/` to your Java class path.

Deploying Cloudera Search

When you deploy Cloudera Search, SolrCloud partitions your data set into multiple indexes and processes, using ZooKeeper to simplify management, resulting in a cluster of coordinating Solr servers.



Note: Before you start

This section assumes that you have already installed Search. Installing Search can be accomplished:

- Using Cloudera Manager as described in [Installing Search](#) on page 122.
- Without Cloudera Manager as described in [Installing Cloudera Search without Cloudera Manager](#) on page 352.

Now you are distributing the processes across multiple hosts. Before completing this process, you may want to review [Choosing where to Deploy the Cloudera Search Processes](#).

Installing and Starting ZooKeeper Server

SolrCloud mode uses a ZooKeeper Service as a highly available, central location for cluster management. For a small cluster, running a ZooKeeper host collocated with the NameNode is recommended. For larger clusters, you may want to run multiple ZooKeeper servers. For more information, see [Installing the ZooKeeper Packages](#) on page 386.

Initializing Solr

Once the ZooKeeper Service is running, configure each Solr host with the ZooKeeper Quorum address or addresses. Provide the ZooKeeper Quorum address for each ZooKeeper server. This could be a single address in smaller deployments, or multiple addresses if you deploy additional servers.

Configure the ZooKeeper Quorum address in `solr-env.sh`. The file location varies by installation type. If you accepted default file locations, the `solr-env.sh` file can be found in:

- Parcels: `/opt/cloudera/parcels/1.0.0+cdh5.3.10+0/etc/default/solr`
- Packages: `/etc/default/solr`

Edit the property to configure the hosts with the address of the ZooKeeper service. You must make this configuration change for every Solr Server host. The following example shows a configuration with three ZooKeeper hosts:

```
SOLR_ZK_ENSEMBLE=<zookeeper1>:2181,<zookeeper2>:2181,<zookeeper3>:2181/solr
```

Configuring Solr for Use with HDFS

To use Solr with your established HDFS service, perform the following configurations:

1. Configure the HDFS URI for Solr to use as a backing store in `/etc/default/solr` or `/opt/cloudera/parcels/1.0.0+cdh5.3.10+0/etc/default/solr`. On every Solr Server host, edit the following property to configure the location of Solr index data in HDFS:

```
SOLR_HDFS_HOME=hdfs://namenodehost:8020/solr
```

Replace `namenodehost` with the hostname of your HDFS NameNode (as specified by `fs.default.name` or `fs.defaultFS` in your `conf/core-site.xml` file). You may also need to change the port number from the default (8020). On an HA-enabled cluster, ensure that the HDFS URI you use reflects the designated name service utilized by your cluster. This value should be reflected in `fs.default.name`; instead of a hostname, you would see `hdfs://nameservice1` or something similar.

2. In some cases, such as for configuring Solr to work with HDFS High Availability (HA), you may want to configure the Solr HDFS client by setting the HDFS configuration directory in `/etc/default/solr` or `/opt/cloudera/parcels/1.0.0+cdh5.3.10+0/etc/default/solr`. On every Solr Server host, locate the appropriate HDFS configuration directory and edit the following property with the absolute path to this directory:

```
SOLR_HDFS_CONFIG=/etc/hadoop/conf
```

Replace the path with the correct directory containing the proper HDFS configuration files, `core-site.xml` and `hdfs-site.xml`.

Configuring Solr to Use Secure HDFS

For information on setting up a secure CDH cluster, see the [CDH 5 Security Guide](#).

In addition to the previous steps for Configuring Solr for use with HDFS, perform the following steps if security is enabled:

1. Create the Kerberos principals and Keytab files for every host in your cluster:

- a. Create the Solr principal using either `kadmin` or `kadmin.local`.

```
kadmin: addprinc -randkey solr/fully.qualified.domain.name@YOUR-REALM.COM
```

```
kadmin: xst -norandkey -k solr.keytab solr/fully.qualified.domain.name
```

For more information, see [Step 4: Create and Deploy the Kerberos Principals and Keytab Files](#)

2. Deploy the Kerberos Keytab files on every host in your cluster:

- a. Copy or move the keytab files to a directory that Solr can access, such as `/etc/solr/conf`.

```
$ sudo mv solr.keytab /etc/solr/conf/
```

```
$ sudo chown solr:hadoop /etc/solr/conf/solr.keytab
$ sudo chmod 400 /etc/solr/conf/solr.keytab
```

3. Add Kerberos-related settings to `/etc/default/solr` or `/opt/cloudera/parcels/1.0.0+cdh5.3.10+0/etc/default/solr` on every host in your cluster, substituting appropriate values. For a package based installation, use something similar to the following:

```
SOLR_KERBEROS_ENABLED=true
SOLR_KERBEROS_KEYTAB=/etc/solr/conf/solr.keytab
SOLR_KERBEROS_PRINCIPAL=solr/fully.qualified.domain.name@YOUR-REALM.COM
```

Creating the /solr Directory in HDFS

Before starting the Cloudera Search server, you need to create the `/solr` directory in HDFS. The Cloudera Search master runs as `solr:solr`, so it does not have the required permissions to create a top-level directory.

To create the `/solr` directory in HDFS:

```
$ sudo -u hdfs hadoop fs -mkdir /solr
$ sudo -u hdfs hadoop fs -chown solr /solr
```

Initializing the ZooKeeper Namespace

Before starting the Cloudera Search server, you need to create the `solr` namespace in ZooKeeper:

```
$ solrctl init
```



Warning: `solrctl init` takes a `--force` option as well. `solrctl init --force` clears the Solr data in ZooKeeper and interferes with any running hosts. If you clear Solr data from ZooKeeper to start over, be sure to stop the cluster first.

Starting Solr

To start the cluster, start Solr Server on each host:

```
$ sudo service solr-server restart
```

After you have started the Cloudera Search Server, the Solr server should be running. To verify that all daemons are running, use the `jps` tool from the Oracle JDK, which you can obtain from the [Java SE Downloads](#) page. If you are running a pseudo-distributed HDFS installation and a Solr search installation on one machine, `jps` shows the following output:

```
$ sudo jps -lm
31407 sun.tools.jps.Jps -lm
31236 org.apache.catalina.startup.Bootstrap start
```

Runtime Solr Configuration

To start using Solr for indexing the data, you must configure a collection holding the index. A configuration for a collection requires a `solrconfig.xml` file, a `schema.xml` and any helper files referenced from the `xml` files. The `solrconfig.xml` file contains all of the Solr settings for a given collection, and the `schema.xml` file specifies the schema that Solr uses when indexing documents. For more details on how to configure a collection for your data set, see <http://wiki.apache.org/solr/SchemaXml>.

Configuration files for a collection are managed as part of the instance directory. To generate a skeleton of the instance directory, run the following command:

```
$ solrctl instancedir --generate $HOME/solr_configs
```

You can customize it by directly editing the `solrconfig.xml` and `schema.xml` files created in `$HOME/solr_configs/conf`.

These configuration files are compatible with the standard Solr tutorial example documents.

After configuration is complete, you can make it available to Solr by issuing the following command, which uploads the content of the entire instance directory to ZooKeeper:

```
$ solrctl instancedir --create collection1 $HOME/solr_configs
```

Use the `solrctl` tool to verify that your instance directory uploaded successfully and is available to ZooKeeper. List the contents of an instance directory as follows:

```
$ solrctl instancedir --list
```

If you used the earlier `--create` command to create `collection1`, the `--list` command should return `collection1`.



Important:

If you are familiar with Apache Solr, you might configure a collection directly in solr home: `/var/lib/solr`. Although this is possible, Cloudera recommends using `solrctl` instead.

Creating Your First Solr Collection

By default, the Solr server comes up with no collections. Make sure that you create your first collection using the `instancedir` that you provided to Solr in previous steps by using the same collection name. `numOfShards` is the number of SolrCloud shards you want to partition the collection across. The number of shards cannot exceed the total number of Solr servers in your SolrCloud cluster:

```
$ solrctl collection --create collection1 -s {{numOfShards}}
```

You should be able to check that the collection is active. For example, for the server `myhost.example.com`, you should be able to navigate to `http://myhost.example.com:8983/solr/collection1/select?q=%3A*&wt=json&indent=true` and verify that the collection is active. Similarly, you should be able to view the topology of your SolrCloud using a URL similar to `http://myhost.example.com:8983/solr/#/~cloud`.

Adding Another Collection with Replication

To support scaling for the query load, create a second collection with replication. Having multiple servers with replicated collections distributes the request load for each shard. Create one shard cluster with a replication factor of two. Your cluster must have at least two running servers to support this configuration, so ensure Cloudera Search is installed on at least two servers. A replication factor of two causes two copies of the index files to be stored in two different locations.

1. Generate the config files for the collection:

```
$ solrctl instancedir --generate $HOME/solr_configs2
```

2. Upload the instance directory to ZooKeeper:

```
$ solrctl instancedir --create collection2 $HOME/solr_configs2
```

3. Create the second collection:

```
$ solrctl collection --create collection2 -s 1 -r 2
```

4. Verify that the collection is live and that the one shard is served by two hosts. For example, for the server `myhost.example.com`, you should receive content from: `http://myhost.example.com:8983/solr/#/~cloud`.

Creating Replicas of Existing Shards

You can create additional replicas of existing shards using a command of the following form:

```
solrctl --zk <zkensemble> --solr <target solr server> core \  
--create <new core name> -p collection=<collection> -p shard=<shard to replicate>
```

For example to create a new replica of collection named `collection1` that is comprised of `shard1`, use the following command:

```
solrctl --zk myZKEnsemble:2181/solr --solr mySolrServer:8983/solr core \  
--create collection1_shard1_replica2 -p collection=collection1 -p shard=shard1
```

Adding a New Shard to a Solr Server

You can use `solrctl` to add a new shard to a specified solr server.

```
solrctl --solr http://<target_solr_server>:8983/solr core --create <core_name> \  
-p dataDir=hdfs://<nameservice>/<index_hdfs_path> -p collection.configName=<config_name> \  
\ \  
-p collection=<collection_name> -p numShards=<int> -p shard=<shard_id>
```

Where:

- `target_solr_server`: The server to host the new shard
- `core_name`: `<collection_name><shard_id><replica_id>`
- `shard_id`: New shard identifier

For example, to add a new second shard named `shard2` to a solr server named `mySolrServer`, where the collection is named `myCollection`, you would use the following command:

```
solrctl --solr http://mySolrServer:8983/solr core --create myCore \  
-p dataDir=hdfs://namenode/solr/myCollection/index -p collection.configName=myConfig \  
-p collection=myCollection -p numShards=2 -p shard=shard2
```

Installing the Spark Indexer

The Spark indexer uses a Spark or MapReduce ETL batch job to move data from HDFS files into Apache Solr. As part of this process, the indexer uses Morphlines to extract and transform data.

To use the Spark indexer, you must install the `solr-crunch` package on hosts where you want to submit a batch indexing job.

To install solr-crunch On RHEL systems:

```
$ sudo yum install solr-crunch
```

To install solr-crunch on Ubuntu and Debian systems:

```
$ sudo apt-get install solr-crunch
```

To install solr-crunch on SLES systems:

```
$ sudo zypper install solr-crunch
```

For information on using Spark to batch index documents, see the [Spark Indexing](#).

Installing MapReduce Tools for use with Cloudera Search

Cloudera Search provides the ability to batch index documents using MapReduce jobs. Install the `solr-mapreduce` package on hosts where you want to submit a batch indexing job.

To install solr-mapreduce On RHEL systems:

```
$ sudo yum install solr-mapreduce
```

To install solr-mapreduce on Ubuntu and Debian systems:

```
$ sudo apt-get install solr-mapreduce
```

To install solr-mapreduce on SLES systems:

```
$ sudo zypper install solr-mapreduce
```

For information on using MapReduce to batch index documents, see the [MapReduce Batch Indexing Reference](#).

Installing the Lily HBase Indexer Service

To query data stored in HBase, you must install the Lily HBase Indexer service. This service indexes the stream of records being added to HBase tables. This process is scalable, fault tolerant, transactional, and operates at near real-time (NRT). The typical delay is a few seconds between the time data arrives and the time the same data appears in search results.

Choosing where to Deploy the Lily HBase Indexer Service Processes

To accommodate the HBase ingest load, you can run as many Lily HBase Indexer services on different hosts as required. See the HBase replication documentation for details on how to plan the capacity. You can co-locate Lily HBase Indexer service processes with SolrCloud on the same set of hosts.

To install the Lily HBase Indexer service on RHEL systems:

```
$ sudo yum install hbase-solr-indexer hbase-solr-doc
```

To install the Lily HBase Indexer service on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-solr-indexer hbase-solr-doc
```

To install the Lily HBase Indexer service on SUSE-based systems:

```
$ sudo zypper install hbase-solr-indexer hbase-solr-doc
```



Important: For the Lily HBase Indexer to work with CDH 5, you may need to run the following command before issuing Lily HBase MapReduce jobs:

```
export HADOOP_CLASSPATH=<Path to hbase-protocol-*.jar>
```

Upgrading Cloudera Search

You can upgrade an existing Cloudera Search installation in several ways. Generally, you stop Cloudera Search services, update Search to the latest version, and then restart Cloudera Search services. You can update Search to the latest version by using the package management tool for your operating system and then restarting Cloudera Search services.

Upgrading with Cloudera Manager

If you are running Cloudera Manager, you can upgrade from within the Cloudera Manager Admin Console using parcels. For Search for CDH 5, search is included in the CDH 5 parcel. To upgrade from previous versions of CDH 5, follow the instructions at [Performing a Rolling Upgrade on a CDH 5 Cluster](#).

Upgrading Manually without Cloudera Manager

Important: Before upgrading, make backup copies of the following configuration files:

- /etc/default/solr
- All collection configurations

Make sure you copy every host that is part of the SolrCloud.

- Cloudera Search for CDH 5 is included as part of CDH 5. Therefore, to upgrade from previous versions of Cloudera Search for CDH 5 to the latest version of Cloudera Search, simply upgrade CDH. For more information, see [Upgrading from an Earlier CDH 5 Release to the Latest Release](#) or [Upgrading from CDH 5.2.0 or Higher to the Latest Release](#).

Installing Hue Search

You must install and configure Hue before you can use Search with Hue.

1. Follow the instructions for [Hue Installation](#) on page 301.
2. Use **one** of the following commands to install Search applications on the Hue machine:

For package installation on RHEL systems:

```
sudo yum install hue-search
```

For package installation on SLES systems:

```
sudo zypper install hue-search
```

For package installation on Ubuntu or Debian systems:

```
sudo apt-get install hue-search
```

For installation using tarballs:

