

# Cloudera Installation and Upgrade



**Important Notice**

© 2010-2016 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, Cloudera Impala, 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.

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.**  
**1001 Page Mill Road, Bldg 3**  
**Palo Alto, CA 94304**  
**[info@cloudera.com](mailto:info@cloudera.com)**  
**US: 1-888-789-1488**  
**Intl: 1-650-362-0488**  
**[www.cloudera.com](http://www.cloudera.com)**

**Release Information**

Version: Cloudera Enterprise 5.7.x  
Date: August 27, 2016

# Table of Contents

<b>About Cloudera Installation and Upgrade.....</b>	<b>9</b>
 <b>Installation Requirements for Cloudera Manager, Cloudera Navigator, and CDH</b>	
<b>5.....</b>	<b>10</b>
Cloudera Manager 5 Requirements and Supported Versions.....	10
<i>Supported Operating Systems.....</i>	10
<i>Supported JDK Versions.....</i>	11
<i>Supported Browsers.....</i>	11
<i>Supported Databases.....</i>	12
<i>Supported CDH and Managed Service Versions.....</i>	12
<i>Supported Transport Layer Security Versions.....</i>	13
<i>Resource Requirements.....</i>	13
<i>Networking and Security Requirements.....</i>	14
<i>Single User Mode Requirements.....</i>	17
Permission Requirements for Package-based Installations and Upgrades of CDH.....	26
Cloudera Navigator 2 Requirements and Supported Versions.....	28
<i>Cloudera Manager Requirements.....</i>	28
<i>Supported Databases.....</i>	28
<i>Supported Browsers.....</i>	29
<i>Supported CDH and Managed Service Versions.....</i>	29
CDH 5 Requirements and Supported Versions.....	31
<i>Supported Operating Systems.....</i>	31
<i>Supported Databases.....</i>	32
<i>Supported JDK Versions.....</i>	33
<i>Supported Browsers.....</i>	34
<i>Supported Network Protocols.....</i>	34
<i>Supported Transport Layer Security Versions.....</i>	34
Supported Configurations with Virtualization and Cloud Platforms.....	35
<i>Amazon Web Services.....</i>	35
<i>Google Cloud Platform.....</i>	35
<i>Microsoft Azure.....</i>	35
<i>VMware.....</i>	35
Filesystem Requirements.....	36
<i>Supported Filesystems.....</i>	36
<i>File Access Time.....</i>	36
Ports.....	36
<i>Ports Used by Cloudera Manager and Cloudera Navigator.....</i>	36

<i>Ports Used by Cloudera Navigator Encryption</i> .....	40
<i>Ports Used by Components of CDH 5</i> .....	40
<i>Ports Used by Components of CDH 4</i> .....	46
<i>Ports Used by Impala</i> .....	51
<i>Ports Used by Cloudera Search</i> .....	52
<i>Ports Used by DistCp</i> .....	52
<i>Ports Used by Third-Party Components</i> .....	53

## **Managing Software Installation Using Cloudera Manager.....55**

<i>Parcels</i> .....	55
<i>Advantages of Parcels</i> .....	55
<i>Parcel Life Cycle</i> .....	56
<i>Parcel Locations</i> .....	57
<i>Managing Parcels</i> .....	57
<i>Viewing Parcel Usage</i> .....	60
<i>Parcel Configuration Settings</i> .....	63
Migrating from Packages to Parcels.....	65
Migrating from Parcels to Packages.....	67
<i>Install CDH and Managed Service Packages</i> .....	67
<i>Deactivate Parcels</i> .....	72
<i>Restart the Cluster</i> .....	73
<i>Remove and Delete Parcels</i> .....	73

## **Installation Overview.....74**

<i>Cloudera Manager Deployment</i> .....	74
<i>Cloudera Manager Installation Phases</i> .....	75
<i>Cloudera Manager Installation Software</i> .....	76
<i>Unmanaged Deployment</i> .....	77
<i>Java Development Kit Installation</i> .....	78
<i>Installing the Oracle JDK</i> .....	78
<i>Cloudera Manager and Managed Service Datastores</i> .....	79
<i>Required Databases</i> .....	79
<i>Setting up the Cloudera Manager Server Database</i> .....	80
<i>External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server</i> .....	83
<i>External Databases for Hue</i> .....	83
<i>Embedded PostgreSQL Database</i> .....	83
<i>External PostgreSQL Database</i> .....	86
<i>MariaDB Database</i> .....	92
<i>MySQL Database</i> .....	98
<i>Oracle Database</i> .....	105
<i>Configuring an External Database for Oozie</i> .....	113

<i>Configuring an External Database for Sqoop</i>	116
<i>Backing Up Databases</i>	117
<i>Data Storage for Monitoring Data</i>	118
<i>Storage Space Planning for Cloudera Manager</i>	121
<b>Installation Path A - Automated Installation by Cloudera Manager (Non-Production Mode)</b>	132
<i>Before You Begin</i>	133
<i>Download and Run the Cloudera Manager Server Installer</i>	134
<i>Start and Log into the Cloudera Manager Admin Console</i>	135
<i>Use the Cloudera Manager Wizard for Software Installation and Configuration</i>	135
<i>Configure Database Settings</i>	139
<i>Review Configuration Changes and Start Services</i>	140
<i>Change the Default Administrator Password</i>	141
<i>Configure Oozie Data Purge Settings</i>	141
<i>Test the Installation</i>	141
<b>Installation Path B - Installation Using Cloudera Manager Parcels or Packages</b>	141
<i>Before You Begin</i>	141
<i>Establish Your Cloudera Manager Repository Strategy</i>	142
<i>Install Cloudera Manager Server Software</i>	143
<i>(Optional) Manually Install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Service Packages</i>	144
<i>Start the Cloudera Manager Server</i>	145
<i>Start and Log into the Cloudera Manager Admin Console</i>	146
<i>Choose Cloudera Manager Edition</i>	146
<i>Choose Cloudera Manager Hosts</i>	146
<i>Choose the Software Installation Type and Install Software</i>	147
<i>Add Services</i>	149
<i>Configure Database Settings</i>	150
<i>Review Configuration Changes and Start Services</i>	150
<i>Change the Default Administrator Password</i>	151
<i>Configure Oozie Data Purge Settings</i>	151
<i>Test the Installation</i>	151
<i>(Optional) Manually Install CDH and Managed Service Packages</i>	151
<b>Installation Path C - Manual Installation Using Cloudera Manager Tarballs</b>	157
<i>Before You Begin</i>	157
<i>Install the Cloudera Manager Server and Agents</i>	157
<i>Create Parcel Directories</i>	159
<i>Start the Cloudera Manager Server</i>	160
<i>Start the Cloudera Manager Agents</i>	161
<i>Install Dependencies</i>	162
<i>Start and Log into the Cloudera Manager Admin Console</i>	163
<i>Choose Cloudera Manager Edition</i>	163
<i>Choose Cloudera Manager Hosts</i>	164
<i>Install CDH and Managed Service Software</i>	164
<i>Add Services</i>	165
<i>Configure Database Settings</i>	166

<i>Review Configuration Changes and Start Services</i> .....	166
<i>(Optional) Change the Cloudera Manager User</i> .....	167
<i>Change the Default Administrator Password</i> .....	167
<i>Configure Oozie Data Purge Settings</i> .....	167
<i>Test the Installation</i> .....	167
<i>Installing Impala</i> .....	167
<i>Installing Search</i> .....	168
<i>Installing Spark</i> .....	168
<i>Installing the GPL Extras Parcel</i> .....	169
<i>Understanding Custom Installation Solutions</i> .....	170
<i>Understanding Parcels</i> .....	170
<i>Understanding Package Management</i> .....	170
<i>Creating and Using a Remote Parcel Repository for Cloudera Manager</i> .....	172
<i>Creating and Using a Package Repository for Cloudera Manager</i> .....	174
<i>Configuring a Custom Java Home Location</i> .....	176
<i>Installing Older Versions of Cloudera Manager 5</i> .....	177
<i>Creating a CDH Cluster Using a Cloudera Manager Template</i> .....	192
<i>Deploying Clients</i> .....	198
<i>Testing the Installation</i> .....	198
<i>Checking Host Heartbeats</i> .....	198
<i>Running a MapReduce Job</i> .....	199
<i>Testing with Hue</i> .....	199
<i>Uninstalling Cloudera Manager and Managed Software</i> .....	199
<i>Uninstalling Cloudera Manager and Managed Software</i> .....	199
<i>Uninstalling a CDH Component From a Single Host</i> .....	204
<i>Installing the Cloudera Navigator Data Management Component</i> .....	204
<i>Installing Cloudera Navigator Key Trustee Server</i> .....	205
<i>Prerequisites</i> .....	205
<i>Setting Up an Internal Repository</i> .....	206
<i>Installing Key Trustee Server</i> .....	206
<i>Securing Key Trustee Server Host</i> .....	208
<i>Leveraging Native Processor Instruction Sets</i> .....	209
<i>Initializing Key Trustee Server</i> .....	210
<i>Installing Cloudera Navigator Key HSM</i> .....	210
<i>Prerequisites</i> .....	211
<i>Setting Up an Internal Repository</i> .....	211
<i>Installing Navigator Key HSM</i> .....	211
<i>Installing Key Trustee KMS</i> .....	211
<i>Setting Up an Internal Repository</i> .....	212
<i>Installing Key Trustee KMS Using Parcels</i> .....	212
<i>Installing Key Trustee KMS Using Packages</i> .....	212
<i>Post-Installation Configuration</i> .....	212
<i>Installing Cloudera Navigator Encrypt</i> .....	212

<i>Prerequisites</i>	213
<i>Setting Up an Internal Repository</i>	213
<i>Installing Navigator Encrypt (RHEL-Compatible)</i>	213
<i>Installing Navigator Encrypt (SLES)</i>	214
<i>Installing Navigator Encrypt (Debian or Ubuntu)</i>	215
<i>Post Installation</i>	216
<i>AES-NI and RDRAND</i>	216
<i>Uninstalling and Reinstalling Navigator Encrypt</i>	217
<i>Installing and Deploying CDH Using the Command Line</i>	218
<i>Before You Install CDH 5 on a Cluster</i>	218
<i>Creating a Local Yum Repository</i>	219
<i>Installing the Latest CDH 5 Release</i>	220
<i>Installing an Earlier CDH 5 Release</i>	230
<i>CDH 5 and MapReduce</i>	233
<i>Migrating from MapReduce (MRv1) to MapReduce (MRv2)</i>	234
<i>Deploying CDH 5 on a Cluster</i>	246
<i>Installing CDH 5 Components</i>	270
<i>Building RPMs from CDH Source RPMs</i>	461
<i>Apache and Third-Party Licenses</i>	461
<i>Uninstalling CDH Components</i>	462
<i>Viewing the Apache Hadoop Documentation</i>	465

## **Upgrade.....466**

<i>Upgrading Cloudera Manager</i>	466
<i>Database Considerations for Cloudera Manager Upgrades</i>	467
<i>Upgrading Cloudera Manager 5 to the Latest Cloudera Manager</i>	469
<i>Upgrading Cloudera Manager 4 to Cloudera Manager 5</i>	481
<i>Upgrading Cloudera Manager 3.7.x</i>	500
<i>Re-Running the Cloudera Manager Upgrade Wizard</i>	500
<i>Reverting a Failed Cloudera Manager Upgrade</i>	500
<i>Upgrading the Cloudera Navigator Data Management Component</i>	503
<i>Upgrading Cloudera Navigator Key Trustee Server</i>	504
<i>Upgrading Cloudera Navigator Key Trustee Server 3.x to 5.4.x</i>	504
<i>Upgrading Cloudera Navigator Key Trustee Server 3.8 to 5.5 Using the ktupgrade Script</i>	509
<i>Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x to the Latest Release</i>	514
<i>Upgrading Cloudera Navigator Key HSM</i>	519
<i>Setting Up an Internal Repository</i>	519
<i>Upgrading Key HSM</i>	519
<i>Upgrading Key Trustee KMS</i>	520
<i>Setting Up an Internal Repository</i>	520
<i>Upgrading Key Trustee KMS Using Parcels</i>	520
<i>Upgrading Key Trustee KMS Using Packages</i>	521
<i>Upgrading Cloudera Navigator Encrypt</i>	521

<i>Setting Up an Internal Repository</i> .....	521
<i>Upgrading Navigator Encrypt (RHEL-Compatible)</i> .....	521
<i>Upgrading Navigator Encrypt (SLES)</i> .....	522
<i>Upgrading Navigator Encrypt (Debian or Ubuntu)</i> .....	523
<i>Best Practices for Upgrading Navigator Encrypt Hosts</i> .....	523
<i>Upgrading CDH and Managed Services Using Cloudera Manager</i> .....	524
<i>Configuring the CDH Version of a Cluster</i> .....	524
<i>Performing a Rolling Upgrade on a CDH 5 Cluster</i> .....	525
<i>Performing a Rolling Upgrade on a CDH 4 Cluster</i> .....	554
<i>Upgrading to CDH Maintenance Releases</i> .....	557
<i>Upgrading to CDH 5.7</i> .....	566
<i>Upgrading to CDH 5.6</i> .....	580
<i>Upgrading to CDH 5.5</i> .....	594
<i>Upgrading to CDH 5.4</i> .....	609
<i>Upgrading to CDH 5.3</i> .....	623
<i>Upgrading to CDH 5.2</i> .....	637
<i>Upgrading to CDH 5.1</i> .....	650
<i>Upgrading CDH 4 to CDH 5</i> .....	660
<i>Upgrading CDH 4</i> .....	681
<i>Upgrading CDH 3</i> .....	690
<i>Upgrading Unmanaged CDH Using the Command Line</i> .....	690
<i>Upgrading from CDH 4 to CDH 5</i> .....	691
<i>Upgrading from an Earlier CDH 5 Release to the Latest Release</i> .....	708
<i>Upgrading to Oracle JDK 1.7</i> .....	738
<i>Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment</i> .....	739
<i>Upgrading to Oracle JDK 1.7 in an Unmanaged Deployment</i> .....	739
<i>Upgrading to Oracle JDK 1.8</i> .....	740
<i>Upgrading to Oracle JDK 1.8 in a Cloudera Manager Deployment</i> .....	740
<i>Upgrading to Oracle JDK 1.8 in an Unmanaged Deployment</i> .....	741
<i>If you are Using AES-256 Encryption, install the JCE Policy File</i> .....	741
<b>Troubleshooting Installation and Upgrade Problems</b> .....	<b>742</b>
<b>Rolling Back a CDH 4-to-CDH 5 Upgrade</b> .....	<b>750</b>
<i>Backing Up Before Upgrading from CDH 4 to CDH 5</i> .....	750
<i>Backup Steps</i> .....	751
<i>Procedure for Rolling Back a CDH 4-to-CDH 5 Upgrade</i> .....	754
<i>Rollback Steps</i> .....	754

## About Cloudera Installation and Upgrade

This guide provides Cloudera software requirements and installation information for production deployments, as well as upgrade procedures. 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

This page describes requirements and supported third-party software for the latest version of Cloudera Manager. Specifically, it lists supported operating systems, browsers, and databases; and it explains which versions of TLS are supported by various components and 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



**Note:** 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.

Cloudera Manager supports the following 64-bit operating systems:

- **RHEL-compatible**
  - Red Hat Enterprise Linux and CentOS, 64-bit (+ SELinux mode in available versions)
    - 5.7
    - 5.10
    - 6.4
    - 6.5
    - 6.6
    - 6.7
    - 7.1
    - 7.2
  - Oracle Enterprise (OEL) Linux with Unbreakable Enterprise Kernel (UEK), 64-bit
    - 5.7 (UEK R2)
    - 5.10
    - 5.11
    - 6.4 (UEK R2)
    - 6.5 (UEK R2, UEK R3)
    - 6.6 (UEK R3)
    - 6.7 (UEK R3)
    - 7.1
    - 7.2



**Important:** Cloudera supports RHEL 7 with the following limitations:

- Only RHEL 7.2 and 7.1 are supported. RHEL 7.0 is not supported.
- Only new installations of RHEL 7.2 and 7.1 are supported by Cloudera. For upgrades to RHEL 7.1 or 7.2, contact your OS vendor and see [Does Red Hat support upgrades between major versions of Red Hat Enterprise Linux?](#)

- **SLES** - SUSE Linux Enterprise Server 11, Service Pack 4, 64-bit is supported by CDH 5.7 and higher. Service Packs 2 and 3 are supported by CDH 5.0 through CDH 5.6. Service Pack 1 is *not* supported by CDH 5, only by CDH 4. Hosts running Cloudera Manager Agents must use [SUSE Linux Enterprise Software Development Kit 11 SP1](#).
- **Debian** - Wheezy 7.0, 7.1, and 7.8, 64-bit. (Squeeze 6.0 is only supported by CDH 4.)
- **Ubuntu** - Trusty 14.04 (LTS) and Precise 12.04 (LTS), 64-bit. (Lucid 10.04 is only supported by CDH 4.)



**Note:** Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled. Cloudera is not responsible for policy support nor policy enforcement. If you experience issues with SELinux, contact your OS provider.

## Supported JDK Versions

The version of Oracle JDK supported by Cloudera Manager depends on the version of CDH being managed. The following table lists the JDK versions supported on a Cloudera Manager 5.7 cluster running the latest CDH 4 and CDH 5. For more information on supported JDK versions for previous versions of Cloudera Manager and CDH, see [Compatibility](#).



**Important:** There is one exception to the minimum supported and recommended JDK versions listed below. 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

CDH Version Managed (Latest)	Minimum Supported JDK Version	Recommended JDK Version
<b>CDH 5</b>	1.7.0_55	1.7.0_67, 1.7.0_75, 1.7.0_80
	1.8.0_31  Cloudera recommends that you <i>not</i> use JDK 1.8.0_40.	1.8.0_60
<b>CDH 4 and CDH 5</b>	1.7.0_55	1.7.0_67, 1.7.0_75, 1.7.0_80
	1.8.0_31	1.8.0_60
<b>CDH 4</b>	1.6.0_31	1.7.0_80

Cloudera Manager can install Oracle JDK 1.7.0\_67 during installation and upgrade. If you prefer to install the JDK yourself, follow the instructions in [Java Development Kit Installation](#) on page 78.

## 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 36 and higher.
- Internet Explorer 9 and higher. Internet Explorer 11 Native Mode.
- Safari 5 and higher.

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

### 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 roles.



**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.

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 \(Non-Production Mode\)](#) on page 132 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 Datastores](#) on page 79.

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:

- MariaDB 5.5
- MySQL - 5.1, 5.5, 5.6, and 5.7
- Oracle 11gR2 and 12c



**Note:** When installing a JDBC driver, only the `ojdbc6.jar` file is supported for both Oracle 11g R2 and Oracle 12c; the `ojdbc7.jar` file is not supported.

- PostgreSQL - 8.1, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4

Cloudera supports the shipped version of MariaDB, 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 31.
- **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](#).

## Supported Transport Layer Security Versions

The following components are supported by Transport Layer Security (TLS):

**Table 1: Components Supported by TLS**

Component	Role	Port	Version
Cloudera Manager	Cloudera Manager Server	7182	TLS 1.2
Cloudera Manager	Cloudera Manager Server	7183	TLS 1.2
Flume		9099	TLS 1.2
HBase	Master	60010	TLS 1.2
HDFS	NameNode	50470	TLS 1.2
HDFS	Secondary NameNode	50495	TLS 1.2
Hive	HiveServer2	10000	TLS 1.2
Hue	Hue Server	8888	TLS 1.2
Cloudera Impala	Impala Daemon	21000	TLS 1.2
Cloudera Impala	Impala Daemon	21050	TLS 1.2
Cloudera Impala	Impala Daemon	22000	TLS 1.2
Cloudera Impala	Impala Daemon	25000	TLS 1.2
Cloudera Impala	Impala StateStore	24000	TLS 1.2
Cloudera Impala	Impala StateStore	25010	TLS 1.2
Cloudera Impala	Impala Catalog Server	25020	TLS 1.2
Cloudera Impala	Impala Catalog Server	26000	TLS 1.2
Oozie	Oozie Server	11443	TLS 1.1
Solr	Solr Server	8983	TLS 1.1
Solr	Solr Server	8985	TLS 1.1
YARN	ResourceManager	8090	TLS 1.2
YARN	JobHistory Server	19890	TLS 1.2

To configure TLS security for the Cloudera Manager Server and Agents, see [Configuring TLS Security for Cloudera Manager](#).

## 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.

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

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 118.
  - **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.  
Python 3.x is not supported.
  - **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 26.

- 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).



**Note:** Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled. Cloudera is not responsible for policy support nor policy enforcement. If you experience issues with SELinux, contact your OS provider.

- 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 RHEL 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 2: 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 #ccc; padding: 5px; margin-top: 10px;"> <b>Note:</b> Applicable to clusters managed by Cloudera Manager only.         </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.

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

Component (Version)	Unix User ID	Groups	Notes
Apache Hive (CDH 4, CDH 5)	hive	hive	The HiveServer2 process and the Hive Metastore processes run as this user.  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 javax.jdo.option.ConnectionUserName in <code>hive-site.xml</code> .
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 <a href="#">HttpFS Security Configuration</a> 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, hive	Impala services run as this user.
Apache Kafka (Cloudera Distribution of Kafka 1.2.0)	kafka	kafka	Kafka services run as this user.
Java KeyStore KMS (CDH 5.2.1 and higher)	kms	kms	The Java KeyStore KMS service runs as this user.
Key Trustee KMS (CDH 5.3 and higher)	kms	kms	The Key Trustee KMS service runs as this user.
Key Trustee Server (CDH 5.4 and higher)	keytrustee	keytrustee	The Key Trustee Server service runs as this user.
Kudu	kudu	kudu	Kudu services run 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.

Component (Version)	Unix User ID	Groups	Notes
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 (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 and higher 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.

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 17.

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.
- [Cloudera Navigator data encryption](#) components are 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:

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

- /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. Select **Category > Parcels**.
  4. 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.
  5. Click **Save Changes** to commit the changes.
  6. [Restart](#) the Cloudera Manager Agent on all hosts.

### 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:

- Edit the `/etc/sudoers` file using the `visudo` command
- 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.

- Edit `/etc/pam.d/su`.
- 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-hdfs`
  - 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:

- 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:
    - Go to **Administration > Settings > Advanced**.
    - Set the **Cloudera Manager Server Local Data Storage Directory** property to a directory owned by the single user.
    - Click **Save Changes** to commit the changes.
- Path B and C when using already managed hosts** - Configure single user mode:
  - Go to **Administration > Settings > Advanced**.
  - Check the **Single User Mode** checkbox.
  - 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`.

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

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 [TLS/SSL encryption](#).
- Do not configure the DataNode Transceiver port and HTTP Web UI port to use privileged ports.
- Configure DataNode data transfer protection.

### Controlling Access to sudo Commands

To comply with security requirements, you may need to control access to the sudo commands run by Cloudera Manager Agents. You can control access by creating a “whitelist” of sudo commands that the Cloudera Manager Agent runs, or you can override the `sudo` command so that a custom script that denies some actions is called instead.

Cloudera Manager Agents use `sudo` for the following regular operations:

- Running the `update-alternatives` command during upgrades and when updating parcels.
- Adding new roles or services that require `sudo` access to the `chown` and `chmod` commands.
- Running the `mount` and `umount` commands when performing resource management activities that use [Linux Control Groups \(cgroups\)](#) and mounting a `tmpfs` mount point for temporary directories for [cm\\_processes](#).
- [Collecting diagnostics](#), which requires reading files from the `/proc` and `/etc` directories and distribution-specific networking configuration files.

### Whitelisting sudo Commands

The following commands may need to be whitelisted for the `cloudera-scm-agent` user. This can be either the default user, `cloudera-scm`, or a [single user you specify](#). Use Linux tools to manage access to these commands as required for your deployment. (See the `man` page for `sudoers`.)

### Regular Operation Commands

- `cat`
- `chmod`
- `chown`
- `cp`
- `mkdir`
- `mount`
- `rm`
- `umount`
- `update-alternatives`

### Support Bundle Collection Commands

- `bash`
- `cat`
- `date`
- `df`

- dmesg
- grep
- hostname
- ls
- netstat
- ps
- rpm
- uname
- chkconfig
- ethtool
- ifconfig
- iptables
- lsmod
- lspci
- lvdisplay
- service
- sysctl
- curl
- dig
- host
- lsb\_release
- lscpu
- nslookup
- ntpstat
- python
- sar
- top
- uptime
- vmstat
- dmidecode
- lsof
- ntpq

### Overriding the sudo Command

You can override the `sudo` command so that a custom script is called instead. This script can deny some actions.

To configure the location of this script:

1. Edit the `/etc/cloudera-scm-agent/config.ini` file on all cluster hosts and add the following line:

```
sudo_command=path_to_script
```

2. Restart the Cloudera Manager Agent on all cluster hosts:

```
service cloudera-scm-agent restart
```

To help determine which commands to override, see the following samples of typical commands run by Cloudera Manager Agent.

Commands run by the Cloudera Manager Agent while it brings up roles for new services on a single host:

```
/bin/cat /proc/cgroups
/bin/chmod -R ugo+r /etc/accumulo/*
/bin/chmod -R ugo+r /etc/hadoop/*
```

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

```
/bin/chmod -R ugo+r /etc/hbase/*
/bin/chmod -R ugo+r /etc/hive/*
/bin/chmod -R ugo+r /etc/solr/*
/bin/chmod -R ugo+r /etc/spark/*
/bin/chmod -R ugo+r /etc/sqoop/*
/bin/chmod 0644 /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* topology.map
/bin/chmod 0755 /ngs/app/searchhp/opt/parcels/CDH*/lib/hue/desktop
/bin/chmod 0755 /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* topology.py
/bin/chmod 4754
/ngs/app/searchhp/opt/parcels/CDH*/lib/hadoop-0.20-mapreduce/sbin/Linux/task-controller
/bin/chmod 6050 /ngs/app/searchhp/opt/parcels/CDH*/lib/hadoop-yarn/bin/container-executor
/bin/chown -R cloudera-scm:cloudera-scm /tmp/*
/bin/chown searchhp:searchhp /etc/hadoop/* topology.map
/bin/chown searchhp:searchhp /etc/hadoop/* topology.py
/bin/chown searchhp:searchhp /ngs/app/searchhp/var/run/searchhp-agent/process/* topology.map
/bin/chown searchhp:searchhp /ngs/app/searchhp/var/run/searchhp-agent/process/* topology.py
/bin/chown root /etc/accumulo/*
/bin/chown root /etc/hadoop/*
/bin/chown root /etc/hbase/*
/bin/chown root /etc/hive/*
/bin/chown root /etc/solr/*
/bin/chown root /etc/spark/*
/bin/chown root /etc/sqoop/*
/bin/chown root
/ngs/app/searchhp/opt/parcels/CDH*/lib/hadoop-0.20-mapreduce/sbin/Linux/task-controller
/bin/chown root /ngs/app/searchhp/opt/parcels/CDH*/lib/hadoop-yarn/bin/container-executor
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* accumulo-conf
/etc/accumulo/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hadoop-conf /etc/hadoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* yarn-conf /etc/hadoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hadoop-conf /etc/hadoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* yarn-conf /etc/hadoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hadoop-conf /etc/hadoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hadoop-conf /etc/hadoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hbase-conf /etc/hbase/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hbase-conf /etc/hbase/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hive-conf /etc/hive/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* hive-conf /etc/hive/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* solr-conf /etc/solr/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* solr-conf /etc/solr/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* spark-conf /etc/spark/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* spark-conf /etc/spark/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* sqoop-conf /etc/sqoop/*
/bin/cp -a /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* sqoop-conf /etc/sqoop/*
/bin/cp -p /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* topology.map
/etc/hadoop/* topology.map
/bin/cp -p /ngs/app/searchhp/var/run/cloudera-scm-agent/process/* topology.py
/etc/hadoop/* topology.py
/bin/mkdir -p /etc/accumulo
/bin/mkdir -p /etc/flume-ng
/bin/mkdir -p /etc/hadoop
/bin/mkdir -p /etc/hadoop-httpfs
/bin/mkdir -p /etc/hadoop-kms
/bin/mkdir -p /etc/hbase
/bin/mkdir -p /etc/hbase-solr
/bin/mkdir -p /etc/hive
/bin/mkdir -p /etc/hive-hcatalog
/bin/mkdir -p /etc/hive-webhcatt
/bin/mkdir -p /etc/hue
/bin/mkdir -p /etc/impala
/bin/mkdir -p /etc/llama
/bin/mkdir -p /etc/mahout
/bin/mkdir -p /etc/oozie
/bin/mkdir -p /etc/pig
/bin/mkdir -p /etc/sentry
/bin/mkdir -p /etc/solr
/bin/mkdir -p /etc/spark
/bin/mkdir -p /etc/sqoop
/bin/mkdir -p /etc/sqoop2
/bin/mkdir -p /etc/zookeeper
/bin/mount -t cgroup -o blkio cm_cgroups /tmp/*
/bin/mount -t cgroup -o cpu cm_cgroups /tmp/*
/bin/mount -t cgroup -o cpuacct cm_cgroups /tmp/*
```

```

/bin/mount -t cgroup -o memory cm_cgroups /tmp/*
/bin/mount -t tmpfs cm_processes /ngs/app/searchp/var/run/cloudera-scm-agent/process -o
mode
/bin/rm
/bin/rm -rf /etc/accumulo/*
/bin/rm -rf /etc/hadoop/*
/bin/rm -rf /etc/hbase/*
/bin/rm -rf /etc/hive/*
/bin/rm -rf /etc/solr/*
/bin/rm -rf /etc/spark/*
/bin/rm -rf /etc/sqoop/*
/bin/umount /tmp/*
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
--display ip6tables.x86_64
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
--display iptables.x86_64
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
--display mta
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
--display print
/usr/sbin/update-alternatives --auto accumulo-conf
/usr/sbin/update-alternatives --auto hadoop-conf
/usr/sbin/update-alternatives --auto hbase-conf
/usr/sbin/update-alternatives --auto hive-conf
/usr/sbin/update-alternatives --auto solr-conf
/usr/sbin/update-alternatives --auto spark-conf
/usr/sbin/update-alternatives --auto sqoop-conf
/usr/sbin/update-alternatives --install /etc/accumulo/conf accumulo-conf /etc/accumulo/*
51
/usr/sbin/update-alternatives --install /etc/flume-ng/conf flume-ng-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/flume-ng/conf.empty 10
/usr/sbin/update-alternatives --install /etc/hadoop-httpfs/conf hadoop-httpfs-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hadoop-httpfs/conf.empty 10
/usr/sbin/update-alternatives --install /etc/hadoop-kms/conf hadoop-kms-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hadoop-kms/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/* 90
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/* 91
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/* 92
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hadoop/conf.empty 10
/usr/sbin/update-alternatives --install /etc/hbase-solr/conf hbase-solr-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hbase-solr/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hbase/conf hbase-conf /etc/hbase/* 90
/usr/sbin/update-alternatives --install /etc/hbase/conf hbase-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hbase/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hive-hcatalog/conf hive-hcatalog-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hive-hcatalog/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hive-webhcat/conf hive-webhcat-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hive-webhcat/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hive/conf hive-conf /etc/hive/* 90
/usr/sbin/update-alternatives --install /etc/hive/conf hive-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hive/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hue/conf hue-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/hue/conf.empty 10
/usr/sbin/update-alternatives --install /etc/impala/conf impala-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/impala/conf.dist 10
/usr/sbin/update-alternatives --install /etc/llama/conf llama-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/llama/conf.dist 10
/usr/sbin/update-alternatives --install /etc/mahout/conf mahout-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/mahout/conf.dist 10
/usr/sbin/update-alternatives --install /etc/oozie/conf oozie-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/oozie/conf.dist 10
/usr/sbin/update-alternatives --install /etc/pig/conf pig-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/pig/conf.dist 10
/usr/sbin/update-alternatives --install /etc/sentry/conf sentry-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/sentry/conf.dist 10
/usr/sbin/update-alternatives --install /etc/solr/conf solr-conf /etc/solr/* 90
/usr/sbin/update-alternatives --install /etc/solr/conf solr-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/solr/conf.dist 10
/usr/sbin/update-alternatives --install /etc/spark/conf spark-conf /etc/spark/* 51
/usr/sbin/update-alternatives --install /etc/spark/conf spark-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/spark/conf.dist 10
/usr/sbin/update-alternatives --install /etc/sqoop/conf sqoop-conf /etc/sqoop/* 50

```

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

```
/usr/sbin/update-alternatives --install /etc/sqoop/conf sqoop-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/sqoop/conf.dist 10
/usr/sbin/update-alternatives --install /etc/sqoop2/conf sqoop2-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/sqoop2/conf.dist 10
/usr/sbin/update-alternatives --install /etc/zookeeper/conf zookeeper-conf
/ngs/app/searchp/opt/parcels/CDH*/etc/zookeeper/conf.dist 10
/usr/sbin/update-alternatives --install /usr/bin/accumulo accumulo
/ngs/app/searchp/opt/parcels/ACCUMULO-1.6.0-1.cdh5.1.0.p0.51/bin/accumulo 10
/usr/sbin/update-alternatives --install /usr/bin/accumulo-tool accumulo-tool
/ngs/app/searchp/opt/parcels/ACCUMULO-1.6.0-1.cdh5.1.0.p0.51/bin/accumulo-tool 10
/usr/sbin/update-alternatives --install /usr/bin/avro-tools avro-tools
/ngs/app/searchp/opt/parcels/CDH*/bin/avro-tools 10
/usr/sbin/update-alternatives --install /usr/bin/beeline beeline
/ngs/app/searchp/opt/parcels/CDH*/bin/beeline 10
/usr/sbin/update-alternatives --install /usr/bin/catalogd catalogd
/ngs/app/searchp/opt/parcels/CDH*/bin/catalogd 10
/usr/sbin/update-alternatives --install /usr/bin/cli_mt cli_mt
/ngs/app/searchp/opt/parcels/CDH*/bin/cli_mt 10
/usr/sbin/update-alternatives --install /usr/bin/cli_st cli_st
/ngs/app/searchp/opt/parcels/CDH*/bin/cli_st 10
/usr/sbin/update-alternatives --install /usr/bin/flume-ng flume-ng
/ngs/app/searchp/opt/parcels/CDH*/bin/flume-ng 10
/usr/sbin/update-alternatives --install /usr/bin/hadoop hadoop
/ngs/app/searchp/opt/parcels/CDH*/bin/hadoop 10
/usr/sbin/update-alternatives --install /usr/bin/hadoop-0.20 hadoop-0.20
/ngs/app/searchp/opt/parcels/CDH*/bin/hadoop-0.20 10
/usr/sbin/update-alternatives --install /usr/bin/hadoop-fuse-dfs hadoop-fuse-dfs
/ngs/app/searchp/opt/parcels/CDH*/bin/hadoop-fuse-dfs 10
/usr/sbin/update-alternatives --install /usr/bin/hbase hbase
/ngs/app/searchp/opt/parcels/CDH*/bin/hbase 10
/usr/sbin/update-alternatives --install /usr/bin/hbase-indexer hbase-indexer
/ngs/app/searchp/opt/parcels/CDH*/bin/hbase-indexer 10
/usr/sbin/update-alternatives --install /usr/bin/hcat hcat
/ngs/app/searchp/opt/parcels/CDH*/bin/hcat 10
/usr/sbin/update-alternatives --install /usr/bin/hdfs hdfs
/ngs/app/searchp/opt/parcels/CDH*/bin/hdfs 10
/usr/sbin/update-alternatives --install /usr/bin/hive hive
/ngs/app/searchp/opt/parcels/CDH*/bin/hive 10
/usr/sbin/update-alternatives --install /usr/bin/hiveserver2 hiveserver2
/ngs/app/searchp/opt/parcels/CDH*/bin/hiveserver2 10
/usr/sbin/update-alternatives --install /usr/bin/impala-shell impala-shell
/ngs/app/searchp/opt/parcels/CDH*/bin/impala-shell 10
/usr/sbin/update-alternatives --install /usr/bin/impalad impalad
/ngs/app/searchp/opt/parcels/CDH*/bin/impalad 10
/usr/sbin/update-alternatives --install /usr/bin/kite-dataset kite-dataset
/ngs/app/searchp/opt/parcels/CDH*/bin/kite-dataset 10
/usr/sbin/update-alternatives --install /usr/bin/llama llama
/ngs/app/searchp/opt/parcels/CDH*/bin/llama 10
/usr/sbin/update-alternatives --install /usr/bin/llamaadmin llamaadmin
/ngs/app/searchp/opt/parcels/CDH*/bin/llamaadmin 10
/usr/sbin/update-alternatives --install /usr/bin/load_gen load_gen
/ngs/app/searchp/opt/parcels/CDH*/bin/load_gen 10
/usr/sbin/update-alternatives --install /usr/bin/mahout mahout
/ngs/app/searchp/opt/parcels/CDH*/bin/mahout 10
/usr/sbin/update-alternatives --install /usr/bin/mapred mapred
/ngs/app/searchp/opt/parcels/CDH*/bin/mapred 10
/usr/sbin/update-alternatives --install /usr/bin/oozie oozie
/ngs/app/searchp/opt/parcels/CDH*/bin/oozie 10
/usr/sbin/update-alternatives --install /usr/bin/pig pig
/ngs/app/searchp/opt/parcels/CDH*/bin/pig 10
/usr/sbin/update-alternatives --install /usr/bin/pyspark pyspark
/ngs/app/searchp/opt/parcels/CDH*/bin/pyspark 10
/usr/sbin/update-alternatives --install /usr/bin/sentry sentry
/ngs/app/searchp/opt/parcels/CDH*/bin/sentry 10
/usr/sbin/update-alternatives --install /usr/bin/solrctl solrctl
/ngs/app/searchp/opt/parcels/CDH*/bin/solrctl 10
/usr/sbin/update-alternatives --install /usr/bin/spark-executor spark-executor
/ngs/app/searchp/opt/parcels/CDH*/bin/spark-executor 10
/usr/sbin/update-alternatives --install /usr/bin/spark-shell spark-shell
/ngs/app/searchp/opt/parcels/CDH*/bin/spark-shell 10
/usr/sbin/update-alternatives --install /usr/bin/spark-submit spark-submit
/ngs/app/searchp/opt/parcels/CDH*/bin/spark-submit 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop sqoop
```

```

/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-codegen sqoop-codegen
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-codegen 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-create-hive-table
sqoop-create-hive-table /ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-create-hive-table
10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-eval sqoop-eval
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-eval 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-export sqoop-export
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-export 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-help sqoop-help
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-help 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-import sqoop-import
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-import 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-import-all-tables
sqoop-import-all-tables /ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-import-all-tables
10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-job sqoop-job
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-job 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-list-databases sqoop-list-databases
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-list-databases 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-list-tables sqoop-list-tables
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-list-tables 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-merge sqoop-merge
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-merge 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-metastore sqoop-metastore
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-metastore 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-version sqoop-version
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop-version 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop2 sqoop2
/ngs/app/searchhp/opt/parcels/CDH*/bin/sqoop2 10
/usr/sbin/update-alternatives --install /usr/bin/statestored statestored
/ngs/app/searchhp/opt/parcels/CDH*/bin/statestored 10
/usr/sbin/update-alternatives --install /usr/bin/whirr whirr
/ngs/app/searchhp/opt/parcels/CDH*/bin/whirr 10
/usr/sbin/update-alternatives --install /usr/bin/yarn yarn
/ngs/app/searchhp/opt/parcels/CDH*/bin/yarn 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-client zookeeper-client
/ngs/app/searchhp/opt/parcels/CDH*/bin/zookeeper-client 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-server zookeeper-server
/ngs/app/searchhp/opt/parcels/CDH*/bin/zookeeper-server 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-server-cleanup
zookeeper-server-cleanup /ngs/app/searchhp/opt/parcels/CDH*/bin/zookeeper-server-cleanup
10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-server-initialize
zookeeper-server-initialize
/ngs/app/searchhp/opt/parcels/CDH*/bin/zookeeper-server-initialize 10
/usr/sbin/update-alternatives --remove accumulo-conf /etc/accumulo/*
/usr/sbin/update-alternatives --remove hadoop-conf /etc/hadoop/*
/usr/sbin/update-alternatives --remove hbase-conf /etc/hbase/*
/usr/sbin/update-alternatives --remove hive-conf /etc/hive/*
/usr/sbin/update-alternatives --remove solr-conf /etc/solr/*
/usr/sbin/update-alternatives --remove spark-conf /etc/spark/*
/usr/sbin/update-alternatives --remove sqoop-conf /etc/sqoop/*