```
$ cd /usr/share/hue
$ sudo tar -xzvf hue-search-####.tar.gz
$ sudo /usr/share/hue/tools/app_reg/app_reg.py \
--install /usr/share/hue/apps/search
```

3. Update the configuration information for the Solr Server:

Cloudera Manager Environment	Environment without Cloudera Manager
<ol style="list-style-type: none"> 1. Connect to Cloudera Manager. 2. Select the Hue service. 3. Click Configuration > View and Edit. 4. Search for the word "safety". 5. Add information about your Solr host to Hue Server (Base) / Advanced. For example, if your hostname is SOLR_HOST, you might add the following: <pre>[search] # URL of the Solr Server solr_url=http://SOLR_HOST:8983/solr</pre> 	<p>Update configuration information in <code>/etc/hue/hue.ini</code>.</p> <ol style="list-style-type: none"> 1. Specify the Solr URL. For example, to use <code>localhost</code> as your Solr host, you would add the following: <pre>[search] # URL of the Solr Server, replace 'localhost' if Solr is running on another host solr_url=http://localhost:8983/solr/</pre>
<ol style="list-style-type: none"> 6. (Optional) To enable Hue in environments where Kerberos authentication is required, update the <code>security_enabled</code> property as follows: <pre># Requires FQDN in solr_url if enabled security_enabled=true</pre> 	<ol style="list-style-type: none"> 2. (Optional) To enable Hue in environments where Kerberos authentication is required, update the <code>security_enabled</code> property as follows: <pre># Requires FQDN in solr_url if enabled security_enabled=true</pre>

4. Configure secure impersonation for Hue.

- If you are using Search in an environment that uses Cloudera Manager 4.8 or later, secure impersonation for Hue is automatically configured. To review secure impersonation settings in the Cloudera Manager home page:
 1. Go to the HDFS service.
 2. Click the **Configuration** tab.
 3. Select **Scope > All**.
 4. Select **Category > All**.

5. Type `hue proxy` in the Search box.
 6. Note the Service-Wide wild card setting for **Hue Proxy Hosts** and **Hue Proxy User Groups**.
- If you are not using Cloudera Manager or are using a version earlier than Cloudera Manager 4.8, configure Hue to impersonate any user that makes requests by modifying `/etc/default/solr`. The changes you make may vary according to the users for which you want to configure secure impersonation. For example, you might make the following changes:

```
SOLR_SECURITY_ALLOWED_PROXYUSERS=hue
SOLR_SECURITY_PROXYUSER_hue_HOSTS=*
SOLR_SECURITY_PROXYUSER_hue_GROUPS=*
```

For more information about Secure Impersonation or to set up additional users for Secure Impersonation, see [Enabling Secure Impersonation](#).

5. (Optional) To view files in HDFS, ensure that the correct `webhdfs_url` is included in `hue.ini` and WebHDFS is properly configured as described in [Configuring CDH Components for Hue](#) on page 306.
6. Restart Hue:

```
$ sudo /etc/init.d/hue restart
```

7. Open `http://hue-host.com:8888/search/` in your browser.

Updating Hue Search

To update Hue search, install updates and restart the Hue service.

1. On the Hue machine, update Hue search:

```
$ cd /usr/share/hue
$ sudo tar -xzvf hue-search-####.tar.gz
$ sudo /usr/share/hue/tools/app_reg/app_reg.py \
--install /usr/share/hue/apps/search
```

2. Restart Hue:

```
$ sudo /etc/init.d/hue restart
```

Sentry Installation

Sentry enables role-based, fine-grained authorization for HiveServer2 and Cloudera Impala. It provides classic database-style authorization for Hive and Impala. For more information, and instructions on configuring Sentry for Hive and Impala, see [The Sentry Service](#).

Installing Sentry

Use the following the instructions, depending on your operating system, to install the latest version of Sentry.



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

OS	Command
RHEL	\$ sudo yum install sentry
SLES	\$ sudo zypper install sentry
Ubuntu or Debian	\$ sudo apt-get update; \$ sudo apt-get install sentry

Upgrading Sentry

Upgrading from CDH 5.x to the Latest CDH 5

1. Stop the Sentry Service

To stop the Sentry service, identify the PID of the Sentry Service and use the `kill` command to end the process:

```
ps -ef | grep sentry
kill -9 <PID>
```

Replace `<PID>` with the PID of the Sentry Service.

2. Remove the previous version of Sentry.

OS	Command
RHEL	\$ sudo yum remove sentry
SLES	\$ sudo zypper remove sentry
Ubuntu or Debian	\$ sudo apt-get remove sentry

3. [Install the new version of Sentry.](#)

4. (From a release earlier than CDH 5.2.0 to CDH 5.2.0 or later) Upgrade Sentry Database Schema

Use the Sentry `schematool` to upgrade the database schema as follows:

```
bin/sentry --command schema-tool --confdir <sentry-site.xml> --dbType <db-type>
--upgradeSchema
```

Where `<db-type>` should be either `mysql`, `postgres` or `oracle`.

5. Start the Sentry Service

- a. Set the `SENTRY_HOME` and `HADOOP_HOME` parameters.
- b. Run the following command:

```
bin/sentry --command service --confdir <sentry-site.xml>
```

Snappy Installation

Snappy is a compression/decompression library. It aims for very high speeds and reasonable compression, rather than maximum compression or compatibility with other compression libraries. Use the following sections to install, upgrade, and use Snappy.

- [Upgrading Snappy](#)
- [Installing Snappy](#)
- [Using Snappy for MapReduce Compression](#)
- [Using Snappy for Pig Compression](#)
- [Using Snappy for Hive Compression](#)
- [Using Snappy Compression in Sqoop Imports](#)

- [Using Snappy Compression with HBase](#)
- [Apache Snappy Documentation](#)

Upgrading Snappy

To upgrade Snappy, simply install the `hadoop` package if you haven't already done so. This applies whether you are upgrading from an earlier CDH 5 release.



Note:

To see which version of Hadoop is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Installing Snappy

Snappy is provided in the `hadoop` package along with the other native libraries (such as native gzip compression).



Warning:

If you install Hadoop from a tarball, Snappy may not work, because the Snappy native library may not be compatible with the version of Linux on your system. If you want to use Snappy, install CDH 5 from the RHEL or Debian packages.

To take advantage of Snappy compression you need to set certain configuration properties, which are explained in the following sections.

Using Snappy for MapReduce Compression

It's very common to enable MapReduce intermediate compression, since this can make jobs run faster without you having to make any application changes. Only the temporary intermediate files created by Hadoop for the shuffle phase are compressed (the final output may or may not be compressed). Snappy is ideal in this case because it compresses and decompresses very fast compared to other compression algorithms, such as Gzip. For information about choosing a compression format, see [Choosing a Data Compression Format](#).

To enable Snappy for MapReduce intermediate compression for the whole cluster, set the following properties in `mapred-site.xml`:

- For MRv1:

```
<property>
  <name>mapred.compress.map.output</name>
  <value>true</value>
</property>
<property>
  <name>mapred.map.output.compression.codec</name>
  <value>org.apache.hadoop.io.compress.SnappyCodec</value>
</property>
```

- For YARN:

```
<property>
  <name>mapreduce.map.output.compress</name>
  <value>true</value>
</property>
<property>
  <name>mapred.map.output.compress.codec</name>
  <value>org.apache.hadoop.io.compress.SnappyCodec</value>
</property>
```

You can also set these properties on a per-job basis.

Use the properties in the following table to compress the final output of a MapReduce job. These are usually set on a per-job basis.

MRv1 Property	YARN Property	Description
<code>mapred.output.compress</code>	<code>mapreduce.output.fileoutputformat.compress</code>	Whether to compress the final job outputs (<i>true</i> or <i>false</i>)
<code>mapred.output.compression.codec</code>	<code>mapreduce.output.fileoutputformat.compress.codec</code>	If the final job outputs are to be compressed, which codec should be used. Set to <code>org.apache.hadoop.io.compress.SnappyCodec</code> for Snappy compression.
<code>mapred.output.compression.type</code>	<code>mapreduce.output.fileoutputformat.compress.type</code>	For SequenceFile outputs, what type of compression should be used (<i>NONE</i> , <i>RECORD</i> , or <i>BLOCK</i>). <i>BLOCK</i> is recommended.

**Note:**

The MRv1 property names are also supported (though deprecated) in MRv2 (YARN), so it's not mandatory to update them in this release.

Using Snappy for Pig Compression

Set the same properties for Pig as for MapReduce (see the table in the previous section).

Using Snappy for Hive Compression

To enable Snappy compression for Hive output when creating `SequenceFile` outputs, use the following settings:

```
SET hive.exec.compress.output=true;
SET mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
SET mapred.output.compression.type=BLOCK;
```

Using Snappy Compression in Sqoop 1 and Sqoop 2 Imports

- **For Sqoop 1:**

On the command line, use the following option to enable Snappy compression:

```
--compression-codec org.apache.hadoop.io.compress.SnappyCodec
```

It is a good idea to use the `--as-sequencefile` option with this compression option.

- **For Sqoop 2:**

When you create a job (`sqoop:000> create job`), choose 7 (*SNAPPY*) as the compression format.

Using Snappy Compression with HBase

If you install Hadoop and HBase from RPM or Debian packages, Snappy requires no HBase configuration.

Viewing the Snappy Documentation

For more information about Snappy, see <http://code.google.com/p/snappy/>.

Spark Installation

Spark is a fast, general engine for large-scale data processing.

The following sections describe how to install and configure Spark.

- [Spark Packaging](#) on page 365

- [Spark Prerequisites](#) on page 365
- [Installing and Upgrading Spark](#) on page 365
- [Configuring and Running Spark \(Standalone Mode\)](#) on page 366

See also the [Apache Spark Documentation](#).

Spark Packaging

The packaging options for installing Spark are:

- RPM packages
- Debian packages

There are five Spark packages:

- `spark-core`: delivers core functionality of spark
- `spark-worker`: init scripts for `spark-worker`
- `spark-master`: init scripts for `spark-master`
- `spark-python`: Python client for Spark
- `spark-history-server`

Spark Prerequisites

- An [operating system supported by CDH 5](#)
- [Oracle JDK](#)
- The `hadoop-client` package (see [Installing the Latest CDH 5 Release](#) on page 155)

Installing and Upgrading Spark



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To see which version of Spark is shipping in the current release, check the [CDH Version and Packaging Information](#). For important information, see the [CDH 5 Release Notes](#), in particular:

- [New Features in CDH 5](#)
- [Apache Spark Incompatible Changes](#)
- [Apache Spark Known Issues](#)

To install or upgrade the Spark packages on a RHEL-compatible system:

```
$ sudo yum install spark-core spark-master spark-worker spark-history-server spark-python
```

To install or upgrade the Spark packages on a SLES system:

```
$ sudo zypper install spark-core spark-master spark-worker spark-history-server spark-python
```

To install or upgrade the Spark packages on an Ubuntu or Debian system:

```
$ sudo apt-get install spark-core spark-master spark-worker spark-history-server spark-python
```

You are now ready to configure Spark. See the [next section](#).

**Note:**

If you uploaded the Spark JAR file as described under [Optimizing YARN Mode](#) on page 370, use the same instructions to upload the new version of the file each time you to a new minor release of CDH (for example, any CDH 5.2.x release, including 5.2.0).

Configuring and Running Spark (Standalone Mode)

Configuring Spark

Before you can run Spark in standalone mode, you must do the following on every host in the cluster:

- Edit the following portion of `/etc/spark/conf/spark-env.sh` to point to the host where the Spark Master runs:

```
###
### === IMPORTANT ===
### Change the following to specify a real cluster's Master host
###
export STANDALONE_SPARK_MASTER_HOST=`hostname`
```

Change `'hostname'` in the last line to the actual hostname of the host where the Spark Master will run.

You can change other elements of the default configuration by modifying `/etc/spark/conf/spark-env.sh`. You can change the following:

- `SPARK_MASTER_PORT` / `SPARK_MASTER_WEBUI_PORT`, to use non-default ports
- `SPARK_WORKER_CORES`, to set the number of cores to use on this machine
- `SPARK_WORKER_MEMORY`, to set how much memory to use (for example 1000MB, 2GB)
- `SPARK_WORKER_PORT` / `SPARK_WORKER_WEBUI_PORT`
- `SPARK_WORKER_INSTANCE`, to set the number of worker processes per node
- `SPARK_WORKER_DIR`, to set the working directory of worker processes

Configuring the Spark History Server

Before you can run the Spark History Server, you must create the `/user/spark/applicationHistory/` directory in HDFS and set ownership and permissions as follows:

```
$ sudo -u hdfs hadoop fs -mkdir /user/spark
$ sudo -u hdfs hadoop fs -mkdir /user/spark/applicationHistory
$ sudo -u hdfs hadoop fs -chown -R spark:spark /user/spark
$ sudo -u hdfs hadoop fs -chmod 1777 /user/spark/applicationHistory
```

On Spark clients (systems from which you intend to launch Spark jobs), do the following:

1. Create `/etc/spark/conf/spark-defaults.conf` on the Spark client:

```
cp /etc/spark/conf/spark-defaults.conf.template /etc/spark/conf/spark-defaults.conf
```

2. Add the following to `/etc/spark/conf/spark-defaults.conf`:

```
spark.eventLog.dir=/user/spark/applicationHistory
spark.eventLog.enabled=true
```

This causes Spark applications running on this client to write their history to the directory that the history server reads.

In addition, if you want the YARN ResourceManager to link directly to the Spark History Server, you can set the `spark.yarn.historyServer.address` property in `/etc/spark/conf/spark-defaults.conf`:

```
spark.yarn.historyServer.address=http://HISTORY_HOST:HISTORY_PORT
```

Starting, Stopping, and Running Spark

- To start Spark, proceed as follows:
 - On one node in the cluster, start the Spark Master:

```
$ sudo service spark-master start
```

- On all the other nodes, start the workers:

```
$ sudo service spark-worker start
```

- On one node, start the History Server:

```
$ sudo service spark-history-server start
```

- To stop Spark, use the following commands on the appropriate hosts:

```
$ sudo service spark-worker stop
$ sudo service spark-master stop
$ sudo service spark-history-server stop
```

Service logs are stored in `/var/log/spark`.

You can use the GUI for the Spark Master at `<master_host>:18080`.

Testing the Spark Service

To test the Spark service, start `spark-shell` on one of the nodes. You can, for example, run a word count application:

```
val file = sc.textFile("hdfs://namenode:8020/path/to/input")
val counts = file.flatMap(line => line.split(" "))
                  .map(word => (word, 1))
                  .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://namenode:8020/output")
```

You can see the application by going to the Spark Master UI, by default at `http://spark-master:18080`, to see the Spark Shell application, its executors and logs.

Running Spark Applications

For details on running Spark applications in the YARN Client and Cluster modes, see [Running Spark Applications](#) on page 368.

Enabling Fault-Tolerant Processing in Spark Streaming

If the driver node for a Spark Streaming application fails, it can lose data that has been received, but not yet processed. To ensure that no data is lost, Spark can write out incoming data to HDFS as it is received and use this data to recover state in the event of a failure. This feature, called *Spark Streaming recovery*, is introduced in CDH 5.3 as a *Beta* feature.

Spark Streaming recovery is not supported for production use in CDH 5.3.

1. To enable Spark Streaming recovery, set the `spark.streaming.receiver.writeAheadLog.enable` parameter to `true` in the `SparkConf` object used to instantiate the `StreamingContext`

```
sparkConf.set("spark.streaming.receiver.writeAheadLog.enable", "true")
```
2. Next, create a `StreamingContext` instance using this `SparkConf`, and specify a checkpoint directory.
3. Finally use the `getOrCreate` method in `StreamingContext` to either create a new context or recover from an old context from the checkpoint directory. The following example shows steps 2 and 3 of this procedure.

```
// Function to create and setup a new StreamingContext
def functionToCreateContext(): StreamingContext = {
  val conf = new SparkConf()
```

```

    sparkConf.set( "spark.streaming.receiver.writeAheadLog.enable", "true ")
    val ssc = new StreamingContext(sparkConf,...) // new context
    val kafkaStream = KafkaUtils.createStream(...)
    // Do some transformations on the stream...and write it out etc.
    ssc.checkpoint(checkpointDirectory) // set checkpoint directory
    ssc
  }

  // Get StreamingContext from checkpoint data or create a new one
  val context = StreamingContext.getOrCreate(checkpointDirectory, functionToCreateContext
  _)

```

To prevent data loss if a receiver fails, the receivers used must be able to replay data from the original data sources if required. The Kafka receiver will automatically replay if the `spark.streaming.receiver.writeAheadLog.enable` parameter is set to `true`. Both the Flume receivers that come packaged with Spark also replay the data automatically on receiver failure.

Running Spark Applications

Spark applications are similar to MapReduce “jobs.” Each application is a self-contained computation which runs some user-supplied code to compute a result. As with MapReduce jobs, Spark applications can make use of the resources of multiple nodes. Spark revolves around the concept of a *resilient distributed dataset* (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel. There are currently two types of RDDs: *parallelized collections*, which take an existing Scala collection and run functions on it in parallel, and *Hadoop datasets*, which run functions on each record of a file in Hadoop distributed file system or any other storage system supported by Hadoop. Both types of RDDs can be operated on through the same methods.

Each application has a driver process which coordinates its execution. This process can run in the foreground (**client mode**) or in the background (**cluster mode**). Client mode is a little simpler, but cluster mode allows you to easily log out after starting a Spark application without terminating the application. For the client mode, the input file path must point to a local file.

Spark starts **executors** to perform computations. There may be many executors, distributed across the cluster, depending on the size of the job. After loading some of the executors, Spark attempts to match tasks to executors.

CDH 5.3 introduces a performance optimization (via [SPARK-1767](#)), which causes Spark to prefer RDDs which are already cached locally in HDFS. This is important enough that Spark will wait for the executors near these caches to be free for a short time.

Note that Spark does not start executors on nodes with cached data, and there is no further chance to select them during the task-matching phase. This is not a problem for most workloads, since most workloads start executors on most or all nodes in the cluster. However, if you do have problems with the optimization, an alternate API, the constructor `DeveloperApi`, is provided for writing a Spark application, which explicitly spells out the preferred locations to start executors. See the following example, as well as

`examples/src/main/scala/org/apache/spark/examples/SparkHdfsLR.scala`, for a working example of using this API.

```

...
val sparkConf = new SparkConf().setAppName( "SparkHdfsLR" )
val inputPath = args(0)
val conf = SparkHadoopUtil.get.newConfiguration()
val sc = new SparkContext(sparkConf,
    InputFormatInfo.computePreferredLocations(
        Seq( new InputFormatInfo(conf, classOf[org.apache.hadoop.mapred.TextInputFormat] ),
        inputPath))
...

```

```

/**
 * :: DeveloperApi ::
 * Alternative constructor for setting preferred locations where Spark will create
 * executors.
 *
 * @param preferredNodeLocationData used in YARN mode to select nodes to launch containers
 * on.

```



```

* Can be generated using
[[org.apache.spark.scheduler.InputFormatInfo.computePreferredLocations]]
* from a list of input files or InputFormats for the application.
*/
@DeveloperApi
def this(config: SparkConf, preferredNodeLocationData: Map[ String, Set[SplitInfo]])
= {
  this(config)
  this.preferredNodeLocationData = preferredNodeLocationData
}

```

Spark can run in two modes:

- **Standalone mode:**

In standalone mode, Spark uses a Master daemon which coordinates the efforts of the Workers, which run the executors. Standalone mode is the default, but it cannot be used on secure clusters.

- **YARN mode:**

In YARN mode, the YARN ResourceManager performs the functions of the Spark Master. The functions of the Workers are performed by the YARN NodeManager daemons, which run the executors. YARN mode is slightly more complex to set up, but it supports security, and provides better integration with YARN's cluster-wide resource management policies.

Multiple Spark applications can run at once. If you decide to run Spark on YARN, you can decide on an application-by-application basis whether to run in YARN client mode or cluster mode. When you run Spark in client mode, the driver process runs locally; in cluster mode, it runs remotely on an ApplicationMaster.



Note:

Some applications that have nested definitions and are run in the Spark shell may encounter a `Task not serializable` exception, because of a limitation in the way Scala compiles code. Cloudera recommends running such applications in a Spark job

The following sections use a sample application, SparkPi, which is packaged with Spark and computes the value of Pi, to illustrate the three modes.

Configuration

The easiest way to configure Spark is by setting `$SPARK_HOME/conf/spark-defaults.conf`.

This file contains lines in the form: "key value". You can create a comment by putting a hash mark (#) at the beginning of a line.



Note: You cannot add comments to the end or middle of a line.

Here is an example of a `spark-defaults.conf` file:

```

spark.master      spark://mysparkmaster.cloudera.com:7077
spark.eventLog.enabled  true
spark.eventLog.dir    hdfs:///user/spark/eventlog
# Set spark executor memory
spark.executor.memory  2g
spark.logConf        true

```

It is a good idea to put configuration keys that you want to use for every application into `spark-defaults.conf`. See [Script](#) for more information about configuration keys.

The Spark-Submit Script

You can start Spark applications with the `spark-submit` script, which is installed in your path when you install the `spark-core` package.



Note: Spark cannot handle command line options of the form `--key=value`; use `--key value` instead. (That is, use a space instead of an equals sign.)

To run `spark-submit`, you need a compiled Spark application JAR. The following sections use a sample JAR, `SparkPi`, which is packaged with Spark. It computes an approximation to the value of Pi.

Running SparkPi in Standalone Mode

Supply the `--master` and `--deploy-mode` client arguments to run `SparkPi` in standalone mode:

```
spark-submit \  
--class org.apache.spark.examples.SparkPi \  
--deploy-mode client \  
--master spark://${SPARK_MASTER_IP}:${SPARK_MASTER_PORT} \  
${SPARK_HOME}/examples/lib/spark-examples_version.jar 10
```

where *version* is, for example, `2.10-1.1.0-cdh5.2.0`.

Arguments that come after the JAR name are supplied to the application. In this case, the argument controls how good we want our approximation to Pi to be.

Running SparkPi in YARN Client Mode

In this case, the command to run `SparkPi` is as follows:

```
spark-submit \  
--class org.apache.spark.examples.SparkPi \  
--deploy-mode client \  
--master yarn \  
${SPARK_HOME}/examples/lib/spark-examples_version.jar 10
```

where *version* is, for example, `2.10-1.1.0-cdh5.2.0`.

Running SparkPi in YARN Cluster Mode

In this case, the command to run `SparkPi` is as follows:

```
spark-submit \  
--class org.apache.spark.examples.SparkPi \  
--deploy-mode cluster \  
--master yarn \  
${SPARK_HOME}/examples/lib/spark-examples_version.jar 10
```

where *version* is, for example, `2.10-1.1.0-cdh5.2.0`.

The command will continue to print out status until the job finishes, or you press `control-C`. Terminating the `spark-submit` process in cluster mode does not terminate the Spark application as it does in client mode. To monitor the status of the running application, run `yarn application -list`.

Optimizing YARN Mode

Normally, Spark copies the Spark assembly JAR file to HDFS each time you run `spark-submit`, as you can see in the following sample log messages:

```
14/06/11 14:21:49 INFO yarn.Client: Uploading  
file:/home/jdoe/spark/b2.4/examples/target/scala-2.10/spark-examples-1.0.0-SNAPSHOT-hadoop2.4.0.jar  
to  
hdfs://spark-02.example.com:6000/user/jdoe/.sparkStaging/application_1402278226964_0012/spark-examples-1.0.0-SNAPSHOT-hadoop2.4.0.jar  
14/06/11 14:21:50 INFO yarn.Client: Uploading  
file:/home/jdoe/spark/b2.4/assembly/target/scala-2.10/spark-assembly-1.0.0-SNAPSHOT-hadoop2.4.0.jar
```

```
to
hdfs://spark-02.example.com:6000/user/joe/.sparkStaging/application_1402278226964_0012/spark-assembly-1.0.0-SNAPSHOT-hadoop2.4.0.jar
```

You can avoid doing this copy each time by manually uploading the Spark assembly JAR file to your HDFS. Then set the `SPARK_JAR` environment variable to this HDFS path:

```
hdfs dfs -mkdir -p /user/spark/share/lib
hdfs dfs -put $SPARK_HOME/assembly/lib/spark-assembly_*.jar \
/user/spark/share/lib/spark-assembly.jar
SPARK_JAR=hdfs://<nn>:<port>/user/spark/share/lib/spark-assembly.jar
```



Note:

Do this manual upload again each time you upgrade Spark to a new to a new minor CDH release (for example, any CDH 5.2.x release, including 5.2.0).

If you are using Cloudera Manager, the Spark assembly JAR is uploaded to HDFS automatically on initial installation, as `/user/spark/share/lib/spark-assembly.jar`, but you need to upload the new version when you upgrade. See the instructions for upgrading Spark using Cloudera Manager under [Upgrading to CDH 5.1](#).

Building Spark Applications

Best practices when compiling your Spark applications include:

- Building a single assembly JAR that includes all the dependencies, except those for Spark and Hadoop.
- Excluding any Spark and Hadoop classes from the assembly JAR, because they are already on the cluster, and part of the runtime classpath. In Maven, you can mark the Spark and Hadoop dependencies as `provided`.
- Always building against the same version of Spark that you are running against, to avoid compatibility issues.

For example, do not assume that applications compiled against Spark 0.9 will run on Spark 1.0 without recompiling. In addition, some applications compiled under Spark 0.9 or earlier will need changes to their source code to compile under Spark 1.0. Applications that compile under Spark 1.0 should compile under all future versions.

Using Spark with HBase

A common use case is to use Spark to process data which is destined for HBase, or which has been extracted from HBase. See [Importing Data Into HBase](#).

Running a Crunch Application with Spark

The blog post [How-to: Run a Simple Apache Spark App in CDH 5](#) provides a tutorial on writing, compiling and running a Spark application. Taking that article as a starting point, do the following to run Crunch with Spark.

1. Add both the `crunch-core` and `crunch-spark` dependencies to your Maven project, along with the other dependencies shown in the blog post.
2. Use the `SparkPipeline` (`org.apache.crunch.impl.spark.SparkPipeline`) where you would have used the `MRPipeline` instance in the declaration of your Crunch pipeline. The `SparkPipeline` will need either a `String` that contains the connection string for the Spark master (`local` for local mode, `yarn-client` for YARN) or an actual `JavaSparkContext` instance.
3. Update the `SPARK_SUBMIT_CLASSPATH`:

```
export
SPARK_SUBMIT_CLASSPATH=./commons-codec-1.4.jar:$SPARK_HOME/assembly/lib/*:./myapp-jar-with-dependencies.jar
```



Important:

The `commons-codec-1.4` dependency must come before the `SPARK_HOME` dependencies.

4. Now you can start the pipeline using your Crunch app *jar-with-dependencies* file using the `spark-submit` script, just as you would for a regular Spark pipeline.

Sqoop 1 Installation

Apache Sqoop 1 is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. You can use Sqoop 1 to import data from external structured datastores into the Hadoop Distributed File System (HDFS) or related systems such as Hive and HBase. Conversely, you can use Sqoop 1 to extract data from Hadoop and export it to external structured datastores such as relational databases and enterprise data warehouses.

**Note:**

To see which version of Sqoop 1 is shipping in CDH 5, check the [CDH Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

See the following sections for information and instructions:

- [Upgrading Sqoop 1 from an Earlier CDH 5 release](#) on page 372
- [Packaging](#)
- [Prerequisites](#)
- [Installing Packages](#)
- [Installing a Tarball](#)
- [Installing the JDBC Drivers](#)
- [Setting HADOOP_MAPRED_HOME](#) on page 375
- [Apache Sqoop 1 Documentation](#)

Upgrading Sqoop 1 from an Earlier CDH 5 release

These instructions assume that you are upgrading Sqoop 1 as part of an upgrade to the latest CDH 5 release, and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

To upgrade Sqoop 1 from an earlier CDH 5 release, install the new version of Sqoop 1 using one of the methods described below: [Installing the Sqoop 1 RPM or Debian Packages](#) on page 373 or [Installing the Sqoop 1 Tarball](#) on page 373

**Important: Configuration files**

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Sqoop 1 Packaging

The packaging options for installing Sqoop 1 are:

- RPM packages
- Tarball
- Debian packages

Sqoop 1 Prerequisites

- An [operating system supported by CDH 5](#)
- [Oracle JDK](#)

- Services which you wish to use with Sqoop, such as HBase, Hive HCatalog, and Accumulo. Sqoop checks for these services when you run it, and finds services which are installed and configured. It logs warnings for services it does not find. These warnings, shown below, are harmless.

```
> Warning: /usr/lib/sqoop/./hbase does not exist! HBase imports will fail.
> Please set $HBASE_HOME to the root of your HBase installation.
> Warning: /usr/lib/sqoop/./hive-hcatalog does not exist! HCatalog jobs will fail.
> Please set $HCAT_HOME to the root of your HCatalog installation.
> Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
> Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

Installing the Sqoop 1 RPM or Debian Packages

Installing the Sqoop 1 RPM or Debian packages is more convenient than installing the Sqoop 1 tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations

The Sqoop 1 packages consist of:

- `sqoop` — Complete Sqoop 1 distribution
- `sqoop-metastore` — For installation of the Sqoop 1 metastore only



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install Sqoop 1 on a RHEL-compatible system:

```
$ sudo yum install sqoop
```

To install Sqoop 1 on an Ubuntu or other Debian system:

```
$ sudo apt-get install sqoop
```

To install Sqoop 1 on a SLES system:

```
$ sudo zypper install sqoop
```

If you have already configured CDH on your system, there is no further configuration necessary for Sqoop 1. You can start using Sqoop 1 by using commands such as:

```
$ sqoop help
$ sqoop version
$ sqoop import
```

Installing the Sqoop 1 Tarball

The Sqoop 1 tarball is a self-contained package containing everything necessary to use Sqoop 1 with YARN on a Unix-like system.



Important:

Make sure you have read and understood the section on tarballs under [How Packaging Affects CDH 5 Deployment](#) on page 156 before you proceed with a tarball installation.

To install Sqoop 1 from the tarball, unpack the tarball in a convenient location. Once it is unpacked, add the `bin` directory to the shell path for easy access to Sqoop 1 commands. Documentation for users and developers can be found in the `docs` directory.

To install the Sqoop 1 tarball on Linux-based systems:

Run the following command:

```
$ (cd /usr/local/ && sudo tar -zxvf _<path_to_sqoop.tar.gz>_)
```



Note:

When installing Sqoop 1 from the tarball package, you must make sure that the environment variables `JAVA_HOME` and `HADOOP_MAPRED_HOME` are configured correctly. The variable `HADOOP_MAPRED_HOME` should point to the root directory of Hadoop installation. Optionally, if you intend to use any Hive or HBase related functionality, you must also make sure that they are installed and the variables `HIVE_HOME` and `HBASE_HOME` are configured correctly to point to the root directory of their respective installation.

Installing the JDBC Drivers for Sqoop 1

Sqoop 1 does not ship with third party JDBC drivers. You must download them separately and save them to the `/var/lib/sqoop/` directory on the server. The following sections show how to install the most common JDBC Drivers.



Note:

The JDBC drivers need to be installed only on the machine where Sqoop is executed; you do not need to install them on all nodes in your Hadoop cluster.

Before you begin:

Make sure the `/var/lib/sqoop` directory exists and has the correct ownership and permissions:

```
mkdir -p /var/lib/sqoop
chown sqoop:sqoop /var/lib/sqoop
chmod 755 /var/lib/sqoop
```

This sets permissions to `drwxr-xr-x`.

For JDBC drivers for Hive, Impala, Teradata, or Netezza, see the [Connectors documentation](#).

Installing the MySQL JDBC Driver

Download the MySQL JDBC driver from <http://www.mysql.com/downloads/connector/j/5.1.html>. You will need to sign up for an account if you do not already have one, and log in, before you can download it. Then copy it to the `/var/lib/sqoop/` directory. For example:

```
$ sudo cp mysql-connector-java-version/mysql-connector-java-version-bin.jar
/var/lib/sqoop/
```

**Note:**

At the time of publication, *version* was 5.1.31, but the version may have changed by the time you read this.