```

Commands run by the Cloudera Manager Agent while creating a diagnostic bundle:

```

/bin/bash -c cd ..; find -maxdepth
/bin/bash -c for x in /etc/security/limits.d/*;
/bin/bash -c PATH
/bin/cat /etc/apt/sources.list
/bin/cat /etc/host.conf
/bin/cat /etc/hostname
/bin/cat /etc/hosts
/bin/cat /etc/issue
/bin/cat /etc/krb5.conf
/bin/cat /etc/nsswitch.conf
/bin/cat /etc/redhat-release
/bin/cat /etc/resolv.conf
/bin/cat /etc/security/limits.conf
/bin/cat /etc/suse-release
/bin/cat /etc/sysconfig/network

```

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

```
/bin/cat /etc/sysconfig/network/ifcfg-eth0
/bin/cat /etc/sysconfig/network-scripts/ifcfg-eth0
/bin/cat /etc/sysconfig/selinux
/bin/cat /proc/cpuinfo
/bin/cat /proc/diskstats
/bin/cat /proc/interrupts
/bin/cat /proc/meminfo
/bin/cat /proc/mounts
/bin/cat /proc/partitions
/bin/cat /proc/swaps
/bin/cat /proc/sys/vm/swappiness
/bin/cat /proc/vmstat
/bin/cat /var/kerberos/krb5kdc/kadm5.acl
/bin/cat /var/kerberos/krb5kdc/kdc.conf
/bin/cat /var/log/kern.log
/bin/cat /var/log/messages
/bin/date
/bin/df -i
/bin/df -k
/bin/dmesg
/bin/grep -r . /sys/kernel/mm
/bin/hostname --fqdn
/bin/ls /etc/yum.repos.d
/bin/netstat -s
/bin/ps aux
/bin/rpm -qa
/bin/uname -a
/bin/uname -r
/sbin/chkconfig --list
/sbin/ethtool eth0
/sbin/ethtool -S eth0
/sbin/ifconfig -a
/sbin/iptables -L -v -n
/sbin/lsmod
/sbin/lspci
/sbin/lvdisplay
/sbin/service --status-all
/sbin/sysctl -A
/usr/bin/curl -m 1 http://169.254.169.254/2011-01-01/meta-data/instance-type
/usr/bin/dig any test-1.ent.cloudera.com
/usr/bin/host -v -t A test-1.ent.cloudera.com
/usr/bin/lsb_release -a
/usr/bin/lscpu
/usr/bin/nslookup -query
/usr/bin/ntpstat
/usr/bin/python -c import socket; print socket.getfqdn();
/usr/bin/sar -A
/usr/bin/top -b -n 1
/usr/bin/uptime
/usr/bin/vmstat
/usr/sbin/dmidecode
/usr/sbin/lsof -n -P
/usr/sbin/ntpq -pn
```

## 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 3: 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.	cloudera-scm
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"> <li>• Direct access to root user using the root password.</li> <li>• Direct access to root user using a SSH key file.</li> <li>• 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.</li> </ul> <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"> <li>• Direct access to root user using the root password.</li> <li>• Direct access to root user using a SSH key file.</li> <li>• 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.</li> </ul> <p>You <i>cannot</i> use another system (such as PowerBroker) that provides root/sudo privileges.</p>
Run the Cloudera Manager Agent.	<p>If <a href="#">single user mode</a> is not enabled, access to the root account during runtime, through one of the following scenarios:</p> <ul style="list-style-type: none"> <li>• 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"> <li>– Direct access to root user using the root password</li> <li>– Direct access to root user using a SSH key file</li> <li>– Passwordless sudo access for a specific user</li> </ul> </li> </ul> <p>Using another system (such as PowerBroker) that provides root/sudo privileges is <i>not</i> acceptable.</p> <ul style="list-style-type: none"> <li>• Through automatic startup during system boot, using init.</li> </ul>
Manually start/stop/restart the Cloudera Manager Agent process.	<p>If <a href="#">single user mode</a> 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>

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

Permission Requirements for Package-Based CDH Installation without Cloudera Manager

**Table 4: 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.

### sudo Commands Run by Cloudera Manager

The sudo commands are:

- yum (RHEL/CentOS/Oracle)
- zypper (SLES)
- apt-get (Debian/Ubuntu)
- apt-key (Debian/Ubuntu)
- sed
- service
- /sbin/chkconfig (RHEL/CentOS/Oracle)
- /usr/sbin/update-rc.d (Debian/Ubuntu)
- id
- rm
- mv
- chown
- install

## 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 2.6 is available with Cloudera Manager 5.7. For information on the requirements for installing Cloudera Manager, see [Cloudera Manager 5 Requirements and Supported Versions](#) on page 10.

### Supported Databases

Cloudera Navigator supports the following databases:

- MariaDB 5.5

- MySQL - 5.1, 5.5, 5.6, and 5.7
- Oracle 11gR2 and 12c



**Note:** When installing a JDBC driver, only the `ojdbc6.jar` file is supported for both Oracle 11g R2 and Oracle 12c; the `ojdbc7.jar` file is not supported.

- PostgreSQL - 8.1, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4

## Supported Browsers

The Cloudera Navigator UI 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 features.

### Cloudera Navigator Auditing

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

Component	Operations (For details, see <a href="#">Cloudera Navigator Auditing</a> ).	Minimum Supported Service Version
HDFS	<ul style="list-style-type: none"> <li>Operations that access or modify a file's or directory's data or metadata</li> <li>Operations denied due to lack of privileges</li> </ul>	CDH 4.0.0
HBase	<ul style="list-style-type: none"> <li>In CDH versions less than 4.2.0, for grant and revoke operations, the operation in log events is <code>ADMIN</code></li> <li>In simple authentication mode, if the HBase Secure RPC Engine property is <code>false</code> (the default), the username in log events is <code>UNKNOWN</code>. To see a meaningful user name:           <ol style="list-style-type: none"> <li>Click the HBase service.</li> <li>Click the <b>Configuration</b> tab.</li> <li>Select <b>Service-wide &gt; Security</b>.</li> <li>Set the HBase Secure RPC Engine property to <code>true</code>.</li> <li>Save the change and restart the service.</li> </ol> </li> </ul>	CDH 4.0.0
Hive	<ul style="list-style-type: none"> <li>Operations (except grant, revoke, and metadata access only) sent to HiveServer2</li> <li>Operations denied due to lack of privileges</li> </ul> <p>Limitations:</p> <ul style="list-style-type: none"> <li>Actions taken against Hive using the Hive CLI are <i>not</i> audited. Therefore if you have enabled auditing you should disable the Hive CLI to prevent actions against Hive that are not audited.</li> <li>In simple authentication mode, the username in log events is the username passed in the HiveServer2 connect command. If you do not</li> </ul>	CDH 4.2.0, CDH 4.4.0 for operations denied due to lack of privileges.

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

Component	Operations (For details, see <a href="#">Cloudera Navigator Auditing</a> ).	Minimum Supported Service Version
	pass a username in the connect command, the username is log events is anonymous.	
Hue	• Operations (except grant, revoke, and metadata access only) sent through the Beeswax Server	CDH 4.4.0
	• User operations such as log in, log out, add and remove user, add and remove LDAP group, add and remove user from LDAP group	CDH 5.5.0
Impala	• Queries denied due to lack of privileges • Queries that pass analysis	Impala 1.2.1 with CDH 4.4.0
Navigator Metadata Server	• Viewing and changing audit reports • Viewing and changing authorization configurations • Viewing and changing metadata • Viewing and changing policies • Viewing and changing saved searches	Cloudera Navigator 2.3
Sentry	• Operations sent to the HiveServer2 and Hive Metastore Server roles and Impala service • Adding and deleting roles, assigning roles to groups and removing roles from groups, creating and deleting privileges, granting and revoking privileges • Operations denied due to lack of privileges  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.	CDH 5.1.0
Solr	• Index creation and deletion • Schema and configuration file modification • Index, service, document tag access	CDH 5.4.0

### Cloudera Navigator Metadata

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

Component	Minimum Supported Version
HDFS. However, federated HDFS is <i>not supported</i> .	CDH 4.4.0
Hive	CDH 4.4.0
Impala	CDH 5.4.0
MapReduce	CDH 4.4.0
Oozie. Supported actions:	CDH 4.4.0
• 2.4 - map-reduce, pig, hive, hive2, sqoop • 2.3 and lower - map-reduce, pig, hive, sqoop	
Pig	CDH 4.6.0
Spark	CDH 5.4.0

Component	Minimum Supported Version
 <b>Important:</b> Spark metadata and lineage is not supported or recommended for production use. By default it is disabled. To try this feature, use it in a test environment until Cloudera resolves currently existing issues and limitations to make it ready for production use.	
Sqoop 1. All <a href="#">Cloudera connectors</a> are supported.	CDH 4.4.0
YARN	CDH 5.0.0

## CDH 5 Requirements and Supported Versions

This page describes requirements and supported third-party software for CDH 5. For the latest information on compatibility across all Cloudera products, see the [Product Compatibility Matrix](#).

### Supported Operating Systems



**Note:** 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.

CDH 5 provides 64-bit packages for RHEL-compatible, SLES, Ubuntu, and Debian systems as listed below.

Operating System	Version	Packages
<b>Red Hat Enterprise Linux (RHEL)-compatible</b>		
RHEL (+ SELinux mode in available versions)	5.7	64-bit
	5.10	64-bit
	6.4	64-bit
	6.5	64-bit
	6.6	64-bit
	6.7	64-bit
	7.1	64-bit
	7.2	64-bit
CentOS (+ SELinux mode in available versions)	5.7	64-bit
	5.10	64-bit
	6.4	64-bit
	6.5	64-bit
	6.6	64-bit
	6.7	64-bit
	7.1	64-bit
	7.2	64-bit

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

Operating System	Version	Packages
Oracle Enterprise (OEL) Linux with Unbreakable Enterprise Kernel (UEK)	5.7 (UEK R2)	64-bit
	5.10	64-bit
	5.11	64-bit
	6.4 (UEK R2)	64-bit
	6.5 (UEK R2, UEK R3)	64-bit
	6.6 (UEK R3)	64-bit
	6.7 (UEK R3)	64-bit
	7.1	64-bit
	7.2	64-bit
<b>SLES</b>		
SUSE Linux Enterprise Server (SLES)	11 with Service Pack 2	64-bit
SUSE Linux Enterprise Server (SLES)	11 with Service Pack 3	64-bit
SUSE Linux Enterprise Server (SLES)	11 with Service Pack 4	64-bit
<b>Ubuntu/Debian</b>		
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, and 7.8	64-bit



**Important:** Cloudera supports RHEL 7 with the following limitations:

- Only RHEL 7.2 and 7.1 are supported. RHEL 7.0 is not supported.
- Only new installations of RHEL 7.2 and 7.1 are supported by Cloudera. For upgrades to RHEL 7.1 or 7.2, contact your OS vendor and see [Does Red Hat support upgrades between major versions of Red Hat Enterprise Linux?](#)



**Note:** Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled. Cloudera is not responsible for policy support nor policy enforcement. If you experience issues with SELinux, contact your OS provider.

## Supported Databases

Component	MariaDB	MySQL	SQLite	PostgreSQL	Oracle	Derby - see Note 5
Oozie	5.5	5.1, 5.5, 5.6, 5.7	–	8.1, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4 See Note 3	11gR2, 12c	Default
Flume	–	–	–	–	–	Default (for the JDBC

Component	MariaDB	MySQL	SQLite	PostgreSQL	Oracle	Derby - see Note 5
						Channel only)
Hue	5.5	5.1, 5.5, 5.6, 5.7 See Note 6	Default	8.1, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4 See Note 3	11gR2, 12c	–
Hive/Impala	5.5	5.1, 5.5, 5.6, 5.7 See Note 1	–	8.1, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4 See Note 3	11gR2, 12c	Default
Sentry	5.5	5.1, 5.5, 5.6, 5.7 See Note 1	–	8.1, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4 See Note 3	11gR2, 12c	–
Sqoop 1	5.5	See Note 4	–	See Note 4	See Note 4	–
Sqoop 2	5.5	–	–	–	–	Default

**Note:**

1. MySQL 5.5 is supported on CDH 5.1. MySQL 5.6 is supported on CDH 5.1 and higher. The InnoDB storage engine must be enabled in the MySQL server.
2. Cloudera Manager installation fails if GTID-based replication is enabled in MySQL.
3. PostgreSQL 9.2 is supported on CDH 5.1 and higher. PostgreSQL 9.3 is supported on CDH 5.2 and higher. PostgreSQL 9.4 is supported on CDH 5.5 and higher.
4. For 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).
5. Derby is supported as shown in the table, but not always recommended. See the pages for individual components in the [Cloudera Installation and Upgrade](#) guide for recommendations.
6. CDH 5 Hue requires the default MySQL version of the operating system on which it is being installed, which is usually MySQL 5.1, 5.5, or 5.6.

## Supported JDK Versions



**Important:** JDK 1.6 is not supported on any CDH 5 release (even though the libraries of CDH 5.0-CDH 5.4 are compatible). **Applications using CDH libraries must run a supported version of JDK 1.7 or higher**, and one that also matches the JDK version of your CDH cluster.

CDH 5.7.x is supported with the versions shown in the following table:

Minimum Supported Version	Recommended Version	Exceptions
1.7.0_55	1.7.0_67, 1.7.0_75, 1.7.0_80	None
1.8.0_31	1.8.0_60	Cloudera recommends that you <i>not</i> use JDK 1.8.0_40.

## Supported Browsers

### Hue

Hue works with the two most recent versions of the following browsers. Cookies and JavaScript must be on.

- **Chrome**
- **Firefox**
- **Safari** (not supported on Windows)
- **Internet Explorer**

Hue could display in older versions and even other browsers, but you might not have access to all of its features.

## Supported Network Protocols

- CDH requires IPv4. IPv6 is not supported.

See also [Configuring Network Names](#) on page 247.

- 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 Transport Layer Security Versions

The following components are supported by the indicated versions of Transport Layer Security (TLS):

**Table 5: Components Supported by TLS**

Component	Role	Name	Port	Version
Flume		Avro Source/Sink		TLS 1.2
Flume		Flume HTTP Source/Sink		TLS 1.2
HBase	Master	HBase Master Web UI Port	60010	TLS 1.2
HDFS	NameNode	Secure NameNode Web UI Port	50470	TLS 1.2
HDFS	Secondary NameNode	Secure Secondary NameNode Web UI Port	50495	TLS 1.2
HDFS	HttpFS	REST Port	14000	TLS 1.1, TLS 1.2
Hive	HiveServer2	HiveServer2 Port	10000	TLS 1.2
Hue	Hue Server	Hue HTTP Port	8888	TLS 1.2
Apache Impala (incubating)	Impala Daemon	Impala Daemon Beeswax Port	21000	TLS 1.2
Apache Impala (incubating)	Impala Daemon	Impala Daemon HiveServer2 Port	21050	TLS 1.2
Apache Impala (incubating)	Impala Daemon	Impala Daemon Backend Port	22000	TLS 1.2
Apache Impala (incubating)	Impala Daemon	Impala Daemon HTTP Server Port	25000	TLS 1.2

Component	Role	Name	Port	Version
Apache Impala (incubating)	Impala StateStore	StateStore Service Port	24000	TLS 1.2
Apache Impala (incubating)	Impala StateStore	StateStore HTTP Server Port	25010	TLS 1.2
Apache Impala (incubating)	Impala Catalog Server	Catalog Server HTTP Server Port	25020	TLS 1.2
Apache Impala (incubating)	Impala Catalog Server	Catalog Server Service Port	26000	TLS 1.2
Oozie	Oozie Server	Oozie HTTPS Port	11443	TLS 1.1, TLS 1.2
Solr	Solr Server	Solr HTTP Port	8983	TLS 1.1, TLS 1.2
Solr	Solr Server	Solr HTTPS Port	8985	TLS 1.1, TLS 1.2
YARN	ResourceManager	ResourceManager Web Application HTTP Port	8090	TLS 1.2
YARN	JobHistory Server	MRv1 JobHistory Web Application HTTP Port	19890	TLS 1.2

## 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.

### Amazon Web Services

For information on deploying Cloudera software on a Amazon Web Services (AWS) cloud infrastructure, see the [Cloudera Enterprise Reference Architecture for AWS Deployments](#).

### Google Cloud Platform

For information on deploying Cloudera software on a Google Cloud Platform infrastructure, see the [Cloudera Enterprise Reference Architecture for Google Cloud Platform Deployments](#).

### Microsoft Azure

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

### VMware

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

Recommendation when deploying on VMware in the current release:

- Use the part of Hadoop Virtual Extensions that has been implemented in [HADOOP-8468](#). This will prevent data loss when a physical node that hosts two or more DataNodes goes down .

### Filesystem Requirements

#### Supported Filesystems

The Hadoop Distributed File System (HDFS) is designed to run on top of an underlying filesystem in an operating system. Cloudera recommends that you use either of the following filesystems tested on the [supported operating systems](#):

- **ext3:** This is the most tested underlying filesystem for HDFS.
- **ext4:** This scalable extension of ext3 is supported in more recent Linux releases.



**Important:** Cloudera does not support in-place upgrades from ext3 to ext4. Cloudera recommends that you format disks as ext4 before using them as data directories.

- **XFS:** This is the default filesystem in RHEL 7.

#### File Access Time

Linux filesystems keep metadata that record when each file was accessed. This means that even reads result in a write to the disk. To speed up file reads, Cloudera recommends that you disable this option, called `atime`, using the mount option in `/etc/fstab`:

```
/dev/sdb1 /data1 ext4 defaults,noatime 0
```

Apply the change without rebooting:

```
mount -o remount /data1
```

### Ports

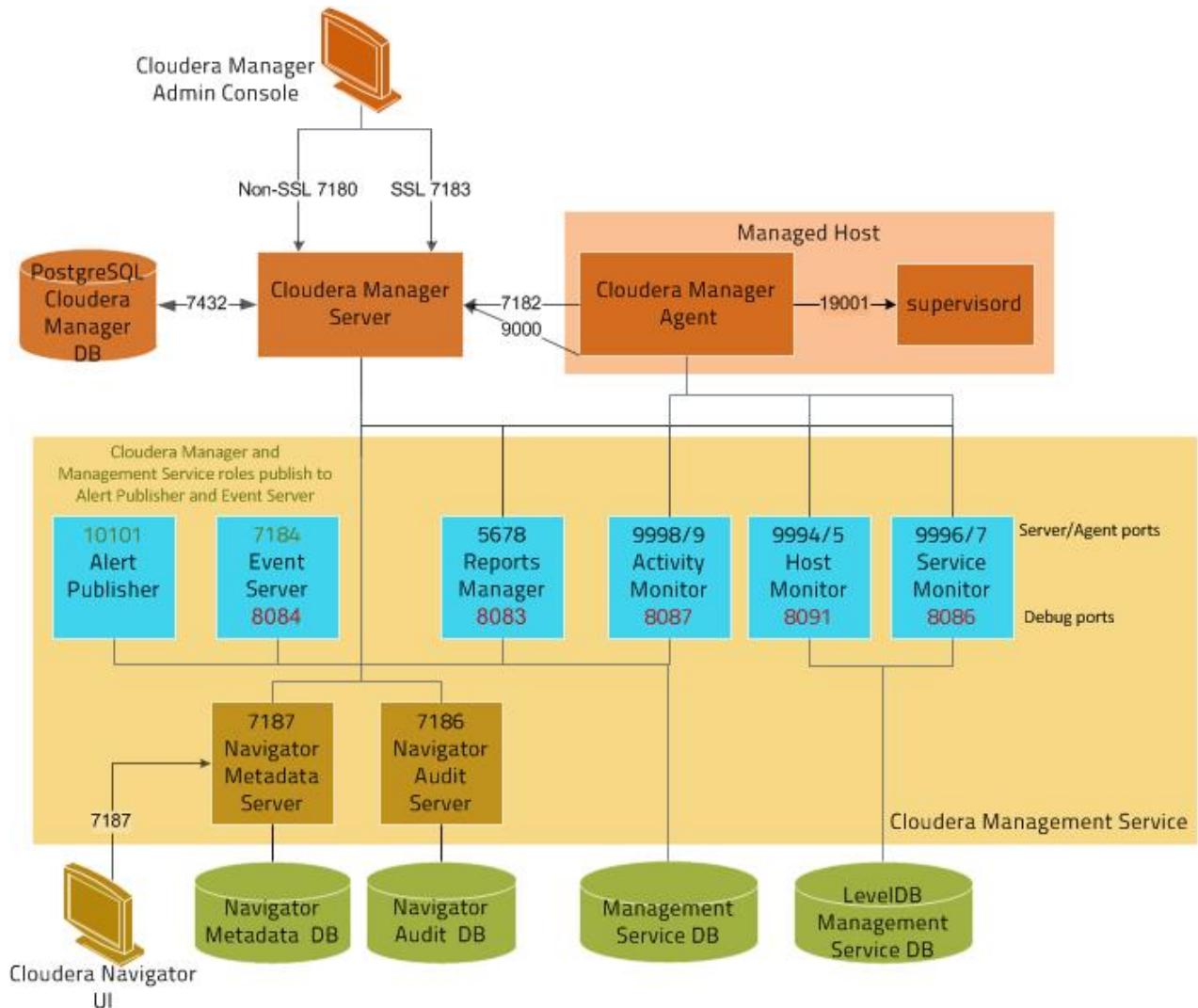
Cloudera Manager, CDH components, managed services, and third-party components use the ports listed in the tables that follow. Before you deploy Cloudera Manager, CDH, and managed services, and third-party components make sure these ports are open on each system. If you are using a firewall, such as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

In the tables in the subsections that follow, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components (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 NodeManager and the JobHistory Server Web UIs).

#### 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:

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



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 cloudera-scm-server-db service	7432	Internal		The optional embedded PostgreSQL database used for storing configuration information for Cloudera Manager Server.

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

Component	Service	Port	Access Requirement	Configuration	Comment
	Peer-to-peer parcel distribution	7191	Internal	Hosts > All Hosts > Configuration > P2P Parcel Distribution Port	Used to distribute parcels to cluster hosts during installation and upgrade operations.
Cloudera Manager Agent	HTTP (Debug)	9000	Internal	/etc/cloudera-scm-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 > 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 > Ports and Addresses	
Service Monitor	HTTP (Debug)	8086	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	Listening for Agent messages (private protocol)	9997	Internal		
	Internal query API (Avro)	9996	Internal		
Activity Monitor	HTTP (Debug)	8087	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	Listening for Agent messages (private protocol)	9999	Internal		
	Internal query API (Avro)	9998	Internal		
Host Monitor	HTTP (Debug)	8091	Internal	Cloudera Management Service > Configuration > Ports and Addresses	

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

Component	Service	Port	Access Requirement	Configuration	Comment
	Listening for Agent messages (private protocol)	9995	Internal		
	Internal query API (Avro)	9994	Internal		
Reports Manager	Queries (Thrift)	5678	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	HTTP (Debug)	8083	Internal		
Cloudera Navigator				Cloudera Management Service > Configuration > Ports and Addresses	
Audit Server	HTTP	7186	Internal		
	HTTP (Debug)	8089	Internal		The port where Navigator Audit Server runs 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 > Ports and Addresses	Used for communication to peer (source) Cloudera Manager.
	HDFS NameNode	8020	External	HDFS > Configuration > 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 destination HDFS NameNode(s).
	HDFS DataNode	50010	External	HDFS > Configuration > 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).

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

### Ports Used by Cloudera Navigator Encryption

All ports listed are TCP.

Component	Service	Port	Access Requirement	Configuration	Comment
Cloudera Navigator Key Trustee Server	HTTPS (key management)	11371	External	<b>Key Trustee Server service &gt; Configuration &gt; Category &gt; Ports and Addresses &gt; Key Trustee Server Port</b>	Navigator Key Trustee Server clients (including Key Trustee KMS and Navigator Encrypt) access this port to store and retrieve encryption keys.
	PostgreSQL database	11381	External	<b>Key Trustee Server service &gt; Configuration &gt; Category &gt; Ports and Addresses &gt; Key Trustee Server Database Port</b>	The Navigator Key Trustee Server database listens on this port. The Passive Key Trustee Server connects to this port on the Active Key Trustee Server for replication in <a href="#">Cloudera Navigator Key Trustee Server High Availability</a> .

### 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	dfs.https.address

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

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

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

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	External	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	
	ApplicationMaster			External		The ApplicationMaster serves an HTTP service using an ephemeral port that cannot be restricted. This port is never accessed directly from outside the cluster by clients. All requests to the ApplicationMaster web server is routed using the YARN ResourceManager (proxy service). Locking down access to ephemeral port ranges within the cluster's network might restrict your access to the ApplicationMaster UI and its logs, along with the ability to look at running applications.
Flume	Flume Agent		41414	External		
Hadoop KMS	Key Management Server		16000	External	kms_http_port	CDH 5.2.1 and higher. Applies to both Java KeyStore KMS and Key Trustee KMS.
	Key Management Server		16001	Localhost	kms_admin_port	CDH 5.2.1 and higher. Applies to both Java KeyStore KMS and Key Trustee KMS.
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	Internal	hbase.zookeeper.property.clientPort	HBase-managed ZooKeeper mode

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	HQuorumPeer		2888	Internal	hbase.zookeeper.peerport	HBase-managed ZooKeeper mode
	HQuorumPeer		3888	Internal	hbase.zookeeper.leaderport	HBase-managed ZooKeeper mode
	REST	Non-Cloudera Manager - managed	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	Cloudera Manager - managed	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		
	Thrift Server	Thrift Server	9090	External	Pass -p <port> on CLI	
	Thrift Server		9095	External		
		Avro server	9090	External	Pass --port <port> on CLI	
	hbase-solr-indexer	Lily Indexer	11060	External		
Hive	Metastore		9083	External		
	HiveServer2		10000	External	hive.server2.thrift.port	The <a href="#">Beeline command interpreter</a> requires that you specify this port on the command line.
	HiveServer2 Web User Interface (UI)		10002	External	hive.server2.webui.port in hive-site.xml	
	WebHCat Server		50111	External	templeton.port	
Hue	Server		8888	External		
Oozie	Oozie Server		11000	External	OOZIE_HTTP_PORT in oozie-env.sh	HTTP
	Oozie Server	SSL	11443	External		HTTPS

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	Oozie Server		11001	localhost	OOZIE_ADMIN_PORT in oozie-env.sh	Shutdown port
Sentry	Sentry Server		8038	External	sentry.service. server.rpc-port	
	Sentry Server		51000	External	sentry.service. web.port	
Spark	Default Master RPC port		7077	External		
	Default Worker RPC port		7078	External		
	Default Master web UI port		18080	External		
	Default Worker web UI port		18081	External		
	History Server		18088	External	history.port	
	Shuffle service		7337	Internal		
Sqoop	Metastore		16000	External	sqoop. metastore. server.port	
Sqoop 2	Sqoop 2 server		8005	Localhost	SQOOP_ADMIN_PORT environment variable	
	Sqoop 2 server		12000	External		
	Sqoop 2		12001	External		Admin port
ZooKeeper	Server (with CDH 5 or Cloudera Manager 5)		2181	External	clientPort	Client port
	Server (with CDH 5 only)		2888	Internal	X in server.N =host:X:Y	Peer
	Server (with CDH 5 only)		3888	Internal	X in server.N =host:X:Y	Peer
	Server (with CDH 5 and Cloudera Manager 5)		3181	Internal	X in server.N =host:X:Y	Peer
	Server (with CDH 5 and Cloudera Manager 5)		4181	Internal	X in server.N =host:X:Y	Peer

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	ZooKeeper JMX port		9010	Internal		<p>ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must do <i>one</i> of the following:</p> <ul style="list-style-type: none"> <li>• Open up all ports when the connection originates from the Cloudera Manager Server</li> <li>• Do the following:           <ol style="list-style-type: none"> <li>1. Open a non-ephemeral port (such as 9011) in the firewall.</li> <li>2. Install Oracle Java 7u4 JDK or higher.</li> <li>3. Add the port configuration to the advanced configuration snippet, for example:  <code>-Dcom.sun.management.jmxremote.port=9011</code></li> <li>4. Restart ZooKeeper.</li> </ol> </li> </ul>

## Ports Used by Components of CDH 4

All ports listed are TCP.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hadoop HDFS	DataNode		50010	External	<code>dfs.datanode.address</code>	DataNode HTTP server port
	DataNode	Secure	1004	External	<code>dfs.datanode.address</code>	
	DataNode		50075	External	<code>dfs.datanode.http.address</code>	
	DataNode		50475	External	<code>dfs.datanode.https.address</code>	
	DataNode	Secure	1006	External	<code>dfs.datanode.http.address</code>	

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	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.namenode.https-address	dfs.https.address is deprecated (but still works)
	Secondary NameNode		50090	Internal	dfs.secondary.http.address or dfs.namenode.secondary.http-address	dfs.secondary.http.address is deprecated (but still works)
	Secondary NameNode	Secure	50495	Internal	dfs.secondary.https.address	
	JournalNode		8485	Internal	dfs.namenode.shared.edits.dir	
	JournalNode		8480	Internal	dfs.journalnode.http-address	
	JournalNode		8481	Internal	dfs.journalnode.https-address	
	Failover Controller		8019	Internal		Used for NameNode HA
	HttpFS		14000	External		
	HttpFS		14001	External		
Hadoop MRv1	JobTracker		8021	External	mapred.job.tracker	
	JobTracker		8023	External	mapred.ha.job.tracker	High Availability service protocol port for the

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
						JobTracker. The JobTracker listens on a separate port for HA operations.
	JobTracker		50030	External	mapred.job.tracker.http.address	
	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	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	External	yarn.resourcemanager.webapp.https.address	
	NodeManager		8040	Internal	yarn.nodemanager.localizer.address	
	NodeManager		8041	External	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	

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	JobHistory Server		10033	Internal	mapreduce.jobhistory.admin.address	
	Shuffle HTTP		13562	Internal		
	JobHistory Server		19888	External	mapreduce.jobhistory.webapp.address	
	JobHistory Server		19890	External	mapreduce.jobhistory.webapp.https.address	
Flume	Flume Agent		41414	External		
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	Internal	hbase.zookeeper.property.clientPort	HBase-managed ZK mode
	HQuorumPeer		2888	Internal	hbase.zookeeper.peerport	HBase-managed ZK mode
	HQuorumPeer		3888	Internal	hbase.zookeeper.leaderport	HBase-managed ZK mode
	REST	Non-CM-managed	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	CM-managed	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 -p <port> on CLI	
	ThriftServer		9095	External		
		Avro server	9090	External	Pass --port <port> on CLI	

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hive	Metastore		9083	External		
	HiveServer		10000	External		
	HiveServer2		10000	External	hive. server2. thrift.port	
	WebHCat Server		50111	External	templeton.port	
Hue	Server		8888	External		
	Beeswax Server		8002	Internal		
	Beeswax Metastore		8003	Internal		
Oozie	Oozie Server		11000	External	OOZIE_HTTP_ PORT in oozie-env.sh	HTTP
	Oozie Server	SSL	11443	External		HTTPS for Oozie is supported in CDH 4.3.0 and higher
	Oozie Server		11001	localhost	OOZIE_ADMIN_ PORT in oozie-env.sh	Shutdown port
Sqoop	Metastore		16000	External	sqoop. metastore. server.port	
Sqoop 2	Sqoop 2 server		8005	Localhost	SQOOP_ADMIN_PORT environment variable	
	Sqoop 2 server		12000	External		
Sqoop 2	Sqoop 2		12001	External		Admin port
ZooKeeper	Server (with CDH4 or Cloudera Manager 4)		2181	External	clientPort	Client port
	Server (with CDH4 only)		2888	Internal	X in server.N =host:X:Y	Peer
	Server (with CDH4 only)		3888	Internal	X in server.N =host:X:Y	Peer
	Server (with CDH4 and Cloudera Manager 4)		3181	Internal	X in server.N =host:X:Y	Peer

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	Server (with CDH4 and Cloudera Manager 4)		4181	Internal	X in server.N =host:X:Y	Peer
	ZooKeeper JMX port		9010	Internal		ZooKeeper will also use another randomly selected port for RMI. In order for Cloudera Manager to monitor ZooKeeper, you must open up all ports when the connection originates from the Cloudera Manager Server.

## 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.

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

Component	Service	Port	Access Requirement	Comment
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	StateStore Service Port	26000	Internal	Internal use only. The catalog service uses this port to communicate with the Impala 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 DistCp

All ports listed are TCP.

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hadoop HDFS	NameNode		8020	External	<code>fs.default.name</code> or <code>fs.defaultFS</code>	<code>fs.default.name</code> is deprecated (but still works)
	DataNode	Secure	1004	External	<code>dfs.datanode.address</code>	
	DataNode		50010	External	<code>dfs.datanode.address</code>	
WebHDFS	NameNode		50070	External	<code>dfs.http.address</code> or <code>dfs.namenode.http-address</code>	<code>dfs.http.address</code> is deprecated (but still works)
	DataNode	Secure	1006	External	<code>dfs.datanode.http.address</code>	
HttpFS	web		14000			

### 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	<code>kdc_ports</code> and <code>kdc_tcp_ports</code> in either the <code>[kdcdefaults]</code> or <code>[realms]</code> sections of <code>kdc.conf</code>	By default only UDP
	KRB5 Admin Server	Secure	749	TCP	External	<code>kadmin_port</code> in the <code>[realms]</code> section of <code>kdc.conf</code>	
	kpasswd		464	UDP/TCP	External		
SSH	ssh		22	TCP	External		
PostgreSQL			5432	TCP	Internal		

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

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
MariaDB			3306	TCP	Internal		
MySQL			3306	TCP	Internal		
LDAP	LDAP Server		389	TCP	External		
	LDAP Server over TLS/SSL	TLS/SSL	636	TCP	External		
	Global Catalog		3268	TCP	External		
	Global Catalog over TLS/SSL	TLS/SSL	3269	TCP	External		

# Managing Software Installation Using Cloudera Manager

A major function of Cloudera Manager is to install CDH and managed service software. Cloudera Manager installs software for new deployments and to upgrade existing deployments. 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. The important differences between parcels and packages are:

- 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.



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

## 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. The important differences between parcels and packages are:

- 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.

For detailed installation instructions using parcels, and other methods, see [Installation Overview](#) on page 74.

Parcels are available for CDH 4.1.3 and higher, for other managed services, and for Sqoop Connectors.



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

## Advantages of Parcels

Because of their unique properties, parcels offer the following advantages over packages:

## Managing Software Installation Using Cloudera Manager

- **Distribution of CDH as a single object** - Instead of having a separate package for each part of CDH, parcels have just a single object to install. This makes it easier to distribute software to a cluster that is not connected to the Internet.
- **Internal consistency** - All CDH components are matched, eliminating the possibility of installing parts from different versions of CDH.
- **Installation outside of /usr** - In some environments, Hadoop administrators do not have privileges to install system packages. These administrators needed to use CDH tarballs, which do not provide the infrastructure that packages do. With parcels, administrators can install to /opt, or anywhere else, without completing the additional manual steps of regular tarballs.



**Note:** With parcels, the path to the CDH libraries is /opt/cloudera/parcels/CDH/lib instead of the usual /usr/lib. Do not link /usr/lib/ elements to parcel-deployed paths, because the links may cause scripts that distinguish between the two paths to not work.

- **Installation of CDH without sudo** - Parcel installation is handled by the Cloudera Manager Agent running as root or another user, so you can install CDH without sudo.
- **Decoupled distribution from activation** - With side-by-side install capabilities, you can stage a new version of CDH across the cluster before switching to it. This allows the most time-consuming part of an upgrade to be done ahead of time without affecting cluster operations, thereby reducing downtime.
- **Rolling upgrades** - Packages require you to shut down the old process, upgrade the package, and then start the new process. Any errors in the process can be difficult to recover from, and upgrading requires extensive integration with the package management system to function seamlessly. With parcels, when a new version is staged side-by-side, you can switch to a new minor version by simply changing which version of CDH is used when restarting each process. You can then perform upgrades with [rolling restarts](#), in which service roles are restarted in the correct order to switch 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. Major version upgrades (for example, CDH 4 to CDH 5) require full service restarts because of substantial changes between the versions. Finally, you can upgrade individual parcels or multiple parcels at the same time.
- **Upgrade management** - Cloudera Manager manages all the steps in a CDH version upgrade. With packages, Cloudera Manager only helps with initial installation.
- **Additional components** - Parcels are not limited to CDH. Cloudera Impala, Cloudera Search, LZO, Apache Kafka, and [add-on service](#) parcels are also available.
- **Compatibility with other distribution tools** - Cloudera Manager works with other tools you use for download and distribution. For example, you can use Puppet. Or, you can download the parcel to Cloudera Manager Server manually if 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 have following phases:

- **Downloaded** - The parcel software is copied to a local parcel directory on the Cloudera Manager Server, where it is available for distribution to other hosts in any of the clusters managed by this Cloudera Manager Server. You can have multiple parcels for a product downloaded to your Cloudera Manager Server. After a parcel has been downloaded to the Server, it is available for distribution on all clusters managed by the Server. A downloaded parcel appears in the cluster-specific section for every cluster managed by this Cloudera Manager Server.
- **Distributed** - The parcel is copied to the cluster hosts, and components of the parcel are unpacked. Distributing a parcel does not upgrade the components running on your cluster; the current services continue to run unchanged. You can have multiple parcels distributed on your cluster. Distributing parcels does not require Internet access; the Cloudera Manager Agent on each cluster member downloads the parcels from the local parcel repository on the Cloudera Manager Server.
- **Activated** - Links to the parcel components are created. Activation does not automatically stop the current services or perform a restart. You can restart services after activation, or the system administrator can determine when to perform those operations.

If you are upgrading CDH or managed services when you activate a parcel, follow the instructions in [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524 to complete the upgrade.

- **In Use** - The parcel components on the cluster hosts are in use when you start or restart the services that use those components.
- **Deactivated** - The links to the parcel components are removed from the cluster hosts.
- **Removed** - The parcel components are removed from the cluster hosts.
- **Deleted** - The parcel is deleted from the local parcel repository on the Cloudera Manager Server.

Cloudera Manager detects when new parcels are available. You can configure Cloudera Manager to download and distribute parcels automatically. See [Configuring Cloudera Manager Server Parcel Settings](#) on page 63.

## Parcel Locations

The default location for the local parcel directory on the Cloudera Manager Server is /opt/cloudera/parcel-repo. To change this location, follow the instructions in [Configuring Cloudera Manager Server Parcel Settings](#) on page 63.

The default location for the distributed parcels on 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 64.



**Note:** With parcels, the path to the CDH libraries is /opt/cloudera/parcels/CDH/lib instead of the usual /usr/lib. Do not link /usr/lib/ elements to parcel-deployed paths, because the links may cause scripts that distinguish between the two paths to not work.

## Managing Parcels

On the Parcels page in Cloudera Manager, you can manage parcel installation and activation and determine which parcel versions are running across your clusters. The Parcels page displays a list of parcels managed by Cloudera Manager. Cloudera Manager displays the name, version, and status of each parcel and provides available actions on the parcel.

### Accessing the Parcels Page

Access the Parcels page by doing one of the following:

- Click the parcel indicator
- Click the **Hosts** in the top navigation bar, then the **Parcels** tab.

Use the selectors on the left side of the console to filter the displayed parcels:

- **Location** selector - View only parcels that are available remotely, only parcels pertaining to a particular cluster, or parcels pertaining to all clusters. When you access the Parcels page, the selector is set to Available Remotely.
- **Error Status** section of the **Filters** selector - Limit the list of displayed parcels by error status.
- **Parcel Name** section of the **Filters** selector - Limit the list of displayed parcels by parcel name.
- **Status** section of the **Filters** selector - Limit the list to parcels that have been distributed, parcels that have not been distributed (**Other**), or all parcels.

When you download a parcel, it appears in the list for each cluster managed by Cloudera Manager, indicating that the parcel is available for distribution on those clusters. Only one copy of the downloaded parcel resides on the Cloudera Manager Server. After you distribute the parcel, Cloudera Manager copies the parcel to the hosts in that cluster.

For example, if Cloudera Manager is managing two clusters, the rows in the All Clusters page list the information about the parcels on the two clusters. The Status column displays the current status of the parcels. The Version column displays version information about the parcel. Click the icon to view the release notes for the parcel. The Actions column shows actions you can perform on the parcels, such as download, distribute, delete, deactivate, and remove from host.

## Managing Software Installation Using Cloudera Manager

Location	Parcel Name	Version	Status	Actions
Available Remotely	CDH 4	4.8.0-1.cdh4.8.0.p0.1150	Available Remotely	Download
Cluster 1				
Cluster 2		4.7.1-1.cdh4.7.1.p0.47	Downloaded	Distribute
All Clusters				

Filters	Parcel Name	Version	Status	Actions
▼ PARCEL NAME	CDH 4	4.8.0-1.cdh4.8.0.p0.1150	Available Remotely	Download
All	16			
ACCUMULO	2			
<b>CDH 4</b>	<b>4</b>	<b>4.7.1-1.cdh4.7.1.p0.47</b>	<b>Distributed, Activated</b>	<b>Deactivate</b>
CDH 5	4			
IMPALA	2			
SOLR	2			
SPARK	2			
▼ STATUS				
All	4			
Distributed	1			
Other	3			

### Downloading a Parcel

1. Go to the Parcels page. In the Location selector, click **ClusterName** or **Available Remotely**. Parcels that are available for download display the Available Remotely status and a Download button.

If the parcel you want is not shown here—for example, you want to upgrade to a version of CDH that is not the most current version—you can make additional remote parcel repositories available. You can also configure the location of the local parcel repository and other settings. See [Parcel Configuration Settings](#) on page 63.

If a parcel version is too new to be supported by the Cloudera Manager version, the parcel appears with a red background and error message:

CDH 5	5.5.0-1.cdh5.5.0.p0.871	Available Remotely
• Local parcel error for parcel CDH-5.5.0-1.cdh5.5.0.p0.871-el6.parcel : The version 5.5.0-1.cdh5.5.0.p0.871 is too new to be supported.		

Such parcels are also listed when you select the Error status in the Error Status section of the Filters selector.

2. Click the **Download** button of the parcel you want to download to your local repository. The status changes to **Downloading**.

After a parcel has been downloaded, it is removed from the Available Remotely page.

### Distributing a Parcel

Downloaded parcels can be distributed to the hosts in your cluster and made available for activation. Parcels are downloaded to 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, you select distribution to a specific cluster's hosts on a cluster-by-cluster basis.

1. From the Parcels page, in the Location selector, select the cluster where you want to distribute the parcel, or select **All Clusters**. (The first cluster in the list is selected by default when you open the Parcels page.)
2. Click **Distribute** for the parcel you want to distribute. The status changes to **Distributing**. During distribution, you can:
  - Click the **Details** link in the Status column to view the **Parcel Distribution Status** page.

- Click **Cancel** to cancel the distribution. When the Distribute action completes, the button changes to **Activate**, and you can click the **Distributed** status link to view the status page.

Distribution does not require Internet access; 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 parcels must be distributed, you can control how many concurrent uploads Cloudera Manager performs. See [Parcel Configuration Settings](#) on page 63.

To delete a parcel that is ready to be distributed, click the triangle at the right end of the **Distribute** button and select **Delete**. This deletes the 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.

- From the Parcels page, in the Location selector, choose **ClusterName** or **All Clusters**, and click the **Activate** button for the parcel you want to activate. This updates Cloudera Manager to point to the new software, which is ready to run the next time a service is restarted. A pop-up indicates which services must be restarted to use the new parcel.
- Choose one of the following:
  - Restart** - Activate the parcel and restart services affected by the new parcel.
  - Activate Only** - Active the parcel. You can restart services at a time that is convenient. If you do not restart services as part of the activation process, you must restart them at a later time. Until you restart services, the current parcel continues to run.
- Click **OK**.

Activating a new parcel also deactivates the previously active parcel for the product you just upgraded. However, until you restart the services, the previously active parcel displays a status of **Still in use** because the services are using that parcel, and you cannot remove the parcel until it is no longer being used.

If the parcel you activate updates the software for only a subset of services, even if you restart all of that subset, the previously active parcel displays **Still in use** until you restart the remaining services. For example, if you are running HDFS, YARN, Oozie, Hue, Impala, and Spark services, and you activate a parcel that updates only the Oozie service, the pop-up that displays instructs you to restart only the Oozie and Hue services. Because the older parcel is still in use by the HDFS, YARN, Impala, and Spark services, the parcel page shows that parcel as **Still in use** until you restart these remaining services.

Sometimes additional upgrade steps may be required. In this case, instead of **Activate**, the button will say **Upgrade**. When you click the **Upgrade** button, the upgrade wizard starts. See [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524.

### Deactivating a Parcel

You can deactivate an active parcel; this updates Cloudera Manager to point to the previous software version, which is ready to run the next time a service is restarted. From the Parcels page, choose **ClusterName** or **All Clusters** in the Location selector, and click the **Deactivate** button on an activated parcel.

To use the previous version of the software, restart your services.



**Important:** If you originally installed from parcels, and one version of the software is installed (that is, no packages, and no previous parcels have been activated and started), when you attempt to restart after deactivating the current version, your roles will be stopped and will not be able to restart.

## Managing Software Installation Using Cloudera Manager

### Removing a Parcel

From the Parcels page, in the Location selector, choose **ClusterName** or **All Clusters**, click the  to the right of an **Activate** button, and select **Remove from Hosts**.

### Deleting a Parcel

From the Parcels page, in the Location selector, choose **ClusterName** or **All Clusters**, and click the  to the right of a **Distribute** button, and select **Delete**.

### Relocating the Host Parcel Directory

The default location of the host parcel directory is `/opt/cloudera/parcels`. To relocate distributed parcels to a different directory, do the following:

1. Stop all services.
2. Deactivate all in-use parcels.
3. Shut down the Cloudera Manager Agent on all hosts.
4. Move the existing parcels to the new location.
5. Configure the host parcel directory.
6. Start the Cloudera Manager Agents.
7. Activate the parcels.
8. Start all services.

### Troubleshooting

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

If a parcel is being distributed but never completes, make sure you have enough free space in the [parcel download directories](#), because Cloudera Manager will try to download and unpack parcels even if there is insufficient space.

## Viewing Parcel Usage

The **Parcel Usage** page shows parcels in current use in your clusters. In a large deployment, this makes it easier to keep track of different versions 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, the cluster is not displayed, and instead you see a message indicating the cluster is not running parcels.

**Hosts** Status Configuration Templates Disks Overview **Packages**

### Parcel Usage

Product

Cluster 1 ▾

CDH ▾

CDH 5.1.0-1.cdh5.1.0.p0.460 (Active, 4)

No CDH processes running on this host

Multiple product versions running on a single host

Hosts with CDH processes running



The screenshot shows the 'Parcels' tab in the Cloudera Manager interface. It displays a 'Parcel Usage' section with dropdown menus for 'Cluster 1' and 'CDH'. Under 'CDH', a checkbox for 'CDH 5.1.0-1.cdh5.1.0.p0.460 (Active, 4)' is checked. Below the dropdowns are two checkboxes: 'No CDH processes running on this host' (unchecked) and 'Multiple product versions running on a single host' (unchecked). To the right, a heading 'Hosts with CDH processes running' is followed by a host map icon consisting of four squares, each containing a blue square, representing four hosts with active parcels.

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 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 occurs, for example, if you have not restarted a service or role after activating a new parcel. If you have individual hosts running components installed as packages, the square is empty.

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, click the square representing the host. This displays the parcel versions installed on that host.

**Parcel Usage**

The screenshot shows the 'Parcels' tab in Cloudera Manager. On the left, a sidebar for 'Cluster 1' lists products: CDH (selected), No CDH processes running, and Multiple product versions running. The main area displays 'Hosts with CDH processes running' for host [tcdn501-1.ent.cloudera.com](#). A tooltip for this host shows 'Versions used by running roles' including CDH 5.1.0-1.cdh5.1.0.p0.460 (Active) and roles like Hive Metastore Server, HiveServer2, JobHistory Server, NameNode, Oozie Server, ResourceManager, SecondaryNameNode, and Server.

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

**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 Soop 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 or click **Hosts** and click the **Parcels** tab.
2. Click the **Configuration** button.

- **Menu**

1. Select **Administration > Settings**.
2. Select **Category > Parcels**.

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 checks 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. 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](#)

## Managing Software Installation Using Cloudera Manager

[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, `latest`, or the substitution variable `{latest_supported}`. The substitution variable appears after the parcel for the CDH version with the same major number as the Cloudera Manager version to enable substitution of the latest supported maintenance version of CDH.

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 checks for new parcels.
- Configure a proxy to access to the remote repositories.
- Configure whether downloads and distribution of parcels should occur automatically when new ones are detected. If automatic downloading and distribution are not enabled (the default), 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 versions and how many parcel versions to retain

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

- Theoretically, the concurrent upload count (**Maximum Parcel Uploads**) is unimportant if all hosts have the same speed Ethernet. Fifty concurrent uploads is acceptable in most cases. However, if the server has more bandwidth (for example, 10 GbE, and the normal hosts are using 1 GbE), then the count is important to maximize bandwidth. It should 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 configured to prevent starving other services, or if you can accept the risk associated with 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. Select **Category > Parcels**.
4. 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.
5. Click **Save Changes** to commit the changes.
6. [Restart](#) the Cloudera Manager Agent on all hosts.

## Configuring Peer-to-Peer Distribution of Parcels

Cloudera Manager uses a peer-to-peer service to efficiently distribute parcels to cluster hosts. The service is enabled by default and is configured to run on port 7191. You can change this port number, and you can disable peer-to-peer distribution.

To modify peer-to-peer distribution of parcels:

1. Open Cloudera Manager and select **Hosts > All Hosts > Configuration**.
2. Change the value of the **P2P Parcel Distribution Port** property to the new port number.  
Set the value to 0 to disable peer-to-peer distribution of parcels.
3. Click **Save Changes** to commit the changes.

## 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](#) on page 524.

### Download, Distribute, and Activate Parcels

1. In the Cloudera Manager Admin Console, click the Parcels indicator  in the top navigation bar.
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 63 page:
  - **CDH 4**
    - CDH - <https://archive.cloudera.com/cdh4/parcels/>
    - Impala - <https://archive.cloudera.com/impala/parcels/>
    - Search <https://archive.cloudera.com/search/parcels/>
    - Spark - <https://archive.cloudera.com/spark/parcels/>
    - GPL Extras - <https://archive.cloudera.com/gplextras/parcels/>
  - **CDH 5** - Impala, Spark, and Search are included in the CDH parcel.
    - CDH - <https://archive.cloudera.com/cdh5/parcels/>
    - GPL Extras - <https://archive.cloudera.com/gplextras5/parcels/>
  - **Key Trustee Server**
    - Go to the Key Trustee Server [download page](#). Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee Server parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz filename.tar.gz` command.
  - **Key Trustee KMS**
    - Go to the Key Trustee KMS [download page](#). Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee KMS parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz filename.tar.gz` command.
  - **Other services**
    - Accumulo - <https://archive.cloudera.com/accumulo/parcels/>
    - Sqoop connectors - <https://archive.cloudera.com/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 Remote Parcel Repository for Cloudera Manager](#) on page 172 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**.

## Managing Software Installation Using Cloudera Manager

### Restart the Cluster and Deploy Client Configuration

#### 1. Restart the cluster:

- On the **Home > Status** tab, click



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

- 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:

- On the **Home > Status** tab, click



to the right of the cluster name and select **Deploy Client Configuration**.

- Click **Deploy Client Configuration**.

### Uninstall Packages

- 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.
- Uninstall the CDH packages on each host:

- Not including Impala and Search

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

- Including Impala and Search

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

- 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:
  - a. Stop the Hue service.
  - b. Copy the backup from the temporary location to the newly created Hue database directory, /var/lib/hue.
  - c. 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 parcels, the path to the CDH libraries is /opt/cloudera/parcels/CDH/lib instead of the usual /usr/lib. Do not link /usr/lib/ elements to parcel-deployed paths, because the links may cause scripts that distinguish between the two paths to not work. 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**)

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](#) on page 524.

### Install CDH and Managed Service Packages

#### Choose a Repository Strategy

To install CDH and Managed Service Packages, choose one of the following repository strategies:

- 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 170. 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.

Do one of the following:

- [Install CDH 5 and Managed Service Packages](#) on page 67
- [Install CDH 4, Impala, and Solr Managed Service Packages](#) on page 70

### Install CDH 5 and Managed Service Packages

Install the packages on all cluster hosts using the following steps:

- **Red Hat**

1. Download and install the "1-click Install" package.

- a. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>

## Managing Software Installation Using Cloudera Manager

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 7	<a href="https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera">RHEL/CentOS/Oracle 7 link</a>

- b. Install the RPM for all RHEL versions:

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

2. (Optional) add a repository key:

- Red Hat/CentOS/Oracle 5

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

- Red Hat/CentOS/Oracle 6

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

3. Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python  
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

- SLES

1. Download and install the "1-click Install" package.

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

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

- b. Install the RPM:

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

- c. Update your system package index by running:

```
$ sudo zypper refresh
```

2. (Optional) add a repository key:

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

### 3. Install the CDH packages:

```
$ 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 hive-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

- Ubuntu and Debian

- Download and install the "1-click Install" package

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

- 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
```

- Optionally add a repository key:

- Debian Wheezy

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

- Ubuntu Precise

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

- Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

## Managing Software Installation Using Cloudera Manager

### Install CDH 4, Impala, and Solr Managed Service Packages

Install the packages on all cluster hosts using the following steps:

- **RHEL-compatible**

1. Click the entry in the table at [CDH Download Information](#) that matches your RHEL or CentOS system.
2. Go to the repo file (`cloudera-cdh4.repo`) for your system and save it in the `/etc/yum.repos.d/` directory.
3. Optionally add a repository key:

- **RHEL/CentOS/Oracle 5**

```
$ sudo rpm --import  
https://archive.cloudera.com/cdh4/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- **RHEL/CentOS 6**

```
$ sudo rpm --import  
https://archive.cloudera.com/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
```

**4. Install packages on every host in your cluster:**

- a. Install CDH 4 packages:

```
$ 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
```

- b. To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo yum install hue
```

**5. (Requires CDH 4.2 and higher) Install Impala**

- a. In the table at [Cloudera Impala Version and Download Information](#), click the entry that matches your RHEL or CentOS system.
- b. Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
- c. Install Impala and the Impala Shell on Impala machines:

```
$ sudo yum -y install impala impala-shell
```

**6. (Requires CDH 4.3 and higher) Install Search**

- a. In the table at [Cloudera Search Version and Download Information](#), click the entry that matches your RHEL or CentOS system.
- b. Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
- c. Install the Solr Server on machines where you want Cloudera Search.

```
$ sudo yum -y install solr-server
```

- **SLES**

**1. Run the following command:**

```
$ sudo zypper addrepo -f  
https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/cloudera-cdh4.repo
```

**2.** Update your system package index by running:

```
$ sudo zypper refresh
```

**3.** Optionally add a repository key:

```
$ sudo rpm --import https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
```

**4.** Install packages on every host in your cluster:

a. Install CDH 4 packages:

```
$ 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
```

b. To install the hue-common package and all Hue applications on the Hue host, install the hue meta-package:

```
$ sudo zypper install hue
```

c. (Requires CDH 4.2 and higher) Install Impala

a. Run the following command:

```
$ sudo zypper addrepo -f  
https://archive.cloudera.com/impala/sles/11/x86_64/impala/cloudera-impala.repo
```

b. Install Impala and the Impala Shell on Impala machines:

```
$ sudo zypper install impala impala-shell
```

d. (Requires CDH 4.3 and higher) Install Search

a. Run the following command:

```
$ sudo zypper addrepo -f  
https://archive.cloudera.com/search/sles/11/x86_64/search/cloudera-search.repo
```

b. Install the Solr Server on machines where you want Cloudera Search.

```
$ sudo zypper install solr-server
```

- Ubuntu or Debian

1. In the table at [CDH Version and Packaging Information](#), click the entry that matches your Ubuntu or Debian system.

2. Go to the list file (cloudera.list) for your system and save it in the /etc/apt/sources.list.d/ directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your cloudera.list file should look like:

```
deb [arch=amd64] https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4  
contrib  
deb-src https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib
```

3. Optionally add a repository key:

## Managing Software Installation Using Cloudera Manager

- Ubuntu Lucid

```
$ curl -s https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh/archive.key | sudo apt-key add -
```

- Ubuntu Precise

```
$ curl -s https://archive.cloudera.com/cdh4/ubuntu/precise/amd64/cdh/archive.key | sudo apt-key add -
```

- Debian Squeeze

```
$ curl -s https://archive.cloudera.com/cdh4/debian/squeeze/amd64/cdh/archive.key | sudo apt-key add -
```

### 4. Install packages on every host in your cluster:

- Install CDH 4 packages:

```
$ 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
```

- To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo apt-get install hue
```

- (Requires CDH 4.2 and higher) Install Impala

- In the table at [Cloudera Impala Version and Download Information](#), click the entry that matches your Ubuntu or Debian system.
- Go to the list file for your system and save it in the `/etc/apt/sources.list.d/` directory.
- Install Impala and the Impala Shell on Impala machines:

```
$ sudo apt-get install impala impala-shell
```

- (Requires CDH 4.3 and higher) Install Search

- In the table at [Cloudera Search Version and Download Information](#), click the entry that matches your Ubuntu or Debian system.
- Install Solr Server on machines where you want Cloudera Search:

```
$ sudo apt-get install solr-server
```

## 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:

- 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 > Status** tab, 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

From the Parcels page, in the Location selector, choose **ClusterName** or **All Clusters**, click the  to the right of an **Activate** button, and select **Remove from Hosts**.

### Deleting a Parcel

From the Parcels page, in the Location selector, choose **ClusterName** or **All Clusters**, and click the  to the right of a **Distribute** button, and select **Delete**.

# Installation Overview

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](#) on page 466.



**Note:** If you intend to deploy Cloudera Manager in a highly-available configuration, see [Configuring Cloudera Manager for High Availability With a Load Balancer](#) before starting your installation.

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 three installation options:
    - [Installation Path A - Automated Installation by Cloudera Manager \(Non-Production Mode\)](#) on page 132 - Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, Cloudera Manager Agent, CDH, and managed service software on cluster hosts. Cloudera Manager also 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 14 for further information.
      - All hosts must have access to standard package repositories and either archive.cloudera.com or a local repository with the required installation files.
    - [Installation Path B - Installation Using Cloudera Manager Parcels or Packages](#) on page 141 - you install the Oracle JDK, 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.
- 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 14 for further information.

- All hosts must have access to standard package repositories and either archive.cloudera.com or a local repository with the required 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 - Installation Using Cloudera Manager Parcels or Packages](#) on page 141 - 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.

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 14 for further information.
  - All hosts must have access to standard package repositories and either archive.cloudera.com or a local repository with the required installation files.
  - [Installation Path C - Manual Installation Using Cloudera Manager Tarballs](#) on page 157 - you install the Oracle JDK, Cloudera Manager Server, and Cloudera Manager Agent software using tarballs and use Cloudera Manager to automate installation of CDH and managed service software as parcels.

## Cloudera Manager Installation Phases

The following table describes the phases of installing 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. 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 74.

**Table 6: Cloudera Installation Phases**

Phase			
<b>Phase 1: Install JDK</b>	<p>There are two options:</p> <ul style="list-style-type: none"> <li>• Install the JDK required by Cloudera Manager Server, Management Service, and CDH.</li> <li>• Use the Cloudera Manager Installer to install a supported version of the Oracle JDK in /usr/java and on all hosts in the cluster.</li> <li>• Use the command line to manually install supported versions of the Oracle JDK and set the JAVA_HOME environment variable to the install directory on all hosts.</li> </ul>		
<b>Phase 2: Set up Databases</b>	<p>There are two options:</p> <ul style="list-style-type: none"> <li>• Install, configure, and start the databases that are required by the Cloudera Manager Server, Cloudera Management Service, and that are optional for some CDH services.</li> <li>• Use the Cloudera Manager Installer to install, configure, and start an embedded PostgreSQL database.</li> <li>• Use command-line package installation tools like yum to install, configure, and install the database</li> </ul>		
	Path A	Path B	Path C
<b>Phase 3: Install Cloudera Manager Server</b>	Use the Cloudera Manager Installer to install its	Use Linux package install commands (like yum) to	Use Linux commands to unpack tarballs and service

## Installation Overview

Phase			
Install and start Cloudera Manager Server on one host.	packages and the server. Requires Internet access and sudo privileges on the host.	install Cloudera Manager Server.  Update database properties.  Use service commands to start Cloudera Manager Server.	commands to start the server.
<b>Phase 4: Install Cloudera Manager Agents</b>  Install and start the Cloudera Manager Agent on all hosts.	Use the Cloudera Manager Installation wizard to install the Agents on all hosts.	There are two options: <ul style="list-style-type: none"><li>• Use Linux package install commands (like <code>yum</code>) to install Cloudera Manager Agents on all hosts.</li><li>• Use the Cloudera Manager Installation wizard to install the Agents on all hosts.</li></ul>	Use Linux commands to unpack tarballs and service commands to start the agents on all hosts.
<b>Phase 5: Install CDH and Managed Service software</b>  Install, configure, and start CDH and managed services on all hosts.	Use the Cloudera Manager Installation wizard to install CDH and other managed services.	There are two options: <ul style="list-style-type: none"><li>• Use the Cloudera Manager Installation wizard to install CDH and other managed services.</li><li>• Use Linux package install commands (like <code>yum</code>) to install CDH and other managed services on all hosts.</li></ul>	Use Linux commands to unpack tarballs and service commands to start CDH and managed services on all hosts.
<b>Phase 6: Create, Configure and Start CDH and Managed Services</b>  Configure and start CDH and managed services.	Use the Cloudera Manager Installation wizard to install CDH and other managed services, assign roles to hosts, and configure the cluster. Many configurations are automated.	Use the Cloudera Manager Installation wizard to install CDH and other managed services, assign roles to hosts, and configure the cluster. Many configurations are automated.	Use the Cloudera Manager Installation wizard to install CDH and other managed services, assign roles to hosts, and configure the cluster. Many configurations are automated.  You can also use the Cloudera Manager API to manage a cluster, which can be useful for scripting preconfigured deployments.

## Cloudera Manager Installation Software

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

- **Installation path A (non-production)** - A small self-executing Cloudera Manager installation program to install the Cloudera Manager Server and other packages. 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.



**Important:** Path A installation is intended for demonstrations and proof-of-concept deployments only. Do not use this method of installation for production environments.

- **Installation paths B and C** - Cloudera Manager package repositories for manually installing the Cloudera Manager Server, Agent, and embedded database packages.
- **Installation path B** - The Cloudera Manager Installation wizard for automating installation of Cloudera Manager Agent package.
- **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 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 Installation 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.)

Installation paths:

- [Installation Path A - Automated Installation by Cloudera Manager \(Non-Production Mode\)](#) on page 132
- [Installation Path B - Installation Using Cloudera Manager Parcels or Packages](#) on page 141
- [Installation Path C - Manual Installation Using Cloudera Manager Tarballs](#) on page 157

## Unmanaged Deployment

In an deployment not managed by Cloudera Manager, you are responsible for managing all phases of the lifecycle of CDH and managed service components on each host: installation, configuration, and service lifecycle 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.

- **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, and so on, 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.

## Installation Overview

See [Installing and Deploying CDH Using the Command Line](#) on page 218 for detailed instructions for each of these options.

## Java Development Kit Installation

Some installation paths require that you install the Oracle Java Development Kit on hosts before deploying Cloudera Manager, CDH, and managed services. To install the Oracle JDK, follow the instructions in [Installing the Oracle JDK](#) on page 78. The completed installation, or any already existing installation, must meet the following requirements.

### Requirements

- Install a supported version:
  - CDH 5 - [Supported JDK Versions](#) on page 33
  - 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](#) on page 741.

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](#) or [Java SE 7 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, for example

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

in the following files on Cloudera Manager Server and cluster hosts:

- Cloudera Manager Server - `/etc/default/cloudera-scm-server`. This change affects only the Cloudera Manager Server process, and does not affect the Cloudera Management Service roles.
  - Cloudera Manager package-based or unmanaged deployment - `/etc/default/bigtop-utils`
4. Follow the instructions in [Configuring a Custom Java Home Location](#) on page 176. This change affects all CDH processes and Cloudera Management Service roles in the cluster.



**Note:** This method of changing the JDK for Cloudera Manager, Cloudera Management Service roles, and CDH processes does not affect the JDK used by other non-Cloudera processes.

## Cloudera Manager and Managed Service Datastores

Cloudera Manager uses various databases and datastores 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 \(Non-Production Mode\)](#) on page 132 and [Embedded PostgreSQL Database](#) on page 83.



**Important:** The embedded PostgreSQL database is not recommended for use in production systems.

Although the embedded database is useful for getting started quickly, you can also use your own PostgreSQL, MariaDB, 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 121.

### Required Databases

The Cloudera Manager Server, Oozie Server, Sqoop 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.

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large.
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small.
- 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.

See [Backing Up Databases](#) on page 117.

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

Cloudera Manager provides three installation paths:

## Installation Overview

- 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 use the embedded PostgreSQL database for the Cloudera Manager Server and can optionally choose to create external databases for Oozie Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server. If you choose to use PostgreSQL for Sqoop Server you must create an external database.
- Path B and Path C require you to create databases for the Cloudera Manager Server, Oozie Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server. If you choose to use PostgreSQL for Sqoop Server you must create an external database.

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. For demonstration and proof-of-concept deployments you can use an embedded PostgreSQL database. See [Embedded PostgreSQL Database](#) on page 83.



**Important:** The embedded PostgreSQL database is not recommended for use in production systems.

#### Preparing a Cloudera Manager Server External Database

Before performing these steps, install and configure a database server as described in [Configuring and Starting the MariaDB Server](#) on page 93, [Configuring and Starting the MySQL Server](#) on page 99, [Configuring the Oracle Server](#) on page 105, or [Configuring and Starting the PostgreSQL Server](#) on page 86.

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/cmft/schema/scm_prepare_database.sh
```

- Tarball install

```
<tarball root>/share/cmft/schema/scm_prepare_database.sh
```

The script prepares the database by:

- Creating the Cloudera Manager Server database configuration file.

- (MariaDB, MySQL, and PostgreSQL) Creating a database for the Cloudera Manager Server to use. This is optional and is only completed if options are specified.
- (MariaDB, MySQL, and PostgreSQL) Setting up a user account for the Cloudera Manager Server. This is optional and is only completed if options are specified.

**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
```

Return to [\(Optional\) Manually Install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Service Packages](#) on page 144.

#### 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 7: Required Parameters**

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

Option	Description
<code>-h</code> or <code>--host</code>	The IP address or hostname of the host where the database is installed. The default is to use the local host.
<code>-P</code> or <code>--port</code>	The port number to use to connect to the database. The default port is 3306 for MariaDB, 3306 for MySQL, 5432 for PostgreSQL, and 1521 for Oracle. This option is used for a remote connection only.
<code>-u</code> or <code>--user</code>	The admin username for the database application. For <code>-u</code> , no space occurs between the option and the provided value. If this option is supplied, the script creates a user and

## Installation Overview

Option	Description
	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/cmfschema/scm_prepare_database.sh postgresql scm scm scm
```

## External Databases for Oozie Server, Sqoop Server, 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 Oozie Server, Sqoop Server, 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 [Configuring an External Database for Oozie](#) on page 113, [Configuring an External Database for Sqoop](#) on page 116, [MariaDB Database](#) on page 92, [MySQL Database](#) on page 98, [Oracle Database](#) on page 105, and [External PostgreSQL Database](#) on page 86.

## External Databases for Hue

By default Hue is configured to use the SQLite database. Cloudera strongly recommends an external database for clusters with multiple Hue users. See [Using an External Database for Hue Using Cloudera Manager](#).

## 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 - Installation Using Cloudera Manager Parcels or Packages](#) on page 141 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
RHEL-compatible, if you have a yum repo configured	\$ sudo yum install cloudera-manager-server-db-2
RHEL-compatible, if you're transferring RPMs manually	sudo yum --nogpgcheck localinstall cloudera-manager-server-db-2.noarch.rpm
SLES	\$ sudo zypper install cloudera-manager-server-db-2
Ubuntu or Debian	\$ sudo apt-get install cloudera-manager-server-db-2

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.

## Installation Overview

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

✓ Skipped. Cloudera Manager will create this database in a later step.

Database Host Name:  
tcdn2-1.ent.cloudera.com:7432

Database Type:  
PostgreSQL

Database Name :  
hive

Username:  
hive

Password:  
t56lwbdk4F

##### Reports Manager

✓ Successful

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

rman

Username:

rman

Password:

Y6S4lWvfnNo

##### Navigator Audit Server

✓ Successful

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

nav

Username:

nav

Password:

QLR2B0qq9O

##### Navigator Metadata Server

✓ Successful

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

navms

Username:

navms

Password:

imo07jxOen

##### Oozie Server

✓ Skipped. Cloudera Manager will create this database in a later step.

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

oozie\_oozie\_se

Username:

oozie\_oozie\_se

Password:

NTF1KNdpPl

Test Connection

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, select **Clusters > Cloudera Management Service**.
- Click the **Configuration** tab.
- In the **Scope** section, select the role where you are configuring the database.
- Select **Category > 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 86 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:

- RHEL

```
$ 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 an available supported version. See [Supported Databases](#) on page 12.

- 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_connections` - 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.

- **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
```

## Installation Overview

- **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

Role	Database	User	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;
```

Return to [Establish Your Cloudera Manager Repository Strategy](#) on page 142.

#### Configuring the Hue Server to Store Data in PostgreSQL

For information about installing and configuring an external PostgreSQL database , see [External PostgreSQL Database](#) on page 86.

1. In the Cloudera Manager Admin Console, go to the Hue service status page.
2. Select **Actions > Stop**. Confirm you want to stop the service by clicking **Stop**.
3. Select **Actions > Dump Database**. Confirm you want to dump the database by clicking **Dump Database**.
4. Note the host to which the dump was written under **Step** in the **Dump Database Command** window. You can also find it by selecting **Commands > Recent Commands > Dump Database**.
5. Open a terminal window for the host and go to the dump file in `/tmp/hue_database_dump.json`.
6. Remove all JSON objects with `useradmin.userprofile` in the `model` field, for example:

```
{
  "pk": 14,
  "model": "useradmin.userprofile",
  "fields": [
    { "creation_method": "EXTERNAL", "user": 14, "home_directory": "/user/tuser2" }
  ],
}
```

7. Install the PostgreSQL server.

#### RHEL

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

#### SLES

```
$ sudo zypper install postgresql-server
```

#### Ubuntu or Debian

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

8. Initialize the data directories.

```
$ service postgresql initdb
```

9. Configure client authentication.

a. Edit `/var/lib/pgsql/data/pg_hba.conf`.

## Installation Overview

- 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
```

- 10** Start the PostgreSQL server.

```
$ su - postgres
# /usr/bin/postgres -D /var/lib/pgsql/data > logfile 2>&1 &
# exit
```

- 11** Configure PostgreSQL to listen on all network interfaces.

- a.** Edit /var/lib/pgsql/data/postgresql.conf and set list\_addresses.

```
listen_addresses = '0.0.0.0'      # Listen on all addresses
```

- 12** 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
```

- 13** Restart the PostgreSQL server.

```
$ sudo service postgresql restart
```

- 14** Verify connectivity.

```
su - postgres
# psql -h localhost -U hue -d hue
Password for user hue: secretpassword
hue=> \q
exit
```

- 15** Configure the PostgreSQL server to start at boot.

### 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
```

- 16** Configure the Hue database:

- a.** In the Cloudera Manager Admin Console, click the **HUE** service.
- b.** Click the **Configuration** tab.
- c.** Select **Scope > Hue Server**.
- d.** Select **Category > Advanced**.

- e. Set Hue Server Advanced Configuration Snippet (Safety Valve) for `hue_safety_valve_server.ini` with the following:

```
[desktop]
[[database]]
engine=postgresql_psycopg2
name=hue
host=localhost
port=5432
user=hue
password=secretpassword
```



**Note:** If you set **Hue Database Hostname**, **Hue Database Port**, **Hue Database Username**, and **Hue Database Password** at the service-level, under **Service-Wide > Database**, you can omit those properties from the server-lever configuration above and avoid storing the Hue password as plain text. In either case, set **engine** and **name** in the server-level safety-valve.

- f. Click **Save Changes**.

17. Optionally restore the Hue data to the new database:

- Select **Actions > Synchronize Database**.
- Determine the foreign key ID.

```
bash# su - postgres
$ psql -h localhost -U hue -d hue
postgres=# \d auth_permission;
```

- Drop the foreign key that you retrieved in the previous step.

```
postgres=# ALTER TABLE auth_permission DROP CONSTRAINT content_type_id_refs_id_xxxxxxx;
```

- Delete the rows in the `django_content_type` table.

```
postgres=# TRUNCATE django_content_type CASCADE;
```

- In Hue service instance page, **Actions > Load Database**. Confirm you want to load the database by clicking **Load Database**.
- 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_xxxxxxx
FOREIGN KEY (content_type_id) REFERENCES django_content_type(id) DEFERRABLE INITIALLY
DEFERRED;
```

18 Start the Hue service.

#### Configuring PostgreSQL for Oozie

Install PostgreSQL 8.4.x or 9.0.x.

See [External PostgreSQL Database](#) on page 86.

#### Create the Oozie User and Oozie Database

For example, using the PostgreSQL `psql` command-line tool:

```
$ psql -U postgres
Password for user postgres: *****
```

## Installation Overview

```
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.UTF-8'  
    LC_CTYPE = 'en_US.UTF-8'  
    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
```

## MariaDB Database

To use a MariaDB database, follow these procedures.

### Installing the MariaDB Server



#### Note:

- If you already have a MariaDB database set up, you can skip to the section [Configuring and Starting the MariaDB Server](#) on page 93 to verify that your MariaDB 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 MariaDB database.

OS	Command
RHEL	\$ sudo yum install mariadb-server
SLES	\$ sudo zypper install mariadb-server
Ubuntu and Debian	\$ sudo apt-get install mariadb-server



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

After issuing the command to install MariaDB, you might need to confirm that you want to complete the installation.

## Configuring and Starting the MariaDB Server

1. Stop the MariaDB server if it is running.

```
$ sudo service mariadb stop
```

2. 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.

3. Determine the location of the [option file](#), `my.cnf`.

4. Update `my.cnf` so that it conforms to the following requirements:

- To prevent deadlocks, set the isolation level to read committed.
- The default settings in the MariaDB 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 MariaDB 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 = 16M
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

binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
```

## Installation Overview

```
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
```

5. If AppArmor is running on the host where MariaDB is installed, you might need to configure AppArmor to allow MariaDB to write to the binary.
6. Ensure the MariaDB server starts 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 and Debian	\$ sudo chkconfig mysql on

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

7. Start the MariaDB server:

```
$ sudo service mariadb start
```

8. Set the MariaDB 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 MariaDB 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.

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.

Cloudera recommends that you use the MySQL JDBC driver for MariaDB.