**Important:**

Make sure you have at least version 5.1.31. Some systems ship with an earlier version that may not work correctly with Sqoop.

Installing the Oracle JDBC Driver

You can download the JDBC Driver from the Oracle website, for example <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>. You must accept the license agreement before you can download the driver. Download the `ojdbc6.jar` file and copy it to the `/var/lib/sqoop/` directory:

```
$ sudo cp ojdbc6.jar /var/lib/sqoop/
```

Installing the Microsoft SQL Server JDBC Driver

Download the Microsoft SQL Server JDBC driver from <http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774> and copy it to the `/var/lib/sqoop/` directory. For example:

```
$ curl -L 'http://download.microsoft.com/download/0/2/A/02AAE597-3865-456C-AE7F-613F99F850A8/sqljdbc_4.0.2206.100_enu.tar.gz' | tar xz
$ sudo cp sqljdbc_4.0/enu/sqljdbc4.jar /var/lib/sqoop/
```

Installing the PostgreSQL JDBC Driver

Download the PostgreSQL JDBC driver from <http://jdbc.postgresql.org/download.html> and copy it to the `/var/lib/sqoop/` directory. For example:

```
$ curl -L 'http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar' -o postgresql-9.2-1002.jdbc4.jar
$ sudo cp postgresql-9.2-1002.jdbc4.jar /var/lib/sqoop/
```

Setting HADOOP_MAPRED_HOME

- For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop 1 in a YARN installation, make sure that the `HADOOP_MAPRED_HOME` environment variable is set correctly, as follows:

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

- For each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop 1 in an MRv1 installation, set the `HADOOP_MAPRED_HOME` environment variable as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

Viewing the Sqoop 1 Documentation

For additional documentation see the [Sqoop 1 User Guide](#) and the [Sqoop 1 Developer Guide](#).

Sqoop 2 Installation



Note: Sqoop 2 is being deprecated. Cloudera recommends using Sqoop 1.

Sqoop 2 is a server-based tool designed to transfer data between Hadoop and relational databases. You can use Sqoop 2 to import data from a relational database management system (RDBMS) such as MySQL or Oracle into the Hadoop Distributed File System (HDFS), transform the data with Hadoop MapReduce, and then export it back into an RDBMS.

Sqoop 2 Packaging

There are three packaging options for installing Sqoop 2:

- Tarball (.tgz) that contains both the Sqoop 2 server and the client.
- Separate RPM packages for Sqoop 2 server (`sqoop2-server`) and client (`sqoop2-client`)
- Separate Debian packages for Sqoop 2 server (`sqoop2-server`) and client (`sqoop2-client`)

Sqoop 2 Installation

- [Upgrading Sqoop 2 from an Earlier CDH 5 Release](#) on page 376
- [Installing Sqoop 2](#)
- [Configuring Sqoop 2](#)
- [Starting, Stopping and Using the Server](#)
- [Apache Documentation](#)

See also [Feature Differences - Sqoop 1 and Sqoop 2](#) on page 380.

Upgrading Sqoop 2 from an Earlier CDH 5 Release



Note: Sqoop 2 is being deprecated. Cloudera recommends using Sqoop 1.

These instructions assume that you are upgrading Sqoop 2 as part of an upgrade to the latest CDH 5 release, and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

For more detailed instructions for upgrading Sqoop 2, see the [Apache Sqoop Upgrade page](#).

To upgrade Sqoop 2 from an earlier CDH 5 release, proceed as follows:

1. Install the new version of Sqoop 2 following directions under [Installing Sqoop 2](#) on page 377.
2. *If you are running MRv1 on CDH 5 Beta 1 and will continue to run it after upgrading:*

- a. Update `/etc/defaults/sqoop2-server` to point to MR1:

```
mv /etc/defaults/sqoop2-server.rpmnew /etc/defaults/sqoop2-server
```

- b. Update alternatives:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.mr1
```

3. Run the upgrade tool:

```
sqoop2-tool upgrade
```

This upgrades the repository database to the latest version.

**Important: Configuration files**

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Installing Sqoop 2*Sqoop 2 Prerequisites*

- An [operating system supported by CDH 5](#)
- [Oracle JDK](#)
- Hadoop must be installed on the node which runs the Sqoop 2 server component.
- Services which you wish to use with Sqoop, such as HBase, Hive HCatalog, and Accumulo. Sqoop checks for these services when you run it, and finds services which are installed and configured. It logs warnings for services it does not find. These warnings, shown below, are harmless.

```
> Warning: /usr/lib/sqoop/./hbase does not exist! HBase imports will fail.
> Please set $HBASE_HOME to the root of your HBase installation.
> Warning: /usr/lib/sqoop/./hive-hcatalog does not exist! HCatalog jobs will fail.
> Please set $HCAT_HOME to the root of your HCatalog installation.
> Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
> Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

Installing Sqoop 2

Sqoop 2 is distributed as two separate packages: a client package (`sqoop2-client`) and a server package (`sqoop2-server`). Install the server package on one node in the cluster; because the Sqoop 2 server acts as a MapReduce client this node must have Hadoop installed and configured.

Install the client package on each node that will act as a client. A Sqoop 2 client will always connect to the Sqoop 2 server to perform any actions, so Hadoop does not need to be installed on the client nodes.

Depending on what you are planning to install, choose the appropriate package and install it using your preferred package manager application.



Note: The Sqoop 2 packages cannot be installed on the same machines as [Sqoop1](#) packages. However you can use both versions in the same Hadoop cluster by installing Sqoop1 and Sqoop 2 on different nodes.

To install the Sqoop 2 server package on a RHEL-compatible system:**Note: Install Cloudera Repository**

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

```
$ sudo yum install sqoop2-server
```

To install the Sqoop 2 client package on a RHEL-compatible system:

```
$ sudo yum install sqoop2-client
```

To install the Sqoop 2 server package on a SLES system:

```
$ sudo zypper install sqoop2-server
```

To install the Sqoop 2 client package on a SLES system:

```
$ sudo zypper install sqoop2-client
```

To install the Sqoop 2 server package on an Ubuntu or Debian system:

```
$ sudo apt-get install sqoop2-server
```

To install the Sqoop 2 client package on an Ubuntu or Debian system:

```
$ sudo apt-get install sqoop2-client
```



Note:

Installing the `sqoop2-server` package creates a `sqoop-server` service configured to start Sqoop 2 at system startup time.

You are now ready to configure Sqoop 2. See the [next section](#).

Configuring Sqoop 2

This section explains how to configure the Sqoop 2 server.



Note: Sqoop 2 is being deprecated. Cloudera recommends using Sqoop 1.

Configuring which Hadoop Version to Use

The Sqoop 2 client does not interact directly with Hadoop MapReduce, and so it does not require any MapReduce configuration.

The Sqoop 2 server can work with either MRv1 or YARN. **It cannot work with both simultaneously.**

You set the MapReduce version the Sqoop 2 server works with by means of the `alternatives` command (or `update-alternatives`, depending on your operating system):

- To use YARN:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.dist
```

- To use MRv1:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.mr1
```

**Important: If you are upgrading from a release earlier than CDH 5 Beta 2**

In earlier releases, the mechanism for setting the MapReduce version was the `CATALINA_BASE` variable in the `/etc/defaults/sqoop2-server` file. This does not work as of CDH 5 Beta 2, and in fact could cause problems. **Check your `/etc/defaults/sqoop2-server` file and make sure `CATALINA_BASE` is not set.**

Installing the JDBC Drivers

Sqoop 2 does not ship with third party JDBC drivers. You must download them separately and save them to the `/var/lib/sqoop2/` directory on the server. The following sections show how to install the most common JDBC drivers. Once you have installed the JDBC drivers, restart the Sqoop 2 server so that the drivers are loaded.

**Note:**

The JDBC drivers need to be installed only on the machine where Sqoop is executed; you do not need to install them on all nodes in your Hadoop cluster.

Installing the MySQL JDBC Driver

Download the MySQL JDBC driver [here](#). You will need to sign up for an account if you don't already have one, and log in, before you can download it. Then copy it to the `/var/lib/sqoop2/` directory. For example:

```
$ sudo cp mysql-connector-java-version/mysql-connector-java-version-bin.jar
/var/lib/sqoop2/
```

At the time of publication, `version` was `5.1.31`, but the version may have changed by the time you read this.

**Important:**

Make sure you have at least version `5.1.31`. Some systems ship with an earlier version that may not work correctly with Sqoop.

Installing the Oracle JDBC Driver

You can download the JDBC Driver from the Oracle website, for example [here](#). You must accept the license agreement before you can download the driver. Download the `ojdbc6.jar` file and copy it to `/var/lib/sqoop2/` directory:

```
$ sudo cp ojdbc6.jar /var/lib/sqoop2/
```

Installing the Microsoft SQL Server JDBC Driver

Download the Microsoft SQL Server JDBC driver [here](#) and copy it to the `/var/lib/sqoop2/` directory. For example:

```
$ curl -L 'http://download.microsoft.com/download/0/2/A/02AAE597-3865-456C-AE7F-613F99F850A8/sqljdbc_4.0.2206.100_enu.tar.gz'
| tar xz
$ sudo cp sqljdbc_4.0/enu/sqljdbc4.jar /var/lib/sqoop2/
```

Installing the PostgreSQL JDBC Driver

Download the PostgreSQL JDBC driver [here](#) and copy it to the `/var/lib/sqoop2/` directory. For example:

```
$ curl -L 'http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar' -o
postgresql-9.2-1002.jdbc4.jar
$ sudo cp postgresql-9.2-1002.jdbc4.jar /var/lib/sqoop2/
```

Starting, Stopping, and Accessing the Sqoop 2 Server

Starting the Sqoop 2 Server

After you have completed all of the required configuration steps, you can start Sqoop 2 server:

```
$ sudo /sbin/service sqoop2-server start
```

Stopping the Sqoop 2 Server

```
$ sudo /sbin/service sqoop2-server stop
```

Checking that the Sqoop 2 Server has Started

You can verify whether the server has started correctly by connecting to its HTTP interface. The simplest way is to get the server version using following command:

```
$ wget -qO - localhost:12000/sqoop/version
```

You should get a text fragment in JSON format similar to the following:

```
{"version": "1.99.2-cdh5.0.0", ...}
```

Accessing the Sqoop 2 Server with the Sqoop 2 Client

Start the Sqoop 2 client:

```
sqoop2
```

Identify the host where your server is running (we will use `localhost` in this example):

```
sqoop:000> set server --host localhost
```

Test the connection by running the command `show version --all` to obtain the version number from server. You should see output similar to the following:

```
sqoop:000> show version --all
server version:
  Sqoop 1.99.2-cdh5.0.0 revision ...
  Compiled by jenkins on ...
client version:
  Sqoop 1.99.2-cdh5.0.0 revision ...
  Compiled by jenkins on ...
Protocol version:
  [1]
```

Viewing the Sqoop 2 Documentation

For more information about Sqoop 2, see [Highlights of Sqoop 2](#) and <https://archive.cloudera.com/cdh5/cdh/5/sqoop2>.

Feature Differences - Sqoop 1 and Sqoop 2



Note: Sqoop 2 is being deprecated. Customers are advised to use Sqoop 1 instead.

Feature	Sqoop 1	Sqoop 2
Connectors for all major RDBMS	Supported.	Not supported. Workaround: Use the generic JDBC Connector which has been tested on the following databases: Microsoft SQL Server, PostgreSQL, MySQL and Oracle.

Feature	Sqoop 1	Sqoop 2
		This connector should work on any other JDBC compliant database. However, performance might not be comparable to that of specialized connectors in Sqoop.
Kerberos Security Integration	Supported.	Not supported.
Data transfer from RDBMS to Hive or HBase	Supported.	Not supported. Workaround: Follow this two-step approach. <ol style="list-style-type: none"> 1. Import data from RDBMS into HDFS 2. Load data into Hive or HBase manually using appropriate tools and commands such as the <code>LOAD DATA</code> statement in Hive
Data transfer from Hive or HBase to RDBMS	Not supported. Workaround: Follow this two-step approach. <ol style="list-style-type: none"> 1. Extract data from Hive or HBase into HDFS (either as a text or Avro file) 2. Use Sqoop to export output of previous step to RDBMS 	Not supported. Follow the same workaround as for Sqoop 1.

Whirr Installation

Apache Whirr is a set of libraries for running cloud services. You can use Whirr to run CDH 5 clusters on cloud providers' clusters, such as Amazon Elastic Compute Cloud (Amazon EC2). There's no need to install the RPMs for CDH 5 or do any configuration; a working cluster will start immediately with one command. It's ideal for running temporary Hadoop clusters to carry out a proof of concept, or to run a few one-time jobs. When you are finished, you can destroy the cluster and all of its data with one command.

Use the following sections to install, upgrade, and deploy Whirr:

- [Upgrading Whirr](#)
- [Installing Whirr](#)
- [Generating an SSH Key Pair](#)
- [Defining a Cluster](#)
- [Launching a Cluster](#)
- [Apache Whirr Documentation](#)

Upgrading Whirr



Note:

To see which version of Whirr is shipping in CDH 5, check the [Version and Packaging Information](#). For important information on new and changed components, see the [CDH 5 Release Notes](#).

Upgrading Whirr from an Earlier CDH 5 Release to the Latest CDH 5 Release

Step 1: Stop the Whirr proxy.

Kill the `hadoop-proxy.sh` process by pressing Control-C.

Installing Cloudera Manager and CDH

Step 2: Destroy the Cluster.

Whirr clusters are normally short-lived. If you have a running cluster, destroy it: see [Destroying a cluster](#) on page 384.

Step 3: Install the New Version of Whirr

See [Installing Whirr](#) on page 382.

The upgrade is now complete. For more information, see [Managing a Cluster with Whirr](#) on page 383, and [Viewing the Whirr Documentation](#) on page 385.

Installing Whirr



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

To install Whirr on an Ubuntu or other Debian system:

```
$ sudo apt-get install whirr
```

To install Whirr on a RHEL-compatible system:

```
$ sudo yum install whirr
```

To install Whirr on a SLES system:

```
$ sudo zypper install whirr
```

To install Whirr on another system: Download a Whirr tarball from [here](#).

To verify Whirr is properly installed:

```
$ whirr version
```

Generating an SSH Key Pair for Whirr

After installing Whirr, generate a password-less SSH key pair to enable secure communication with the Whirr cluster.

```
ssh-keygen -t rsa -P ''
```



Note:

If you specify a non-standard location for the key files in the `ssh-keygen` command (that is, not `~/.ssh/id_rsa`), then you must specify the location of the private key file in the `whirr.private-key-file` property and the public key file in the `whirr.public-key-file` property. For more information, see the next section.

Defining a Whirr Cluster



Note:

For information on finding your cloud credentials, see the [Whirr FAQ](#).

After generating an SSH key pair, the only task left to do before using Whirr is to define a cluster by creating a properties file. You can name the properties file whatever you like. The example properties file used in these instructions is named `hadoop.properties`. Save the properties file in your home directory. After defining a cluster in the properties file, you will be ready to launch a cluster and run MapReduce jobs.



Important:

The properties shown below are sufficient to get a bare-bones cluster up and running, but you will probably need to do more configuration to do real-life tasks, especially if you are using HBase and ZooKeeper. You can find more comprehensive template files in the `recipes` directory, for example `recipes/hbase-cdh.properties`.