OS	Command
RHEL	<ol style="list-style-type: none"> <li>Download the MySQL JDBC driver from <a href="http://www.mysql.com/downloads/connector/j/5.1.html">http://www.mysql.com/downloads/connector/j/5.1.html</a>.</li> <li>Extract the JDBC driver JAR file from the downloaded file. For example:           <pre>tar zxvf mysql-connector-java-5.1.31.tar.gz</pre> </li> <li>Copy the JDBC driver, renamed, to the relevant host. For example:           <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> </li> </ol> <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> <p><b>Note:</b> Do not use the <code>yum install</code> command to install the MySQL driver package, because it installs openJDK, and then uses the Linux alternatives command to set the system JDK to be openJDK.</p>
SLES	<code>\$ sudo zypper install mysql-connector-java</code>
Ubuntu or Debian	<code>\$ sudo apt-get install libmysql-java</code>

Return to [Establish Your Cloudera Manager Repository Strategy](#) on page 142.

[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

## Installation Overview

- 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 MariaDB 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

### Configuring the Hue Server to Store Data in MariaDB

For information about installing and configuring a MariaDB database , see [MariaDB Database](#) on page 92.

1. In the Cloudera Manager Admin Console, go to the Hue service status page.
2. Select **Actions > Stop**. Confirm you want to stop the service by clicking **Stop**.
3. Select **Actions > Dump Database**. Confirm you want to dump the database by clicking **Dump Database**.
4. Note the host to which the dump was written under **Step** in the **Dump Database Command** window. You can also find it by selecting **Commands > Recent Commands > Dump Database**.
5. Open a terminal window for the host and go to the dump file in `/tmp/hue_database_dump.json`.
6. Remove all JSON objects with `useradmin.userprofile` in the `model` field, for example:

```
{  
  "pk": 14,
```

```
"model": "useradmin.userprofile",
"fields":
{ "creation_method": "EXTERNAL", "user": 14, "home_directory": "/user/tuser2" }
},
```

**7.** Set strict mode in /etc/my.cnf and restart MySQL:

```
[mysqld]
sql_mode=STRICT_ALL_TABLES
```

**8.** Create a new database and grant privileges to a Hue user to manage this database. For example:

```
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)
```

**9.** In the Cloudera Manager Admin Console, click the Hue service.

**10** Click the **Configuration** tab.

**11** Select **Scope > All**.

**12** Select **Category > Database**.

**13** Specify the settings for **Hue Database Type**, **Hue Database Hostname**, **Hue Database Port**, **Hue Database Username**, **Hue Database Password**, and **Hue Database Name**. For example, for a MySQL database on the local host, you might use the following values:

- Hue Database Type = mysql
- Hue Database Hostname = *host*
- Hue Database Port = 3306
- Hue Database Username = hue
- Hue Database Password = *secretpassword*
- Hue Database Name = hue

**14** Optionally restore the Hue data to the new database:

- a. Select **Actions > Synchronize Database**.
- b. Determine the foreign key ID.

```
$ mysql -uhue -psecretpassword
mysql > SHOW CREATE TABLE auth_permission;
```

- c. (InnoDB only) Drop the foreign key that you retrieved in the previous step.

```
mysql > ALTER TABLE auth_permission DROP FOREIGN KEY content_type_id_refs_id_XXXXXX;
```

- d. Delete the rows in the django\_content\_type table.

```
mysql > DELETE FROM hue.django_content_type;
```

- e. In Hue service instance page, click **Actions > Load Database**. Confirm you want to load the database by clicking **Load Database**.
- f. (InnoDB only) Add back the foreign key.

```
mysql > ALTER TABLE auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES
django_content_type (id);
```

**15** Start the Hue service.

## Installation Overview

### Configuring MariaDB for Oozie

#### Install and Start MariaDB 5.5

See [MariaDB Database](#) on page 92.

#### Create the Oozie Database and Oozie MariaDB User

For example, using the MariaDB mysql command-line tool:

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

mysql> create database oozie;
Query OK, 1 row affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> exit
Bye
```

#### Add the MariaDB JDBC Driver JAR to Oozie

Cloudera recommends that you use the MySQL JDBC driver for MariaDB. Copy or symbolically link the MySQL JDBC driver JAR to the /var/lib/oozie/ directory.



**Note:** You must manually download the MySQL JDBC driver JAR file.

## 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 99 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.
- Cloudera Manager installation fails if GTID-based replication is enabled in MySQL.

#### 1. Install the MySQL database.

OS	Command
RHEL	\$ sudo yum install mysql-server
SLES	\$ sudo zypper install mysql \$ sudo zypper install libmysqlclient_r15

OS	Command
	 <b>Note:</b> Some SLES systems encounter errors when using the preceding <code>zypper install</code> command. For more information on resolving this issue, see the Novell Knowledgebase topic, <a href="#">error running chkconfig</a> .
<b>Ubuntu and Debian</b>	<code>\$ sudo apt-get install mysql-server</code>

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
<b>RHEL</b>	<code>\$ sudo service mysqld stop</code>
<b>SLES, Ubuntu, and Debian</b>	<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](#).

## Installation Overview

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 = 16M
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. Comment out binlog_format for older versions.
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

sql_mode=STRICT_ALL_TABLES
```

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 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 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	<ol style="list-style-type: none"> <li>Download the MySQL JDBC driver from <a href="http://www.mysql.com/downloads/connector/j/5.1.html">http://www.mysql.com/downloads/connector/j/5.1.html</a>.</li> <li>Extract the JDBC driver JAR file from the downloaded file. For example:           <pre>tar zxvf mysql-connector-java-5.1.31.tar.gz</pre> </li> <li>Copy the JDBC driver, renamed, to the relevant host. For example:           <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> </li> </ol> <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</pre>

## Installation Overview

OS	Command
	<pre>mysql-connector-java-5.1.31/mysql-connector-java-5.1.31-bin.jar /usr/share/java/mysql-connector-java.jar</pre>
	<p> <b>Note:</b> Do not use the <code>yum install</code> command to install the MySQL driver package, because it installs openJDK, and then uses the <code>Linux alternatives</code> command to set the system JDK to be openJDK.</p>
SLES	\$ sudo zypper install mysql-connector-java
Ubuntu or Debian	\$ sudo apt-get install libmysql-java

Return to [Establish Your Cloudera Manager Repository Strategy](#) on page 142.

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

### Configuring the Hue Server to Store Data in MySQL



**Note:** Cloudera recommends InnoDB over MyISAM as the Hue MySQL engine. On CDH 5, Hue *requires* InnoDB.

For information about installing and configuring a MySQL database , see [MySQL Database](#) on page 98.

1. In the Cloudera Manager Admin Console, go to the Hue service status page.
2. Select **Actions > Stop**. Confirm you want to stop the service by clicking **Stop**.
3. Select **Actions > Dump Database**. Confirm you want to dump the database by clicking **Dump Database**.
4. Note the host to which the dump was written under **Step** in the **Dump Database Command** window. You can also find it by selecting **Commands > Recent Commands > Dump Database**.
5. Open a terminal window for the host and go to the dump file in `/tmp/hue_database_dump.json`.
6. Remove all JSON objects with `useradmin.userprofile` in the `model` field, for example:

```
{
  "pk": 14,
  "model": "useradmin.userprofile",
  "fields":
  {
    "creation_method": "EXTERNAL", "user": 14, "home_directory": "/user/tuser2" }
},
```

7. Set strict mode in `/etc/my.cnf` and restart MySQL:

```
[mysqld]
sql_mode=STRICT_ALL_TABLES
```

8. Create a new database and grant privileges to a Hue user to manage this database. For example:

```
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)
```

9. In the Cloudera Manager Admin Console, click the Hue service.

- 10 Click the **Configuration** tab.

- 11 Select **Scope > All**.

- 12 Select **Category > Database**.

- 13 Specify the settings for **Hue Database Type**, **Hue Database Hostname**, **Hue Database Port**, **Hue Database Username**, **Hue Database Password**, and **Hue Database Name**. For example, for a MySQL database on the local host, you might use the following values:

- Hue Database Type = mysql

## Installation Overview

- Hue Database Hostname = *host*
- Hue Database Port = 3306
- Hue Database Username = hue
- Hue Database Password = *secretpassword*
- Hue Database Name = hue

**14** Optionally restore the Hue data to the new database:

- a. Select **Actions > Synchronize Database**.
- b. Determine the foreign key ID.

```
$ mysql -uhue -psecretpassword
mysql > SHOW CREATE TABLE auth_permission;
```

- c. **(InnoDB only)** Drop the foreign key that you retrieved in the previous step.

```
mysql > ALTER TABLE auth_permission DROP FOREIGN KEY content_type_id_refs_id_XXXXXX;
```

- d. Delete the rows in the django\_content\_type table.

```
mysql > DELETE FROM hue.django_content_type;
```

- e. In Hue service instance page, click **Actions > Load Database**. Confirm you want to load the database by clicking **Load Database**.
- f. **(InnoDB only)** Add back the foreign key.

```
mysql > ALTER TABLE auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES
django_content_type (id);
```

**15** Start the Hue service.

### Configuring MySQL for Oozie

#### Install and Start MySQL 5.x

See [MySQL Database](#) on page 98.

#### 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;
Query OK, 1 row affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> exit
Bye
```

#### 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.

## 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
```

## Installation Overview

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;
```

### Ensuring 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>.



**Note:** Only the `ojdbc6.jar` file is supported for both Oracle 11g R2 and Oracle 12c; the `ojdbc7.jar` file is not supported.

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 80.

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;
```

## Installation Overview

```
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.

Return to [Establish Your Cloudera Manager Repository Strategy](#) on page 142.

### Configuring the Hue Server to Store Data in Oracle (Parcel Installation)

Use the following instructions to configure the Hue Server with an Oracle database if you are working on a parcel-based deployment. If you are using packages, see [Configuring the Hue Server to Store Data in Oracle \(Package Installation\)](#).

For information about installing and configuring an Oracle database , see [Oracle Database](#) on page 105.



**Important:** Configure the database for character set AL32UTF8 and national character set UTF8.

1. Install the required packages.

#### RHEL

```
$ sudo yum install gcc python-devel python-pip python-setuptools libaio
```

#### SLES

```
$ sudo zypper install gcc python-devel python-pip python-setuptools libaio
```

#### Ubuntu or Debian

```
$ sudo apt-get install gcc python-devel python-pip python-setuptools libaio1
```

2. Add <http://tiny.cloudera.com/hue-oracle-client-db> to the Cloudera Manager remote parcel repository URL list and download, distribute, and activate the parcel.
3. For CDH versions lower than 5.3, install the Python Oracle library:



**Note:** HUE\_HOME is a reference to the location of your Hue installation. For package installs, this is usually /usr/lib/hue; for parcel installs, this is usually, /opt/cloudera/parcels/<parcel version>/lib/hue/.

```
$ HUE_HOME/build/env/bin/pip install cx_Oracle
```

4. For CDH versions lower than 5.3, upgrade django south:

```
$ HUE_HOME/build/env/bin/pip install south --upgrade
```

5. In the Cloudera Manager Admin Console, go to the Hue service status page.
6. Select **Actions > Stop**. Confirm you want to stop the service by clicking **Stop**.
7. Select **Actions > Dump Database**. Confirm you want to dump the database by clicking **Dump Database**.
8. Click the **Configuration** tab.
9. Select **Scope > All**.
10. Select **Category > Advanced**.
11. Set the **Hue Service Advanced Configuration Snippet (Safety Valve)** for **hue\_safety\_valve.ini** property.



**Note:** If you set **Hue Database Hostname**, **Hue Database Port**, **Hue Database Username**, and **Hue Database Password** at the service-level, under **Service-Wide > Database**, you can omit those properties from the server-lever configuration above and avoid storing the Hue password as plain text. In either case, set **engine** and **name** in the server-level safety-valve.

Add the following options (and modify accordingly for your setup):

```
[desktop]
[[database]]
host=localhost
port=1521
engine=oracle
user=hue
password=secretpassword
name=<SID of the Oracle database, for example, 'XE'>
```

For CDH 5.1 and higher you can use an Oracle service name. To use the Oracle service name instead of the SID, use the following configuration instead:

```
port=0
engine=oracle
user=hue
password=secretpassword
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}
```

- 12 Grant required permissions to the `hue` user in Oracle:

```
GRANT CREATE <sequence> TO <user>;
GRANT CREATE <session> TO <user>;
GRANT CREATE <table> TO <user>;
GRANT CREATE <view> TO <user>;
GRANT CREATE <procedure> TO <user>;
GRANT CREATE <trigger> TO <user>;
GRANT EXECUTE ON sys.dbms_crypto TO <user>;
GRANT EXECUTE ON SYS.DBMS_LOB TO <user>;
```

- 13 Go to the Hue Server instance in Cloudera Manager and select **Actions > Synchronize Database**.

## Installation Overview

- 14** Ensure you are connected to Oracle as the `hue` user, then run the following command to delete all data from Oracle tables:

```
> set pagesize 100;
> SELECT 'DELETE FROM ' || table_name || ';' FROM user_tables;
```

- 15** Run the statements generated in the preceding step.

- 16** Commit your changes.

```
commit;
```

- 17** Load the data that you dumped. Go to the Hue Server instance and select **Actions > Load Database**. This step is not necessary if you have a fresh Hue install with no data or if you don't want to save the Hue data.

- 18** Start the Hue service.

### Configuring the Hue Server to Store Data in Oracle (Package Installation)

If you have a parcel-based environment, see [Configuring the Hue Server to Store Data in Oracle \(Parcel Installation\)](#).



**Important:** Configure the database for character set AL32UTF8 and national character set UTF8.

1. Download the Oracle 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.
2. Unzip the Oracle client zip files.
3. Set environment variables to reference the libraries.

```
$ export ORACLE_HOME=oracle_download_directory
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME
```

- 4.** Create a symbolic link for the shared object:

```
$ cd $ORACLE_HOME
$ ln -sf libclntsh.so.11.1 libclntsh.so
```

- 5.** Install the required packages.

#### RHEL

```
$ sudo yum install gcc python-devel python-pip python-setuptools libaio
```

#### SLES

```
$ sudo zypper install gcc python-devel python-pip python-setuptools libaio
```

#### Ubuntu or Debian

```
$ sudo apt-get install gcc python-devel python-pip python-setuptools libaio1
```

- 6.** For CDH versions lower than 5.3, install the Python Oracle library:



**Note:** `HUE_HOME` is a reference to the location of your Hue installation. For package installs, this is usually `/usr/lib/hue`; for parcel installs, this is usually, `/opt/cloudera/parcels/<parcel version>/lib/hue/`.

```
$ HUE_HOME/build/env/bin/pip install cx_Oracle
```

**7.** For CDH versions lower than 5.3, upgrade django south:

```
$ HUE_HOME/build/env/bin/pip install south --upgrade
```

- 8.** In the Cloudera Manager Admin Console, go to the Hue service status page.
- 9.** Select **Actions > Stop**. Confirm you want to stop the service by clicking **Stop**.
- 10** Select **Actions > Dump Database**. Confirm you want to dump the database by clicking **Dump Database**.
- 11** Click the **Configuration** tab.
- 12** Select **Scope > All**.
- 13** Select **Category > Advanced**.
- 14** Set the **Hue Service Environment Advanced Configuration Snippet (Safety Valve)** property to

```
ORACLE_HOME=oracle_download_directory
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$oracle_download_directory
```

**15** Set the **Hue Service Advanced Configuration Snippet (Safety Valve)** for **hue\_safety\_valve.ini** property.



**Note:** If you set **Hue Database Hostname**, **Hue Database Port**, **Hue Database Username**, and **Hue Database Password** at the service-level, under **Service-Wide > Database**, you can omit those properties from the server-lever configuration above and avoid storing the Hue password as plain text. In either case, set **engine** and **name** in the server-level safety-valve.

Add the following options (and modify accordingly for your setup):

```
[desktop]
[[database]]
host=localhost
port=1521
engine=oracle
user=hue
password=secretpassword
name=<SID of the Oracle database, for example, 'XE'>
```

For CDH 5.1 and higher you can use an Oracle service name. To use the Oracle service name instead of the SID, use the following configuration instead:

```
port=0
engine=oracle
user=hue
password=secretpassword
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}
```

**16** Grant required permissions to the `hue` user in Oracle:

```
GRANT CREATE <sequence> TO <user>;
GRANT CREATE <session> TO <user>;
GRANT CREATE <table> TO <user>;
GRANT CREATE <view> TO <user>;
GRANT CREATE <procedure> TO <user>;
GRANT CREATE <trigger> TO <user>;
GRANT EXECUTE ON sys.dbms_crypto TO <user>;
GRANT EXECUTE ON SYS.DBMS_LOB TO <user>;
```

## Installation Overview

17. Go to the Hue Server instance in Cloudera Manager and select **Actions > Synchronize Database**.
18. Ensure you are connected to Oracle as the `hue` user, then run the following command to delete all data from Oracle tables:

```
> set pagesize 100;
> SELECT 'DELETE FROM ' || table_name || ';' FROM user_tables;
```

19. Run the statements generated in the preceding step.

20. Commit your changes.

```
commit;
```

21. Load the data that you dumped. Go to the Hue Server instance and select **Actions > Load Database**. This step is not necessary if you have a fresh Hue install with no data or if you don't want to save the Hue data.

22. Start the Hue service.

### Configuring Oracle for Oozie

#### 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;
```

#### 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.

## Configuring an External Database for Oozie

The default database for Oozie is the embedded PostgreSQL database. You can also choose to use an external database. The databases that Oozie supports are listed at:

- [CDH 4 supported databases](#)
- [CDH 5 supported databases](#)

See the following sections for the procedures for setting one of the supported database types for Oozie and configuring database purge settings.

### Configuring PostgreSQL for Oozie

Install PostgreSQL 8.4.x or 9.0.x.

See [External PostgreSQL Database](#) on page 86.

### 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.UTF-8'
  LC_CTYPE = 'en_US.UTF-8'
  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
```

### Configuring MariaDB for Oozie

#### Install and Start MariaDB 5.5

See [MariaDB Database](#) on page 92.

## Installation Overview

### Create the Oozie Database and Oozie MariaDB User

For example, using the MariaDB mysql command-line tool:

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

mysql> create database oozie;
Query OK, 1 row affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> exit
Bye
```

### Add the MariaDB JDBC Driver JAR to Oozie

Cloudera recommends that you use the MySQL JDBC driver for MariaDB. Copy or symbolically link the MySQL JDBC driver JAR to the /var/lib/oozie/ directory.



**Note:** You must manually download the MySQL JDBC driver JAR file.

### Configuring MySQL for Oozie

#### Install and Start MySQL 5.x

See [MySQL Database](#) on page 98.

### 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;
Query OK, 1 row affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> exit
Bye
```

### 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 Oracle for Oozie

### 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;
```

## 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.

## Configuring Oozie Data Purge Settings

You can change your Oozie configuration to control when data is purged in order to improve performance, cut down on database disk usage, or to keep the history for a longer period of time. Limiting the size of the Oozie database can also improve performance during upgrades. See [Configuring Oozie Data Purge Settings Using Cloudera Manager](#).

### Configuring an External Database for Sqoop

Sqoop 2 has a built-in Derby database, but Cloudera recommends that you use a PostgreSQL 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 that do not happen with server-based databases.

See [Supported Databases](#) on page 32 for tested database versions.



**Note:**

There is currently no recommended way to migrate data from an existing Derby database into the new PostgreSQL database.

Use the procedure that follows to configure Sqoop 2 to use PostgreSQL instead of Apache Derby.

Install PostgreSQL 8.4.x or 9.0.x.

See [External PostgreSQL Database](#) on page 86.

#### Create the Sqoop User and Sqoop Database

```
$ psql -U postgres
Password for user postgres: *****

postgres=# CREATE ROLE sqoop LOGIN ENCRYPTED PASSWORD 'sqoop'
  NOSUPERUSER INHERIT CREATEDB NOCREATEROLE;
CREATE ROLE

postgres=# CREATE DATABASE "sqoop" WITH OWNER = sqoop
  ENCODING = 'UTF8'
  TABLESPACE = pg_default
  LC_COLLATE = 'en_US.UTF8'
  LC_CTYPE = 'en_US.UTF8'
  CONNECTION LIMIT = -1;
CREATE DATABASE

postgres=# \q
```

#### Configure Sqoop 2 to use PostgreSQL

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

1. Go to the Sqoop service.
2. Click the **Configuration** tab.
3. Select **Scope > Sqoop 2 Server**.
4. Select **Category > Database**.
5. Set the following properties:
  - Sqoop Repository Database Type - postgresql
  - Sqoop Repository Database Host - the hostname on which you installed the PostgreSQL server. If the port is non-default for your database type, use host:port notation.
  - Sqoop Repository Database Name, User, Password - the properties you specified in [Create the Sqoop User and Sqoop Database](#) on page 116.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

## 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.

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large.
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small.
- 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. Get the name, user, and password properties for the Cloudera Manager database from `/etc/cloudera-scm-server/db.properties`:

```
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 88, 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 MariaDB Databases

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

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

## Installation Overview

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 95, 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 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 102, 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.

### Database Vendor Resources

Use the following links to access vendor documentation on backing up and restoring databases.

- **MariaDB 5.5:** <http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/>
- **MySQL 5.5:** <http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html>
- **MySQL 5.6:** <http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html>
- **PostgreSQL 8.4:** <https://www.postgresql.org/docs/8.4/static/backup.html>
- **PostgreSQL 9.2:** <https://www.postgresql.org/docs/9.2/static/backup.html>
- **PostgreSQL 9.3:** <https://www.postgresql.org/docs/9.3/static/backup.html>
- **Oracle 11gR2:** [http://docs.oracle.com/cd/E11882\\_01/backup.112/e10642/toc.htm](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm)

## 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 120.

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)
- 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 120.

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 120.

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)

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 120.

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.

## Installation Overview

### 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 9: Small Clusters: No More Than 10 Hosts**

	<b>Required</b>	<b>Recommended</b>
Java Heap Size	256 MB	512 MB
Non-Java Memory	768 MB	1.5 GB

**Table 10: Medium Clusters: Between 11 and 100 Hosts**

	<b>Required</b>	<b>Recommended</b>
Java Heap Size	1 GB	2 GB
Non-Java Memory	2 GB	4 GB

**Table 11: Large Clusters: More Than 100 Hosts**

	<b>Required</b>	<b>Recommended</b>
Java Heap Size	2 GB	4 GB
Non-Java Memory	6 GB	12 GB

## Storage Space Planning for Cloudera Manager

### Minimum Required Role: [Full Administrator](#)

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. Note that storage locations and types have changed for some roles between Cloudera Manager 4 and 5.

## Installation Overview

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 12: Cloudera Manager Server**

Entity	Cloudera Manager Server Configuration
Default Storage Location	<b>RDBMS:</b> Use any supported RDBMS to store the core configuration of your Cloudera Manager database and all cluster, service, and role configurations.  <b>Disk:</b> Cloudera Manager Server Local Data Storage Directory (command_storage_path) 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.  /var/lib/cloudera-scm-server/
Storage Configuration Defaults, Minimum, or Maximum	There are no direct storage defaults relevant to this entity.
Where to Control Data Retention or Size	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 <b>Administration &gt; Settings</b> and edit the following property:  <b>Command Eviction Age</b> Length of time after which inactive commands are evicted from the database.  Default is two years.
Sizing, Planning & Best Practices	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.  You should perform regular, verified, remotely-stored backups of the Cloudera Manager Server database.

### Cloudera Management Service

**Table 13: Cloudera Management Service - Activity Monitor Configuration**

Entity	Activity Monitor
Default Storage Location	Any supported RDBMS.  See <a href="#">Cloudera Manager and Managed Service Datastores</a> on page 79.
Storage Configuration Defaults / Minimum / Maximum	Default: 14 Days worth of MapReduce (MRv1) jobs/tasks

Entity	Activity Monitor
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:</p> <ol style="list-style-type: none"> <li>1. Go the Cloudera Management Service.</li> <li>2. Click the <b>Configuration</b> tab.</li> <li>3. Select <b>Scope &gt; Activity Monitor or Cloudera Management Service (Service-Wide)</b>.</li> <li>4. Select <b>Category &gt; Main</b>.</li> <li>5. Locate the <b><i>propertyName</i></b> property or search for it by typing its name in the Search box.</li> </ol> <p><b>Purge Activities Data at This Age</b></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><b>Purge Attempts Data at This Age</b></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><b>Purge MapReduce Service Data at This Age</b></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> <p>6. Click <b>Save Changes</b> to commit the changes.</p>
Sizing, Planning, and Best Practices	<p>The Activity Monitor only monitors MapReduce jobs, and does not monitor YARN applications. If you no longer use MapReduce (MRv1) in your cluster, the Activity Monitor is not required for Cloudera Manager 5 (or higher) or CDH 5 (or higher).</p> <p>The amount of storage space needed for 14 days worth of MapReduce activities can vary greatly and directly depends on the size of your cluster and the level of activity that uses MapReduce. 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"> <li>• A simulated 1000-host cluster, each host with 32 slots</li> <li>• Synthetic MapReduce jobs with 200 attempts (tasks) per activity (job)</li> </ul> <p>Sizing observations for this cluster:</p> <ul style="list-style-type: none"> <li>• Each attempt takes 10 minutes to complete.</li> <li>• This usage results in roughly 20 thousand jobs a day with some 5 million total attempts.</li> <li>• For a retention period of 7 days, this Activity Monitor database required 200 GB.</li> </ul>

**Table 14: 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"> <li>• 10 GiB Services Time Series Storage</li> <li>• 1 GiB Impala Query Storage</li> <li>• 1 GiB YARN Application Storage</li> </ul> <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:</p> <ol style="list-style-type: none"> <li>1. Go the Cloudera Management Service.</li> <li>2. Click the <b>Configuration</b> tab.</li> <li>3. Select <b>Scope &gt; Service Monitor or Cloudera Management Service (Service-Wide)</b>.</li> <li>4. Select <b>Category &gt; Main</b>.</li> <li>5. Locate the <b>propertyName</b> property or search for it by typing its name in the Search box.</li> </ol> <p><b>Time-Series Storage</b></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 <a href="#">metric data</a> 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 <b>Cloudera Management Service &gt; Charts Library</b> 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><b>Impala Storage</b></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><b>YARN Storage</b></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</p>

Entity	Service Monitor Configuration
	<p>approximate because Cloudera Manager only begins deleting data when it reaches the limit.</p> <p><b>6.</b> Click <b>Save Changes</b> to commit the changes.</p>
Sizing, Planning, and Best Practices	<p>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.</p>

**Table 15: Cloudera Management Service - Host Monitor**

Entity	Host Monitor
Default Storage Location	<p>/var/lib/cloudera-host-monitor/ on the host where the Host Monitor role is configured to run.</p>
Storage Configuration Defaults / Minimum/ Maximum	<p>Default + Minimum: 10 GiB Host Time Series Storage</p>
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 <a href="#">Data Storage for Monitoring Data</a> on page 118.</p> <p>To configure these data retention in Cloudera Manager Administration Console:</p> <ol style="list-style-type: none"> <li><b>1.</b> Go the Cloudera Management Service.</li> <li><b>2.</b> Click the <b>Configuration</b> tab.</li> <li><b>3.</b> Select <b>Scope &gt; Host Monitor or Cloudera Management Service (Service-Wide)</b>.</li> <li><b>4.</b> Select <b>Category &gt; Main</b>.</li> <li><b>5.</b> Locate each property or search for it by typing its name in the Search box.</li> </ol> <p><b>Time-Series Storage</b></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 <a href="#">metric data</a> 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 <b>Cloudera Management Service &gt; Charts Library</b> 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> <p><b>6.</b> Click <b>Save Changes</b> to commit the changes.</p>

## Installation Overview

Entity	Host Monitor
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 16: Cloudera Management Service - Event Server**

Entity	Event Server
Default Storage Location	/var/lib/cloudera-scm-eventserver/ 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,</p> <ol style="list-style-type: none"> <li>1. Go the Cloudera Management Service.</li> <li>2. Click the <b>Configuration</b> tab.</li> <li>3. Select <b>Scope &gt; Event Server or Cloudera Management Service (Service-Wide)</b>.</li> <li>4. Select <b>Category &gt; Main</b>.</li> <li>5. Edit the following property:</li> </ol> <p><b>Maximum Number of Events in the Event Server Store</b></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> <ol style="list-style-type: none"> <li>6. Click <b>Save Changes</b> to commit the changes.</li> </ol>
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 <a href="#">matches a set of rules</a> for identifying messages of interest and makes them available for searching, filtering and additional action. You can view and filter events on the <b>Diagnostics &gt; Events</b> 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> <b>Note:</b> The Cloudera Management Service role Alert Publisher sources all the content for its work by regularly polling the Event Server for entries that are marked to be sent out using SNMP or SMTP(S). The Alert Publisher is not discussed because it has no noteworthy storage requirements of its own.</p> </div>

**Table 17: Cloudera Management Service - Reports Manager**

Entity	Reports Manager
Default Storage Location	<p><b>RDBMS:</b></p> <p>Any Supported RDBMS.</p> <p>See <a href="#">Installing and Configuring Databases</a>.</p>

Entity	Reports Manager
	<p><b>Disk:</b>  <code>/var/lib/cloudera-scm-headlamp/</code> on the host where the Reports Manager role is configured to run.</p>
Storage Configuration Defaults	<p><b>RDBMS:</b>  There are no exposed defaults or configurations to directly cull or purge the size of this data set.</p> <p><b>Disk:</b>  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 Manager for its local storage. Periodically monitor, review and readjust the local storage allocation.</p>

#### Cloudera Navigator

By default, during the Cloudera Manager Installation wizard the Navigator Audit Server and Navigator Metadata Server are assigned to the same host as the Cloudera Management Service monitoring roles. This configuration works for a small cluster, but should be updated before the cluster grows. You can either change the configuration at installation time or [move the Navigator Metadata Server](#) if necessary.

**Table 18: Cloudera Navigator - Navigator Audit Server**

Entity	Navigator Audit Server
Default Storage Location	Any Supported RDBMS. See <a href="#">Installing and Configuring Databases</a> .
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:</p> <ol style="list-style-type: none"> <li>1. Go the Cloudera Management Service.</li> <li>2. Click the <b>Configuration</b> tab.</li> <li>3. Select <b>Scope &gt; Navigator Audit Server or Cloudera Management Service (Service-Wide)</b>.</li> <li>4. Select <b>Category &gt; Main</b>.</li> </ol>

## Installation Overview

Entity	Navigator Audit Server
	<p>5. Locate the <b>Navigator Audit Server Data Expiration Period</b> property or search for it by typing its name in the Search box.</p> <p><b>Navigator Audit Server Data Expiration Period</b></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> <p>6. Click <b>Save Changes</b> to commit the changes.</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"> <li>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.</li> <li>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.</li> </ul> <p>Use this table to map <a href="#">Product Compatibility Matrix for Cloudera Navigator</a> versions to Cloudera Manager versions.</p>

**Table 19: Cloudera Navigator - Navigator Metadata Server**

Entity	Navigator Metadata Server
Default Storage Location	<p><b>RDBMS:</b> Any Supported RDBMS. See <a href="#">Installing and Configuring Databases</a>.</p> <p><b>Disk:</b> <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><b>RDBMS:</b> There are no exposed defaults or configurations to directly cull or purge the size of this data set.</p> <p><b>Disk:</b></p>

Entity	Navigator Metadata Server
	<p>There are no configuration defaults to influence the size of this location. You can change the location itself with the <b>Navigator Metadata Server Storage Dir</b> 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><b>RDBMS:</b></p> <p>There is no maximum size of this data and no way to purge data that is old.</p> <p><b>Disk:</b></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 <b>Navigator Metadata Server Storage Dir</b> property. To purge old data, see <a href="#">Managing Metadata Capacity</a>.</p>
Sizing, Planning, and Best Practices	<p><b>Memory:</b></p> <p>Two activities determine Navigator Metadata Server resource requirements:</p> <ul style="list-style-type: none"> <li>• Extracting metadata from the cluster and creating relationships</li> <li>• Querying</li> </ul> <p>The Navigator Metadata Server uses Solr to store, index, and query metadata. Indexing happens during extraction. Querying is fast and efficient because the data is indexed. The Navigator Metadata Server memory requirements are based on amount of data that is stored and indexed. The Solr instance runs in process with Navigator, so you should set Java heap for the Navigator Metadata Server accordingly.</p> <p>When the Navigator Metadata Server starts up it logs the number of documents contained in Solr. For example:</p> <pre data-bbox="691 1184 1475 1360">2015-11-11 09:24:58,013 INFO com.cloudera.nav.server.NavServerUtil: Found 68813088 documents in solr core nav_elements 2015-11-11 09:24:58,705 INFO com.cloudera.nav.server.NavServerUtil: Found 78813930 documents in solr core nav_relations</pre> <p>To compute the memory required by the Metadata Server during <i>normal operation</i>, use the number of documents in <code>nav_elements</code> * 200. So for the above example, the recommended amount of memory would be <math>(68813088 * 200)</math> or about 14 GB.</p> <p>For <i>upgrade</i>, use the number of documents in <code>nav_elements</code> + <code>nav_relations</code>. If you use the number in the above example, for upgrade you would need <math>((68813088 + 78813930) * 200)</math> or about 30 GB.</p> <p><b>RDBMS:</b></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 <a href="#">matrix</a> to map Cloudera Navigator and Cloudera Manager versions.</p> <p><b>Disk:</b></p>

Entity	Navigator Metadata Server
	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.

**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 20: 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 (<code>cloudera-scm-server</code>) 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 using the Cloudera Manager Administration Console or the Cloudera Manager API.</p> <p><b>Sizing and Planning</b></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 <a href="#">Parcel Configuration Settings</a> on page 63.</p> <p>Provide sufficient space to hold all the parcels you download from all configured Remote Parcel Repository URLs (See <a href="#">Parcel Configuration Settings</a> on page 63). 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><b>Lifecycle Management and Best Practices</b></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><b>Backup Considerations</b></p>

Parcel Lifecycle Path (default)	Notes
	<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><b>Sizing and Planning</b></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 <code>.parcel</code> 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 <b>Administration &gt; Settings &gt; Parcels &gt; Retain Downloaded Parcel Files</b></p>
Host Parcel Directory <code>/opt/cloudera/parcels</code>	<p>Managed cluster hosts running a Cloudera Manager Agent extract parcels from the <code>/opt/cloudera/parcel-cache</code> 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><b>Sizing and Planning</b></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 <b>Administration &gt; Settings &gt; Parcels</b> and configure the following property:</p> <p><b>Automatically Remove Old Parcels</b></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. The default value is Disabled.</p> <p><b>Number of Old Parcel Versions to Retain</b></p> <p>If you enable <b>Automatically Remove Old Parcels</b>, 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. The default value is 3.</p>

**Table 21: 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: In your pre-upgrade planning, you likely saw a <a href="#">warning in the Upgrade Guide</a> advising that Cloudera Manager did this migration work for you automatically. The Service Monitor and Host Monitor are 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 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.  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 <a href="#">using the documented recommendations</a> .
Ongoing Space Reclamation	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.

### 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.

## Installation Path A - Automated Installation by Cloudera Manager (Non-Production Mode)



**Important:** Path A installation is intended for demonstrations and proof-of-concept deployments only. Do not use this method of installation for production environments.

Before proceeding with this path for a new installation, review [Cloudera Manager Deployment](#) on page 74. If you are upgrading an existing Cloudera Manager installation, see [Upgrading Cloudera Manager](#) on page 466.

In Installation Path A, Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, Cloudera Manager Agent, CDH, and managed service software on cluster hosts. Cloudera Manager also 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 14 for further information.
- All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.

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

## Before You Begin

### Install and Configure Databases

By default, Installation Path A installs an embedded PostgreSQL database. You can also choose to configure an external database. Read [Cloudera Manager and Managed Service Datastores](#) on page 79. 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 Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 83.

### Perform Configuration Required by Single User Mode

If you are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 17.

### (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 system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue.

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>

### Install the Oracle JDK

If you choose not to have the Oracle JDK installed by Cloudera Manager, install the JDK on all hosts in the cluster according to the following instructions:

- CDH 5 - [Java Development Kit Installation](#) on page 78.

## Installation Overview

- CDH 4 - [Java Development Kit Installation](#).

### Download and Run the Cloudera Manager Server Installer

Download the Cloudera Manager installer to the cluster host where you want to install the Cloudera Manager Server:

1. Open [Cloudera Manager Downloads](#) in a web browser.
2. In the **Cloudera Manager** box, click **Download Now**.
3. Click **Download Cloudera Manager** to download the most recent version of the installer or click **Select a Different Version** to download an earlier version.

The product interest dialog box displays.

4. Click **Sign in** and enter your email address and password or complete the product interest form and click **Continue**.

The **Cloudera Standard License** page displays.

5. Accept the license agreement and click **Submit**.

The **Automated Installation** instructions display. You can also view system requirements, release notes, and you can go to the documentation.

6. Download the installer:

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

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

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

8. Run the Cloudera Manager Server installer by doing one of the following:

- 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
```

9. Read the Cloudera Manager README and then press **Return** or **Enter** to choose **Next**.

10. 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.

11. Read the Oracle Binary Code License Agreement and then press **Return** or **Enter** to choose **Next**.

12. 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:

- a. The installer installs the Oracle JDK and the Cloudera Manager repository files.
- b. The installer installs the Cloudera Manager Server and embedded PostgreSQL packages.
- c. The installer starts the Cloudera Manager Server and embedded PostgreSQL database.

13. When the installation completes, the complete URL for the Cloudera Manager Admin Console displays, including the port number, which is 7180 by default. Press **Return** or **Enter** to choose **OK** to continue.

14. 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 199.

## 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 start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.
  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. Although 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.
  4. After logging in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
  5. Click **Continue**.

The **Welcome to Cloudera Manager** page displays.

## 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 install the Oracle JDK on the cluster hosts.
- 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 174 for more information about this option.

### Choose Cloudera Manager Edition

From the **Welcome to Cloudera Manager** page, you can select the edition of Cloudera Manager to install and, optionally, install a license:

1. Choose which [edition](#) to install:
  - Cloudera Express, which does not require a license, but provides a 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:

## Installation Overview

- Basic Edition
- Flex Edition
- Data Hub Edition

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

### 2. If you elect 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. Go to the location of your license file, click the file, and click **Open**.
  - d. Click **Upload**.
3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.
  4. Click **Continue** to proceed with the installation.

## Choose Cloudera Manager Hosts

Choose which hosts will run CDH and managed services:

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 with 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 do not 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.

## 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. In the **Choose Method** section select one of the following:
  - **Use Parcels**

- 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.

- To specify the **Parcel Directory** or **Local Parcel Repository Path**, 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 63 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.

- Click **OK**. Parcels available from the configured remote parcel repository URLs are displayed in the parcels list. If you specify a URL for a parcel version too new to be supported by the Cloudera Manager version, the parcel is not displayed in the parcel list.

- Use Packages**

- Select the major release of CDH to install.
- Select the specific release of CDH to install. You can choose either the latest version, a specific version, or use a custom repository. If you specify a custom repository for a CDH version too new to be supported by the Cloudera Manager version, Cloudera Manager will install the packages but you will not be able to create services using those packages and will have to manually uninstall those packages and manually reinstall packages for a supported CDH version.
- Select the specific releases of other services to install. You can choose either the latest version or use a custom repository. Choose **None** if you do not want to install that service.
- If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. 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 **Cluster Installation JDK Installation Options** screen displays.

- Select **Install Oracle Java SE Development Kit (JDK)** to allow Cloudera Manager to install the JDK on each cluster host. If you have already installed the JDK, do not select this option. If 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.



**Note:** If you already manually installed the JDK on each cluster host, this option to install the JDK does not display.

- Click **Continue**.
- (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

## Installation Overview

create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**.

**7.** Specify host installation properties:

- Select **root** or enter the username for an account that has password-less sudo permission.
- Select an authentication method:
  - If you choose password authentication, enter and confirm the password.
  - If you choose public-key authentication, provide a passphrase and path to the required key files.
- You can 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.

**8.** 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 parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you 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.

**9.** 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**.

## Add Services

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

- Select the combination of services to install:

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

As you select 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 lifecycle.

- 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 [Cloudera Navigator 2 Overview](#).

## 2. Click **Continue**.

3. 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. You can reassigned role instances if necessary.

Click a field below a role to display a dialog box 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 box.

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.

## 4. When you are satisfied with the assignments, click **Continue**.

## Configure Database Settings

### 1. Choose the database type:

- Keep the default setting of **Use Embedded Database** to have Cloudera Manager create and configure required databases. Record the auto-generated passwords.

## Installation Overview

### 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:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL	hive	hive	t56lwbdk4F

✓ Skipped. Cloudera Manager will create this database in a later step.

✓ Successful

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL	rman	rman	Y6S4IWvfNo

#### Navigator Audit Server

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL	nav	nav	QLR2B0qqO9

✓ Successful

#### Navigator Metadata Server

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL	navms	navms	lmo07jxOen

✓ Successful

#### Oozie Server

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL	oozie_oozie_se	oozie_oozie_se	NTF1KNdpPl

✓ Skipped. Cloudera Manager will create this database in a later step.

Test Connection

- Select **Use Custom Databases** to specify external database host, enter the database type, database name, username, and password for the database that you created when you set up the database.
- If you are adding the Oozie service, you can change your Oozie configuration to control when data is purged in order to improve performance, cut down on database disk usage, improve upgrade performance, or to keep the history for a longer period of time. See [Configuring Oozie Data Purge Settings Using Cloudera Manager](#).

2. 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 **Cluster Setup Review Changes** screen displays.

### Review Configuration Changes and Start Services

1. 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. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.



**Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

## 2. Click **Continue**.

The wizard starts the services.

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

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

## Change the Default Administrator Password

As soon as possible, change the default administrator password:

- 1.** 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 **OK**.

## Configure Oozie Data Purge Settings

If you added an Oozie service, you can change your Oozie configuration to control when data is purged in order to improve performance, cut down on database disk usage, or to keep the history for a longer period of time. Limiting the size of the Oozie database can also improve performance during upgrades. See [Configuring Oozie Data Purge Settings Using Cloudera Manager](#).

## Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 198.

## Installation Path B - Installation Using Cloudera Manager Parcels or Packages

Installation Path B installs Cloudera Manager using packages downloaded from a repository. There are several options for installing the JDK, Agents, CDH, and Managed Service packages:

- Install these items manually using packages. You can use utilities such as Puppet or Chef to help with the installation of these items across all the hosts in a cluster.
- Cloudera Manager can install them for you on all of the hosts in your cluster. If you choose Cloudera Manager installation, you can select installation using *packages* or Cloudera Manager *parcels*. 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 14 for further information.
  - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.

You can also install Cloudera Manager and CDH using tarballs. See [Installation Path C - Manual Installation Using Cloudera Manager Tarballs](#) on page 157.

Before proceeding with this path for a new installation, review [Cloudera Manager Deployment](#) on page 74. If you are upgrading a Cloudera Manager existing installation, see [Upgrading Cloudera Manager](#) on page 466.

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

## Before You Begin

### Perform Configuration Required by Single User Mode

If you are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 17.

## Installation Overview

### (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 system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue.

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 and Configure External Databases

Read [Cloudera Manager and Managed Service Datastores](#) on page 79. Install and configure an external database for services or Cloudera Management Service roles using the instructions in [External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 83.

Cloudera Manager also requires a database. Prepare the Cloudera Manager Server database as described in [Preparing a Cloudera Manager Server External Database](#) on page 80.

### Establish Your Cloudera Manager Repository Strategy

Cloudera recommends installing products using package management tools such as `yum` for RHEL 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 .
- 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 170. 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.

#### RHEL-compatible

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

OS Version	Repo URL
RHEL/CentOS/Oracle 5	<a href="https://archive.cloudera.com/cm5/redhat/5/x86_64/cm/cloudera-manager.repo">https://archive.cloudera.com/cm5/redhat/5/x86_64/cm/cloudera-manager.repo</a>
RHEL/CentOS 6	<a href="https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo">https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo</a>
RHEL/CentOS 7	<a href="https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo">https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo</a>

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

#### SLES

1. Run the following command:

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

2. Update your system package index by running:

```
$ sudo zypper refresh
```

## Ubuntu or Debian

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

OS Version	Repo URL
Ubuntu Trusty (14.04)	<a href="https://archive.cloudera.com/cm5/ubuntu/trusty/amd64/cm/cloudera.list">https://archive.cloudera.com/cm5/ubuntu/trusty/amd64/cm/cloudera.list</a>
Ubuntu Precise (12.04)	<a href="https://archive.cloudera.com/cm5/ubuntu/precise/amd64/cm/cloudera.list">https://archive.cloudera.com/cm5/ubuntu/precise/amd64/cm/cloudera.list</a>
Ubuntu Lucid (10.04)	<a href="https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm/cloudera.list">https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm/cloudera.list</a>
Debian Wheezy (7.0 and 7.1)	<a href="https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm/cloudera.list">https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm/cloudera.list</a>
Debian Squeeze (6.0)	<a href="https://archive.cloudera.com/cm5/debian/squeeze/amd64/cm/cloudera.list">https://archive.cloudera.com/cm5/debian/squeeze/amd64/cm/cloudera.list</a>

- Copy the content of that file to the `cloudera-manager.list` file in the `/etc/apt/sources.list.d/` directory.

- Update your system package index by running:

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

## Install Cloudera Manager Server Software

In this step you install the JDK and Cloudera Manager Server packages on the Cloudera Manager host.

### Install the Oracle JDK on the Cloudera Manager Server Host

Install the Oracle Java Development Kit (JDK) on the Cloudera Manager Server host. You can install the JDK from a repository, or you can download the JDK from Oracle and install it yourself:

- Install the JDK from a repository

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	\$ sudo yum install oracle-j2sdk1.7
SLES	\$ sudo zypper install oracle-j2sdk1.7
Ubuntu or Debian	\$ sudo apt-get install oracle-j2sdk1.7

- Install the JDK manually

See [Java Development Kit Installation](#) on page 78.

### Install the Cloudera Manager Server Packages

- 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	\$ sudo yum install cloudera-manager-daemons cloudera-manager-server
RHEL, if you're manually transferring RPMs	\$ sudo yum --nogpgcheck localinstall cloudera-manager-daemons-*.*.rpm \$ sudo yum --nogpgcheck localinstall cloudera-manager-server-*.*.rpm

## Installation Overview

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

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`.

### (Optional) Manually Install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Service Packages

You can use Cloudera Manager to install the Oracle JDK, Cloudera Manager Agent packages, CDH, and managed service packages or you can install any of these packages manually. To use Cloudera Manager to install the packages, you must meet the requirements described in [Cloudera Manager Deployment](#) on page 74.



**Important:** If you are installing CDH and managed service software using packages and you want to manually install Cloudera Manager Agent or CDH packages, you must manually install them both following the procedures in this section; you cannot choose to install only one of them this way.

If you are going to use Cloudera Manager to install all of the software, *skip this section* and continue with [Start the Cloudera Manager Server](#) on page 145. Otherwise, to manually install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Services, continue with the procedures linked below and then return to this page to continue the installation. in this section. You can choose to manually install any of the following software and, in a later step, Cloudera Manager installs any software that you do not install manually:

#### Manually Install the Oracle JDK

You can use Cloudera Manager to install the Oracle JDK on all cluster hosts or you can install the JDKs manually. If you choose to have Cloudera Manager install the JDKs, *skip this section*. To use Cloudera Manager to install the JDK, you must meet the requirements described in [Cloudera Manager Deployment](#) on page 74.

Install the Oracle JDK on every 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 78.
- CDH 4 - [Java Development Kit Installation](#).

#### Manually Install Cloudera Manager Agent Packages

The Cloudera Manager **Agent** is responsible for starting and stopping processes, unpacking configurations, triggering installations, and monitoring all hosts in a cluster. You can install the Cloudera Manager agent manually on all hosts, or Cloudera Manager can install the Agents in a later step. To use Cloudera Manager to install the agents, skip this section and continue with

To install the Cloudera Manager Agent packages manually, do the following on every cluster 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:	\$ sudo yum install cloudera-manager-agent cloudera-manager-daemons
RHEL, if you're manually transferring RPMs:	\$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons

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

2. On every cluster 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](#).

3. Start the Agents by running the following command on all hosts:

```
$ 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 742. When the Agent hosts reboot, `cloudera-scm-agent` starts automatically.

#### Manually Install CDH and Managed Service Packages

The CDH and Managed Service Packages contain all of the CDH software. You can choose to manually install CDH and the Managed Service Packages, or you can choose to let Cloudera Manager perform this installation in a later step. To use Cloudera Manager perform the installation, continue with [Start the Cloudera Manager Server](#) on page 145. Otherwise, follow the steps in [\(Optional\) Manually Install CDH and Managed Service Packages](#) on page 151 and then return to this page to continue the installation.

#### 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 742.

## Installation Overview

### 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 start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.
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. Although 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.
4. After logging in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

The **Welcome to Cloudera Manager** page displays.

### Choose Cloudera Manager Edition

From the **Welcome to Cloudera Manager** page, you can select the edition of Cloudera Manager to install and, optionally, install a license:

1. Choose which [edition](#) to install:
  - Cloudera Express, which does not require a license, but provides a 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 upgrade the license at a later time. See [Managing Licenses](#).

2. If you elect 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. Go to the location of your license file, click the file, and click **Open**.
  - d. Click **Upload**.
3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.
4. Click **Continue** to proceed with the installation.

### Choose Cloudera Manager Hosts

Choose which hosts will run CDH and managed services

1. Do one of the following depending on whether you are using Cloudera Manager to install software:

- If you are using Cloudera Manager to install software, 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 with 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 do not 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.
- If you installed Cloudera Agent packages in [Manually Install Cloudera Manager Agent Packages](#) on page 144, choose from among hosts with the packages installed:
    1. Click the **Currently Managed Hosts** tab.
    2. Choose the hosts to add to the cluster.

## 2. Click **Continue**.

The **Cluster Installation Select Repository** screen displays.

## Choose the Software Installation Type and Install Software

Choose a software installation type (parcels or packages) and install the software. If you have already installed the CDH and Managed Service packages, you cannot choose **Parcel** installation.



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

1. Choose the software installation type and CDH and managed service version:
  - **Use Parcels**
    1. Choose the parcels to install. The choices depend on the repositories you have chosen; a repository can contain multiple parcels. Only the parcels for the latest supported service versions are configured by default.

## Installation Overview

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, specify the 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 63 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you 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 are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. 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.

- **Use Packages** - Do one of the following:

- If Cloudera Manager is installing the packages:

1. Click the package version.

2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. 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.

- If you manually installed packages in [Manually Install CDH and Managed Service Packages](#) on page 145 , select the CDH version (CDH 4 or CDH 5) that matches the packages you installed manually.

2. If you installed the Agent and JDK manually on all cluster hosts:

- 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**.

- Skip the remaining steps in this section and continue with [Add Services](#) on page 149

3. Select **Install Oracle Java SE Development Kit (JDK)** to allow Cloudera Manager to install the JDK on each cluster host. If you have already installed the JDK, do not select this option. If 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.



**Note:** If you already manually installed the JDK on each cluster host, this option to install the JDK does not display.

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**.
5. If you chose to have Cloudera Manager install software, specify host installation properties:
  - Select **root** or enter the username for an account that has password-less sudo permission.
  - Select an authentication method:
    - If you choose password authentication, enter and confirm the password.
    - If you choose public-key authentication, provide a passphrase and path to the required key files.
  - You can 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 chose to have Cloudera Manager install software, Cloudera Manager installs the Oracle JDK, Cloudera Manager Agent, packages and CDH and managed service parcels or packages. During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.
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**.

## Add Services

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

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

As you select 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.

## Installation Overview

- 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 lifecycle.

- 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 [Cloudera Navigator 2 Overview](#).

### 2. Click **Continue**.

3. 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. You can reassign role instances if necessary.

Click a field below a role to display a dialog box 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 box.

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.

4. When you are satisfied with the assignments, click **Continue**.

## Configure Database Settings

On the Database Setup page, configure settings for required 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.
2. 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 Configuration Changes and Start Services

1. 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. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.



**Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

## 2. Click **Continue**.

The wizard starts the services.

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

## Change the Default Administrator Password

As soon as possible, change the default administrator password:

1. 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 **OK**.

## Configure Oozie Data Purge Settings

If you added an Oozie service, you can change your Oozie configuration to control when data is purged in order to improve performance, cut down on database disk usage, or to keep the history for a longer period of time. Limiting the size of the Oozie database can also improve performance during upgrades. See [Configuring Oozie Data Purge Settings Using Cloudera Manager](#).

## Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 198.

## (Optional) Manually Install CDH and Managed Service Packages

The procedures in this topic are an optional part of the Path B Installation instructions. Begin with the steps in [Installation Path B - Installation Using Cloudera Manager Parcels or Packages](#) on page 141 before following the steps in this topic. For an overview of the installation process, see [Installation Overview](#) on page 74.

The CDH and Managed Service Packages contain all of the CDH software. You can choose to manually install CDH and the Managed Service Packages, or you can choose to let Cloudera Manager perform this installation in a later step. Otherwise, follow the steps in this topic to manually install CDH and Managed Service packages and then continue the installation with [Start the Cloudera Manager Server](#) on page 145.



**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 Using Cloudera Manager](#) on page 55.

Choose one of the following procedures, depending which version of CDH you are installing:

- [Install CDH 5 and Managed Service Packages](#) on page 151
- [Install CDH 4, Impala, and Solr Managed Service Packages](#) on page 154

### Install CDH 5 and Managed Service Packages

Install the packages on all cluster hosts using the following steps:

- **Red Hat**

1. Download and install the "1-click Install" package.
  - a. Download the CDH 5 "1-click Install" package (or RPM).

## Installation Overview

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

- b. Install the RPM for all RHEL versions:

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

2. (Optional) add a repository key:

- Red Hat/CentOS/Oracle 5

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

- Red Hat/CentOS/Oracle 6

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

3. Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python  
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

- SLES

1. Download and install the "1-click Install" package.

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

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

- b. Install the RPM:

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

- c. Update your system package index by running:

```
$ sudo zypper refresh
```

## 2. (Optional) add a repository key:

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

## 3. Install the CDH packages:

```
$ 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 hive-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

### • Ubuntu and Debian

#### 1. Download and install the "1-click Install" package

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

##### b. 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
```

## 2. Optionally add a repository key:

### • Debian Wheezy

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

### • Ubuntu Precise

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

## 3. Install the CDH packages:

```
$ 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-webhcate 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
```

## Installation Overview

```
solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python  
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

### Install CDH 4, Impala, and Solr Managed Service Packages

Install the packages on all cluster hosts using the following steps:

- **RHEL-compatible**

1. Click the entry in the table at [CDH Download Information](#) that matches your RHEL or CentOS system.
2. Go to the repo file (`cloudera-cdh4.repo`) for your system and save it in the `/etc/yum.repos.d/` directory.
3. Optionally add a repository key:

- **RHEL/CentOS/Oracle 5**

```
$ sudo rpm --import  
https://archive.cloudera.com/cdh4/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- **RHEL/CentOS 6**

```
$ sudo rpm --import  
https://archive.cloudera.com/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
```

4. Install packages on every host in your cluster:

- a. Install CDH 4 packages:

```
$ 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
```

- b. To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo yum install hue
```

5. **(Requires CDH 4.2 and higher)** Install Impala

- In the table at [Cloudera Impala Version and Download Information](#), click the entry that matches your RHEL or CentOS system.
- Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
- Install Impala and the Impala Shell on Impala machines:

```
$ sudo yum -y install impala impala-shell
```

6. **(Requires CDH 4.3 and higher)** Install Search

- In the table at [Cloudera Search Version and Download Information](#), click the entry that matches your RHEL or CentOS system.
- Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
- Install the Solr Server on machines where you want Cloudera Search.

```
$ sudo yum -y install solr-server
```

- SLES

1. Run the following command:

```
$ sudo zypper addrepo -f
https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/cloudera-cdh4.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

3. Optionally add a repository key:

```
$ sudo rpm --import
https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
```

4. Install packages on every host in your cluster:

- a. Install CDH 4 packages:

```
$ 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
```

- b. To install the hue-common package and all Hue applications on the Hue host, install the hue meta-package:

```
$ sudo zypper install hue
```

- c. (Requires CDH 4.2 and higher) Install Impala

- a. Run the following command:

```
$ sudo zypper addrepo -f
https://archive.cloudera.com/impala/sles/11/x86_64/impala/cloudera-impala.repo
```

- b. Install Impala and the Impala Shell on Impala machines:

```
$ sudo zypper install impala impala-shell
```

- d. (Requires CDH 4.3 and higher) Install Search

- a. Run the following command:

```
$ sudo zypper addrepo -f
https://archive.cloudera.com/search/sles/11/x86_64/search/cloudera-search.repo
```

- b. Install the Solr Server on machines where you want Cloudera Search.

```
$ sudo zypper install solr-server
```

- Ubuntu or Debian

1. In the table at [CDH Version and Packaging Information](#), click the entry that matches your Ubuntu or Debian system.

## Installation Overview

2. Go to the list file (`cloudera.list`) for your system and save it in the `/etc/apt/sources.list.d/` directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your `cloudera.list` file should look like:

```
deb [arch=amd64] https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4  
contrib  
deb-src https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib
```

3. Optionally add a repository key:

- Ubuntu Lucid

```
$ curl -s https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh/archive.key | sudo  
apt-key add -
```

- Ubuntu Precise

```
$ curl -s https://archive.cloudera.com/cdh4/ubuntu/precise/amd64/cdh/archive.key | sudo  
apt-key add -
```

- Debian Squeeze

```
$ curl -s https://archive.cloudera.com/cdh4/debian/squeeze/amd64/cdh/archive.key | sudo  
apt-key add -
```

4. Install packages on every host in your cluster:

- Install CDH 4 packages:

```
$ 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
```

- To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo apt-get install hue
```

- (Requires CDH 4.2 and higher) Install Impala

- In the table at [Cloudera Impala Version and Download Information](#), click the entry that matches your Ubuntu or Debian system.
- Go to the list file for your system and save it in the `/etc/apt/sources.list.d/` directory.
- Install Impala and the Impala Shell on Impala machines:

```
$ sudo apt-get install impala impala-shell
```

- (Requires CDH 4.3 and higher) Install Search

- In the table at [Cloudera Search Version and Download Information](#), click the entry that matches your Ubuntu or Debian system.
- Install Solr Server on machines where you want Cloudera Search:

```
$ sudo apt-get install solr-server
```

Continue the installation with [Start the Cloudera Manager Server](#) on page 145.

## Installation Path C - Manual Installation Using Cloudera Manager Tarballs

Before proceeding with this path for a new installation, review [Cloudera Manager Deployment](#) on page 74. If you are upgrading an existing Cloudera Manager installation, see [Upgrading Cloudera Manager](#) on page 466.

In this procedure, 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. For a full discussion of deployment options, see [Installation Overview](#) on page 74.

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

### Before You Begin

#### Install the Oracle JDK

See [Java Development Kit Installation](#) on page 78.

#### Install and Configure External Databases

Read [Cloudera Manager and Managed Service Datastores](#) on page 79. Install and configure an external database for services or Cloudera Management Service roles using the instructions in [External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server](#) on page 83.

Cloudera Manager also requires a database. Prepare the Cloudera Manager Server database as described in [Preparing a Cloudera Manager Server External Database](#) on page 80.

#### (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 system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue.

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
```

Extract the contents of the tarball, to this directory. 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 be extracted to /opt/cloudera-manager/cm-5.0/. This full path is needed later and is referred to as *tarball\_root* directory.

## Installation Overview

### Perform Configuration Required by Single User Mode

If you are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 17.

#### Create Users

The Cloudera Manager Server and managed services require 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. This created user, is used automatically after installation is complete.

To create user `cloudera-scm`, use a command such as the following:

```
$ sudo useradd --system --home=/opt/cloudera-manager/cm-5.6.0/run/cloudera-scm-server  
--no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm
```

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-5.6.0/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 installation has a `var` subdirectory where state is stored. In a non-tarball installation, state is stored in `/var`. Cloudera recommends that you reconfigure the tarball installation 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 installation 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 do not reuse the state directory between different tarball installations, duplicate Cloudera Manager Agent entries can occur in the Cloudera Manager database.

#### Configuring for a Custom Cloudera Manager User and Custom Directories

You can change the default username and directories used by Cloudera Manager. If you do not change the default, skip to [Cloudera Manager and Managed Service Datastores](#) on page 79. By default, Cloudera Manager creates the following directories in `/var/log` and `/var/lib`:

- `/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 username and custom directories for Cloudera Manager, you must create these directories on the Cloudera Manager Server host and assign ownership of these directories to the custom username. Cloudera Manager installer makes no changes to any directories that already exist. Cloudera Manager cannot write to any existing directories for which it does not have proper permissions, and if you do not change ownership, Cloudera Management Service roles may not perform as expected. To resolve these issues, do one of the following:

- **Change ownership of existing directories:**

1. Use the `chown` command to change ownership of all existing directories to the Cloudera Manager user. If the Cloudera Manager username and group are `cloudera-scm`, to change the ownership of the headlamp log directory, you issue a command similar to the following:

```
$ sudo chown -R cloudera-scm:cloudera-scm /var/log/cloudera-scm-headlamp
```

- **Use alternate directories:**

1. If the directories you plan to use do not exist, create them. For example, to create `/var/cm_logs/cloudera-scm-headlamp` for use by the `cloudera-scm` user, you can use the following commands:

```
mkdir /var/cm_logs/cloudera-scm-headlamp
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.

## Create Parcel Directories

1. On the Cloudera Manager Server host, create a parcel repository directory:

```
$ sudo mkdir -p /opt/cloudera/parcel-repo
```

## Installation Overview

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 run the command:

```
$ sudo 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 run 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 this case, you have the 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 742.

## Start the Cloudera Manager Agents

Start the Cloudera Manager Agent according to the 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, to run the Cloudera Manager Agent as `cloudera-scm`, you have the 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, and then run the script as root:

1. Remove the following line from `tarball_root/etc/default/cloudera-scm-agent`:

```
export CMF_SUDO_CMD= "
```

## Installation Overview

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 you install with tarballs and parcels, some services may require additional dependencies that are not provided by Cloudera. On each host, install the required packages:

- **Red-hat compatible**

- `chkconfig`
- `python` (2.6 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` (2.6 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
- 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 start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.
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. Although 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.
4. After logging in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

The **Welcome to Cloudera Manager** page displays.

## Choose Cloudera Manager Edition

From the **Welcome to Cloudera Manager** page, you can select the edition of Cloudera Manager to install and, optionally, install a license:

1. Choose which [\*\*edition\*\*](#) to install:
  - Cloudera Express, which does not require a license, but provides a 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.

## Installation Overview

- 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 upgrade the license at a later time. See [Managing Licenses](#).

### 2. If you elect 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. Go to the location of your license file, click the file, and click **Open**.
- d. Click **Upload**.

### 3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.

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

## Choose Cloudera Manager Hosts

1. Click the **Currently Managed Hosts** tab.
2. Choose the hosts to add to the cluster.
3. Click **Continue**.

The **Cluster Installation Select Repository** screen displays.

## Install CDH and Managed Service Software

### 1. Install CDH and managed services using parcels:

#### a. Use Parcels

- a. Choose the parcels to install. The choices depend on the repositories you have chosen; a repository can 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, specify the 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 63 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you see all the unique parcels contained in all your repositories.
- **Proxy Server** - Specify the properties of a proxy server.

2. Click **OK**.

- b.** If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. 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.
  - b.** Click **Continue**. Cloudera Manager installs the CDH and managed service parcels. During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you 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**.

## Add Services

- In the first page of the Add Services wizard, choose the combination of services to install and whether to install Cloudera Navigator:

- Select the combination of services to install:

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

As you select 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 lifecycle.

- 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 [Cloudera Navigator 2 Overview](#).

## Installation Overview

### 2. Click **Continue**.

3. 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. You can reassigned role instances if necessary.

Click a field below a role to display a dialog box 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 box.

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.

### 4. When you are satisfied with the assignments, click **Continue**.

## Configure Database Settings

On the Database Setup page, configure settings for required 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.
2. 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 Configuration Changes and Start Services

1. 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. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.



**Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

### 2. Click **Continue**.

The wizard starts the services.

3. When all of the services are started, click **Continue**. You see a success message indicating that your cluster has been successfully started.
4. 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 automatically starts the Cloudera Management Service, assuming that it runs using `cloudera-scm`. If you configured this service to run using a user other than `cloudera-scm`, the Cloudera Management Service roles do not start automatically. To change the service configuration to use the user account that you selected:

1. Connect to the Cloudera Manager Admin Console.
2. Do one of the following:
  - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
  - On the **Home > Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
3. Click the **Configuration** tab.
4. Use the search box to find the property to change. For example, you might enter "system" to find the **System User** and **System Group** properties.
5. Make any changes required to the System User and System Group to ensure Cloudera Manager uses the proper user accounts.
6. Click **Save Changes**.
7. Start the Cloudera Management Service roles.

## Change the Default Administrator Password

As soon as possible, change the default administrator password:

1. 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 **OK**.

## Configure Oozie Data Purge Settings

If you added an Oozie service, you can change your Oozie configuration to control when data is purged in order to improve performance, cut down on database disk usage, or to keep the history for a longer period of time. Limiting the size of the Oozie database can also improve performance during upgrades. See [Configuring Oozie Data Purge Settings Using Cloudera Manager](#).

## Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 198.

## Installing Impala

Cloudera Impala is included with CDH 5. In a parcel-based configuration, it is part of the CDH parcel rather than a separate parcel. Starting with CDH 5.4 (corresponding to Impala 2.2 in the Impala versioning scheme) new releases of Impala are only available on CDH 5, not CDH 4.

Although these installation instructions primarily focus on CDH 5, you can also manage CDH 4 clusters using Cloudera Manager 5. In CDH 4, Impala has packages and parcels that you download and install separately from CDH. 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 12 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 74.

## Installation Overview

### 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 55 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 63 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, which is included with CDH 5. There are various ways to install Search with CDH. For example, you might select **Core with Search** or **All Services** as described in Add Services in [Installation Path A - Automated Installation by Cloudera Manager \(Non-Production Mode\)](#) on page 132.



#### Note:

- See [Supported CDH and Managed Service Versions](#) on page 12 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 74.

Cloudera Search supports one instance of the Solr service on each host in a cluster. Using multiple Solr instances on a host is not supported.

### 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 55 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 63 for more details.

#### Post Installation Configuration

See [Managing Solr](#) 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 12 for supported versions.
- Before proceeding, review the installation options described in [Cloudera Manager Deployment](#) on page 74.

## 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 55 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 63 for more details.

## Post Installation Configuration

See [Managing Spark Using Cloudera Manager](#) for instructions on adding the Spark service.

## Installing the GPL Extras Parcel

GPL Extras contains LZO functionality for [compressing data](#).

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 5.4 and higher** - [https://archive.cloudera.com/gplextras5/parcels/{latest\\_supported}](https://archive.cloudera.com/gplextras5/parcels/{latest_supported}). The substitution variable {latest\_supported} appears after the parcel to enable substitution of the latest supported maintenance version of the parcel.
  - **CDH 5.0-5.3** - <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 22: CDH 5**

Impala Version	Parcels Version Subdirectory	GPL Extras Parcel Version
CDH 5.x.y	5.x.y/	GPLEXTRAS-5.x.y

**Table 23: 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)

## Installation Overview

<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:

RHEL:

```
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 55.

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 Remote Parcel Repository for Cloudera Manager](#) on page 172.

## 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.

- RHEL/CentOS `yum - /etc/yum.repos.d/`
- SLES `zypper - /etc/zypp/zypper.conf`
- 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:

- RHEL/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/`.

## Installation Overview

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 Remote Parcel Repository for Cloudera Manager

This topic describes how to create a repository and direct hosts in your Cloudera Manager deployment to use that repository. There are two options for publishing the repository:

- [Creating a Permanent Remote Repository](#) on page 172
- [Creating a Temporary Remote Repository](#) on page 173

Once you have created a repository, go to [Configuring the Cloudera Manager Server to Use the Parcel URL](#) on page 174.

After completing these steps, you have established the environment required 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.

### Creating a Permanent Remote Repository

#### Installing 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 [Downloading the Parcel and Publishing Files](#) on page 173.

#### Installing Apache HTTPD

You may need to respond to some prompts to confirm you want to complete the installation.

OS	Command
RHEL	[root@localhost yum.repos.d]\$ yum install httpd
SLES	[root@localhost zypp]\$ zypper install httpd
Ubuntu or Debian	[root@localhost apt]\$ apt-get install httpd

#### Starting Apache HTTPD

OS	Command	
RHEL	[root@localhost tmp]\$ service httpd start Starting httpd: [ ]	[ OK ]
SLES	[root@localhost tmp]\$ service apache2 start Starting httpd: [ ]	[ OK ]

OS	Command
Ubuntu or Debian	[root@localhost tmp]\$ service apache2 start Starting httpd: [ ]

### Downloading the Parcel and Publishing Files

1. Download the parcel and manifest.json files for your OS distribution from

- **CDH 4**
  - CDH - <https://archive.cloudera.com/cdh4/parcels/>
  - Impala - <https://archive.cloudera.com/impala/parcels/>
  - Search <https://archive.cloudera.com/search/parcels/>
  - Spark - <https://archive.cloudera.com/spark/parcels/>
  - GPL Extras - <https://archive.cloudera.com/gplextras/parcels/>
- **CDH 5** - Impala, Spark, and Search are included in the CDH parcel.
  - CDH - <https://archive.cloudera.com/cdh5/parcels/>
  - GPL Extras - <https://archive.cloudera.com/gplextras5/parcels/>
- **Key Trustee Server**
  - Go to the Key Trustee Server [download page](#). Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee Server parcels and manifest.json files in a .tar.gz file. Extract the files with the tar xvfz filename.tar.gz command.
- **Key Trustee KMS**
  - Go to the Key Trustee KMS [download page](#). Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee KMS parcels and manifest.json files in a .tar.gz file. Extract the files with the tar xvfz filename.tar.gz command.
- **Other services**
  - Accumulo - <https://archive.cloudera.com/accumulo/parcels/>
  - Sqoop connectors - <https://archive.cloudera.com/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.

### Creating a Temporary Remote Repository

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.

## Installation Overview

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-4.6.0.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.

### Configuring 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 or click **Hosts** and click the **Parcels** tab.
2. Click the **Configuration** button.

- **Menu**

1. Select **Administration > Settings**.
2. Select **Category > Parcels**.

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 remote package repository and direct hosts in your Cloudera Manager deployment to use that repository. There are two options for publishing the repository:

- [Creating a Permanent Remote Repository](#) on page 174
- [Creating a Temporary Remote Repository](#) on page 175

Once you have created a repository, go to [Modifying Clients to Find the Repository](#) on page 176.

After completing these steps, you have established the environment required 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.

### Creating a Permanent Remote Repository

#### Installing 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 [Downloading the Tarball and Publishing Repository Files](#) on page 175.

## Installing Apache HTTPD

You may need to respond to some prompts to confirm you want to complete the installation.

OS	Command
RHEL	[root@localhost yum.repos.d]\$ yum install httpd
SLES	[root@localhost zypp]\$ zypper install httpd
Ubuntu or Debian	[root@localhost apt]\$ apt-get install httpd

## Starting Apache HTTPD

OS	Command	
RHEL	[root@localhost tmp]\$ service httpd start Starting httpd: [ ]	[ OK ]
SLES	[root@localhost tmp]\$ service apache2 start Starting httpd: [ ]	[ OK ]
Ubuntu or Debian	[root@localhost tmp]\$ service apache2 start Starting httpd: [ ]	[ OK ]

## Downloading the Tarball and Publishing Repository Files

1. Download the tarball for your OS distribution from the [repo as tarball archive](#).

For Cloudera Navigator data encryption components, go to the download page for each component, select your OS version, and click **Download**:

- [Cloudera Navigator Key Trustee Server](#)
- [Cloudera Navigator Key HSM](#)
- [Cloudera Navigator Key Trustee KMS](#)
- [Cloudera Navigator Encrypt](#)

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]$ tar xvfz cm5.0.0-centos6.tar.gz
[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.

## Creating a Temporary Remote Repository

You can quickly create a temporary remote 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](#).

For Cloudera Navigator data encryption components, go to the download page for each component, select your OS version, and click **Download**:

- [Cloudera Navigator Key Trustee Server](#)
- [Cloudera Navigator Key HSM](#)
- [Cloudera Navigator Key Trustee KMS](#)
- [Cloudera Navigator Encrypt](#)

## Installation Overview

2. Unpack the tarball and modify file permissions. For example, you might use the following commands:

```
[root@localhost tmp]$ tar xvfz cm5.0.0-centos6.tar.gz  
[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.

### Modifying Clients to Find the 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 codename components</pre> <p>You can find the <code>codename</code> and <code>component</code> variables in the <code>./conf/distributions</code> file in the repository.</p> <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>

## 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 78.
- 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. Select **Category > Advanced**.
5. Set the **Java Home Directory** property to the custom location.
6. Click **Save Changes**.
7. Restart all services.



**Note:** This procedure changes the JDK for Cloudera Management Services and CDH cluster processes only. It does not affect the JDK used by other non-Cloudera processes.

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 Datastores](#) on page 79.

#### (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 system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue.

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

## Installation Overview

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 RHEL 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 on the Cloudera Manager Server Host

Install the Oracle Java Development Kit (JDK) on the Cloudera Manager Server host. You can install the JDK from a repository, or you can download the JDK from Oracle and install it yourself:

- **Install the JDK from a repository**

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 JDK manually**

See [Java Development Kit Installation](#) on page 78.

### 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>
RHEL, if you're manually transferring RPMs	<code>\$ sudo yum --nogpgcheck localinstall cloudera-manager-daemons-*.*.rpm</code> <code>\$ sudo yum --nogpgcheck localinstall cloudera-manager-server-*.*.rpm</code>

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

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 80.
- Embedded database - Install an embedded PostgreSQL database as described in [Embedded PostgreSQL Database](#) on page 83.

### (Optional) Manually Install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Service Packages

You can use Cloudera Manager to install the Oracle JDK, Cloudera Manager Agent packages, CDH, and managed service packages or you can install any of these packages manually. To use Cloudera Manager to install the packages, you must meet the requirements described in [Cloudera Manager Deployment](#) on page 74.



**Important:** If you are installing CDH and managed service software using packages and you want to manually install Cloudera Manager Agent or CDH packages, you must manually install them both following the procedures in this section; you cannot choose to install only one of them this way.

If you are going to use Cloudera Manager to install all of the software, *skip this section* and continue with [Start the Cloudera Manager Server](#) on page 145. Otherwise, to manually install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Services, continue with the procedures linked below and then return to this page to continue the installation. in this section. You can choose to manually install any of the following software and, in a later step, Cloudera Manager installs any software that you do not install manually:

#### Manually Install the Oracle JDK

You can use Cloudera Manager to install the Oracle JDK on all cluster hosts or you can install the JDKs manually. If you choose to have Cloudera Manager install the JDKs, *skip this section*. To use Cloudera Manager to install the JDK, you must meet the requirements described in [Cloudera Manager Deployment](#) on page 74.

Install the Oracle JDK on every 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 78.
- CDH 4 - [Java Development Kit Installation](#).

#### Manually Install Cloudera Manager Agent Packages

The Cloudera Manager **Agent** is responsible for starting and stopping processes, unpacking configurations, triggering installations, and monitoring all hosts in a cluster. You can install the Cloudera Manager agent manually on all hosts, or Cloudera Manager can install the Agents in a later step. To use Cloudera Manager to install the agents, skip this section and continue with

To install the Cloudera Manager Agent packages manually, do the following on every cluster 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:

## Installation Overview

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

2. On every cluster 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](#).

3. Start the Agents by running the following command on all hosts:

```
$ 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 742. When the Agent hosts reboot, cloudera-scm-agent starts automatically.

### Manually Install CDH and Managed Service Packages

The CDH and Managed Service Packages contain all of the CDH software. You can choose to manually install CDH and the Managed Service Packages, or you can choose to let Cloudera Manager perform this installation in a later step. To use Cloudera Manager perform the installation, continue with [Start the Cloudera Manager Server](#) on page 145. Otherwise, follow the steps in [\(Optional\) Manually Install CDH and Managed Service Packages](#) on page 151 and then return to this page to continue the installation.

#### Install CDH and Managed Service Packages

##### Choose a Repository Strategy

To install CDH and Managed Service Packages, choose one of the following repository strategies:

- 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 170. 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.