MRv1 Cluster

The following file defines a cluster with a single machine for the NameNode and JobTracker, and another machine for a DataNode and TaskTracker.

```
whirr.cluster-name=myhadoopcluster
whirr.instance-templates=1 hadoop-jobtracker+hadoop-namenode,1
hadoop-datanode+hadoop-tasktracker
whirr.provider=aws-ec2
whirr.identity=<cloud-provider-identity>
whirr.credential=<cloud-provider-credential>
whirr.private-key-file=${sys:user.home}/.ssh/id_rsa
whirr.public-key-file=${sys:user.home}/.ssh/id_rsa.pub
whirr.env.repo=cdh5
whirr.hadoop-install-function=install_cdh_hadoop
whirr.hadoop-configure-function=configure_cdh_hadoop
whirr.hardware-id=m1.large
whirr.image-id=us-east-1/ami-ccb35ea5
whirr.location-id=us-east-1
```

YARN Cluster

The following configuration provides the essentials for a YARN cluster. Change the number of instances for `hadoop-datanode+yarn-nodemanager` from 2 to a larger number if you need to.

```
whirr.cluster-name=myhadoopcluster
whirr.instance-templates=1 hadoop-namenode+yarn-resource-manager+mapreduce-historyserver,2
hadoop-datanode+yarn-nodemanager
whirr.provider=aws-ec2
whirr.identity=<cloud-provider-identity>
whirr.credential=<cloud-provider-credential>
whirr.private-key-file=${sys:user.home}/.ssh/id_rsa
whirr.public-key-file=${sys:user.home}/.ssh/id_rsa.pub
whirr.env.mapreduce_version=2
whirr.env.repo=cdh5
whirr.hadoop.install-function=install_cdh_hadoop
whirr.hadoop.configure-function=configure_cdh_hadoop
whirr.mr_jobhistory.start-function=start_cdh_mr_jobhistory
whirr.yarn.configure-function=configure_cdh_yarn
whirr.yarn.start-function=start_cdh_yarn
whirr.hardware-id=m1.large
whirr.image-id=us-east-1/ami-ccb35ea5
whirr.location-id=us-east-1
```

Managing a Cluster with Whirr

To launch a cluster:

```
$ whirr launch-cluster --config hadoop.properties
```

As the cluster starts up, messages are displayed in the console. You can see debug-level log messages in a file named `whirr.log` in the directory where you ran the `whirr` command. After the cluster has started, a message appears in the console showing the URL you can use to access the web UI for Whirr.

Running a Whirr Proxy

For security reasons, traffic from the network where your client is running is proxied through the master node of the cluster using an SSH tunnel (a SOCKS proxy on port 6666). A script to launch the proxy is created when you launch the cluster, and may be found in `~/ .whirr/<cluster-name>`.

To launch the Whirr proxy:

1. Run the following command in a new terminal window:

```
$ . ~/.whirr/myhadoopcluster/hadoop-proxy.sh
```

2. To stop the proxy, kill the process by pressing Ctrl-C.

Running a MapReduce job

After you launch a cluster, a `hadoop-site.xml` file is automatically created in the directory `~/ .whirr/<cluster-name>`. You need to update the local Hadoop configuration to use this file.

To update the local Hadoop configuration to use `hadoop-site.xml`:

1. On all systems, type the following commands:

```
$ cp -r /etc/hadoop/conf.empty /etc/hadoop/conf.whirr
$ rm -f /etc/hadoop/conf.whirr/*-site.xml
$ cp ~/.whirr/myhadoopcluster/hadoop-site.xml /etc/hadoop/conf.whirr
```

2. If you are using an Ubuntu, Debian, or SLES system, type these commands:

```
$ sudo update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.whirr
50
$ update-alternatives --display hadoop-conf
```

3. If you are using a Red Hat system, type these commands:

```
$ sudo alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.whirr 50
$ alternatives --display hadoop-conf
```

4. You can now browse HDFS:

```
$ hadoop fs -ls /
```

To run a MapReduce job, run these commands:

- For MRv1:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
$ hadoop fs -mkdir input
$ hadoop fs -put $HADOOP_MAPRED_HOME/CHANGES.txt input
$ hadoop jar $HADOOP_MAPRED_HOME/hadoop-examples.jar wordcount input output
$ hadoop fs -cat output/part-* | head
```

- For YARN:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
$ hadoop fs -mkdir input
$ hadoop fs -put $HADOOP_MAPRED_HOME/CHANGES.txt input
$ hadoop jar $HADOOP_MAPRED_HOME/hadoop-mapreduce-examples.jar wordcount input output
$ hadoop fs -cat output/part-* | head
```

Destroying a cluster

When you are finished using a cluster, you can terminate the instances and clean up the resources using the commands shown in this section.

WARNING

All data will be deleted when you destroy the cluster.

To destroy a cluster:

1. Run the following command to destroy a cluster:

```
$ whirr destroy-cluster --config hadoop.properties
```

2. Shut down the SSH proxy to the cluster if you started one earlier.

Viewing the Whirr Documentation

For additional documentation see the [Whirr Documentation](#).

ZooKeeper Installation

**Note: Running Services**

When starting, stopping and restarting CDH components, always use the `service (8)` command rather than running scripts in `/etc/init.d` 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 scripts in `/etc/init.d`, 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).)

Apache ZooKeeper is a highly reliable and available service that provides coordination between distributed processes.

**Note: For More Information**

From the Apache ZooKeeper site:

ZooKeeper is a high-performance coordination service for distributed applications. It exposes common services — such as naming, configuration management, synchronization, and group services - in a simple interface so you do not have to write them from scratch. You can use it off-the-shelf to implement consensus, group management, leader election, and presence protocols. And you can build on it for your own, specific needs.

To learn more about Apache ZooKeeper, visit <http://zookeeper.apache.org/>.

**Note:**

To see which version of ZooKeeper is shipping in CDH 5, check the [CDH Version and Packaging Information](#). For important information on new and changed components, see the [Cloudera Release Guide](#).

Use the following sections to install, upgrade and administer ZooKeeper:

- [Upgrading ZooKeeper from an Earlier CDH 5 Release](#) on page 386
- [Installing the ZooKeeper Packages](#) on page 386
- [Maintaining a ZooKeeper Server](#) on page 389
- [Viewing the ZooKeeper Documentation](#) on page 389

Upgrading ZooKeeper from an Earlier CDH 5 Release

Cloudera recommends that you use a **rolling upgrade** process to upgrade ZooKeeper: that is, upgrade one server in the ZooKeeper ensemble at a time. This means bringing down each server in turn, upgrading the software, then restarting the server. The server will automatically rejoin the quorum, update its internal state with the current ZooKeeper leader, and begin serving client sessions.

This method allows you to upgrade ZooKeeper without any interruption in the service, and also lets you monitor the ensemble as the upgrade progresses, and roll back if necessary if you run into problems.

The instructions that follow assume that you are upgrading ZooKeeper as part of a CDH 5 upgrade, and have already performed the steps under [Upgrading from an Earlier CDH 5 Release to the Latest Release](#).

Performing a ZooKeeper Rolling Upgrade

Follow these steps to perform a rolling upgrade.

Step 1: Stop the ZooKeeper Server on the First Node

To stop the ZooKeeper server:

```
$ sudo service zookeeper-server stop
```

Step 2: Install the ZooKeeper Base Package on the First Node

See [Installing the ZooKeeper Base Package](#).

Step 3: Install the ZooKeeper Server Package on the First Node

See [Installing the ZooKeeper Server Package](#).



Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version; for details, see [Automatic handling of configuration files by dpkg](#).

Step 4: Restart the Server

See [Installing the ZooKeeper Server Package](#) for instructions on starting the server.

The upgrade is now complete on this server and you can proceed to the next.

Step 5: Upgrade the Remaining Nodes

Repeat Steps 1-4 above on each of the remaining nodes.

The ZooKeeper upgrade is now complete.

Installing the ZooKeeper Packages

There are two ZooKeeper server packages:

- The `zookeeper` base package provides the basic libraries and scripts that are necessary to run ZooKeeper servers and clients. The documentation is also included in this package.

- The `zookeeper-server` package contains the `init.d` scripts necessary to run ZooKeeper as a daemon process. Because `zookeeper-server` depends on `zookeeper`, installing the server package automatically installs the base package.



Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade, install the Cloudera `yum`, `zypper`/`YaST` or `apt` repository, and install or upgrade CDH 5 and make sure it is functioning correctly. For instructions, see [Installing the Latest CDH 5 Release](#) on page 155 and [Upgrading Unmanaged CDH Using the Command Line](#).

Installing the ZooKeeper Base Package

To install ZooKeeper On Red Hat-compatible systems:

```
$ sudo yum install zookeeper
```

To install ZooKeeper on Ubuntu and other Debian systems:

```
$ sudo apt-get install zookeeper
```

To install ZooKeeper on SLES systems:

```
$ sudo zypper install zookeeper
```

Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server

The instructions provided here deploy a single ZooKeeper server in "standalone" mode. This is appropriate for evaluation, testing and development purposes, but may not provide sufficient reliability for a production application. See [Installing ZooKeeper in a Production Environment](#) on page 388 for more information.

To install the ZooKeeper Server On Red Hat-compatible systems:

```
$ sudo yum install zookeeper-server
```

To install a ZooKeeper server on Ubuntu and other Debian systems:

```
$ sudo apt-get install zookeeper-server
```

To install ZooKeeper on SLES systems:

```
$ sudo zypper install zookeeper-server
```

To create `/var/lib/zookeeper` and set permissions:

```
mkdir -p /var/lib/zookeeper
chown -R zookeeper /var/lib/zookeeper/
```

To start ZooKeeper



Note:

ZooKeeper may start automatically on installation on Ubuntu and other Debian systems. This automatic start will happen only if the data directory exists; otherwise you will be prompted to initialize as shown below.

- To start ZooKeeper after an upgrade:

```
$ sudo service zookeeper-server start
```

- To start ZooKeeper after a first-time install:

```
$ sudo service zookeeper-server init
$ sudo service zookeeper-server start
```



Note:

If you are deploying multiple ZooKeeper servers after a fresh install, you need to create a `myid` file in the data directory. You can do this by means of an `init` command option: `$ sudo service zookeeper-server init --myid=1`

Installing ZooKeeper in a Production Environment

In a production environment, you should deploy ZooKeeper as an ensemble with an odd number of servers. As long as a majority of the servers in the ensemble are available, the ZooKeeper service will be available. The minimum recommended ensemble size is three ZooKeeper servers, and Cloudera recommends that each server run on a separate machine. In addition, the ZooKeeper server process should have its own dedicated disk storage if possible.

Deploying a ZooKeeper ensemble requires some additional configuration. The configuration file (`zoo.cfg`) on each server must include a list of all servers in the ensemble, and each server must also have a `myid` file in its data directory (by default `/var/lib/zookeeper`) that identifies it as one of the servers in the ensemble. Proceed as follows *on each server*.

1. Use the commands under [Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server](#) on page 387 to install `zookeeper-server` on each host.
2. Test the expected loads to set the Java heap size so as to avoid swapping. Make sure you are well below the threshold at which the system would start swapping; for example 12GB for a machine with 16GB of RAM.
3. Create a configuration file. This file can be called anything you like, and must specify settings for at least the parameters shown under "Minimum Configuration" in the [ZooKeeper Administrator's Guide](#). You should also configure values for `initLimit`, `syncLimit`, and `server.n`; see the [explanations](#) in the administrator's guide. For example:

```
tickTime=2000
dataDir=/var/lib/zookeeper/
clientPort=2181
initLimit=5
syncLimit=2
server.1=zoo1:2888:3888
server.2=zoo2:2888:3888
server.3=zoo3:2888:3888
```

In this example, the final three lines are in the form `server.id=hostname:port:port`. The first port is for a follower in the ensemble to listen on for the leader; the second is for leader election. You set `id` for each server in the next step.

4. Create a file named `myid` in the server's `DataDir`; in this example, `/var/lib/zookeeper/myid`. The file must contain only a single line, and that line must consist of a single unique number between 1 and 255; this is the `id` component mentioned in the previous step. In this example, the server whose hostname is `zoo1` must have a `myid` file that contains only 1.
5. Start each server as described in the [previous section](#).
6. Test the deployment by running a ZooKeeper client:

```
zookeeper-client -server hostname:port
```

For example:

```
zookeeper-client -server zool:2181
```

For more information on configuring a multi-server deployment, see [Clustered \(Multi-Server\) Setup](#) in the ZooKeeper Administrator's Guide.

Setting up Supervisory Process for the ZooKeeper Server

The ZooKeeper server is designed to be both highly reliable and highly available. This means that:

- If a ZooKeeper server encounters an error it cannot recover from, it will "fail fast" (the process will exit immediately)
- When the server shuts down, the ensemble remains active, and continues serving requests
- Once restarted, the server rejoins the ensemble without any further manual intervention.

Cloudera recommends that you fully automate this process by configuring a supervisory service to manage each server, and restart the ZooKeeper server process automatically if it fails. See the [ZooKeeper Administrator's Guide](#) for more information.

Maintaining a ZooKeeper Server

The ZooKeeper server continually saves `znode` snapshot files and, optionally, transactional logs in a Data Directory to enable you to recover data. It's a good idea to back up the ZooKeeper Data Directory periodically. Although ZooKeeper is highly reliable because a persistent copy is replicated on each server, recovering from backups may be necessary if a catastrophic failure or user error occurs.

When you use the default configuration, the ZooKeeper server does not remove the snapshots and log files, so they will accumulate over time. You will need to clean up this directory occasionally, taking into account on your backup schedules and processes. To automate the cleanup, a `zkCleanup.sh` script is provided in the `bin` directory of the `zookeeper` base package. Modify this script as necessary for your situation. In general, you want to run this as a `cron` task based on your backup schedule.

The data directory is specified by the `dataDir` parameter in the ZooKeeper [configuration file](#), and the data log directory is specified by the `dataLogDir` parameter.

For more information, see [Ongoing Data Directory Cleanup](#).

Viewing the ZooKeeper Documentation

For additional ZooKeeper documentation, see <https://archive.cloudera.com/cdh5/cdh/5/zookeeper/>.

Avro Usage

[Apache Avro](#) is a serialization system. Avro supports rich data structures, a compact binary encoding, and a container file for sequences of Avro data (often referred to as "Avro data files"). Avro is designed to be language-independent and there are several language bindings for it, including Java, C, C++, Python, and Ruby.

Avro does not rely on generated code, which means that processing data imported from Flume or Sqoop 1 is simpler than using Hadoop Writables in Sequence Files, where you have to take care that the generated classes are on the processing job's classpath. Furthermore, Pig and Hive cannot easily process Sequence Files with custom Writables, so users often revert to using text, which has disadvantages from a compactness and compressibility point of view (compressed text is not generally splittable, making it difficult to process efficiently using MapReduce).

All components in CDH 5 that produce or consume files support Avro data files as a file format. But bear in mind that because uniform Avro support is new, there may be some rough edges or missing features.

The following sections contain brief notes on how to get started using Avro in the various CDH 5 components:

- [Avro Data Files](#)
- [Compression](#)
- [Flume](#)
- [Sqoop](#)

- [MapReduce](#)
- [Streaming](#)
- [Pig](#)
- [Hive](#)

Avro Data Files

Avro data files have the `.avro` extension. Make sure the files you create have this extension, since some tools look for it to determine which files to process as Avro (e.g. `AvroInputFormat` and `AvroAsTextInputFormat` for MapReduce and Streaming).

Compression for Avro Data Files

By default Avro data files are not compressed, but it is generally advisable to enable compression to reduce disk usage and increase read and write performance. Avro data files support Deflate and [Snappy](#) compression. Snappy is faster, while Deflate is slightly more compact.