Do one of the following:

- [Install CDH 5 and Managed Service Packages](#) on page 180
- [Install CDH 4, Impala, and Solr Managed Service Packages](#) on page 183

#### Install CDH 5 and Managed Service Packages

Install the packages on all cluster hosts using the following steps:

- Red Hat

1. Download and install the "1-click Install" package.

- a. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

- b. Install the RPM for all RHEL versions:

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

2. (Optionally) add a repository key:

- Red Hat/CentOS/Oracle 5

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

- Red Hat/CentOS/Oracle 6

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

3. Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

- SLES

1. Download and install the "1-click Install" package.

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

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

- b. Install the RPM:

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

## Installation Overview

- c. Update your system package index by running:

```
$ sudo zypper refresh
```

2. (Optional) add a repository key:

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

3. Install the CDH packages:

```
$ sudo zypper clean --all  
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3  
hadoop-httfs hadoop-kms hbase-solr hive-hbase hive-webhcate 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-core spark-master spark-worker spark-history-server spark-python  
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

- Ubuntu and Debian

1. Download and install the "1-click Install" package

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

- b. 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
```

2. Optionally add a repository key:

- **Debian Wheezy**

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

- **Ubuntu Precise**

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

### 3. Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

## Install CDH 4, Impala, and Solr Managed Service Packages

Install the packages on all cluster hosts using the following steps:

- **RHEL-compatible**

1. Click the entry in the table at [CDH Download Information](#) that matches your RHEL or CentOS system.
2. Go to the repo file (`cloudera-cdh4.repo`) for your system and save it in the `/etc/yum.repos.d/` directory.
3. Optionally add a repository key:

- **RHEL/CentOS/Oracle 5**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh4/redhat/5/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- **RHEL/CentOS 6**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh4/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
```

#### 4. Install packages on every host in your cluster:

- a. Install CDH 4 packages:

```
$ sudo yum -y install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-0.20-mapreduce hue-plugins
hbase hive oozie oozie-client pig zookeeper
```

- b. To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo yum install hue
```

#### 5. (Requires CDH 4.2 and higher) Install Impala

- a. In the table at [Cloudera Impala Version and Download Information](#), click the entry that matches your RHEL or CentOS system.
- b. Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
- c. Install Impala and the Impala Shell on Impala machines:

```
$ sudo yum -y install impala impala-shell
```

#### 6. (Requires CDH 4.3 and higher) Install Search

- a. In the table at [Cloudera Search Version and Download Information](#), click the entry that matches your RHEL or CentOS system.

## Installation Overview

- b.** Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
- c.** Install the Solr Server on machines where you want Cloudera Search.

```
$ sudo yum -y install solr-server
```

- **SLES**

- 1.** Run the following command:

```
$ sudo zypper addrepo -f  
https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/cloudera-cdh4.repo
```

- 2.** Update your system package index by running:

```
$ sudo zypper refresh
```

- 3.** Optionally add a repository key:

```
$ sudo rpm --import  
https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- 4.** Install packages on every host in your cluster:

- a.** Install CDH 4 packages:

```
$ 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
```

- b.** To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo zypper install hue
```

- c. (Requires CDH 4.2 and higher)** Install Impala

- a.** Run the following command:

```
$ sudo zypper addrepo -f  
https://archive.cloudera.com/impala/sles/11/x86_64/impala/cloudera-impala.repo
```

- b.** Install Impala and the Impala Shell on Impala machines:

```
$ sudo zypper install impala impala-shell
```

- d. (Requires CDH 4.3 and higher)** Install Search

- a.** Run the following command:

```
$ sudo zypper addrepo -f  
https://archive.cloudera.com/search/sles/11/x86_64/search/cloudera-search.repo
```

- b.** Install the Solr Server on machines where you want Cloudera Search.

```
$ sudo zypper install solr-server
```

- **Ubuntu or Debian**

1. In the table at [CDH Version and Packaging Information](#), click the entry that matches your Ubuntu or Debian system.
2. Go to the list file (`cloudera.list`) for your system and save it in the `/etc/apt/sources.list.d/` directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your `cloudera.list` file should look like:

```
deb [arch=amd64] https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4
contrib
deb-src https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh lucid-cdh4 contrib
```

**3. Optionally add a repository key:**

- **Ubuntu Lucid**

```
$ curl -s https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh/archive.key | sudo
apt-key add -
```

- **Ubuntu Precise**

```
$ curl -s https://archive.cloudera.com/cdh4/ubuntu/precise/amd64/cdh/archive.key | sudo
apt-key add -
```

- **Debian Squeeze**

```
$ curl -s https://archive.cloudera.com/cdh4/debian/squeeze/amd64/cdh/archive.key | sudo
apt-key add -
```

**4. Install packages on every host in your cluster:**

a. Install CDH 4 packages:

```
$ 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
```

b. To install the `hue-common` package and all Hue applications on the Hue host, install the `hue` meta-package:

```
$ sudo apt-get install hue
```

c. **(Requires CDH 4.2 and higher)** Install Impala

- a. In the table at [Cloudera Impala Version and Download Information](#), click the entry that matches your Ubuntu or Debian system.
- b. Go to the list file for your system and save it in the `/etc/apt/sources.list.d/` directory.
- c. Install Impala and the Impala Shell on Impala machines:

```
$ sudo apt-get install impala impala-shell
```

d. **(Requires CDH 4.3 and higher)** Install Search

- 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:

```
$ sudo apt-get install solr-server
```

## Installation Overview

### 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 742.

### Start the Cloudera Manager Agents

If you using Cloudera Manager to install the Cloudera Manager Agent packages, *skip this section*. Otherwise, run the following 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 742.

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 start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.
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. Although 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.
4. After logging in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

The **Welcome to Cloudera Manager** page displays.

## Choose Cloudera Manager Edition

From the **Welcome to Cloudera Manager** page, you can select the edition of Cloudera Manager to install and, optionally, install a license:

**1.** Choose which [edition](#) to install:

- Cloudera Express, which does not require a license, but provides a 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 upgrade the license at a later time. See [Managing Licenses](#).

**2.** If you elect 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. Go to the location of your license file, click the file, and click **Open**.
- d. Click **Upload**.

**3.** Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.

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

## Choose Cloudera Manager Hosts

Choose which hosts will run CDH and managed services

**1.** Do one of the following depending on whether you are using Cloudera Manager to install software:

- If you are using Cloudera Manager to install software, 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 with 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 do not 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

## Installation Overview

**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.

- If you installed Cloudera Agent packages in [Manually Install Cloudera Manager Agent Packages](#) on page 144, choose from among hosts with the packages installed:

1. Click the **Currently Managed Hosts** tab.
2. Choose the hosts to add to the cluster.

2. Click **Continue**.

The **Cluster Installation Select Repository** screen displays.

### Choose the Software Installation Type and Install Software

Choose a software installation type (parcels or packages) and install the software. If you have already installed the CDH and Managed Service packages, you cannot choose **Parcel** installation.



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

1. Choose the software installation type and CDH and managed service version:

- **Use Parcels**

1. Choose the parcels to install. The choices depend on the repositories you have chosen; a repository can 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, specify the 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 63 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you 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 are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. 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.

- **Use Packages** - Do one of the following:

- If Cloudera Manager is installing the packages:
  1. Click the package version.
  2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. 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.
- If you manually installed packages in [Manually Install CDH and Managed Service Packages](#) on page 145 , select the CDH version (CDH 4 or CDH 5) that matches the packages you installed manually.

2. If you installed the Agent and JDK manually on all cluster hosts:

- 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**.

- Skip the remaining steps in this section and continue with [Add Services](#) on page 149

3. Select **Install Oracle Java SE Development Kit (JDK)** to allow Cloudera Manager to install the JDK on each cluster host. If you have already installed the JDK, do not select this option. If 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.



**Note:** If you already manually installed the JDK on each cluster host, this option to install the JDK does not display.

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**.

5. If you chose to have Cloudera Manager install software, specify host installation properties:

- Select **root** or enter the username for an account that has password-less sudo permission.
- Select an authentication method:
  - If you choose password authentication, enter and confirm the password.
  - If you choose public-key authentication, provide a passphrase and path to the required key files.
- You can 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 chose to have Cloudera Manager install software, Cloudera Manager installs the Oracle JDK, Cloudera Manager Agent, packages and CDH and managed service parcels or packages. During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.

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**.

## Installation Overview

### Add Services

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

- Select the combination of services to install:

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

As you select 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 lifecycle.

- 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 [Cloudera Navigator 2 Overview](#).

### 2. Click Continue.

3. 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. You can reassign role instances if necessary.

Click a field below a role to display a dialog box 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 box.

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.

- When you are satisfied with the assignments, click **Continue**.

#### Configure Database Settings

On the Database Setup page, configure settings for required databases:

- Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
- 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 Configuration Changes and Start Services

- 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. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.



**Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

- Click **Continue**.

The wizard starts the services.

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

#### Change the Default Administrator Password

As soon as possible, change the default administrator password:

- 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 **OK**.

#### Test the Installation

You can test the installation following the instructions in [Testing the Installation](#) on page 198.

### Creating a CDH Cluster Using a Cloudera Manager Template

You can create a new CDH cluster by exporting a *cluster template* from an existing CDH cluster managed by Cloudera Manager. You can then modify the template and use it to create new clusters with the same configuration on a new set of hosts. Use cluster templates to:

- Duplicate clusters for use in developer, test, and production environments.
- Quickly create a cluster for a specific workload.
- Reproduce a production cluster for testing and debugging.

Follow these general steps to create a template and a new cluster:

1. Export the cluster configuration from the source cluster. The exported configuration is a JSON file that details all of the configurations of the cluster. The JSON file includes an `instantiator` section that contains some values you must provide before creating the new cluster.

See [Exporting the Cluster Configuration](#) on page 192.

2. Set up the hosts for the new cluster by installing Cloudera Manager agents and the JDK on all hosts. For secure clusters, also configure a Kerberos key distribution center (KDC) in Cloudera Manager.

See [Preparing a New Cluster](#) on page 193

3. Create any local repositories required for the cluster.

See [Establish Your Cloudera Manager Repository Strategy](#) on page 142.

4. Complete the `instantiator` section of the cluster configuration JSON document to create a template.

See [Creating the Template](#) on page 193.

5. Import the cluster template to the new cluster.

See [Importing the Template to a New Cluster](#) on page 196.

#### Exporting the Cluster Configuration

To create a cluster template, you begin by exporting the configuration from the source cluster. The cluster must be running and managed by Cloudera Manager 5.7 or higher.

To export the configuration:

1. Any [Host Templates](#) you have created are used to export the configuration. If you do not want to use those templates in the new cluster, delete them. In Cloudera Manager, go to **Hosts > Host Templates** and click **Delete** next to the Host Template you want to delete.
2. Delete any Host Templates created by the Cloudera Manager Installation Wizard. They typically have a name like `Template - 1`.
3. Run the following command to download the JSON configuration file to a convenient location for editing:

```
curl -u admin_username:admin_user_password  
"http://Cloudera Manager URL/api/v12/clusters/Cluster name/export" >  
path_to_file/file_name.json
```

For example:

```
curl -u adminuser:adminpass  
"http://myCluster-1.myDomain.com:7180/api/v12/clusters/Cluster1/export" >  
myCluster1-template.json
```



**Note:** Add the `?exportAutoConfig=true` parameter to the command above to include configurations made by [Autoconfiguration](#). These configurations are included for reference only and are not used when you import the template into a new cluster. For example:

```
curl -u admin_username:admin_user_password
"http://Cloudera Manager URL/api/v12/clusters/Cluster name/export"
>
path_to_file/file_name.json?exportAutoConfig=true
```

## Preparing a New Cluster

The new cluster into which you import the cluster template must meet the following requirements:

- Database for Cloudera Manager is installed and configured.
- Cloudera Manager 5.7 or higher is installed and running.
- All required databases for CDH services are installed. See [Cloudera Manager and Managed Service Datastores](#) on page 79.
- The JDK is installed on all cluster hosts.
- The Cloudera Manager Agent is installed and configured on all cluster hosts.
- If the source cluster uses Kerberos, the new cluster must have KDC properties and privileges configured in Cloudera Manager.
- If the source cluster used *packages* to install CDH and managed services, install those packages manually before importing the template. See [Managing Software Installation Using Cloudera Manager](#) on page 55.

## Creating the Template

To create a template, modify the `instantiator` section of the JSON file you downloaded. Lines that contain the string `<changeme>` require a value that you must supply. Here is a sample `instantiator` section:

```
"instantiator" : {
  "clusterName" : "<changeme>",
  "hosts" : [ {
    "hostName" : "<changeme>",
    "hostTemplateRefName" : "<changeme>",
    "roleRefNames" : [ "HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e" ]
  }, {
    "hostName" : "<changeme>",
    "hostTemplateRefName" : "<changeme>"
  }, {
    "hostNameRange" : "<HOST[0001-0002]>",
    "hostTemplateRefName" : "<changeme>"
  } ],
  "variables" : [ {
    "name" : "HDFS-1-NAMENODE-BASE-dfs_name_dir_list",
    "value" : "/dfs/nn"
  }, {
    "name" : "HDFS-1-SECONDARYNAMENODE-BASE-fs_checkpoint_dir_list",
    "value" : "/dfs/snn"
  }, {
    "name" : "HIVE-1-hive_metastore_database_host",
    "value" : "myCluster-1.myDomain.com"
  }, {
    "name" : "HIVE-1-hive_metastore_database_name",
    "value" : "hivel"
  }, {
    "name" : "HIVE-1-hive_metastore_database_password",
    "value" : "<changeme>"
  }, {
    "name" : "HIVE-1-hive_metastore_database_port",
    "value" : "3306"
  }, {
    "name" : "HIVE-1-hive_metastore_database_type",
    "value" : "mysql"
```

## Installation Overview

```
}, {
    "name" : "HIVE-1-hive_metastore_database_user",
    "value" : "hive1"
}, {
    "name" : "HUE-1-database_host",
    "value" : "myCluster-1.myDomain.com"
}, {
    "name" : "HUE-1-database_name",
    "value" : "hueserver0be88b55f5dedbf7bc74d61a86c0253e"
}, {
    "name" : "HUE-1-database_password",
    "value" : "<changeme>"
}, {
    "name" : "HUE-1-database_port",
    "value" : "3306"
}, {
    "name" : "HUE-1-database_type",
    "value" : "mysql"
}, {
    "name" : "HUE-1-database_user",
    "value" : "hueserver0be88b5"
}, {
    "name" : "IMPALA-1-IMPALAD-BASE-scratch_dirs",
    "value" : "/impala/impalad"
}, {
    "name" : "KUDU-1-KUDU_MASTER-BASE-fs_data_dirs",
    "value" : "/var/lib/kudu/master"
}, {
    "name" : "KUDU-1-KUDU_MASTER-BASE-fs_wal_dir",
    "value" : "/var/lib/kudu/master"
}, {
    "name" : "KUDU-1-KUDU_TSERVER-BASE-fs_data_dirs",
    "value" : "/var/lib/kudu/tserver"
}, {
    "name" : "KUDU-1-KUDU_TSERVER-BASE-fs_wal_dir",
    "value" : "/var/lib/kudu/tserver"
}, {
    "name" : "MAPREDUCE-1-JOBTRACKER-BASE-jobtracker_mapred_local_dir_list",
    "value" : "/mapred/jt"
}, {
    "name" : "MAPREDUCE-1-TASKTRACKER-BASE-tasktracker_mapred_local_dir_list",
    "value" : "/mapred/local"
}, {
    "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_host",
    "value" : "myCluster-1.myDomain.com:3306"
}, {
    "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_name",
    "value" : "oozieserver0be88b55f5dedbf7bc74d61a86c0253e"
}, {
    "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_password",
    "value" : "<changeme>"
}, {
    "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_type",
    "value" : "mysql"
}, {
    "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_user",
    "value" : "oozieserver0be88"
}, {
    "name" : "YARN-1-NODEMANAGER-BASE-yarn_nodemanager_local_dirs",
    "value" : "/yarn/nm"
}, {
    "name" : "YARN-1-NODEMANAGER-BASE-yarn_nodemanager_log_dirs",
    "value" : "/yarn/container-logs"
} ]}
```

To modify the template:

1. Update the hosts section.

If you have host templates defined in the source cluster, they appear in the hostTemplates section of the JSON template. For hosts that do not use host templates, the export process creates host templates based on role

assignments to facilitate creating the new cluster. In either case, you must match the items in the `hostTemplates` section with the `hosts` sections in the `instantiator` section.

Here is a sample of the `hostTemplates` section from the same JSON file as the `instantiator` section, above:

```
"hostTemplates" : [ {
    "refName" : "HostTemplate-0-from-myCluster-1.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-GATEWAY-BASE",
"HBASE-1-HBASETHRIFTSERVER-BASE", "HBASE-1-MASTER-BASE", "HDFS-1-BALANCER-BASE",
"HDFS-1-GATEWAY-BASE", "HDFS-1-NAMENODE-BASE", "HDFS-1-NFSGATEWAY-BASE",
"HDFS-1-SECONDARYNAMENODE-BASE", "HIVE-1-GATEWAY-BASE", "HIVE-1-HIVEMETASTORE-BASE",
"HIVE-1-HIVESERVER2-BASE", "HUE-1-HUE_LOAD_BALANCER-BASE", "HUE-1-HUE_SERVER-BASE",
"IMPALA-1-CATALOGSERVER-BASE", "IMPALA-1-STATESTORE-BASE", "KAFKA-1-KAFKA_BROKER-BASE",
"KS_INDEXER-1-HBASE_INDEXER-BASE", "KUDU-1-KUDU_MASTER-BASE", "MAPREDUCE-1-GATEWAY-BASE",
"MAPREDUCE-1-JOBTRACKER-BASE", "OOZIE-1-OOZIE_SERVER-BASE", "SOLR-1-SOLR_SERVER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SPARK_ON_YARN-1-SPARK_YARN_HISTORY_SERVER-BASE",
"SQOOP-1-SQOOP_SERVER-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-GATEWAY-BASE",
"YARN-1-JOBHISTORY-BASE", "YARN-1-RESOURCEMANAGER-BASE", "ZOOKEEPER-1-SERVER-BASE" ],
}, {
    "refName" : "HostTemplate-1-from-myCluster-4.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE" ],
}, {
    "refName" : "HostTemplate-2-from-myCluster-[2-3].myDomain.com",
    "cardinality" : 2,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KAFKA-1-KAFKA_BROKER-BASE", "KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE" ]
} ]
```

The value of `cardinality` indicates how many hosts are assigned to the host template in the source cluster.

The value of `roleConfigGroupsRefNames` indicates which role groups are assigned to the host(s).

Do the following for each host template in the `hostTemplates` section:

1. Locate the entry in the `hosts` section of the `instantiator` where you want the roles to be installed.
2. Copy the value of the `refName` to the value for `hostTemplateRefName`.
3. Enter the hostname in the new cluster as the value for `hostName`. Some host sections may instead use `hostNameRange` for clusters with multiple hosts that have the same set of roles. Indicate a range of hosts using brackets; for example, `myhost[1-4].foo.com`.

Here is an example of the `hostTemplates` and the `hosts` section of the `instantiator` completed correctly:

```
"hostTemplates" : [ {
    "refName" : "HostTemplate-0-from-myCluster-1.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-GATEWAY-BASE",
"HBASE-1-HBASETHRIFTSERVER-BASE", "HBASE-1-MASTER-BASE", "HDFS-1-BALANCER-BASE",
"HDFS-1-GATEWAY-BASE", "HDFS-1-NAMENODE-BASE", "HDFS-1-NFSGATEWAY-BASE",
"HDFS-1-SECONDARYNAMENODE-BASE", "HIVE-1-GATEWAY-BASE", "HIVE-1-HIVEMETASTORE-BASE",
"HIVE-1-HIVESERVER2-BASE", "HUE-1-HUE_LOAD_BALANCER-BASE", "HUE-1-HUE_SERVER-BASE",
"IMPALA-1-CATALOGSERVER-BASE", "IMPALA-1-STATESTORE-BASE", "KAFKA-1-KAFKA_BROKER-BASE",
"KS_INDEXER-1-HBASE_INDEXER-BASE", "KUDU-1-KUDU_MASTER-BASE", "MAPREDUCE-1-GATEWAY-BASE",
"MAPREDUCE-1-JOBTRACKER-BASE", "OOZIE-1-OOZIE_SERVER-BASE", "SOLR-1-SOLR_SERVER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SPARK_ON_YARN-1-SPARK_YARN_HISTORY_SERVER-BASE",
"SQOOP-1-SQOOP_SERVER-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-GATEWAY-BASE",
"YARN-1-JOBHISTORY-BASE", "YARN-1-RESOURCEMANAGER-BASE", "ZOOKEEPER-1-SERVER-BASE" ],
}, {
    "refName" : "HostTemplate-1-from-myCluster-4.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE" ] }
```

## Installation Overview

```
"KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE"
],
},
{
  "refName" : "HostTemplate-2-from-myCluster-[2-3].myDomain.com",
  "cardinality" : 2,
  "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KAFKA-1-KAFKA_BROKER-BASE", "KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE"
],
}
],
"instantiator" : {
  "clusterName" : "myCluster_new",
  "hosts" : [
    {
      "hostName" : "myNewCluster-1.myDomain.com",
      "hostTemplateRefName" : "HostTemplate-0-from-myCluster-1.myDomain.com",
      "roleRefNames" : [ "HDFS-1-NAMENODE-c975a0b51fd36e914896cd5e0adb1b5b" ]
    },
    {
      "hostName" : "myNewCluster-5.myDomain.com",
      "hostTemplateRefName" : "HostTemplate-1-from-myCluster-4.myDomain.com"
    },
    {
      "hostNameRange" : "myNewCluster-[3-4].myDomain.com",
      "hostTemplateRefName" : "HostTemplate-2-from-myCluster-[2-3].myDomain.com"
    }
  ],
}
```

2. For host sections that have a `roleRefNames` line, determine the role type and assign the appropriate host for the role. If there are multiple instances of a role, you must select the correct hosts. To determine the role type, search the template file for the value of `roleRefNames`.

For example: For a role ref named `HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e`, if you search for that string, you find a section similar to the following:

```
"roles": [
{
  "refName": "HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e",
  "roleType": "NAMENODE"
}]
```

In this case, the role type is NAMENODE.

3. Modify the `variables` section. This section contains various properties from the source cluster. You can change any of these values to be different in the new cluster, or you can leave the values as copied from the source. For any values shown as `<changeme>`, you must provide the correct value.



**Note:** Many of these variables contain information about databases used by the Hive Metastore and other CDH components. Change the values of these variables to match the databases configured for the new cluster.

4. Enter the internal name of the new cluster on the line with `"clusterName" : "<changeme>"`. For example:

```
"clusterName" : "QE_test_cluster"
```

5. (Optional) Change the display name for the cluster. Edit the line that begins with `"displayName"` (near the top of the JSON file); for example:

```
"displayName" : "myNewCluster",
```

### Importing the Template to a New Cluster

To import the cluster template:

1. Log in to the Cloudera Manager server as root.

- Run the following command to import the template. If you have remote repository URLs configured in the source cluster, append the command with `?addRepositories=true`.

```
curl -X POST -H "Content-Type: application/json" -d
@path_to_template/template_filename.json
http://admin_user:admin_password@cloudera_manager_url:cloudera_manager_port/api/v12/cm/importClusterTemplate
```

You should see a response similar to the following:

```
{
  "id" : 17,
  "name" : "ClusterTemplateImport",
  "startTime" : "2016-03-09T23:44:38.491Z",
  "active" : true,
  "children" : {
    "items" : [ ]
  }
}
```

#### Examples:

```
curl -X POST -H "Content-Type: application/json" -d @myTemplate.json
http://admin:admin@myNewCluster-1.mydomain.com:7182/api/v12/cm/importClusterTemplate
```

```
curl -X POST -H "Content-Type: application/json" -d @myTemplate.json
http://admin:admin@myNewCluster-1.mydomain.com:7182/api/v12/cm/importClusterTemplate?addRepositories=true
```

If there is no response, or you receive an error message, the JSON file may be malformed, or the template may have invalid hostnames or invalid references. Inspect the JSON file, correct any errors, and then re-run the command.

- Open Cloudera Manager for the new cluster in a web browser and click the Cloudera Manager logo to go to the home page.
- Click the **All Recent Commands** tab.

If the import is proceeding, you should see a link labeled **Import Cluster Template**. Click the link to view the progress of the import.

If any of the commands fail, correct the problem and click **Retry**. You may need to edit some properties in Cloudera Manager.

After you import the template, Cloudera Manager applies the [Autoconfiguration](#) rules that set properties such as memory and CPU allocations for various roles. If the new cluster has different hardware or operational requirements, you may need to modify these values.

#### Sample Python Code

You can perform the steps to export and import a cluster template programmatically using a client written in Python or other languages. (You can also use the `curl` commands provided above.)

#### Python export example:

```
resource = ApiResource("myCluster-1.myDomain.com", 7180, "admin", "admin", version=12)
cluster = resource.get_cluster("Cluster1");
template = cluster.export(False)
pprint(template)
```

#### Python import example:

```
resource = ApiResource("localhost", 8180, "admin", "admin", version=12)
with open('~/cluster-template.json') as data_file:
    data = json.load(data_file)
```

## Installation Overview

```
template = ApiClusterTemplate(resource).from_json_dict(data, resource)
cms = ClouderaManager(resource)
cms.import_cluster_template(template)
```

## Deploying Clients

Client configuration files are generated automatically by Cloudera Manager based on the services you install.

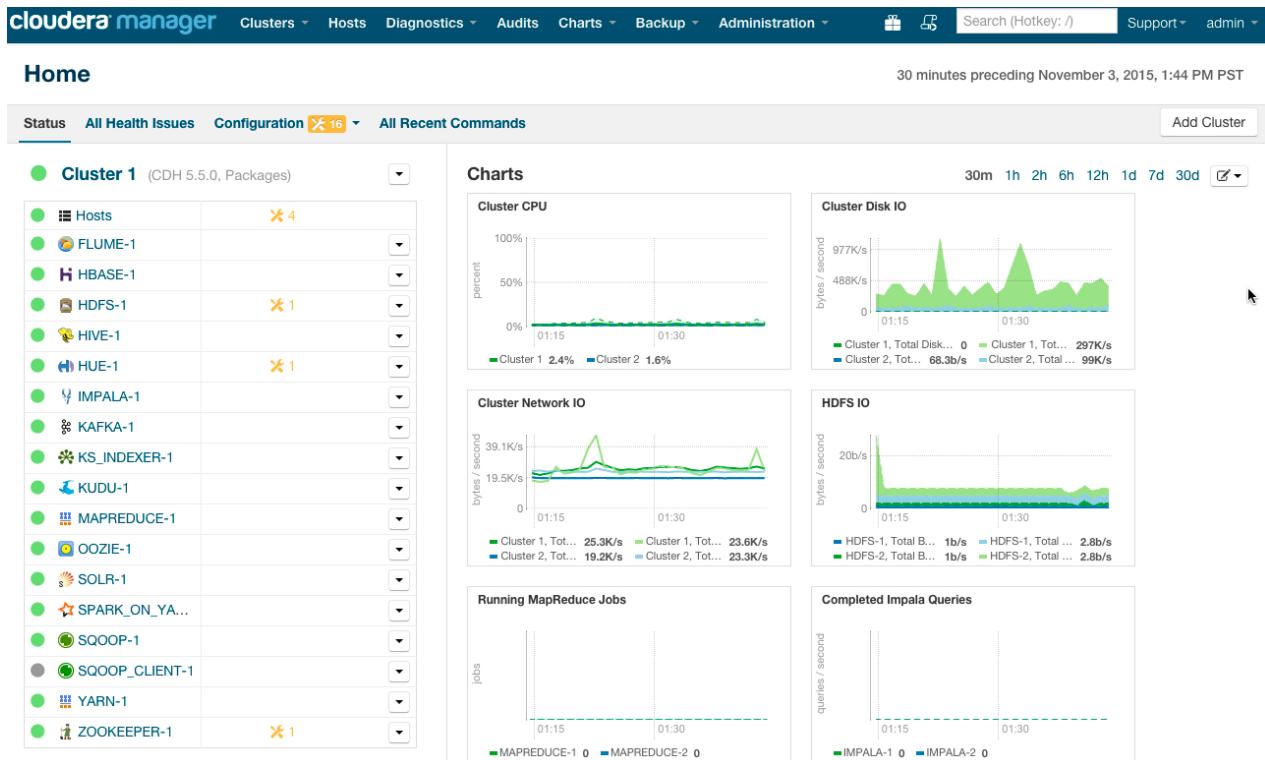
Cloudera Manager deploys these configurations automatically at the end of the installation workflow. You can also download the client configuration files to deploy them manually.

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**. You can click 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	<span>Actions</span>	<span>Details</span>
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** > **Status** tab, click the Hue service.
2. Click the **Hue Web UI** link, which opens Hue in a new window.
3. Log in with the credentials, **username**: hdfs, **password**: hdfs.
4. Choose an application in the navigation bar at the top of the browser window.

For more information, see the [Hue User Guide](#).

## Uninstalling Cloudera Manager and Managed Software

Use the following instructions to uninstall the Cloudera Manager Server, Agents, managed software, and databases.

### 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 204, /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

## Installation Overview

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 > Status** tab, 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 > Status** tab, 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. a. Do one of the following:

- 1. Select **Clusters > Cloudera Management Service > Cloudera Management Service**.

- 2. Select **Actions > Stop**.

- 1. On the **Home > Status** tab, click



to the right of **Cloudera Management Service** and select **Stop**.

- b. Click **Stop** to confirm. The **Command Details** window shows the progress of stopping the roles.

- c. When **Command completed with n/n successful subcommands** appears, the task is complete. Click **Close**.

### Deactivate and Remove Parcels

If you installed using packages, skip this step and go to [Uninstall the Cloudera Manager Server](#) on page 201; you will remove packages in [Uninstall Cloudera Manager Agent and Managed Software](#) on page 201. If you installed using parcels remove them as follows:

1. Click the parcel indicator in the main navigation bar.
2. In the **Location** selector on the left, select **All Clusters**.
3. For each activated parcel, select **Actions > Deactivate**. When this action has completed, the parcel button changes to **Activate**.
4. For each activated parcel, select **Actions > Remove from Hosts**. When this action has completed, the parcel button changes to **Distribute**.
5. 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
```

- **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.

### RHEL 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.

### RHEL-compatible 7 and higher

```
$ sudo service cloudera-scm-agent next_stop_hard
$ sudo service cloudera-scm-agent stop
```

### All other RHEL/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
<b>RHEL</b>	\$ sudo yum remove 'cloudera-manager-*'	<ul style="list-style-type: none"> <li>• <b>CDH 4</b></li> </ul>

## Installation Overview

OS	Parcel Install	Package Install
		<pre>\$ sudo yum remove 'cloudera-manager-*' bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httplfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper hue impala impala-shell solr-server</pre> <ul style="list-style-type: none"> <li>• <b>CDH 5</b></li> </ul> <pre>\$ sudo yum remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httplfs 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-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>
<b>SLES</b>	<pre>\$ sudo zypper remove 'cloudera-manager-*'</pre>	<ul style="list-style-type: none"> <li>• <b>CDH 4</b></li> </ul> <pre>\$ sudo zypper remove 'cloudera-manager-*' bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httplfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper hue impala impala-shell solr-server</pre> <ul style="list-style-type: none"> <li>• <b>CDH 5</b></li> </ul> <pre>\$ sudo zypper remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httplfs 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-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>
<b>Debian/Ubuntu</b>	<pre>\$ sudo apt-get purge 'cloudera-manager-*'</pre>	<ul style="list-style-type: none"> <li>• <b>CDH 4</b></li> </ul> <pre>\$ sudo apt-get purge 'cloudera-manager-*' bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httplfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper hue impala impala-shell solr-server</pre> <ul style="list-style-type: none"> <li>• <b>CDH 5</b></li> </ul>

OS	Parcel Install	Package Install
		<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-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>

### 3. Run the `clean` command:

#### RHEL

```
$ 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

If you are uninstalling on RHEL, run the following commands on all Agent hosts to permanently remove 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`.

```
$ sudo umount cm_processes
$ sudo rm -Rf /usr/share/cmfc /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
```

## Installation Overview

### 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 the [CDH4 Installation Guide](#) or [Cloudera Installation and Upgrade](#).

### 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 the Cloudera Navigator Data Management Component

### Minimum Required Role: [Full Administrator](#)

The Cloudera Navigator data management component is implemented as two roles in the [Cloudera Management Service](#): Navigator Audit Server and Navigator Metadata Server. You can add Cloudera Navigator data management 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.

### Configuring a Database for the Cloudera Navigator

When you install the Cloudera Navigator data management component 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 28. For information on setting up an external database, see [Cloudera Manager and Managed Service Datastores](#) on page 79.

### Adding Cloudera Navigator Roles in a New Cloudera Manager Installation

1. Install Cloudera Manager following the instructions in [Cloudera Manager Deployment](#) on page 74.
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 elect 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. Go to the location of your license file, click the file, and click **Open**.
  - d. Click **Upload**.
- 4.** Click **Continue** to proceed with the installation.
- 5.** In the first page of the Add Services procedure, check the **Include Cloudera Navigator** checkbox.
- 6.** 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 Data Management 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](#)

#### Related Information

- [Cloudera Navigator 2 Overview](#)
- [Upgrading the Cloudera Navigator Data Management Component](#) on page 503
- [Cloudera Navigator Data Management Component Administration](#)
- [Cloudera Data Management](#)
- [Configuring Authentication in the Cloudera Navigator Data Management Component](#)
- [Configuring TLS/SSL for the Cloudera Navigator Data Management Component](#)
- [Cloudera Navigator Data Management Component User Roles](#)

## Installing Cloudera Navigator Key Trustee Server



**Important:** Before installing Cloudera Navigator Key Trustee Server, see [Deployment Planning for Data at Rest Encryption](#) for important considerations.

You can install Navigator Key Trustee Server using Cloudera Manager with parcels or using the command line with packages. See [Parcels](#) on page 55 for more information on parcels.



**Note:** If you are using or planning to use Key Trustee Server in conjunction with a CDH cluster, Cloudera strongly recommends using Cloudera Manager to install and manage Key Trustee Server to take advantage of Cloudera Manager's robust deployment, management, and monitoring capabilities.

## Prerequisites

See [Data at Rest Encryption Requirements](#) for more information about encryption and Key Trustee Server requirements.

### Setting Up an Internal Repository

You must create an internal repository to install or upgrade the Cloudera Navigator data encryption components. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see the following topics:

- [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172
- [Creating and Using a Package Repository for Cloudera Manager](#) on page 174

### Installing Key Trustee Server



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

#### Installing Key Trustee Server Using Cloudera Manager



**Note:** These instructions apply to using Cloudera Manager only. To install Key Trustee Server using packages, skip to [Installing Key Trustee Server Using the Command Line](#) on page 207.

If you are installing Key Trustee Server for use with [HDFS Transparent Encryption](#), the **Set up HDFS Data At Rest Encryption** wizard installs and configures Key Trustee Server. See [Enabling HDFS Encryption Using the Wizard](#) for instructions.

1. **(Recommended)** Create a new cluster in Cloudera Manager containing only the hosts Key Trustee Server will be installed on. Cloudera strongly recommends installing Key Trustee Server in a dedicated cluster to enable multiple clusters to share the same Key Trustee Server and to avoid restarting the Key Trustee Server when restarting a cluster. See [Adding and Deleting Clusters](#) for instructions on how to create a new cluster in Cloudera Manager.



**Important:** The **Add Cluster** wizard prompts you to install CDH and other cluster services. To exit the wizard without installing CDH, select a version of CDH to install and continue. When the installation begins, click the Cloudera Manager logo in the upper left corner and confirm you want to exit the wizard. This allows you to create the dedicated cluster with the Key Trustee Server hosts without installing CDH or other services that are not required for Key Trustee Server.

2. Add the internal parcel repository you created in [Setting Up an Internal Repository](#) on page 206 to Cloudera Manager following the instructions in [Configuring Cloudera Manager Server Parcel Settings](#) on page 63.
3. Download, distribute, and activate the Key Trustee Server parcel on the cluster containing the Key Trustee Server host, following the instructions in [Managing Parcels](#) on page 57.



**Important:** The KEYTRUSTEE parcel in Cloudera Manager is *not* the Key Trustee Server parcel; it is the Key Trustee KMS parcel. The parcel name for Key Trustee Server is KEYTRUSTEE\_SERVER.

After you activate the Key Trustee Server parcel, Cloudera Manager prompts you to restart the cluster. Click the **Close** button to ignore this prompt. You *do not* need to restart the cluster after installing Key Trustee Server.

After installing Key Trustee Server using Cloudera Manager, continue to [Securing Key Trustee Server Host](#) on page 208.

## Installing Key Trustee Server Using the Command Line



**Note:** These instructions apply to package-based installations using the command line only. To install Key Trustee Server using Cloudera Manager, see [Installing Key Trustee Server Using Cloudera Manager](#) on page 206.

If you are using or planning to use Key Trustee Server in conjunction with a CDH cluster, Cloudera strongly recommends using Cloudera Manager to install and manage Key Trustee Server to take advantage of Cloudera Manager's robust deployment, management, and monitoring capabilities.

### 1. Install the EPEL Repository

Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the `epel-release` package:

1. Copy the URL for the `epel-release-<version>.noarch` located at the bottom of the [EPEL 6](#) page.
2. Run the following commands to install the EPEL repository:

```
$ sudo wget <epel_rpm_url>
$ sudo yum install epel-release-<version>.noarch.rpm
```

Replace `<version>` with the version number of the downloaded RPM (for example, 6-8).

If the `epel-release` package is already installed, you see a message similar to the following:

```
Examining /var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
/var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: does not update installed package.
Error: Nothing to do
```

Confirm that the EPEL repository is installed:

```
$ sudo yum repolist | grep -i epel
```

### 2. (RHEL 7 Only) Enable the `extras` Repository

Key Trustee Server requires the `python-flask` package. For RHEL 6, this package is provided in the EPEL repository. For RHEL 7, it is provided in the RHEL `extras` repository. To enable this repository, run the following command:

```
$ sudo subscription-manager repos --enable=rhel-7-server-extras-rpms
```

### 3. Install the PostgreSQL 9.3 Repository



**Note:** Cloudera Navigator Key Trustee Server currently supports only PostgreSQL version 9.3. If you have a different version of PostgreSQL installed on the Key Trustee Server host, remove it before proceeding or select a different host on which to install Key Trustee Server.

To install the PostgreSQL 9.3 repository, run the following command:

```
$ sudo yum install
http://yum.postgresql.org/9.3/redhat/rhel-6-x86_64/pgdg-redhat93-9.3-1.noarch.rpm
```



**Important:** If you are using CentOS, add the following line to the CentOS base repository:

```
exclude=python-psycopg2*
```

By default, the base repository is located at `/etc/yum.repos.d/CentOS-Base.repo`. If you have an internal mirror of the base repository, update the correct file for your environment.

## Installation Overview

### 4. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
```

### 5. Install the CDH Repository

Key Trustee Server and Key HSM depend on the `bigtop-utils` package, which is included in the CDH repository. For instructions on adding the CDH repository, see [To add the CDH repository](#). To create a local CDH repository, see [Creating a Local Yum Repository](#) on page 219 for instructions.

### 6. Install NTP

The Network Time Protocol (NTP) service synchronizes system time. Cloudera recommends using NTP to ensure that timestamps in system logs, cryptographic signatures, and other auditable events are consistent across systems.

Install and start NTP with the following commands:

```
$ sudo yum install ntp
$ sudo service ntpd start
## For RHEL/CentOS 7, use 'sudo systemctl start ntpd' instead ##
```

### 7. Install Key Trustee Server

Run the following command to install the Key Trustee Server:

```
$ sudo yum install keytrustee-server
```

Installing the Key Trustee Server also installs required dependencies, including PostgreSQL 9.3. After the installation completes, confirm that the PostgreSQL version is 9.3 by running the command `createuser -V`.

### 8. Configure Services to Start at Boot

Ensure that `ntpd`, `keytrustee-db`, and `keytrusteed` start automatically at boot:

```
$ sudo chkconfig ntpd on
$ sudo chkconfig keytrustee-db on
$ sudo chkconfig keytrusteed on
```

The `chkconfig` command provides no output if successful.



**Note:** The `/etc/init.d/postgresql` script does not work when the PostgreSQL database is started by Key Trustee Server, and cannot be used to monitor the status of the database. Use `/etc/init.d/keytrustee-db` instead.

After installing Key Trustee Server, continue to [Securing Key Trustee Server Host](#) on page 208.

## Securing Key Trustee Server Host

Cloudera strongly recommends securing the Key Trustee Server host to protect against unauthorized access to Key Trustee Server. Red Hat provides security guides for RHEL:

- [RHEL 6 Security Guide](#)
- [RHEL 7 Security Guide](#)

Cloudera also recommends configuring the Key Trustee Server host to allow network communication only over certain ports. See [Ports](#) on page 36 for more information about the ports used by Cloudera Manager and CDH. You can use

the following examples to create iptables rules for an EDH cluster. Add any other ports required by your environment, subject to your organization security policies.

```
# Flush iptables
iptables -F
iptables -X

# Allow unlimited traffic on loopback (localhost) connection
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Allow established, related connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Open all Cloudera Manager and CDH ports to allow Key Trustee Server to work properly

iptables -A INPUT -p tcp -m tcp --dport 4867 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 5432 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 5678 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7180 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7180 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7182 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7183 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7184 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7185 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7186 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7187 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 7432 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 8020 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 8083 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 8084 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 8086 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 8087 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 8091 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9000 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9001 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9994 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9995 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9996 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9997 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9998 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 9999 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 10101 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 11371 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 11381 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 19001 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 50010 -j ACCEPT

# Drop all other connections
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

# Save iptables rules so that they're loaded if the system is restarted
sed 's/IPTABLES_SAVE_ON_STOP="no"/IPTABLES_SAVE_ON_STOP="yes"/' -i
/etc/sysconfig/iptables-config
sed 's/IPTABLES_SAVE_ON_RESTART="no"/IPTABLES_SAVE_ON_RESTART="yes"/' -i
/etc/sysconfig/iptables-config
```

## Leveraging Native Processor Instruction Sets

### AES-NI

The Advanced Encryption Standard New Instructions (AES-NI) instruction set is designed to improve the speed of encryption and decryption using AES. Some newer processors come with AES-NI, which can be enabled on a per-server basis. If you are uncertain whether AES-NI is available on a device, run the following command to verify:

```
$ grep -o aes /proc/cpuinfo
```

## Installation Overview

To determine whether the AES-NI kernel module is loaded, run the following command:

```
$ sudo lsmod | grep aesni
```

If the CPU supports AES-NI but the kernel module is not loaded, see your operating system documentation for instructions on installing the `aesni-intel` module.

### Intel RDRAND

The Intel RDRAND instruction set, along with its underlying Digital Random Number Generator (DRNG), is useful for generating keys for cryptographic protocols without using `haveged`.

To determine whether the CPU supports RDRAND, run the following command:

```
$ grep -o rdrand /proc/cpuinfo
```

To enable RDRAND, install `rng-tools` version 4 or higher:

1. Download the source code:

```
$ sudo wget  
http://downloads.sourceforge.net/project/gkernel/rng-tools/4/rng-tools-4.tar.gz
```

2. Extract the source code:

```
tar xvfz rng-tools-4.tar.gz
```

3. Enter the `rng-tools-4` directory:

```
$ cd rng-tools-4
```

4. Run `./configure`.

5. Run `make`.

6. Run `make install`.

Start `rngd` with the following command:

```
$ sudo rngd --no-tpm=1 -o /dev/random
```

## Initializing Key Trustee Server

After installing Key Trustee Server, you must initialize it before it is operational. Continue to [Initializing Standalone Key Trustee Server](#) or [Cloudera Navigator Key Trustee Server High Availability](#) for instructions.

## Installing Cloudera Navigator Key HSM



**Important:** Before installing Cloudera Navigator Key HSM, see [Deployment Planning for Data at Rest Encryption](#) for important considerations.

Cloudera Navigator Key HSM is a universal hardware security module (HSM) driver that translates between the target HSM platform and Cloudera Navigator Key Trustee Server.

With Navigator Key HSM, you can use a Key Trustee Server to securely store and retrieve encryption keys and other secure objects, without being limited solely to a hardware-based platform.

## Prerequisites

You must install Key HSM on the same host as Key Trustee Server. See [Data at Rest Encryption Requirements](#) for more information about encryption and Key HSM requirements.

## Setting Up an Internal Repository

You must create an internal repository to install or upgrade Cloudera Navigator Key HSM. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see [Creating and Using a Package Repository for Cloudera Manager](#) on page 174.

## Installing Navigator Key HSM



**Important:** If you have implemented Key Trustee Server high availability, install and configure Key HSM on each Key Trustee Server host.

### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
```

### 2. Install the CDH Repository

Key Trustee Server and Key HSM depend on the `bigtop-utils` package, which is included in the CDH repository. For instructions on adding the CDH repository, see [To add the CDH repository](#). To create a local CDH repository, see [Creating a Local Yum Repository](#) on page 219 for instructions.

### 3. Install Navigator Key HSM

Install the Navigator Key HSM package using `yum`:

```
$ sudo yum install keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the `/usr/share/keytrustee-server-keyhsm` directory by default.

## Installing Key Trustee KMS



**Important:** Following these instructions installs the required software to add the Key Trustee KMS service to your cluster; this enables you to use Cloudera Navigator Key Trustee Server as the underlying keystore for [HDFS Transparent Encryption](#). This *does not* install Key Trustee Server. See [Installing Cloudera Navigator Key Trustee Server](#) on page 205 for instructions on installing Key Trustee Server. You must install Key Trustee Server before installing and using Key Trustee KMS.

Key Trustee KMS is a custom Key Management Server (KMS) that uses Cloudera Navigator Key Trustee Server as the underlying keystore, instead of the file-based Java KeyStore (JKS) used by the default Hadoop KMS.



**Important:** Key Trustee KMS is supported only in Cloudera Manager deployments. You can install the software using parcels or packages, but running Key Trustee KMS outside of Cloudera Manager is not supported.

The **KMS (Navigator Key Trustee)** service in Cloudera Manager 5.3 is renamed to **Key Trustee KMS** in Cloudera Manager 5.4.

## Installation Overview

### Setting Up an Internal Repository

You must create an internal repository to install Key Trustee KMS. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172 if you are using parcels, or [Creating and Using a Package Repository for Cloudera Manager](#) on page 174 if you are using packages.

### Installing Key Trustee KMS Using Parcels

1. Go to **Hosts > Parcels**.
2. Click **Configuration** and add your internal repository to the **Remote Parcel Repository URLs** section. See [Configuring the Cloudera Manager Server to Use the Parcel URL](#) on page 174 for more information.
3. Download, distribute, and activate the Key Trustee KMS parcel. See [Managing Parcels](#) on page 57 for detailed instructions on using parcels to install or upgrade components.



**Note:** The KEYTRUSTEE\_SERVER parcel in Cloudera Manager is *not* the Key Trustee KMS parcel; it is the Key Trustee Server parcel. The parcel name for Key Trustee KMS is KEYTRUSTEE.

### Installing Key Trustee KMS Using Packages

1. After [Setting Up an Internal Repository](#) on page 212, configure the Key Trustee KMS host to use the repository. See [Modifying Clients to Find the Repository](#) on page 176 for more information.
2. Because the keytrustee-keyprovider package depends on the hadoop-kms package, you must add the CDH repository. See [To add the CDH repository](#) for instructions. If you want to create an internal CDH repository, see [Creating a Local Yum Repository](#) on page 219.
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
```

### Post-Installation Configuration

For instructions on installing Key Trustee Server and configuring Key Trustee KMS to use Key Trustee Server, see the following topics:

- [Installing Cloudera Navigator Key Trustee Server](#) on page 205
- [Enabling HDFS Encryption Using the Wizard](#)

### Installing Cloudera Navigator Encrypt



**Important:** Before installing Cloudera Navigator Encrypt, see [Deployment Planning for Data at Rest Encryption](#) for important considerations.

## Prerequisites

See [Data at Rest Encryption Requirements](#) for more information about encryption and Navigator Encrypt requirements.

## Setting Up an Internal Repository

You must create an internal repository to install or upgrade Navigator Encrypt. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see [Creating and Using a Package Repository for Cloudera Manager](#) on page 174.

## Installing Navigator Encrypt (RHEL-Compatible)



**Important:** Cloudera supports RHEL 7 with the following limitations:

- Only RHEL 7.2 and 7.1 are supported. RHEL 7.0 is not supported.
- Only new installations of RHEL 7.2 and 7.1 are supported by Cloudera. For upgrades to RHEL 7.1 or 7.2, contact your OS vendor and see [Does Red Hat support upgrades between major versions of Red Hat Enterprise Linux?](#)

### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

### 2. Install Kernel Libraries

For Navigator Encrypt to run as a kernel module, you must download and install the kernel development headers. Each kernel module is compiled specifically for the underlying kernel version. Running as a kernel module allows Navigator Encrypt to provide high performance and completely transparency to user-space applications.

To determine your current kernel version, run `uname -r`.

To install the development headers for your current kernel version, run:

```
$ sudo yum install kernel-headers-$(uname -r) kernel-devel-$(uname -r)
```

For OEL with the Unbreakable Enterprise Kernel (UEK), run:

```
$ sudo yum install kernel-uek-headers-$(uname -r) kernel-uek-devel-$(uname -r)
```



**Note:** For UEK3, you do not need to install `kernel-uek-headers-*`

If `yum` cannot find these packages, it displays an error similar to the following:

```
Unable to locate package <packagename>.
```

In this case, do one of the following to proceed:

- Find and install the kernel headers package by using a tool like [RPM Pbone](#).
- Upgrade your kernel to the latest version. If you upgrade the kernel, you must reboot after upgrading and select the kernel from the grub menu to make it active.

### 3. (RHEL or CentOS 6.6 Only) Install dkms

## Installation Overview

Because of a broken dependency in RHEL or CentOS 6.6, you must manually install the `dkms` package:

```
$ sudo yum install http://pkgs.repoforge.org/dkms/dkms-2.1.1.2-1.el6.rf.noarch.rpm
```

### 4. Install Navigator Encrypt

Install the Navigator Encrypt client using the `yum` package manager:

```
$ sudo yum install navencrypt
```

If you attempt to install Navigator Encrypt with incorrect or missing kernel headers, you see a message like the following:

```
Building navcryptfs 3.8.0 DKMS kernel module...
#####
# BUILDING ERROR #####
Creating symlink /var/lib/dkms/navcryptfs/3.8.0/source ->
/usr/src/navcryptfs-3.8.0
DKMS: add completed.
Error! echo
Your kernel headers for kernel 3.10.0-229.4.2.el7.x86_64 cannot be found at
/lib/modules/3.10.0-229.4.2.el7.x86_64/build or
/lib/modules/3.10.0-229.4.2.el7.x86_64/source.
#####
# BUILDING ERROR #####
Failed installation of navcryptfs 3.8.0 DKMS kernel module !
```

To recover, see [Navigator Encrypt Kernel Module Setup](#).

## Installing Navigator Encrypt (SLES)

### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

### 2. Install NTP

The Network Time Protocol (NTP) service synchronizes system time. Cloudera recommends using NTP to ensure that timestamps in system logs, cryptographic signatures, and other auditible events are consistent across systems. Install and start NTP with the following commands:

```
$ sudo zypper install ntp
# /etc/init.d/ntp start
```

### 3. Install the Kernel Module Package and Navigator Encrypt Client

Install the kernel module package (KMP) and Navigator Encrypt client with `zypper`:

#### 1. For SP2:

```
$ sudo zypper install
http://internal-repo.example.com/path/to/cloudera-navcryptfs-kmp-SP2-<kernel_flavor>-3.9.0_3.0.13_0.27-4.1.x86_64.rpm
$ sudo zypper install navencrypt
```

Replace *internal-repo.example.com/path/to/* with the hostname and path of the internal repository you created, and replace <kernel\_flavor> with the [kernel flavor](#) for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

## 2. For SP3:

```
$ sudo zypper install
http://internal-repo.example.com/path/to/cloudera-navencryptfs-kmp-SP3-<kernel_flavor>-3.9.0_3.0.76_0.11-4.1.x86_64.rpm
$ sudo zypper install navencrypt
```

Replace *internal-repo.example.com/path/to/* with the hostname and path of the internal repository you created, and replace <kernel\_flavor> with the [kernel flavor](#) for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

## 3. For SP4:

```
$ sudo zypper install
http://internal-repo.example.com/path/to/cloudera-navencryptfs-kmp-SP4-<kernel_flavor>-3.9.0_3.0.101_63-4.1.x86_64.rpm
$ sudo zypper install navencrypt
```

Replace *internal-repo.example.com/path/to/* with the hostname and path of the internal repository you created, and replace <kernel\_flavor> with the [kernel flavor](#) for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

## 4. Enable Unsupported Modules

Edit `/etc/modprobe.d/unsupported-modules` and set `allow_unsupported_modules` to 1. For example:

```
# Every kernel module has a flag 'supported'. If this flag is not set loading
# this module will taint your kernel. You will not get much help with a kernel
# problem if your kernel is marked as tainted. In this case you firstly have
# to avoid loading of unsupported modules.
#
# Setting allow_unsupported_modules 1 enables loading of unsupported modules
# by modprobe, setting allow_unsupported_modules 0 disables it. This can
# be overridden using the --allow-unsupported-modules command line switch.
allow_unsupported_modules 1
```

## Installing Navigator Encrypt (Debian or Ubuntu)

### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

- **Ubuntu**

```
$ echo "deb http://repo.example.com/path/to/ubuntu/stable $DISTRO_CODENAME main" | sudo
tee -a /etc/apt/sources.list
```

- **Debian**

```
$ echo "deb http://repo.example.com/path/to/debian/stable $DISTRO_CODENAME main" | sudo
tee -a /etc/apt/sources.list
```

Import the GPG key by running the following command:

```
$ wget -O - http://repo.example.com/path/to/gpg_gazzang.asc | apt-key add -
```

Update the repository index with `apt-get update`.

## Installation Overview

### 2. Install NTP

The Network Time Protocol (NTP) service synchronizes system time. Cloudera recommends using NTP to ensure that timestamps in system logs, cryptographic signatures, and other auditible events are consistent across systems. Install and start NTP with the following commands:

```
$ sudo apt-get install ntp  
$ sudo /etc/init.d/ntp start
```

### 3. Install Kernel Headers

Determine your kernel version by running `uname -r`, and install the appropriate headers:

```
$ sudo apt-get install linux-headers-$(uname -r)
```

### 4. Install the Navigator Encrypt Client

Install Navigator Encrypt:

```
$ sudo apt-get install navencrypt
```

## Post Installation

To ensure that Navigator Encrypt and NTP start after a reboot, add them to the start order with `chkconfig`:

```
$ sudo chkconfig --level 235 navencrypt-mount on  
$ sudo chkconfig --level 235 ntpd on
```

## AES-NI and RDRAND

The Advanced Encryption Standard New Instructions (AES-NI) instruction set is designed to improve the speed of encryption and decryption using AES. Some newer processors come with AES-NI, which can be enabled on a per-server basis.

Both the eCryptfs and dm-crypt back ends for Navigator Encrypt can automatically detect and use AES-NI if it is available. If you are uncertain whether AES-NI is available on a device, run the following command to verify:

```
$ grep -o aes /proc/cpuinfo
```

To determine whether the AES-NI kernel module is loaded, run the following command:

```
$ sudo lsmod | grep aesni
```

If the CPU supports AES-NI but the kernel module is not loaded, see your operating system documentation for instructions on installing the `aesni-intel` module.

Navigator Encrypt needs a source of random numbers if it is using dm-crypt as its back end. Use `rng-tools` version 4 or higher to seed the system's entropy pool, using the RDRAND instruction. To install and start `rngd`:

#### 1. Download the source code:

```
$ sudo wget  
http://downloads.sourceforge.net/project/gkernel/rng-tools/4/rng-tools-4.tar.gz
```

#### 2. Extract the source code:

```
tar xvfz rng-tools-4.tar.gz
```

**3.** Enter the `rng-tools-4` directory:

```
$ cd rng-tools-4
```

**4.** Run `./configure`

**5.** Run `make`

**6.** Run `make install`

Once you have installed `rng-tools`, start the `rngd` daemon by running the following command as root:

```
$ sudo rngd --no-tpm=1 -o /dev/random
```

For improved performance, Cloudera recommends configuring Navigator Encrypt to read directly from `/dev/random` instead of `/dev/urandom`.

To configure Navigator Encrypt to use `/dev/random` as an entropy source, add `--use-random` to the `nav encrypt-prepare` command when you are setting up Navigator Encrypt.

## Uninstalling and Reinstalling Navigator Encrypt

### Uninstalling Navigator Encrypt

For RHEL-compatible OSes:

```
$ sudo yum remove nav encrypt  
$ sudo yum remove nav encrypt-kernel-module
```

These commands remove the software itself. On RHEL-compatible OSes, the `/etc/nav encrypt` directory is not removed as part of the uninstallation. Remove it manually if required.

### Reinstalling Navigator Encrypt

After uninstalling Navigator Encrypt, repeat the installation instructions for your distribution in [Installing Cloudera Navigator Encrypt](#) on page 212.

When Navigator Encrypt is uninstalled, the configuration files and directories located in `/etc/nav encrypt` are not removed. Consequently, you do not need to use the `nav encrypt register` command during reinstallation. If you no longer require the previous installation configuration information in the directory `/etc/nav encrypt`, you can remove its contents.

# Installing and Deploying CDH Using the Command Line

## Before You Install CDH 5 on a Cluster

**Important:**

- Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).
- **Upgrading from CDH 4:** If you are upgrading from CDH 4, you must first uninstall CDH 4, then install CDH 5; see [Upgrading from CDH 4 to CDH 5](#) on page 691.
- 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 \(MRv1\) to MapReduce \(MRv2\)](#) on page 234 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 31.
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 Resource Manager.

- For more information and instructions on setting up a new HA configuration, see [High Availability](#).



**Important:**

If you decide to configure [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`. After completing the [HDFS HA software configuration](#), follow the installation instructions under [Deploying HDFS High Availability](#).

- To upgrade an existing configuration, follow the instructions under [Upgrading to CDH 5](#) on page 693.

## Creating a Local Yum Repository



**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

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 machines in your cluster do 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 also need an Internet connection to download updated RPMs to your local repository.

1. Download the repo file. Click the link for your RHEL or CentOS system in the table, find the appropriate repo file, and save in `/etc/yum.repos.d/`.

For OS Version	Link to CDH 5 Repository
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

2. Install a web server such as `apache/lighttpd` on the machine that hosts 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 RPM packages, `yum-utils` and `createrepo`, if 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 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/`.

## Installation Overview

6. On your web server, issue the following command from the 5 / subdirectory of your RPM directory:

```
createrepo .
```

This creates 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 220, starting at Step 2 (you have already done Step 1).

## Installing the Latest CDH 5 Release

### CDH 5 Installation Options

There are multiple ways to install CDH 5:



**Note:** Cloudera recommends automatically installing CDH 5 and dependencies with Cloudera Manager.

- Automatically install CDH 5 with a [Cloudera Manager Deployment](#) on page 74. This is the simplest and preferred method.
- Manually install the CDH 5 package or repository in one of three ways:
  - Install the CDH 5 "1-click" package (preferred manual method) *OR*
  - Add the CDH 5 repository *OR*
  - Build your own CDH 5 repository.
- Manually install the CDH 5 tarball. See "Package and Tarball Binaries" below.

### Package and Tarball Binaries

#### Installing from Packages

- To install and deploy YARN, see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).
- To install and deploy MRv1, see [Deploying MapReduce v1 \(MRv1\) on a Cluster](#).

#### Installing from a Tarball

- The CDH 5 [tarball](#) deploys YARN and includes the MRv1 binaries. There is no separate tarball for MRv1. The MRv1 scripts are in the directory, `bin-mapreduce1`, and examples are in `examples-mapreduce1`.

#### Before You Begin Installing CDH 5 Manually

- This page explains new installations. To upgrade from an earlier release, see [Upgrading from CDH 4 to CDH 5](#).
- To migrate from MRv1 to YARN, see [Migrating from MapReduce \(MRv1\) to MapReduce \(MRv2\)](#) on page 234.
- For a list of supported operating systems, see [CDH 5 Requirements and Supported Versions](#) on page 31.

- Installing CDH 5 requires `sudo` privileges. If necessary, use root user (superuser) to configure `sudo` privileges.
- CDH 5 requires the Oracle Java Development Kit (JDK). See [Java Development Kit Installation](#).
- In CDH 5, both the NameNode and Resource Manager (or Job Tracker) can be configured for [High Availability](#).
- Use the `service` (8) command to start and stop services rather than running scripts in `/etc/init.d` directly.

**Note: Running Services**

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

**Steps to Install CDH 5 Manually****Step 1: Add or Build the CDH 5 Repository or Download the "1-click Install" package.**

- To install CDH 5 on a [RHEL](#) system, download packages with `yum` or use a web browser.
- To install CDH 5 on a [SLES](#) system, download packages with `zypper` or YaST or use a web browser.
- To install CDH 5 on an [Ubuntu or Debian](#) system, download packages with `apt` or use a web browser.

**On RHEL-compatible Systems**

Use one of the following methods to install CDH 5 on RHEL-compatible systems.

- [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. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

2. Install the RPM for all RHEL versions:

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

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps to install both implementations.

**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

## Installation Overview

### **OR: To add the CDH 5 repository:**

Download the repo file. Click the link for your RHEL or CentOS system in the table, find the appropriate repo file, and save in /etc/yum.repos.d/.

For OS Version	Link to CDH 5 Repository
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps to install both implementations.



#### **Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

### **OR: To build a Yum repository:**

Follow the instructions at [Creating a Local Yum Repository](#) to create your own yum repository:

- Download the appropriate repo file
- Create the repo
- Distribute the repo and set up a web server.

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps to install both implementations.



#### **Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

## On SLES Systems

Use one of the following methods to download the CDH 5 repository or package on SLES systems.

- [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.

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

2. Install the RPM:

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

**3.** Update your system package index by running:

```
$ sudo zypper refresh
```

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps 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
```

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps to install both implementations.



**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

**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.

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps to install both implementations.



**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

## 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](#)

**To download and install the CDH 5 "1-click Install" package:**

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>

OS Version	Package Link
Trusty	<a href="#">Trusty package</a>

**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: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo apt-get update
```

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps 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)**

For Ubuntu Trusty systems, you must perform an extra step after adding the repository. See "Additional Step for Trusty" below.

OS Version	Command
Debian Wheezy	\$ sudo wget 'https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list
Ubuntu Precise	\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list
Ubuntu Lucid	\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/lucid/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list
Ubuntu Trusty	\$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list



**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo apt-get update
```

### 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.

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps 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](#).

Continue with [Step 2: Optionally Add a Repository Key](#) on page 225. Then choose [Step 3: Install CDH 5 with YARN](#) on page 226, or [Step 4: Install CDH 5 with MRv1](#) on page 228; or do both steps to install both implementations.

### Step 2: Optionally Add a Repository Key

**Before installing YARN or MRv1:** (Optional) 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 RHEL/CentOS/Oracle 5 systems:**

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

- **For RHEL/CentOS/Oracle 6 systems:**

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

- **For RHEL/CentOS/Oracle 7 systems:**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh5/redhat/7/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</pre>

## Installation Overview

OS Version	Command
	<pre>-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:** When configuring [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
<b>Resource Manager host</b> (analogous to MRv1 JobTracker) running:	
<i>RHEL/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>
<b>NameNode host</b> running:	
<i>RHEL/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>

Where to install	Install commands
<i>Ubuntu or Debian</i>	<code>sudo apt-get install hadoop-hdfs-namenode</code>
<b>Secondary NameNode host (if used) running:</b>	
<i>RHEL/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>
<b>All cluster hosts except the Resource Manager running:</b>	
<i>RHEL/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>
<b>One host in the cluster running:</b>	
<i>RHEL/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>
<b>All client hosts running:</b>	
<i>RHEL/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.

## Installation Overview

### Step 4: Install CDH 5 with MRv1



**Note:** If installing both MRv1 and YARN, do not install packages that you already installed in [Step 3: Install CDH 5 with YARN](#) on page 226. If installing YARN *only*, skip this step and go to [Step 3: Install CDH 5 with YARN](#) on page 226.



**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:** Ubuntu systems may try to start the service immediately after you install it. This should fail harmlessly, but you can find information at [askubuntu](#) on how to prevent this.

Where to install	Install commands
<b>JobTracker host running:</b>	
<i>RHEL/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>
<b>NameNode host running:</b>	

Where to install	Install commands
<i>RHEL/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>
<b>Secondary NameNode host (if used) running:</b>	
<i>RHEL/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>
<b>All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts running:</b>	
<i>RHEL/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>
<b>All client hosts running:</b>	
<i>RHEL/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

This section explains how to install LZO (Lempel–Ziv–Oberhumer) compression. For more information, see [Choosing and Configuring Data Compression](#)



**Note:** If upgrading (rather than installing for the first time), remove the old LZO version first. 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:

## Installation Overview

For OS Version	Do this
RHEL/CentOS/Oracle 5	Go to <a href="#">this link</a> and save the file in the /etc/yum.repos.d/ directory.
RHEL/CentOS/Oracle 6	Go to <a href="#">this link</a> and save the file in the /etc/yum.repos.d/ directory.
RHEL/CentOS/Oracle 7	Go to <a href="#">this link</a> and save the file in the /etc/yum.repos.d/ directory.
SLES	<ol style="list-style-type: none"><li>Run the following command: <pre>\$ sudo zypper addrepo -f https://archive.cloudera.com/gplextras5/sles/11/x86_64/gplextras/ cloudera-gplextras5.repo</pre></li><li>Update your system package index by running: <pre>\$ sudo zypper refresh</pre></li></ol>
Ubuntu or Debian	Go to <a href="#">this link</a> and save the file as /etc/apt/sources.list.d/gplextras.list. <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><p><b>Important:</b> Make sure you do not let the file name default to cloudera.list, as that will overwrite your existing cloudera.list.</p></div>

- Install the package on each host as follows:

For OS version	Install commands
RHEL/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>

- 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 259.



**Important:** Be sure to 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

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.

## Downloading and Installing an Earlier Release

Choose your Linux version and proceed as follows to install an earlier release:

- [On RHEL-compatible systems](#)
- [On SLES systems](#)
- [On Ubuntu and Debian systems](#)

### On RHEL-compatible systems

#### Step 1. Download and save the Yum repo file

Click the entry in the table below that matches your RHEL or CentOS system, go 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	<a href="#">Red Hat/CentOS/Oracle 5 link</a>
RHEL/CentOS 6 (64-bit)	<a href="#">Red Hat/CentOS 6 link</a>

#### 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.

## Installation Overview

4. Follow the instructions for RHEL 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
```

- 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, Cloudera refers to MapReduce (MRv2) as YARN. You can use the instructions later in this section to install:

- YARN (MRv2)
- MapReduce (MRv1)
- 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)

The MapReduce (MRv2) or YARN architecture splits the two primary responsibilities of the JobTracker — resource management and job scheduling/monitoring — into separate daemons: a global ResourceManager and per-application ApplicationMasters. With MRv2, the ResourceManager and per-host NodeManagers form the data-computation framework. The ResourceManager service effectively replaces the functions of the JobTracker, and NodeManagers

## Installation Overview

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 run and monitor the tasks. For details of this architecture, see [Apache Hadoop NextGen MapReduce \(YARN\)](#).

See also [Migrating from MapReduce \(MRv1\) to MapReduce \(MRv2\)](#) on page 234.

## Migrating from MapReduce (MRv1) to MapReduce (MRv2)

This is a guide to migrating from Apache MapReduce 1 (MRv1) to MapReduce (MRv2) (or YARN).

### Introduction

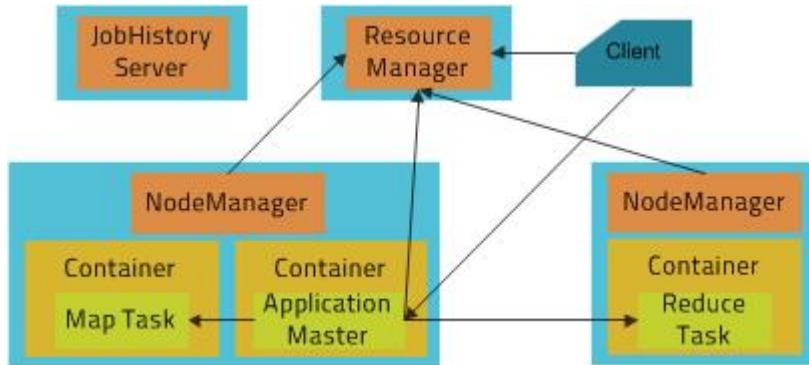
MapReduce 2, or Next Generation MapReduce, is a long needed upgrade to the way that scheduling, resource management, and execution occur in Hadoop. At their core, the improvements separate cluster resource management capabilities from MapReduce-specific logic. They enable Hadoop to share resources dynamically 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 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 ResourceManager 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 ApplicationMaster, 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 NodeManager, 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 can 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 are 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 use 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 are required to compile. It is orthogonal to binary compatibility - binaries for an application that is binary-compatible, but not source-compatible, continues to run fine on the new framework, but code changes are 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](#).

If you want your job to run against both MRv1 and MRv2, compile it against MRv2.

## 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&lt;version&gt;.jar</code>	<code>/usr/lib/hadoop-mapreduce/hadoop-streaming.jar</code>

Name	MapReduce MRv1 location	YARN location
Rumen	N/A	/usr/lib/hadoop-mapreduce/hadoop-rumen.jar
Hadoop Examples	/usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar	/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
DistCp v1	/usr/lib/hadoop-0.20-mapreduce/hadoop-tools.jar	/usr/lib/hadoop-mapreduce/hadoop-extras.jar
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.



#### Note:

As of CDH 5.4.0, configuring MapReduce jobs is simpler than before: instead of having to set both the heap size (`mapreduce.map.java.opts` or `mapreduce.reduce.java.opts`) and the container size (`mapreduce.map.memory.mb` or `mapreduce.reduce.memory.mb`), you can now choose to set only one of them; the other is inferred from `mapreduce.job.heap.memory-mb.ratio`. If you do not specify either of them, container size defaults to 1 GiB and the heap size is inferred.

The impact on user jobs is as follows: for jobs that do not set heap size, this increases the JVM size from 200 MB to a default 820 MB. This should be okay for most jobs, but streaming tasks might need more memory because their Java process takes their total usage over the container size. Even in that case, this would likely happen only for those tasks relying on aggressive GC to keep the heap under 200 MB.

### 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 261 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 YARN \(MRv2\) and MapReduce \(MRv1\)](#) for more configuration specifics; and, for detailed tuning advice with sample configurations, see [Tuning YARN](#).

**Table 24: YARN and MRv2 Memory Configuration**

Cloudera Manager Property Name	CDH Property Name	Default Configuration	Cloudera Tuning Guidelines
Container Memory Minimum	yarn.scheduler.minimum-allocation-mb	1 GB	0
Container Memory Maximum	yarn.scheduler.maximum-allocation-mb	64 GB	amount of memory on largest node
Container Memory Increment	yarn.scheduler.increment-allocation-mb	512 MB	Use a fairly large value, such as 128 MB
Container Memory	yarn.nodemanager.resource.memory-mb	8 GB	8 GB
Map Task Memory	mapreduce.map.memory.mb	1 GB	1 GB
Reduce Task Memory	mapreduce.reduce.memory.mb	1 GB	1 GB
Map Task Java Opts Base	mapreduce.map.java.opts	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m
Reduce Task Java Opts Base	mapreduce.reduce.java.opts	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m
ApplicationMaster Memory	yarn.app.mapreduce.am.resource.mb	1 GB	1 GB

Cloudera Manager Property Name	CDH Property Name	Default Configuration	Cloudera Tuning Guidelines
ApplicationMaster Java Opts Base	yarn.app.mapreduce.am.command-opts	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m

### 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.

For further information on tuning and resource management, see [Tuning YARN](#) and [YARN \(MRv2\) and MapReduce \(MRv1\) Schedulers](#).

## 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 rmadmin` 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 the MRv1 `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 *
<code>security.inter.tracker.protocol.acl</code>	<code>security.resourcemanager.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.resourcemanager-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>

## Installation Overview

Port	Use	Property
8031	ResourceManager Resource Tracker RPC (for NodeManagers)	yarn.resourcemanager.resource-tracker.address
8040	NodeManager Localizer RPC	yarn.nodemanager.localizer.address
8042	NodeManager Web UI and REST APIs	yarn.nodemanager.webapp.address
10020	Job History RPC	mapreduce.jobhistory.address
19888	Job History Web UI and REST APIs	mapreduce.jobhistory.webapp.address
13562	Shuffle HTTP	mapreduce.shuffle.port



**Note:** You can set `yarn.resourcemanager.hostname.id` for each ResourceManager instead of setting the ResourceManager values; this will cause YARN to use the default ports on those hosts.

### 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>	
<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 <a href="#">Configuring YARN (MRv2)</a> <a href="#">ResourceManager High Availability Using the Command Line</a> )	YARN / MRv2 has different RPC ports for different functionalities. Each port-related configuration must be suffixed with an id. Note that there is no <i>name</i> component in YARN.
<code>mapred.ha.jobtracker.rpc-address.name.id</code>	<code>yarn.resourcemanager.ha.admin.address</code>	
<code>mapred.ha.fencing.methods</code>	<code>yarn.resourcemanager.ha.fencer</code>	Not required to be specified
<code>mapred.client.failover.*</code>	<code>None</code>	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

MRv1	YARN / MRv2	Comment
	yarn.resourcemanager.store.class	org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore
mapred.ha.automatic-failover.enabled	yarn.resourcemanager.ha.automatic-failover.enabled	Enable automatic failover
mapred.ha.zkfc.port	yarn.resourcemanager.ha.automatic-failover.port	
mapred.job.tracker	yarn.resourcemanager.cluster.id	Cluster name

### Upgrading an MRv1 Installation Using Cloudera Manager

See [Importing MapReduce Configurations to YARN](#) for instructions.

### Upgrading an MRv1 Installation Using the Command Line

1. Uninstall the following packages: hadoop-0.20-mapreduce, hadoop-0.20-mapreduce-jobtracker, hadoop-0.20-mapreduce-tasktracker, hadoop-0.20-mapreduce-zkfc, hadoop-0.20-mapreduce-jobtrackerha.
2. **Install** the following additional packages : hadoop-yarn, hadoop-mapreduce, hadoop-mapreduce-historyserver, hadoop-yarn-resourcemanager, hadoop-yarn-nodemanager.
3. 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 [Resource Configuration](#) 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`.
4. 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 MRv1.

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 ApplicationMaster; if the job has completed, then this proxy is the JobHistoryServer. Thus, the user experience is similar to that of MRv1, 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

## Installation Overview

MRv1	YARN / MRv2
	or all of the following: <code>yarn.resourcemanager.address</code> <code>yarn.resourcemanager.scheduler.address</code> <code>yarn.resourcemanager.resource-tracker.address</code> <code>yarn.resourcemanager.admin.address</code>
<code>mapred.job.tracker.http.address</code>	<code>yarn.resourcemanager.webapp.address</code> or <code>yarn.resourcemanager.hostname</code>
<code>mapred.job.tracker.handler.count</code>	<code>yarn.resourcemanager.resource-tracker.client.thread-count</code>
<code>mapred.hosts</code>	<code>yarn.resourcemanager.nodes.include-path</code>
<code>mapred.hosts.exclude</code>	<code>yarn.resourcemanager.nodes.exclude-path</code>
<code>mapred.cluster.max.map.memory.mb</code>	<code>yarn.scheduler.maximum-allocation-mb</code>
<code>mapred.cluster.max.reduce.memory.mb</code>	<code>yarn.scheduler.maximum-allocation-mb</code>
<code>mapred.acls.enabled</code>	<code>yarn.acl.enable</code>
<code>mapreduce.cluster.acls.enabled</code>	<code>yarn.acl.enable</code>

### JobTracker Properties and JobHistoryServer Equivalents

MRv1	YARN / MRv2	Comment
<code>mapred.job.tracker.retiredjobs.cache.size</code>	<code>mapreduce.jobhistory.joblist.cache.size</code>	
<code>mapred.job.tracker.jobhistory.lru.cache.size</code>	<code>mapreduce.jobhistory.loadedjobs.cache.size</code>	
<code>mapred.job.tracker.history.completed.location</code>	<code>mapreduce.jobhistory.done-dir</code>	Local FS in MR1; stored in HDFS in MR2
<code>hadoop.job.history.user.location</code>	<code>mapreduce.jobhistory.done-dir</code>	
<code>hadoop.job.history.location</code>	<code>mapreduce.jobhistory.done-dir</code>	

### JobTracker Properties and MapReduce ApplicationMaster Equivalents

MRv1	YARN / MRv2	Comment
<code>mapreduce.jobtracker.staging.root.dir</code>	<code>yarn.app.mapreduce.am.staging-dir</code>	Now configurable per job

### TaskTracker Properties and NodeManager Equivalents

MRv1	YARN / MRv2
<code>mapred.tasktracker.map.tasks.maximum</code>	<code>yarn.nodemanager.resource.memory-mb</code> and <code>yarn.nodemanager.resource.cpu-vcores</code>
<code>mapred.tasktracker.reduce.tasks.maximum</code>	<code>yarn.nodemanager.resource.memory-mb</code> and <code>yarn.nodemanager.resource.cpu-vcores</code>
<code>mapred.tasktracker.expiry.interval</code>	<code>yarn.nm.liveliness-monitor.expiry-interval-ms</code>
<code>mapred.tasktracker.resourcecalculatorplugin</code>	<code>yarn.nodemanager.container-monitor.resource-calculator.class</code>
<code>mapred.tasktracker.taskmemorymanager.monitoring-interval</code>	<code>yarn.nodemanager.container-monitor.interval-ms</code>

MRv1	YARN / MRv2
mapred.tasktracker.tasks.sleeptime-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
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

## Installation Overview

MRv1	YARN / MRv2	Comment
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
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.queuename	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	

MRv1	YARN / MRv2	Comment
mapred.combine.recordsBeforeProgress	mapreduce.task.combine.progress.records	MRv1 name should still work - see <a href="#">MAPREDUCE-510</a>

### 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
mapreduce.task.tmp.dir	No longer configurable. Now always tmp/ (under container's local dir)
mapred.child.tmp	No longer configurable. Now always tmp/ (under container's local dir)
mapred.temp.dir	
mapred.jobtracker.instrumentation	ResourceManager does not accept instrumentation
mapred.jobtracker.plugins	ResourceManager does not accept plugins
mapred.task.cache.level	
mapred.queue.names	These go in the scheduler-specific configuration files
mapred.system.dir	
mapreduce.tasktracker.cache.local.numberdirectories	
mapreduce.reduce.input.limit	
io.sort.record.percent	Tuned automatically ( <a href="#">MAPREDUCE-64</a> )
mapred.cluster.map.memory.mb	Not necessary; MRv2 uses resources instead of slots
mapred.cluster.reduce.memory.mb	Not necessary; MRv2 uses resources instead of slots
mapred.max.tracker.blacklists	
mapred.jobtracker.maxtasks.per.job	Related configurations go in scheduler-specific configuration files

## Installation Overview

MRv1	Comment
mapred.jobtracker.taskScheduler.maxRunningTasksPerJob	Related configurations go in scheduler-specific configuration files
io.map.index.skip	
mapred.user.jobconf.limit	
mapred.local.dir.minspacestart	
mapred.local.dir.minspacekill	
hadoop.rpc.socket.factory.class.JobSubmissionProtocol	
mapreduce.tasktracker.outofband.heartbeat	Always on
mapred.jobtracker.job.history.block.size	

## 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 220.

To deploy CDH 5 on a cluster, do the following:

1. [Configuring Dependencies Before Deploying CDH on a Cluster](#) on page 246
2. [Deploying HDFS on a Cluster](#) on page 249
3. Deploy [YARN with MapReduce v2 \(YARN\)](#) or [MapReduce v1 \(MRv1\)](#)

See also:

- [Configuring the Daemons to Start on Boot](#) on page 269
- [Optimizing Performance in CDH](#)
- [Configuring Centralized Cache Management in HDFS](#)
- [Managing HDFS Snapshots](#)
- [Configuring an NFSv3 Gateway Using the Command Line](#)

### Configuring Dependencies Before Deploying CDH on a Cluster

This section explains the tasks you must perform before deploying CDH on a cluster.