You do not need to do any additional configuration to read a compressed Avro data file rather than an uncompressed one. However, to write an Avro data file you need to specify the type of compression to use. How you specify compression depends on the component being used, as explained in the sections below.

Using Flume with Avro

The [HDFSEventSink](#) that is used to serialize event data onto HDFS supports plugin implementations of [EventSerializer](#) interface. Implementations of this interface have full control over the serialization format and can be used in cases where the default serialization format provided by the Sink does not suffice.

An abstract implementation of the `EventSerializer` interface is provided along with Flume, called the [AbstractAvroEventSerializer](#). This class can be extended to support custom schema for Avro serialization over HDFS. A simple implementation that maps the events to a representation of String header map and byte payload in Avro is provided by the class [FlumeEventAvroEventSerializer](#) which can be used by setting the `serializer` property of the Sink as follows:

```
<agent-name>.sinks.<sink-name>.serializer = AVRO_EVENT
```

Importing Avro Files with Sqoop 1 Using the Command Line

On the command line, use the following option to import to Avro data files:

```
--as-avrodatafile
```

Sqoop 1 will automatically generate an Avro schema that corresponds to the database table being exported from.

To enable Snappy compression, add the following option:

```
--compression-codec snappy
```



Note:

Sqoop 2 does not currently support Avro.

Using Avro with MapReduce

The Avro MapReduce API is an Avro module for running MapReduce programs which produce or consume Avro data files.

If you are using Maven, simply add the following dependency to your POM:

```
<dependency>
  <groupId>org.apache.avro</groupId>
  <artifactId>avro-mapred</artifactId>
  <version>1.7.3</version>
```

```
<classifier>hadoop2</classifier>
</dependency>
```

Then write your program using the [Avro MapReduce javadoc](#) for guidance.

At runtime, include the `avro` and `avro-mapred` JARs in the `HADOOP_CLASSPATH`; and the `avro`, `avro-mapred` and `paranamer` JARs in `-libjars`.

To enable Snappy compression on output files call `AvroJob.setOutputCodec(job, "snappy")` when configuring the job. You will also need to include the `snappy-java` JAR in `-libjars`.

Streaming

To read from Avro data files from a streaming program, specify `org.apache.avro.mapred.AvroAsTextInputFormat` as the input format. This input format will convert each datum in the Avro data file to a string. For a "bytes" schema, this will be the raw bytes, while in the general case it will be a single-line [JSON](#) representation of the datum.

To write to Avro data files from a streaming program, specify `org.apache.avro.mapred.AvroTextOutputFormat` as the output format. This output format will create Avro data files with a "bytes" schema, where each datum is a tab-delimited key-value pair.

At runtime specify the `avro`, `avro-mapred` and `paranamer` JARs in `-libjars` in the streaming command.

To enable Snappy compression on output files, set the property `avro.output.codec` to `snappy`. You will also need to include the `snappy-java` JAR in `-libjars`.

Using Avro with Pig

CDH provides `AvroStorage` for Avro integration in Pig.

To use it, first register the `piggybank` JAR file and supporting libraries:

```
REGISTER piggybank.jar
REGISTER lib/avro-1.7.3.jar
REGISTER lib/json-simple-1.1.jar
REGISTER lib/snappy-java-1.0.4.1.jar
```

Then you can load Avro data files as follows:

```
a = LOAD 'my_file.avro' USING org.apache.pig.piggybank.storage.avro.AvroStorage();
```

Pig maps the Avro schema to a corresponding Pig schema.

You can store data in Avro data files with:

```
store b into 'output' USING org.apache.pig.piggybank.storage.avro.AvroStorage();
```

In the case of `store`, Pig generates an Avro schema from the Pig schema. It is possible to override the Avro schema, either by specifying it literally as a parameter to `AvroStorage`, or by using the same schema as an existing Avro data file. See the [Pig wiki](#) for details.

To store two relations in one script, specify an index to each `store` function. Here is an example:

```
set1 = load 'input1.txt' using PigStorage() as ( ... );
store set1 into 'set1' using org.apache.pig.piggybank.storage.avro.AvroStorage('index',
'1');

set2 = load 'input2.txt' using PigStorage() as ( ... );
store set2 into 'set2' using org.apache.pig.piggybank.storage.avro.AvroStorage('index',
'2');
```

For more information, see the [AvroStorage wiki](#); look for "index".

To enable Snappy compression on output files do the following before issuing the `STORE` statement:

```
SET mapred.output.compress true
SET mapred.output.compression.codec org.apache.hadoop.io.compress.SnappyCodec
SET avro.output.codec snappy
```

There is some additional documentation on the [Pig wiki](#). Note, however, that the version numbers of the JAR files to register are different on that page, so you should adjust them as shown above.

Using Avro with Hive

The following example demonstrates how to create a Hive table that is backed by Avro data files:

```
CREATE TABLE doctors
ROW FORMAT
SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
STORED AS
INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'
TBLPROPERTIES ('avro.schema.literal'='{
  "namespace": "testing.hive.avro.serde",
  "name": "doctors",
  "type": "record",
  "fields": [
    {
      "name": "number",
      "type": "int",
      "doc": "Order of playing the role"
    },
    {
      "name": "first_name",
      "type": "string",
      "doc": "first name of actor playing role"
    },
    {
      "name": "last_name",
      "type": "string",
      "doc": "last name of actor playing role"
    },
    {
      "name": "extra_field",
      "type": "string",
      "doc": "an extra field not in the original file",
      "default": "fishfingers and custard"
    }
  ]
}');

LOAD DATA LOCAL INPATH '/usr/share/doc/hive-0.7.1+42.55/examples/files/doctors.avro'
INTO TABLE doctors;
```

You could also create an Avro backed Hive table by using an Avro schema file:

```
CREATE TABLE my_avro_table(notused INT)
ROW FORMAT SERDE
'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
WITH SERDEPROPERTIES (
  'avro.schema.url'='file:///tmp/schema.avsc')
STORED as INPUTFORMAT
'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat' ;
```

The `avro.schema.url` is a URL (here a `file://` URL) pointing to an Avro schema file that is used for reading and writing, it could also be an `hdfs` URL, eg. `hdfs://hadoop-namenode-uri/examplefile`

To enable Snappy compression on output files, run the following before writing to the table:

```
SET hive.exec.compress.output=true;
SET avro.output.codec=snappy;
```

You will also need to include the `snappy-java` JAR in `--auxpath`. The `snappy-java` JAR is located at:

```
/usr/lib/hive/lib/snappy-java-1.0.4.1.jar
```

[Haivvreo SerDe](#) has been merged into Hive as `AvroSerDe`, and it is no longer supported in its original form. `schema.url` and `schema.literal` have been changed to `avro.schema.url` and `avro.schema.literal` as a result of the merge. If you were you using [Haivvreo SerDe](#), you can use the new Hive `AvroSerDe` with tables created with the `Haivvreo SerDe`. For example, if you have a table `my_avro_table` that uses the `Haivvreo SerDe`, you can do the following to make the table use the new `AvroSerDe`:

```
ALTER TABLE my_avro_table SET SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe';

ALTER TABLE my_avro_table SET FILEFORMAT
INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat';
```

Using the Parquet File Format with Impala, Hive, Pig, and MapReduce

Parquet is automatically installed when you install any of the above components, and the necessary libraries are automatically placed in the classpath for all of them. Copies of the libraries are in `/usr/lib/parquet` or inside the parcels in `/lib/parquet`.

The Parquet file format incorporates several features that make it highly suited to data warehouse-style operations:

- Columnar storage layout. A query can examine and perform calculations on all values for a column while reading only a small fraction of the data from a data file or table.
- Flexible compression options. The data can be compressed with any of several codecs. Different data files can be compressed differently. The compression is transparent to applications that read the data files.
- Innovative encoding schemes. Sequences of identical, similar, or related data values can be represented in ways that save disk space and memory. The encoding schemes provide an extra level of space savings beyond the overall compression for each data file.
- Large file size. The layout of Parquet data files is optimized for queries that process large volumes of data, with individual files in the multi-megabyte or even gigabyte range.

Among components of the CDH distribution, Parquet support originated in Impala. Impala can create Parquet tables, insert data into them, convert data from other file formats to Parquet, and then perform SQL queries on the resulting data files. Parquet tables created by Impala can be accessed by Hive, and vice versa.

The CDH software stack lets you use the tool of your choice with the Parquet file format, for each phase of data processing. For example, you can read and write Parquet files using Pig and MapReduce jobs. You can convert, transform, and query Parquet tables through Impala and Hive. And you can interchange data files between all of those components.

Using Parquet Tables with Impala

The Cloudera Impala component can create tables that use Parquet data files; insert data into those tables, converting the data into Parquet format; and query Parquet data files produced by Impala or by other components. The only syntax required is the `STORED AS PARQUET` clause on the `CREATE TABLE` statement. After that, all `SELECT`, `INSERT`, and other statements recognize the Parquet format automatically. For example, a session in the `impala-shell` interpreter might look as follows:

```
[localhost:21000] > create table parquet_table (x int, y string) stored as parquet;
[localhost:21000] > insert into parquet_table select x, y from some_other_table;
Inserted 50000000 rows in 33.52s
[localhost:21000] > select y from parquet_table where x between 70 and 100;
```

Once you create a Parquet table this way in Impala, you can query it or insert into it through either Impala or Hive.

Remember that Parquet format is optimized for working with large data files. In Impala 2.0 and later, the default size of Parquet files written by Impala is 256 MB; in earlier releases, 1 GB. Avoid using the `INSERT ... VALUES` syntax, or partitioning the table at too granular a level, if that would produce a large number of small files that cannot take advantage of the Parquet optimizations for large data chunks.

Inserting data into a partitioned Impala table can be a memory-intensive operation, because each data file requires a memory buffer to hold the data before being written. Such inserts can also exceed HDFS limits on simultaneous open files, because each node could potentially write to a separate data file for each partition, all at the same time. Always make sure table and column statistics are in place for any table used as the source for an `INSERT ... SELECT` operation into a Parquet table. If capacity problems still occur, consider splitting up such insert operations into one `INSERT` statement per partition.

Impala can query Parquet files that use the `PLAIN`, `PLAIN_DICTIONARY`, `BIT_PACKED`, and `RLE` encodings. Currently, Impala does not support `RLE_DICTIONARY` encoding. When creating files outside of Impala for use by Impala, make sure to use one of the supported encodings. In particular, for MapReduce jobs, `parquet.writer.version` must not be defined (especially as `PARQUET_2_0`) for writing the configurations of Parquet MR jobs. Use the default version (or format). The default format, 1.0, includes some enhancements that are compatible with older versions. Data using the 2.0 format might not be consumable by Impala, due to use of the `RLE_DICTIONARY` encoding.

If you use Sqoop to convert RDBMS data to Parquet, be careful with interpreting any resulting values from `DATE`, `DATETIME`, or `TIMESTAMP` columns. The underlying values are represented as the Parquet `INT64` type, which is represented as `BIGINT` in the Impala table. The Parquet values represent the time in milliseconds, while Impala interprets `BIGINT` as the time in seconds. Therefore, if you have a `BIGINT` column in a Parquet table that was imported this way from Sqoop, divide the values by 1000 when interpreting as the `TIMESTAMP` type.

For complete instructions and examples, see [Using the Parquet File Format with Impala Tables](#).

Using Parquet Tables in Hive

To create a table named `PARQUET_TABLE` that uses the Parquet format, you would use a command like the following, substituting your own table name, column names, and data types:

```
hive> CREATE TABLE parquet_table_name (x INT, y STRING)
      STORED AS PARQUET;
```



Note:

- Once you create a Parquet table this way in Hive, you can query it or insert into it through either Impala or Hive. Before the first time you access a newly created Hive table through Impala, issue a one-time `INVALIDATE METADATA` statement in the `impala-shell` interpreter to make Impala aware of the new table.
- `dfs.block.size` should be set to 256MB in `hdfs-site.xml`.

If the table will be populated with data files generated outside of Impala and Hive, it is often useful to create the table as an external table pointing to the location where the files will be created:

```
hive> create external table parquet_table_name (x INT, y STRING)
      ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
      STORED AS
      INPUTFORMAT "parquet.hive.DeprecatedParquetInputFormat"
      OUTPUTFORMAT "parquet.hive.DeprecatedParquetOutputFormat"
      LOCATION '/test-warehouse/tinytable';
```

To populate the table with an `INSERT` statement, and to read the table with a `SELECT` statement, see [Using the Parquet File Format with Impala Tables](#).

Select the compression to use when writing data with the `parquet.compression` property, for example:

```
set parquet.compression=GZIP;
INSERT OVERWRITE TABLE tinytable SELECT * FROM texttable;
```

The valid options for compression are:

- UNCOMPRESSED
- GZIP
- SNAPPY

Using Parquet Files in Pig

Reading Parquet Files in Pig

Assuming the external table was created and populated with Impala or Hive as described above, the Pig instruction to read the data is:

```
grunt> A = LOAD '/test-warehouse/tinytable' USING parquet.pig.ParquetLoader AS (x: int,
y int);
```

Writing Parquet Files in Pig

Create and populate a Parquet file with the `ParquetStorer` class:

```
grunt> store A into '/test-warehouse/tinytable' USING parquet.pig.ParquetStorer;
```

There are three compression options: `uncompressed`, `snappy`, and `gzip`. The default is `snappy`. You can specify one of them once before the first store instruction in a Pig script:

```
SET parquet.compression gzip;
```

Using Parquet Files in MapReduce

MapReduce needs Thrift in its `CLASSPATH` and in `libjars` to access Parquet files. It also needs `parquet-format` in `libjars`. Perform the following setup before running MapReduce jobs that access Parquet data files:

```
if [ -e /opt/cloudera/parcels/CDH ] ; then
    CDH_BASE=/opt/cloudera/parcels/CDH
else
    CDH_BASE=/usr
fi
THRIFTJAR=`ls -l $CDH_BASE/lib/hive/lib/libthrift*jar | awk '{print $9}' | head -1`
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$THRIFTJAR
export LIBJARS=`echo "$CLASSPATH" | awk 'BEGIN { RS = ":" } { print }' | grep
parquet-format | tail -1`
export LIBJARS=$LIBJARS,$THRIFTJAR

hadoop jar my-parquet-mr.jar -libjars $LIBJARS
```

Reading Parquet Files in MapReduce

Taking advantage of the `Example` helper classes in the Parquet JAR files, a simple map-only MapReduce job that reads Parquet files can use the `ExampleInputFormat` class and the `Group` value class. There is nothing special about the reduce phase when using Parquet files. The following example demonstrates how to read a Parquet file in a MapReduce job; portions of code specific to the Parquet aspect are shown in bold.

```
import static java.lang.Thread.sleep;
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
```

```

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import parquet.Log;
import parquet.example.data.Group;
import parquet.hadoop.example.ExampleInputFormat;

public class TestReadParquet extends Configured
    implements Tool {
    private static final Log LOG =
        Log.getLog(TestReadParquet.class);

    /*
     * Read a Parquet record
     */
    public static class MyMap extends
        Mapper<LongWritable, Group, NullWritable, Text> {

        @Override
        public void map(LongWritable key, Group value, Context context) throws IOException,
            InterruptedException {
            NullWritable outKey = NullWritable.get();
            String outputRecord = "";
            // Get the schema and field values of the record
            String inputRecord = value.toString();
            // Process the value, create an output record
            // ...
            context.write(outKey, new Text(outputRecord));
        }
    }

    public int run(String[] args) throws Exception {

        Job job = new Job(getConf());

        job.setJarByClass(getClass());
        job.setJobName(getClass().getName());
        job.setMapOutputKeyClass(LongWritable.class);
        job.setMapOutputValueClass(Text.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);
        job.setMapperClass(MyMap.class);
        job.setNumReduceTasks(0);

        job.setInputFormatClass(ExampleInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.setInputPaths(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.waitForCompletion(true);
        return 0;
    }

    public static void main(String[] args) throws Exception {
        try {
            int res = ToolRunner.run(new Configuration(), new TestReadParquet(), args);
            System.exit(res);
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(255);
        }
    }
}

```

Writing Parquet Files in MapReduce

When writing Parquet files you will need to provide a schema. The schema can be specified in the run method of the job before submitting it, for example:

```
...
import parquet.Log;
import parquet.example.data.Group;
import parquet.hadoop.example.GroupWriteSupport;
import parquet.hadoop.example.ExampleInputFormat;
import parquet.hadoop.example.ExampleOutputFormat;
import parquet.hadoop.metadata.CompressionCodecName;
import parquet.hadoop.ParquetFileReader;
import parquet.hadoop.metadata.ParquetMetadata;
import parquet.schema.MessageType;
import parquet.schema.MessageTypeParser;
import parquet.schema.Type;
...
public int run(String[] args) throws Exception {
...

    String writeSchema = "message example {\n" +
        "required int32 x;\n" +
        "required int32 y;\n" +
        "}";
    ExampleOutputFormat.setSchema(
        job,
        MessageTypeParser.parseMessageType(writeSchema));

    job.submit();
}
```

or it can be extracted from the input file(s) if they are in Parquet format:

```
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.LocatedFileStatus;
import org.apache.hadoop.fs.RemoteIterator;
...

public int run(String[]
    args) throws Exception {
...

    String inputFile = args[0];
    Path parquetFilePath = null;
    // Find a file in case a directory was passed

    RemoteIterator<LocatedFileStatus> it = FileSystem.get(getConf()).listFiles(new
    Path(inputFile), true);
    while(it.hasNext()) {
        FileStatus fs = it.next();

        if(fs.isFile()) {
            parquetFilePath = fs.getPath();
            break;
        }
    }
    if(parquetFilePath == null) {
        LOG.error("No file found for " + inputFile);
        return 1;
    }
    ParquetMetadata readFooter =
        ParquetFileReader.readFooter(getConf(), parquetFilePath);
    MessageType schema =
        readFooter.getFileMetaData().getSchema();
    GroupWriteSupport.setSchema(schema, getConf());

    job.submit();
}
```

Records can then be written in the mapper by composing a `Group` as value using the `Example` classes and no key:

```
protected void map(LongWritable key, Text value,
    Mapper<LongWritable, Text, Void, Group>.Context context)
    throws java.io.IOException, InterruptedException {
    int x;
    int y;
    // Extract the desired output values from the input text
    //
    Group group = factory.newGroup()
        .append("x", x)
        .append("y", y);
    context.write(null, group);
}
```

Compression can be set before submitting the job with:

```
ExampleOutputFormat.setCompression(job, codec);
```

The codec should be one of the following:

- `CompressionCodecName.UNCOMPRESSED`
- `CompressionCodecName.SNAPPY`
- `CompressionCodecName.GZIP`

Parquet File Interoperability

Impala has included Parquet support from the beginning, using its own high-performance code written in C++ to read and write the Parquet files. The Parquet JARs for use with Hive, Pig, and MapReduce are available with CDH 4.5 and higher. Using the Java-based Parquet implementation on a CDH release prior to CDH 4.5 is not supported.

A Parquet table created by Hive can typically be accessed by Impala 1.1.1 and higher with no changes, and vice versa. Prior to Impala 1.1.1, when Hive support for Parquet was not available, Impala wrote a dummy `SerDes` class name into each data file. These older Impala data files require a one-time `ALTER TABLE` statement to update the metadata for the `SerDes` class name before they can be used with Hive. See [Cloudera Impala Incompatible Changes](#) for details.

A Parquet file written by Hive, Impala, Pig, or MapReduce can be read by any of the others. Different defaults for file and block sizes, compression and encoding settings, and so on might cause performance differences depending on which component writes or reads the data files. For example, Impala typically sets the HDFS block size to 256 MB and divides the data files into 256 MB chunks, so that each I/O request reads an entire data file.

There may be limitations in a particular release. The following are current limitations in CDH:

- Parquet has not been tested with HCatalog. Without HCatalog, Pig cannot correctly read dynamically partitioned tables; that is true for all file formats.
- Currently, Impala does not support table columns using nested data types or composite data types such as `map`, `struct`, or `array`. Any Parquet data files that include such types cannot be queried through Impala.
- Cloudera supports some but not all of the object models from the upstream `Parquet-MR` project. Currently, the supported object models are:

- `parquet-avro` (recommended for Cloudera users)
- `parquet-thrift`
- `parquet-protobuf`
- `parquet-pig`
- The Impala and Hive object models that are built into those components, not available in external libraries. (CDH does not include the `parquet-hive` module of the `parquet-mr` project, because recent versions of Hive have Parquet support built in.)

Parquet File Structure

To examine the internal structure and data of Parquet files, you can use the `parquet-tools` command that comes with CDH. Make sure this command is in your `$PATH`. (Typically it is symlinked from `/usr/bin`; sometimes, depending

on your installation setup, you might need to locate it under a CDH-specific directory.) Use `parquet-tools -h` to see usage information for all the subcommands. The arguments to this command let you perform operations such as:

- `cat`: Print a file's contents to standard out. In CDH 5.5 and higher, you can use the `-j` option to output JSON.
- `head`: Print the first few records of a file to standard output.
- `schema`: Print the Parquet schema for the file.
- `meta`: Print the file footer metadata, including key-value properties (like Avro schema), compression ratios, encodings, compression used, and row group information.
- `dump`: Print all data and metadata.

Here are some examples showing `parquet-tools` usage:

```
$ # Be careful doing this for a big file! Use parquet-tools head to be safe.
$ parquet-tools cat sample.parq
year = 1992
month = 1
day = 2
dayofweek = 4
dep_time = 748
crs_dep_time = 750
arr_time = 851
crs_arr_time = 846
carrier = US
flight_num = 53
actual_elapsed_time = 63
crs_elapsed_time = 56
arrdelay = 5
depdelay = -2
origin = CMH
dest = IND
distance = 182
cancelled = 0
diverted = 0

year = 1992
month = 1
day = 3
...
```

```
$ parquet-tools head -n 2 sample.parq
year = 1992
month = 1
day = 2
dayofweek = 4
dep_time = 748
crs_dep_time = 750
arr_time = 851
crs_arr_time = 846
carrier = US
flight_num = 53
actual_elapsed_time = 63
crs_elapsed_time = 56
arrdelay = 5
depdelay = -2
origin = CMH
dest = IND
distance = 182
cancelled = 0
diverted = 0

year = 1992
month = 1
day = 3
```

...

```
$ parquet-tools schema sample.parq
message schema {
  optional int32 year;
  optional int32 month;
  optional int32 day;
  optional int32 dayofweek;
  optional int32 dep_time;
  optional int32 crs_dep_time;
  optional int32 arr_time;
  optional int32 crs_arr_time;
  optional binary carrier;
  optional int32 flight_num;
}
```

...

```
$ parquet-tools meta sample.parq
creator:          impala version 2.2.0-cdh5.4.3 (build
1517bb0f71cd604a00369254ac6d88394df83e0f6)
```

```
file schema:      schema
```

```
-----
year:              OPTIONAL INT32 R:0 D:1
month:             OPTIONAL INT32 R:0 D:1
day:              OPTIONAL INT32 R:0 D:1
dayofweek:        OPTIONAL INT32 R:0 D:1
dep_time:         OPTIONAL INT32 R:0 D:1
crs_dep_time:     OPTIONAL INT32 R:0 D:1
arr_time:         OPTIONAL INT32 R:0 D:1
crs_arr_time:     OPTIONAL INT32 R:0 D:1
carrier:          OPTIONAL BINARY R:0 D:1
flight_num:       OPTIONAL INT32 R:0 D:1
...

```

```
row group 1:      RC:20636601 TS:265103674
```

```
-----
year:              INT32 SNAPPY DO:4 FPO:35 SZ:10103/49723/4.92 VC:20636601
ENC:PLAIN_DICTIONARY,RLE,PLAIN
month:             INT32 SNAPPY DO:10147 FPO:10210 SZ:11380/35732/3.14 VC:20636601
ENC:PLAIN_DICTIONARY,RLE,PLAIN
day:              INT32 SNAPPY DO:21572 FPO:21714 SZ:3071658/9868452/3.21 VC:20636601
ENC:PLAIN_DICTIONARY,RLE,PLAIN
dayofweek:        INT32 SNAPPY DO:3093276 FPO:3093319 SZ:2274375/5941876/2.61
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
dep_time:         INT32 SNAPPY DO:5367705 FPO:5373967 SZ:28281281/28573175/1.01
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
crs_dep_time:     INT32 SNAPPY DO:33649039 FPO:33654262 SZ:10220839/11574964/1.13
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
arr_time:         INT32 SNAPPY DO:43869935 FPO:43876489 SZ:28562410/28797767/1.01
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
crs_arr_time:     INT32 SNAPPY DO:72432398 FPO:72438151 SZ:10908972/12164626/1.12
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
carrier:          BINARY SNAPPY DO:83341427 FPO:83341558 SZ:114916/128611/1.12
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
flight_num:       INT32 SNAPPY DO:83456393 FPO:83488603 SZ:10216514/11474301/1.12
VC:20636601 ENC:PLAIN_DICTIONARY,RLE,PLAIN
...

```

Examples of Java Programs to Read and Write Parquet Files

You can find full examples of Java code at the Cloudera [Parquet examples](#) Github repository.

The [TestReadWriteParquet.java](#) example demonstrates the “identity” transform. It reads any Parquet data file and writes a new file with exactly the same content.

The [TestReadParquet.java](#) example reads a Parquet data file, and produces a new text file in CSV format with the same content.

Building RPMs from CDH Source RPMs

This section describes how to build binary packages (RPMs) from published CDH source packages (SRPMs):

- [Prerequisites](#)
- [Setting up an Environment for Building RPMs](#)
- [Building an RPM](#)

Prerequisites

- Oracle Java Development Kit (JDK) version 6.
- [Apache Ant](#) version 1.7 or higher.
- [Apache Maven](#) 3.0 or higher.
- The following environment variables must be set: JAVA_HOME, JAVA5_HOME, FORREST_HOME, and ANT_HOME.
- Your PATH must include the JAVA_HOME, ANT_HOME, FORREST_HOME and maven bin directories.
- If you are using Red Hat or CentOS systems, the rpmdevtools package is required for the rpmdev-setuptree command used below.

Setting up an environment for building RPMs

RHEL or CentOS systems

Users of these systems can run the following command to set up their environment:

```
$ rpmdev-setuptree # Creates ~/rpmbuild and ~/.rpmmacros
```

SLES systems

Users of these systems can run the following command to set up their environment:

```
$ mkdir -p ~/rpmbuild/{BUILD,RPMS,S{OURCE,PEC,RPM}S}
$ echo "%_topdir $HOME/rpmbuild"> ~/.rpmmacros
```

Building an RPM

Download SRPMs from archive.cloudera.com. The source RPMs for CDH 5 reside at

https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/5/SRPMS/,

https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/SRPMS/ or

https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/SRPMS/. Run the following commands as a non-root user, substituting the particular SRPM that you intend to build:

```
$ export SRPM=hadoop-0.20-0.20.2+320-1.src.rpm
$ rpmbuild --nodeps --rebuild $SRPM # Builds the native RPMs
$ rpmbuild --nodeps --rebuild --target noarch $SRPM # Builds the java RPMs
```

The built packages can be found in \$HOME/rpmbuild/RPMS.

Apache and Third-Party Licenses

This section describes the licenses that apply to CDH 5.

Apache License

All software developed by Cloudera for CDH is released with an Apache 2.0 license. Please let us know if you find any file that doesn't explicitly state the Apache license at the top and we'll immediately fix it.

Apache License Version 2.0, January 2004 <http://www.apache.org/licenses/>

Copyright 2010-2013 Cloudera

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Third-Party Licenses

For a list of third-party licenses associated with CDH, see

<http://www.cloudera.com/content/cloudera-content/cloudera-docs/Licenses/Third-Party-Licenses/Third-Party-Licenses.html>.

Uninstalling CDH Components

Before uninstalling CDH, stop all Hadoop processes, following the instructions in [Stopping Services](#).

Here are the commands to use to uninstall the Hadoop components on different Linux systems.

Operating System	Commands	Comments
Red-Hat-compatible	<code>yum remove</code>	
Debian and Ubuntu	<code>apt-get remove</code> or <code>apt-get purge</code>	<code>apt-get</code> can be run with the <code>remove</code> option to remove only the installed packages or with the <code>purge</code> option to remove packages and configuration
SLES	<code>zypper remove</code>	

Uninstalling from Red Hat, CentOS, and Similar Systems

Component to remove	Command
Mahout	<code>\$ sudo yum remove mahout</code>
Whirr	<code>\$ sudo yum remove whirr</code>
Hue	<code>\$ sudo yum remove hue</code>
Pig	<code>\$ sudo yum remove pig</code>
Sqoop 1	<code>\$ sudo yum remove sqoop</code>
Sqoop 2	<code>\$ sudo yum remove sqoop2-server sqoop2-client</code>
Flume	<code>\$ sudo yum remove flume</code>
Oozie client	<code>\$ sudo yum remove oozie-client</code>
Oozie server	<code>\$ sudo yum remove oozie</code>
Hive	<code>\$ sudo yum remove hive hive-metastore hive-server hive-server2</code>
HBase	<code>\$ sudo yum remove hadoop-hbase</code>
ZooKeeper server	<code>\$ sudo yum remove hadoop-zookeeper-server</code>
ZooKeeper client	<code>\$ sudo yum remove hadoop-zookeeper</code>

Component to remove	Command
ZooKeeper Failover Controller (ZKFC)	<code>\$ sudo yum remove hadoop-hdfs-zkfc</code>
HDFS HA Journal Node	<code>\$ sudo yum remove hadoop-hdfs-hadoop-hdfs-journalnode</code>
Hadoop repository packages	<code>\$ sudo yum remove cloudera-cdh<n></code>
HttpFS	<code>\$ sudo yum remove hadoop-httpfs</code>
Hadoop core packages	<code>\$ sudo yum remove hadoop</code>

Uninstalling from Debian and Ubuntu

Use the `apt-get` command to uninstall software on Debian and Ubuntu systems. You can use `apt-get remove` or `apt-get purge`; the difference is that `apt-get remove` removes all your configuration data as well as the package files.



Warning:

For this reason, you should `apt-get remove` only with great care, and after making sure you have backed up all your configuration data.

The `apt-get remove` commands to uninstall the Hadoop components from a Debian or Ubuntu system are:

Component to remove	Command
Whirr	<code>\$ sudo apt-get remove whirr</code>
Hue	<code>\$ sudo apt-get remove hue</code>
Pig	<code>\$ sudo apt-get remove pig</code>
Sqoop 1	<code>\$ sudo apt-get remove sqoop</code>
Sqoop 2	<code>\$ sudo apt-get remove sqoop2-server sqoop2-client</code>
Flume	<code>\$ sudo apt-get remove flume</code>
Oozie client	<code>\$ sudo apt-get remove oozie-client</code>
Oozie server	<code>\$ sudo apt-get remove oozie</code>
Hive	<code>\$ sudo apt-get remove hive hive-metastore hive-server hive-server2</code>
HBase	<code>\$ sudo apt-get remove hadoop-hbase</code>
ZooKeeper server	<code>\$ sudo apt-get remove hadoop-zookeeper-server</code>
ZooKeeper client	<code>\$ sudo apt-get remove hadoop-zookeeper</code>
ZooKeeper Failover Controller (ZKFC)	<code>\$ sudo apt-get remove hadoop-hdfs-zkfc</code>
HDFS HA Journal Node	<code>\$ apt-get remove hadoop-hdfs-hadoop-hdfs-journalnode</code>

Component to remove	Command
HttpFS	<code>\$ sudo apt-get remove hadoop-httpfs</code>
Hadoop repository packages	<code>\$ sudo apt-get remove cdh<n>-repository</code>
Hadoop core packages	<code>\$ sudo apt-get remove hadoop</code>

Uninstalling from SLES

Component removed	Command
Whirr	<code>\$ sudo zypper remove whirr</code>
Hue	<code>\$ sudo zypper remove hue</code>
Pig	<code>\$ sudo zypper remove pig</code>
Sqoop	<code>\$ sudo zypper remove sqoop</code>
Sqoop	<code>\$ sudo zypper remove sqoop2-server sqoop2-client</code>
Flume	<code>\$ sudo zypper remove flume</code>
Oozie server	<code>\$ sudo zypper remove oozie</code>
Oozie client	<code>\$ sudo zypper remove oozie-client</code>
Hive	<code>\$ sudo zypper remove hive hive-metastore hive-server hive-server2</code>
HBase	<code>\$ sudo zypper remove hadoop-hbase</code>
ZooKeeper server	<code>\$ sudo zypper remove hadoop-zookeeper-server</code>
ZooKeeper client	<code>\$ sudo zypper remove hadoop-zookeeper</code>
ZooKeeper Failover Controller (ZKFC)	<code>\$ sudo zypper remove hadoop-hdfs-zkfc</code>
HDFS HA Journal Node	<code>\$ sudo zypper remove hadoop-hdfs-hadoop-hdfs-journalnode</code>
HttpFS	<code>\$ sudo zypper remove hadoop-httpfs</code>
Hadoop repository packages	<code>\$ sudo zypper remove cloudera-cdh</code>
Hadoop core packages	<code>\$ sudo zypper remove hadoop</code>

Additional clean-up

The uninstall commands may not remove all traces of Hadoop from your system. The `apt-get purge` commands available for Debian and Ubuntu systems delete more files than the commands that use the `remove` option but are still not comprehensive. If you want to remove all vestiges of Hadoop from your system, look for the following and remove them manually:

- log files
- modified system configuration files
- Hadoop configuration files in directories under `/etc` such as `hadoop`, `hbase`, `hue`, `hive`, `oozie`, `sqoop`, `zookeeper`, and `zookeeper.dist`

- user/group identifiers
- Oozie and Hue databases
- Documentation packages

Viewing the Apache Hadoop Documentation

- For additional Apache Hadoop documentation, see <https://archive.cloudera.com/cdh5/cdh/5/hadoop>.
- For more information about YARN, see the Apache Hadoop NextGen MapReduce (YARN) page at <https://archive.cloudera.com/cdh5/cdh/5/hadoop/hadoop-yarn/hadoop-yarn-site/YARN.html>.

Troubleshooting Installation and Upgrade Problems

For information on known issues, see [Known Issues and Workarounds in Cloudera Manager 5](#).

Symptom	Reason	Solution
The Cloudera Manager Server fails to start after upgrade.	There were active commands running before upgrade. This includes commands a user might have run and also for commands Cloudera Manager automatically triggers, either in response to a state change, or something that's on a schedule.	<p>Downgrade the Cloudera Manager Server, stop the commands, and reapply the upgrade. If you must proceed without downgrade, active commands can be stopped if you start the Cloudera Manager Server with the following command:</p> <pre>service cloudera-scm-server force_start</pre>
"Failed to start server" reported by <code>cloudera-manager-installer.bin</code> . <code>/var/log/cloudera-scm-server/cloudera-scm-server.log</code> contains a message beginning <code>Caused by:</code> <code>java.lang.ClassNotFoundException: com.mysql.jdbc.Driver...</code>	You may have SELinux enabled.	Disable SELinux by running <code>sudo setenforce 0</code> on the Cloudera Manager Server host. To disable it permanently, edit <code>/etc/selinux/config</code> .
Installation interrupted and installer won't restart.	You need to do some manual cleanup.	See Uninstalling Cloudera Manager and Managed Software on page 147.
Cloudera Manager Server fails to start and the Server is configured to use a MySQL database to store information about service configuration.	Tables may be configured with the ISAM engine. The Server will not start if its tables are configured with the MyISAM engine, and an error such as the following will appear in the log file: <code>Tables ... have unsupported engine type InnoDB is required.</code>	Make sure that the InnoDB engine is configured, not the MyISAM engine. To check what engine your tables are using, run the following command from the MySQL shell: <code>mysql> show table status;</code> For more information, see MySQL Database on page 48.
Agents fail to connect to Server. Error 113 ('No route to host') in <code>/var/log/cloudera-scm-agent/cloudera-scm-agent.log</code>	You may have SELinux or iptables enabled.	Check <code>/var/log/cloudera-scm-server/cloudera-scm-server.log</code> on the Server host and <code>/var/log/cloudera-scm-agent/cloudera-scm-agent.log</code> on the Agent hosts. Disable SELinux and iptables.
Some cluster hosts do not appear when you click Find Hosts in install or update wizard.	You may have network connectivity problems.	<ul style="list-style-type: none"> • Make sure all cluster hosts have SSH port 22 open. • Check other common causes of loss of connectivity such as firewalls and interference from SELinux.
"Access denied" in install or update wizard during database configuration for Activity Monitor or Reports Manager.	Hostname mapping or permissions are incorrectly set up.	<ul style="list-style-type: none"> • For hostname configuration, see Configuring Network Names (CDH 4) or Configuring Network Names on page 190 (CDH 5).

Symptom	Reason	Solution
		<ul style="list-style-type: none"> For permissions, make sure the values you enter into the wizard match those you used when you configured the databases. The value you enter into the wizard as the database hostname <i>must</i> match the value you entered for the hostname (if any) when you configured the database. <p>For example, if you had entered the following when you created the database</p> <pre>grant all on activity_monitor.* TO 'amon_user'@'myhost1.myco.com' IDENTIFIED BY 'amon_password';</pre> <p>the value you enter here for the database hostname must be myhost1.myco.com. If you did not specify a host, or used a wildcard to allow access from any host, you can enter either the fully-qualified domain name (FQDN), or localhost. For example, if you entered</p> <pre>grant all on activity_monitor.* TO 'amon_user'@'%' IDENTIFIED BY 'amon_password';</pre> <p>the value you enter for the database hostname can be either the FQDN or localhost.</p>
Activity Monitor, Reports Manager, or Service Monitor databases fail to start.	MySQL binlog format problem.	Set binlog_format=mixed in /etc/my.cnf. For more information, see this MySQL bug report . See also Cloudera Manager and Managed Service Data Stores on page 38.
You have upgraded the Cloudera Manager Server, but now cannot start services.	You may have mismatched versions of the Cloudera Manager Server and Agents.	Make sure you have upgraded the Cloudera Manager Agents on all hosts. (The previous version of the Agents will heartbeat with the new version of the Server, but you cannot start HDFS and MapReduce with this combination.)
Cloudera services fail to start.	Java may not be installed or may be installed at a custom location.	See Configuring a Custom Java Home Location on page 131 for more information on resolving this issue.

Symptom	Reason	Solution
<p>The Activity Monitor displays a status of BAD in the Cloudera Manager Admin Console. The log file contains the following message:</p> <pre>ERROR 1436 (HY000): Thread stack overrun: 7808 bytes used of a 131072 byte stack, and 128000 bytes needed. Use 'mysqld -O thread_stack=#' to specify a bigger stack.</pre>	<p>The MySQL thread stack is too small.</p>	<ol style="list-style-type: none"> 1. Update the <code>thread_stack</code> value in <code>my.cnf</code> to 256KB. The <code>my.cnf</code> file is normally located in <code>/etc</code> or <code>/etc/mysql</code>. 2. Restart the <code>mysql</code> service: <pre>\$ sudo service mysql restart</pre> 3. Restart Activity Monitor.
<p>The Activity Monitor fails to start. Logs contain the error <code>read-committed isolation not safe for the statement binlog format</code>.</p>	<p>The <code>binlog_format</code> is not set to <code>mixed</code>.</p>	<p>Modify the <code>mysql.cnf</code> file to include the entry for <code>binlog format</code> as specified in MySQL Database on page 48.</p>
<p>Attempts to reinstall lower versions of CDH or Cloudera Manager using <code>yum</code> fails.</p>	<p>It is possible to install, uninstall, and reinstall CDH and Cloudera Manager. In certain cases, this does not complete as expected. If you install Cloudera Manager 5 and CDH 5, then uninstall Cloudera Manager and CDH, and then attempt to install CDH 4 and Cloudera Manager 4, incorrect cached information may result in the installation of an incompatible version of the Oracle JDK.</p>	<p>Clear information in the yum cache:</p> <ol style="list-style-type: none"> 1. Connect to the CDH host. 2. Execute either of the following commands: <pre>\$ yum --enablerepo='*' clean all</pre> or <pre>\$ rm -rf /var/cache/yum/cloudera*</pre> 3. After clearing the cache, proceed with installation.
<p>The Create Hive Metastore Database Tables command fails due to a problem with an escape string.</p>	<p>PostgreSQL versions 9 and higher require special configuration for Hive because of a backward-incompatible change in the default value of the <code>standard_conforming_strings</code> property. Versions up to PostgreSQL 9.0 defaulted to <code>off</code>, but starting with version 9.0 the default is <code>on</code>.</p>	<p>As the administrator user, use the following command to turn <code>standard_conforming_strings</code> off:</p> <pre>ALTER DATABASE <hive_db_name> SET standard_conforming_strings = off;</pre>
<p>After upgrading to CDH 5, HDFS DataNodes fail to start with exception:</p> <pre>Exception in thread "main" java.lang.RuntimeException: Cannot start datanode because the configured max locked memory size (dfs.datanode.max.locked.memory) of 4294967296 bytes is more than the datanode's available RLIMIT_MEMLOCK ulimit of 65536 bytes.</pre>	<p>HDFS caching, which is enabled by default in CDH 5, requires new memlock functionality from Cloudera Manager Agents.</p>	<p>Do the following:</p> <ol style="list-style-type: none"> 1. Stop all CDH and managed services. 2. On all hosts with Cloudera Manager Agents, hard restart the Agents. Before performing this step, ensure you understand the semantics of the <code>hard_restart</code> command by reading Hard Stopping and Restarting Agents. <ul style="list-style-type: none"> • Packages <pre>\$ sudo service cloudera-scm-agent hard_restart</pre> • Tarballs

Symptom	Reason	Solution
		<ul style="list-style-type: none">- To stop the Cloudera Manager Agent, run this command on each Agent host: <pre data-bbox="1110 342 1472 474">\$ sudo tarball_root/etc/init.d/cloudera-scm-agent hard_restart</pre>- If you are running single user mode, start Cloudera Manager Agent using the user account you chose. For example, you might run the Cloudera Manager Agent as <code>cloudera-scm</code>. In such a case there are following options:<ul style="list-style-type: none">- Run the following command: <pre data-bbox="1110 936 1472 1068">\$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent hard_restart</pre>- Edit the configuration files so the script internally changes the user, then run the script as root:<ol style="list-style-type: none">1. Remove the following line from tarball_root/etc/init.d/cloudera-scm-agent <pre data-bbox="1110 1430 1472 1493">export CMF_SUDO_CMD=" "</pre>2. Change the user and group in tarball_root/etc/init.d/cloudera-scm-agent to the user you want the Agent to run as. For example, to run as <code>cloudera-scm</code>, change the user and

Symptom	Reason	Solution
		<p style="text-align: right;">group as follows:</p> <pre style="border: 1px dashed blue; padding: 5px;">USER=cloudera-scm GROUP=cloudera-scm</pre> <p style="text-align: right;">3. Run the Agent script as root:</p> <pre style="border: 1px dashed blue; padding: 5px;">\$ sudo tarball_root/etc/init.d/cloudera-scm-agent hard_restart</pre> <p>3. Start all services.</p>
<p>You see the following error in NameNode log:</p> <pre style="border: 1px dashed blue; padding: 5px;">2014-10-16 18:36:29,112 WARN org.apache.hadoop.hdfs.server.namenode.NameNode Encountered exception loading fsimage java.io.IOException:File system image contains an old layout version -55.An upgrade to version -59 is required. Please restart NameNode with the "-rollingUpgrade started" option if a rolling upgrade is already started; or restart NameNode with the "-upgrade" option to start a new upgrade. at org.apache.hadoop.hdfs.server.namenode.NameNode at org.apache.hadoop.hdfs.server.namenode.NameNode at org.apache.hadoop.hdfs.server.namenode.NameNode at org.apache.hadoop.hdfs.server.namenode.NameNode at org.apache.hadoop.hdfs.server.namenode.NameNode at org.apache.hadoop.hdfs.server.namenode.NameNode at org.apache.hadoop.hdfs.server.namenode.NameNode</pre>	<p>You upgraded CDH to 5.2 using Cloudera Manager and did not run the HDFS Metadata Upgrade command.</p>	<p>Stop the HDFS service in Cloudera Manager and follow the steps for upgrade (depending on whether you are using packages or parcels) described in Upgrading to CDH 5.2.</p>

Symptom	Reason	Solution
<pre>at org.apache.hadoop.hdfs.NameNode-1 (NameNode) at org.apache.hadoop.hdfs.NameNode-1 (NameNode) at org.apache.hadoop.hdfs.NameNode-1 (NameNode) 2014-10-16 18:36:29,126 INFO org.mortbay.log: Stopped HttpServer2 at org.apache.hadoop.hdfs.NameNode-1 (NameNode) 2014-10-16 18:36:29,127 WARN org.apache.hadoop.http.HttpServer2: HttpServer Acceptor: isRunning is false. Rechecking. 2014-10-16 18:36:29,127 WARN org.apache.hadoop.http.HttpServer2: HttpServer Acceptor: isRunning is false 2014-10-16 18:36:29,127 INFO org.apache.hadoop.metrics2.impl.MetricsSystem: Stopping NameNode metrics system... 2014-10-16 18:36:29,128 INFO org.apache.hadoop.metrics2.impl.MetricsSystem: NameNode metrics system stopped. 2014-10-16 18:36:29,128 INFO org.apache.hadoop.metrics2.impl.MetricsSystem: NameNode metrics system shutdown complete. 2014-10-16 18:36:29,128 FATAL org.apache.hadoop.hdfs.server.namenode.NameNode: Exception in namenode join java.io.IOException: File system image contains an old layout version -55.An upgrade to version -59 is required. Please restart NameNode with the "-rollingUpgrade started" option if a rolling upgrade is already started; or restart NameNode with the "-upgrade" option to start a new upgrade. at org.apache.hadoop.hdfs.server.namenode.NameNode (NameNode)</pre>		

Troubleshooting Installation and Upgrade Problems

Symptom	Reason	Solution
<pre> at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: at org.apache.hadoop.util.ExitUtil: 2014-10-16 18:36:29,130 INFO org.apache.hadoop.util.ExitUtil: Exiting with status 1 2014-10-16 18:36:29,132 INFO org.apache.hadoop.util.ExitUtil: SHUTDOWN_MSG: </pre>		

Appendix: Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```