#### Enabling NTP

CDH requires that you configure the [Network Time Protocol](#) (NTP) service on each machine in your cluster. To start NTP and configure it to run automatically on reboot, perform the following steps on each node in your cluster.

##### 1. Install NTP.

- For RHEL, CentOS, and Oracle:

```
yum install ntp
```

- For SLES:

```
zypper install ntp
```

- For Debian and Ubuntu:

```
apt-get install ntp
```

2. Open the `/etc/ntp.conf` file and add NTP servers, as in the following example.

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
```

3. Save and close the file.
4. Configure the NTP service to run at reboot.

```
chkconfig ntpd on
```

5. Start the NTP service.

```
service ntpd start
```

6. Synchronize the node.

```
ntpdate -u <your_ntp_server>
```

7. Synchronize the system clock (to prevent synchronization problems).

```
hwclock --systohc
```

## Configuring Network Names

**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

**Important:** CDH requires IPv4. IPv6 is not supported.

To ensure that the members of the cluster can communicate with each other, do the following on every system.

1. Set the hostname of each system to a unique name (not `localhost`). Using `hostname` is temporary and does not survive across reboots.

```
sudo hostname <hostname>
```



**Note:** To permanently set a hostname, use `sudo hostnamectl set-hostname <hostname>`.

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.

## Installation Overview

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](#)

### Disabling SELinux

[Security-Enhanced Linux](#) (SELinux) allows you to set access control through policies. You must disable SELinux on each host before you deploy CDH on your cluster.

To disable SELinux, perform the following steps on each host.

1. Check the SELinux state.

```
getenforce
```

If the output is either `permissive` or `disabled`, you can skip this task and go to [Disabling the Firewall](#) on page 248. If the output is `enforcing`, continue to the next step.

2. Open the `/etc/selinux/config` file (in some systems, the `/etc/sysconfig/selinux` file).
3. Change the line `SELINUX=enforcing` to `SELINUX=permissive`.
4. Save and close the file.
5. Restart your system or run the following command to disable SELinux immediately:

```
setenforce 0
```

### Disabling the Firewall

To disable the firewall on each host in your cluster, perform the following steps on each host.

- Save the existing iptables rule set.

```
iptables-save > /root/firewall.rules
```

- Disable iptables.

- For RHEL, CentOS, Oracle, and Debian:

```
chkconfig iptables off
```

and

```
/etc/init.d/iptables stop
```

- For SLES:

```
chkconfig SuSEfirewall2_setup off
```

and

```
rcSuSEfirewall2 stop
```

- For Ubuntu:

```
service ufw stop
```

## 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 [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 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

**Note: Running Services**

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

## Copying the Hadoop Configuration and Setting Alternatives

To customize the Hadoop configuration:

## Installation Overview

### 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`.

### 2. 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 RHEL-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 RHEL-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
fs.defaultFS	core-site.xml	<p>Note: <code>fs.default.name</code> is deprecated. Specifies the NameNode and the default file system, in the form <code>hdfs://&lt;namenode host&gt;:&lt;namenode port&gt;/</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>.</p> <p>Note: for the cluster to function correctly, the <code>&lt;namenode&gt;</code> part of the string <b>must</b> be the hostname (for example <code>mynamenode</code>), or the <a href="#">HA-enabled logical URI</a>, not the IP address.</p>
dfs.permissions.superusergroup	hdfs-site.xml	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

Property	Configuration File Location	Description
		point, unless you will be using a <a href="#">HDFS HA configuration</a> .
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:**

- 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.

- 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
```

- 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

## Installation Overview

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

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.

**Important:**

- The trash feature is disabled by default. Cloudera recommends that you enable it on all production clusters.
- 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"> <li>• If trash is enabled on the server configuration, then the value configured on the server is used and the client configuration is ignored.</li> <li>• If trash is disabled in the server configuration, then the client side configuration is checked.</li> <li>• If the value of this property is zero (the default), then the trash feature is disabled.</li> </ul>
<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 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.

## Installation Overview

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 358.

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](#).

## Installation Overview

### 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 RHEL-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 RHEL-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 261 or
- [Deploying MapReduce v1 \(MRv1\) on a Cluster](#) on page 266

### 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 5.7.x. If you use a lower 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](#):



#### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is 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 worker nodes instead of TaskTracker daemons. The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the tasks. For details of the new architecture, see [Apache Hadoop NextGen MapReduce \(YARN\)](#).

See also [Selecting Appropriate JAR files for Your Jobs](#) on page 235.



#### 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.

## Installation Overview

### 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: <code>yarn.resourcemanager.address</code> , <code>yarn.resourcemanager.admin.address</code> , <code>yarn.resourcemanager.scheduler.address</code> , <code>yarn.resourcemanager.resource-tracker.address</code> , <code>yarn.resourcemanager.webapp.address</code>
yarn.application.classpath	<code>\$HADOOP_CONF_DIR</code> , <code>\$HADOOP_COMMON_HOME/*</code> , <code>\$HADOOP_COMMON_HOME/lib/*</code> , <code>\$HADOOP_HDFS_HOME/*</code> , <code>\$HADOOP_HDFS_HOME/lib/*</code> , <code>\$HADOOP_MAPRED_HOME/*</code> , <code>\$HADOOP_MAPRED_HOME/lib/*</code> , <code>\$HADOOP_YARN_HOME/*</code> , <code>\$HADOOP_YARN_HOME/lib/*</code>	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
<code>yarn.nodemanager.local-dirs</code>	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> .
<code>yarn.nodemanager.log-dirs</code>	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> .
<code>yarn.nodemanager.remote-app-log-dir</code>	Specifies the URI of the directory where logs are aggregated. Set the value to either <code>hdfs://namenode-host.company.com:8020/var/log/hadoop-yarn/apps</code> using the fully-qualified domain name of your NameNode host, or <code>hdfs:/var/log/hadoop-yarn/apps</code> .

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>

```

## Installation Overview

```
</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 host:port
<code>mapreduce.jobhistory.webapp.address</code>	<code>historyserver.company.com:19888</code>	The address of the JobHistory Server web application host:port

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

Property	Recommended value	Description
hadoop.proxyuser.mapred.hosts	*	Allows the mapreduser 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`:

- Configure `yarn.app.mapreduce.am.staging-dir` in `mapred-site.xml`:

```
<property>
  <name>yarn.app.mapreduce.am.staging-dir</name>
  <value>/user</value>
</property>
```

- 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:

- Configure `mapreduce.jobhistory.intermediate-done-dir` and `mapreduce.jobhistory.done-dir` in `mapred-site.xml`.
- Create these two directories.
- Set permissions on `mapreduce.jobhistory.intermediate-done-dir` to 1777.
- 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 260 if you have not already done so.

#### Step 6: If Necessary, Start HDFS on Every Node in the Cluster

[Start HDFS](#) on page 259 if you have not already done so.

#### Step 7: If Necessary, Create the HDFS /tmp Directory

[Create the /tmp directory](#) on page 260 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: Start YARN and the MapReduce JobHistory Server

**To start YARN, start the ResourceManager and NodeManager services:**

**Note:**

Make sure you always start ResourceManager before starting NodeManager services.

On the ResourceManager 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 10: 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 11: 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

This topic describes configuration and startup tasks for MRv1 clusters only.

**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- Do not run MRv1 and YARN on the same set of nodes at the same time. This will degrade performance and may result in an unstable cluster deployment. To deploy YARN instead, see [Deploying MapReduce v2 \(YARN\) on a Cluster](#) on page 261. If you have installed CDH 5 from tarballs, the default deployment is YARN.

1. Make sure you have [configured and deployed HDFS](#).

2. Configure the JobTracker's RPC server.

- a. Open the `mapred-site.xml` file in the custom directory you created when you [copied the Hadoop configuration](#).
- b. Specify the hostname and (optionally) port of the JobTracker's RPC server, in the form `<host><port>`. The default value is `local`. With the default value, JobTracker runs on demand when you run a MapReduce job. Do not try to start the JobTracker yourself in this case. If you specify the host other than `local`, use the hostname (for example `mynamenode`) not the IP address.

For example:

```
<property>
  <name>mapred.job.tracker</name>
  <value>jobtracker-host.company.com:8021</value>
</property>
```



**Note:** For instructions on configuring a highly available JobTracker, see [MapReduce \(MRv1\) JobTracker High Availability](#).

### 3. Configure local storage directories for use by MRv1 daemons.

- Open the `mapred-site.xml` file in the custom directory you created when you [copied the Hadoop configuration](#).
- Edit the `mapred.local.dir` property to specify the directories where the TaskTracker will store temporary data and intermediate map output files while running MapReduce jobs. Cloudera recommends that you specify a directory on each of the JBOD mount points: `/data/1/mapred/local` through `/data/N/mapred/local`. For example:

```
<property>
  <name>mapred.local.dir</name>
  <value>/data/1/mapred/local,/data/2/mapred/local,/data/3/mapred/local</value>
</property>
```

- 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
```

- 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
```

- Set the permissions to `drwxr-xr-x`.

- Configure a health check script for DataNode processes.

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`.

For more information, go to the section titled "Configuring the Node Health Check Script" in [the Apache cluster setup documentation](#).

- Set the `mapreduce.jobtracker.restart.recover` property to `true`. This ensures that running jobs that fail because of a system crash or hardware failure are re-run when the JobTracker restarts. A recovered job has the following properties:
  - It will have the same job ID as when it was submitted.

## Installation Overview

- 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.

**h.** Repeat for each TaskTracker.

### 4. Configure a health check script for DataNode processes.

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](#) on page 253, the DataNode will shut down after the configured number of directory failures). The following 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
```

For more information, go to the section titled "Configuring the Node Health Check Script" in [the Apache cluster setup documentation](#).

### 5. Configure JobTracker recovery.

Set the property `mapreduce.jobtracker.restart.recover` to `true` in `mapred-site.xml`.

JobTracker ensures that running jobs that fail because of a system crash or hardware failure are re-run when the JobTracker restarts. A recovered job has 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.

### 6. 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
```

### 7. 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
```

### 8. Create and configure the `mapred.system.dir` directory in HDFS. 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.

**9. Start MapReduce by starting 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
```

**10 Create a home directory for each MapReduce user. On the NameNode, enter:**

```
$ 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
```

**11 Set HADOOP\_MAPRED\_HOME.**

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

Set this environment variable for each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop in an MRv1 installation.

**12 Configure the Hadoop daemons to start at boot time. For more information, see [Configuring the Daemons to Start on Boot](#) on page 269.****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 5.7.x. If you use a lower 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.

## Installation Overview

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 220 and [Deploying CDH 5 on a Cluster](#) on page 246.



#### Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower 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 270
- [Flume Installation](#) on page 272
- [HBase Installation](#) on page 281
- [HCatalog Installation](#) on page 310
- [Hive Installation](#) on page 329
- [HttpFS Installation](#) on page 358
- [Hue Installation](#) on page 362
- [Impala Installation](#) on page 316
- [KMS Installation and Upgrade](#) on page 394
- [Mahout Installation](#) on page 396
- [Oozie Installation](#) on page 398
- [Pig Installation](#) on page 418
- [Search Installation](#) on page 422
- [Sentry Installation](#) on page 434
- [Snappy Installation](#) on page 436
- [Spark Installation](#) on page 436
- [Sqoop 1 Installation](#) on page 438
- [Sqoop 2 Installation](#) on page 442
- [Whirr Installation](#) on page 450
- [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](#), 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 271
- [Crunch Packaging](#) on page 271
- [Installing and Upgrading Crunch](#) on page 271
- [Crunch Documentation](#) on page 272

### 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### 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*`.

## Installation Overview

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 datastore.



#### Note:

To install Flume using Cloudera Manager, see [Managing Flume](#).

### Upgrading Flume

Use the instructions that follow to upgrade Flume.



#### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

### Upgrading Flume from CDH 4 to CDH 5

Use one of the following sets of instructions to upgrade Flume from CDH 4 to CDH 5, depending on which version of Flume you have been running under CDH 4:

- If you are running Flume 1.x, use [these instructions](#).
- If you are running Flume 0.9.x, use [these instructions](#).

### Upgrading Flume 1.x to CDH 5

Use the instructions that follow to upgrade Flume 1.x from CDH 4 to CDH 5. You must remove the CDH 4 version and then install the CDH 5 version.

Step 1: Remove Flume 1.x from your cluster.

1. Stop the Flume Node processes on each node where they are running:

```
$ sudo service flume-ng-agent stop
```

2. Uninstall the old Flume components:

**On Red Hat-compatible systems:**

```
$ sudo yum remove flume-ng agent flume-ng
```

**On SLES systems:**

```
$ sudo zypper remove flume-ng agent flume-ng
```

**On Ubuntu systems:**

```
$ sudo apt-get remove flume-ng agent flume-ng
```

**Step 2. Install the new version of Flume**

Follow the instructions in the rest of this document to install Flume 1.x under CDH 5.

**Migrating from Flume 0.9.x under CDH 4 to Flume 1.x under CDH 5**

Flume 1.x is a significant departure from Flume 0.9.x in its implementation although many of the original concepts are the same. If you're already familiar with Flume, here are some significant points.

- You still have sources and sinks and they still do the same thing. They are now connected by channels.
- Channels are pluggable, and dictate durability. Flume 1.x ships with an in-memory channel for fast, but non-durable event delivery. There are also JDBC-based and file-based channels for durable event delivery.
- There are no more logical or physical nodes. All physical nodes are "agents," and agents can run zero or more sources and sinks.
- There is no longer a Master, and no ZooKeeper dependency. At this time, Flume runs with a simple file-based configuration system.
- Just about everything is a plugin — some end user facing, some for tool and system developers.
- Thrift and Avro legacy Flume sources are provided to enable sending events from Flume 0.9.4 to Flume 1.x.

You must uninstall Flume 0.9.x and then install Flume 1.x, as follows.

**Step 1: Remove Flume 0.9.x from your cluster.**

1. Stop the Flume Node processes on each node where they are running:

```
$ sudo service flume-node stop
```

2. Stop the Flume Master:

```
$ sudo service flume-master stop
```

3. Uninstall the old Flume components:

**On Red Hat-compatible systems:**

```
$ sudo yum remove flume
```

**On SLES systems:**

```
$ sudo zypper remove flume
```

**On Ubuntu systems:**

```
$ sudo apt-get remove flume
```

**Step 2. Install the new version of Flume**

Follow the instructions in the rest of this document to install Flume 1.x from a [tarball](#) or [packages](#).

***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](#) on page 708.

## Installation Overview

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 274 or [Installing the Flume Tarball](#) on page 274.



### 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

**To install Flume on Ubuntu and other Debian systems:**

```
$ sudo apt-get install flume-ng
```

**To install Flume On RHEL-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 RHEL-compatible systems:**

```
$ sudo yum install flume-ng-doc
```

**To install the documentation on SLES systems:**

```
$ sudo zypper install flume-ng-doc
```

## Installation Overview

### 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, 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	/etc/flume-ng/conf	
Configuration File	/etc/flume-ng/conf/flume.conf	This configuration will be picked-up by the flume agent startup script.
Template of User Customizable Configuration File	/etc/flume-ng/conf/flume-conf.properties.template	Contains a sample config. To use this configuration you should copy this file onto /etc/flume-ng/conf/flume.conf and then modify as appropriate
Template of User Customizable environment file	/etc/flume-ng/conf/flume-env.sh.template	If you want modify this file, copy it first and modify the copy
Daemon Log Directory	/var/log/flume-ng	Contains log files generated by flume agent
Default Flume Home	/usr/lib/flume-ng	Provided by RPMS and DEBS
Flume Agent startup script	/etc/init.d/flume-ng-agent	Provided by RPMS and DEBS
Recommended tar.gz Flume Home	/usr/local/lib/flume-ng	Recommended but installation dependent
Flume Wrapper Script	/usr/bin/flume-ng	Called by the Flume Agent startup script
Flume Agent configuration file	/etc/default/flume-ng-agent	Allows you to specify non-default values for the agent name and for the configuration file location

## Installation Overview

### 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	Ingests 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 <a href="#">Flume Developer Guide</a> for more details.	EmbeddedSource
org.apache.flume.source.kafka.KafkaSource	Streams data from Kafka to Hadoop or from any Flume source to Kafka.	KafkaSource
org.apache.flume.source.taildir.TaildirSource	<p>Watches specified files, and tails them in near real-time when it detects appends to these files.</p> <ul style="list-style-type: none"> <li>This source is reliable and does not miss data, even when the tailing files rotate.</li> <li>It periodically writes the last read position of each file in a position file using the JSON format.</li> <li>If Flume is stopped or down for some reason, it can restart tailing from the position written in the existing position file.</li> <li>It can add event headers to each tailing file group.</li> </ul>	TaildirSource

**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 AsyncHBaseSink is recommended. See <a href="#">Importing Data Into HBase</a> .	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	AsyncHBaseSink

## Installation Overview

Type	Description	Implementation Class
	recommended HBase sink, but note that it does not support Kerberos. See <a href="#">Importing Data Into HBase</a> .	
org.apache.flume.sink.solr.MorphlineSolrSink	Extracts and transforms data from Flume events, and loads it into Apache Solr servers. See the section on MorphlineSolrSink in the Flume User Guide listed under <a href="#">Viewing the Flume Documentation</a> on page 280.	MorphlineSolrSink
org.apache.flume.sink.kafka.KafkaSink	Used to send data to Kafka from a Flume source. You can use the Kafka sink in addition to Flume sinks such as HBase or HDFS.	KafkaSink

### 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
org.apache.flume.channel.kafka.KafkaChannel	Use the Kafka channel: <ul style="list-style-type: none"><li>• To write to Hadoop directly from Kafka without using a source.</li><li>• To write to Kafka directly from Flume sources without additional buffering.</li><li>• As a reliable and highly available channel for any source/sink combination.</li></ul>	KafkaChannel

### 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 to 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.

### 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.



### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

Use the following sections to install, update, and configure HBase:

## 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 5.4 HBase Changes

CDH 5.4 introduces HBase 1.0, which represents a major 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.

## Highly-Available Read Replicas

CDH 5.4 introduces highly-available read replicas. Using read replicas, clients can request, on a per-read basis, a read result using a new consistency model, timeline consistency, rather than strong consistency. The read request is sent to the RegionServer serving the region, but also to any RegionServers hosting replicas of the region. The client receives the read from the fastest RegionServer to respond, and receives an indication of whether the response was from the primary RegionServer or from a replica. See [HBase Read Replicas](#) for more details.

## MultiWAL Support

CDH 5.4 introduces support for writing multiple write-ahead logs (MultiWAL) on a given RegionServer, allowing you to increase throughput when a region writes to the WAL. See [Configuring HBase MultiWAL Support](#).

## Installation Overview

### Medium-Object (MOB) Storage

CDH 5.4 introduces a mechanism for storing objects between 100 KB and 10 MB in a default configuration, or *medium objects*, directly in HBase. Storing objects up to 50 MB is possible with additional configuration. Previously, storing these medium objects directly in HBase could degrade performance due to write amplification caused by splits and compactions.

MOB storage requires HFile V3.

### doAs Impersonation for the Thrift Gateway

Prior to CDH 5.4, the Thrift gateway could be configured to authenticate to HBase on behalf of the client as a static user. A new mechanism, doAs Impersonation, allows the client to authenticate as any HBase user on a per-call basis for a higher level of security and flexibility.

### Namespace Create Authorization

Prior to CDH 5.4, only global admins could create namespaces. Now, a Namespace Create authorization can be assigned to a user, who can then create namespaces.

### Authorization to List Namespaces and Tables

Prior to CDH 5.4, authorization checks were not performed on list namespace and list table operations, so you could list the names of many tables or namespaces, regardless of your authorization. In CDH 5.4, you are not able to list namespaces or tables you do not have authorization to access.

### Crunch API Changes for HBase

In CDH 5.4, Apache Crunch adds the following API changes for HBase:

- `HBaseTypes.cells()` was added to support serializing HBase Cell objects.
- Each method of `HFileUtils` now supports `PCollection<C extends Cell>`, which includes both `PCollection<KeyValue>` and `PCollection<Cell>`, on their method signatures.
- `HFileTarget`, `HBaseTarget`, and `HBaseSourceTarget` each support any subclass of `Cell` as an output type. `HFileSource` and `HBaseSourceTarget` still return `KeyValue` as the input type for backward-compatibility with existing Crunch pipelines.

### ZooKeeper 3.4 Is Required

HBase 1.0 requires ZooKeeper 3.4.

### HBase API Changes for CDH 5.4

CDH 5.4.0 introduces HBase 1.0, which includes some major changes to the HBase APIs. Besides the changes listed above, some APIs have been deprecated in favor of new public APIs.

- The `HConnection` API is deprecated in favor of [Connection](#).
- The `HConnectionManager` API is deprecated in favor of [ConnectionFactory](#).
- The `HTable` API is deprecated in favor of [Table](#).
- The `HTableAdmin` API is deprecated in favor of [Admin](#).

### HBase 1.0 API Example

```
Configuration conf = HBaseConfiguration.create();
try (Connection connection = ConnectionFactory.createConnection(conf)) {
    try (Table table = connection.getTable(TableDescriptor.create(tableName).build())) {
        // use table as needed, the table returned is lightweight
    }
}
```

### CDH 5.3 HBase Changes

CDH 5.4 introduces HBase 0.98.6, which represents a minor upgrade to HBase. CDH 5.3 provides `checkAndMutate(RowMutations)`, in addition to existing support for atomic `checkAndPut` as well as `checkAndDelete` operations on individual rows ([HBASE-11796](#)).

### *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](#).

### *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 78 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 285

### 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
ipc.server.listen.queue.size	hbase.ipc.server.listen.queue.size
ipc.server.max.callqueue.size	hbase.ipc.server.max.callqueue.size
ipc.server.max.callqueue.length	hbase.ipc.server.max.callqueue.length
ipc.server.read.threadpool.size	hbase.ipc.server.read.threadpool.size
ipc.server.tcpkeepalive	hbase.ipc.server.tcpkeepalive
ipc.server.tcpnodelay	hbase.ipc.server.tcpnodelay
ipc.client.call.purge.timeout	hbase.ipc.client.call.purge.timeout
ipc.client.connection.maxidletime	hbase.ipc.client.connection.maxidletime
ipc.client.idlethreshold	hbase.ipc.client.idlethreshold
ipc.client.kill.max	hbase.ipc.client.kill.max

### 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 288.

### 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.

#### *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.

## Installation Overview

- 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

## Installation Overview

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-1083](#).

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.striped.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.striped.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.

## Installation Overview

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 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
  </description>
```

```

    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 281, 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 CDH 4 to CDH 5*

CDH 5.0 HBase is based on Apache HBase 0.96.1.1 Remember that once a cluster has been upgraded to CDH 5, it cannot be reverted to CDH 4. To ensure a smooth upgrade, this section guides you through the steps involved in upgrading HBase from the older CDH 4.x releases to CDH 5.

These instructions also apply to upgrading HBase from CDH 4.x directly to CDH 5.1.0, which is a supported path.

When upgrading from CDH 4.x to CDH 5.5.1, extra steps are required. See [Extra steps must be taken when upgrading from CDH 4.x to CDH 5.5.1..](#)

### Prerequisites

HDFS and ZooKeeper should be available while upgrading HBase.

### Overview of Upgrade Procedure

Before you can upgrade HBase from CDH 4 to CDH 5, your HFiles must be upgraded from HFile v1 format to HFile v2, because CDH 5 no longer supports HFile v1. The upgrade procedure itself is different if you are using Cloudera Manager or the command line, but has the same results. The first step is to check for instances of HFile v1 in the HFiles and mark them to be upgraded to HFile v2, and to check for and report about corrupted files or files with unknown versions, which need to be removed manually. The next step is to rewrite the HFiles during the next major compaction. After the HFiles are upgraded, you can continue the upgrade. After the upgrade is complete, you must recompile custom coprocessors and JARs.

### Upgrade HBase Using the Command Line

CDH 5 comes with an upgrade script for HBase. You can run `/usr/lib/hbase/bin/hbase --upgrade` to see its Help section. The script runs in two modes: `-check` and `-execute`.

#### Step 1: Check for HFile v1 files and compact if necessary

- Run the upgrade command in `-check` mode, and examine the output.

```
$ /usr/lib/hbase/bin/hbase upgrade -check
```

## Installation Overview

Your output should be similar to the following:

```
Tables Processed:  
hdfs://localhost:41020/myHBase/.META.  
hdfs://localhost:41020/myHBase/usertable  
hdfs://localhost:41020/myHBase/TestTable  
hdfs://localhost:41020/myHBase/t  
  
Count of HFileV1: 2  
HFileV1:  
hdfs://localhost:41020/myHBase/usertable  
/fa02dac1f38d03577bd0f7e666f12812/family/249450144068442524  
hdfs://localhost:41020/myHBase/usertable  
/ecdd3eaee2d2fcf8184ac025555bb2af/family/249450144068442512  
  
Count of corrupted files: 1  
Corrupted Files:  
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812/family/1  
Count of Regions with HFileV1: 2  
Regions to Major Compact:  
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812  
hdfs://localhost:41020/myHBase/usertable/ecdd3eaee2d2fcf8184ac025555bb2af
```

In the example above, you can see that the script has detected two HFile v1 files, one corrupt file and the regions to major compact.

By default, the script scans the root directory, as defined by `hbase.rootdir`. To scan a specific directory, use the `--dir` option. For example, the following command scans the `/myHBase/testTable` directory.

```
/usr/lib/hbase/bin/hbase upgrade --check --dir /myHBase/testTable
```

- Trigger a major compaction on each of the reported regions. This major compaction rewrites the files from HFile v1 to HFile v2 format. To run the major compaction, start HBase Shell and issue the `major_compact` command.

```
$ /usr/lib/hbase/bin/hbase shell  
hbase> major_compact 'usertable'
```

You can also do this in a single step by using the `echo` shell built-in command.

```
$ echo "major_compact 'usertable'" | /usr/lib/hbase/bin/hbase shell
```

- Once all the HFileV1 files have been rewritten, running the upgrade script with the `-check` option again will return a "No HFile v1 found" message. It is then safe to proceed with the upgrade.

Step 2: Gracefully shut down CDH 4 HBase cluster

Shut down your CDH 4 HBase cluster before you run the upgrade script in `-execute` mode.

**To shut down HBase gracefully:**

- Stop the REST and Thrift server and clients, then stop the cluster.

- Stop the Thrift server and clients:

```
sudo service hbase-thrift stop
```

Stop the REST server:

```
sudo service hbase-rest stop
```

- Stop the cluster by shutting down the master and the RegionServers:

- Use the following command on the master node:

```
sudo service hbase-master stop
```

- b.** Use the following command on each node hosting a RegionServer:

```
sudo service hbase-regionserver stop
```

**2.** Stop the ZooKeeper Server:

```
$ sudo service zookeeper-server stop
```

Step 3: Uninstall the old version of HBase and replace it with the new version.

**1. To remove HBase on Red-Hat-compatible systems:**

```
$ sudo yum remove hadoop-hbase
```

**To remove HBase on SLES systems:**

```
$ sudo zypper remove hadoop-hbase
```

**To remove HBase on Ubuntu and Debian systems:**

```
$ sudo apt-get purge hadoop-hbase
```



**Warning:**

If you are upgrading an Ubuntu or Debian system from CDH3u3 or lower, you **must** use `apt-get purge` (rather than `apt-get remove`) to make sure the re-install succeeds, but be aware that `apt-get purge` removes all your configuration data. If you have modified any configuration files, DO NOT PROCEED before backing them up.

- 2.** Follow the instructions for installing the new version of HBase at [HBase Installation](#) on page 281.

Step 4: Run the HBase upgrade script in -execute mode



**Important:** Before you proceed with Step 4, upgrade your CDH 4 cluster to CDH 5. See [Upgrading to CDH 5](#) on page 693 for instructions.

This step executes the actual upgrade process. It has a verification step which checks whether or not the Master, RegionServer and backup Master znodes have expired. If not, the upgrade is aborted. This ensures no upgrade occurs while an HBase process is still running. If your upgrade is aborted even after shutting down the HBase cluster, retry after some time to let the znodes expire. Default znode expiry time is 300 seconds.

As mentioned earlier, ZooKeeper and HDFS should be available. If ZooKeeper is managed by HBase, then use the following command to start ZooKeeper.

```
/usr/lib/hbase/bin/hbase-daemon.sh start zookeeper
```

The upgrade involves three steps:

- **Upgrade Namespace:** This step upgrades the directory layout of HBase files.
- **Upgrade Znodes:** This step upgrades /hbase/replication (znodes corresponding to peers, log queues and so on) and table znodes (keep table enable/disable information). It deletes other znodes.
- **Log Splitting:** In case the shutdown was not clean, there might be some Write Ahead Logs (WALs) to split. This step does the log splitting of such WAL files. It is executed in a “non distributed mode”, which could make the upgrade process longer in case there are too many logs to split. To expedite the upgrade, ensure you have completed a clean shutdown.

## Installation Overview

Run the upgrade command in `-execute` mode.

```
$ /usr/lib/hbase/bin/hbase upgrade -execute
```

Your output should be similar to the following:

```
Starting Namespace upgrade
Created version file at hdfs://localhost:41020/myHBase with version=7
Migrating table testTable to hdfs://localhost:41020/myHBase/.data/default/testTable
...
Created version file at hdfs://localhost:41020/myHBase with version=8
Successfully completed NameSpace upgrade.
Starting Znode upgrade
...
Successfully completed Znode upgrade
Starting Log splitting
...
Successfully completed Log splitting
```

The output of the `-execute` command can either return a success message as in the example above, or, in case of a clean shutdown where no log splitting is required, the command would return a "No log directories to split, returning" message. Either of those messages indicates your upgrade was successful.



### 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 5 (Optional): Move Tables to Namespaces

CDH 5 introduces namespaces for HBase tables. As a result of the upgrade, all tables are automatically assigned to namespaces. The `root`, `meta`, and `acl` tables are added to the `hbase` system namespace. All other tables are assigned to the `default` namespace.

To move a table to a different namespace, take a snapshot of the table and clone it to the new namespace. After the upgrade, do the snapshot and clone operations before turning the modified application back on.



**Warning:** Do not move datafiles manually, as this can cause data corruption that requires manual intervention to fix.

## Step 6: Recompile coprocessors and custom JARs.

Recompile any coprocessors and custom JARs, so that they will work with the new version of HBase.

## FAQ

[In order to prevent upgrade failures because of unexpired znodes, is there a way to check/force this before an upgrade?](#)

The upgrade script "executes" the upgrade when it is run with the `-execute` option. As part of the first step, it checks for any live HBase processes (RegionServer, Master and backup Master), by looking at their znodes. If any such znode is still up, it aborts the upgrade and prompts the user to stop such processes, and wait until their znodes have expired. This can be considered an inbuilt check.

The `-check` option has a different use case: To check for HFile v1 files. This option is to be run on live CDH 4 clusters to detect HFile v1 and major compact any regions with such files.

What are the steps for Cloudera Manager to do the upgrade?

See [Upgrade to CDH 5](#) for instructions on upgrading HBase within a Cloudera Manager deployment.

#### *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](#) on page 708.

During a rolling upgrade from CDH 5.0.x to CDH 5.4.x the HBase Master UI will display the URLs to the old HBase RegionServers using an incorrect info port number. Once the rolling upgrade completes the HBase master UI will use the correct port number.

#### 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 RegionServers:

- Use the following command on the master node:

```
sudo service hbase-master stop
```

- Use the following command on each node hosting a RegionServer:

```
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 457.

## Installation Overview

To install the new version of HBase, follow directions in the next section, [HBase Installation](#) on page 281.



### 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 RHEL-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 301, [Configuring HBase in Pseudo-Distributed Mode](#) on page 303, and [Deploying HBase on a Cluster](#) on page 305 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 RHEL 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
increateBlockOutputStream 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 Using Cloudera Manager*

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

1. Go to the HBase service.
2. Click the **Configuration** tab.
3. Select **Scope > Master** or **Scope > RegionServer**.
4. Locate the **Maximum Process File Descriptors** property or search for it by typing its name in the Search box.
5. Edit the property value.

If more than one role group applies to this configuration, edit the value for the appropriate role group. See [Modifying Configuration Properties Using Cloudera Manager](#).

6. Click **Save Changes** to commit the changes.
7. Restart the role.
8. Restart the service.

### *Configuring ulimit for HBase Using the Command Line*



#### **Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

Cloudera recommends increasing the maximum number of file handles to more than 10,000. Increasing the file handles for the user running the HBase process is an operating system configuration, not an HBase configuration. A common mistake is to increase the number of file handles for a particular user when HBase is 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 user, use the `ulimit -n` command while logged in as that user.

To set the maximum number of processes a user can start, use the `ulimit -u` command. You can also use the `ulimit` command to set many other limits. For more information, see the online documentation for your operating system, or the output of the `man ulimit` command.

## Installation Overview

To make the changes persistent, 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 Using the Command Line



#### Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

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 -      nproc   2048
hdfs -      nofile  32768
hbase -     nproc   2048
hbase -     nofile  32768
```



#### 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.DFSClient: Could not obtain block
blk_XXXXXXXXXXXXXXXXXXXX_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](#) 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 Transparent 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.

### Using Hedged Reads



#### Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).



#### Note:

To enable hedged reads for HBase, edit the `hbase-site.xml` file on each server. Set `dfs.client.hedged.read.threadpool.size` to the number of threads to dedicate to running hedged threads, and set the `dfs.client.hedged.read.threshold.millis` configuration property to the number of milliseconds to wait before starting a second read against a different block replica. Set `dfs.client.hedged.read.threadpool.size` to 0 or remove it from the configuration to disable the feature. After changing these properties, restart your cluster.

The following is an example configuration for hedged reads for HBase.

```
<property>
  <name>dfs.client.hedged.read.threadpool.size</name>
  <value>20</value>  <!-- 20 threads -->
</property>
<property>
  <name>dfs.client.hedged.read.threshold.millis</name>
  <value>10</value>  <!-- 10 milliseconds -->
</property>
```

### 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
```

## Installation Overview

### HBase Shell Overview

- To get help and to see all available commands, use the `help` command.
- To get help on a specific command, use `help "command"`. For example:

```
hbase> help "create"
```

- To remove an attribute from a table or column family or reset it to its default value, set its value to `nil`. For example, use the following command to remove the `KEEP_DELETED_CELLS` attribute from the `f1` column of the `users` table:

```
hbase> alter 'users', { NAME => 'f1', KEEP_DELETED_CELLS => nil }
```

- To exit the HBase Shell, type `quit`.

### Setting Virtual Machine Options for HBase Shell

HBase in CDH 5.2 and higher 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:+PrintGCAppliedTime -XX:+PrintGCDateStamps  
-XX:+PrintGCDetails -Xloggc:$HBASE_HOME/logs/gc-hbase.log" ./bin/hbase shell
```

### Scripting with HBase Shell

CDH 5.2 and higher include 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.

CDH 5.7 and higher include the `get_splits` command, which returns the split points for a given table:

```
hbase> get_splits 't2'  
Total number of splits = 5  
=> [ "", "10", "20", "30", "40" ]
```

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
```

### Troubleshooting HBase

See [Troubleshooting HBase](#).

### Configuring the BlockCache

See [Configuring the HBase BlockCache](#).

### Configuring the Scanner Heartbeat

See [Configuring the HBase Scanner Heartbeat](#).

### 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 RegionServer, 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 RHEL-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
```

## Installation Overview

### Starting the HBase Master

- On RHEL 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 RegionServer 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 307, [Using MapReduce with HBase](#) on page 308 and [Troubleshooting HBase](#) on page 308.

### Installing and Starting the HBase Thrift Server

#### To install Thrift on RHEL-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 RHEL-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>
```

## Installation Overview

```
<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 459

#### 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 RHEL-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 go 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 307, [Using MapReduce with HBase](#) on page 308 and [Troubleshooting HBase](#) on page 308.

**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 RegionServer](#) 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 306.

### *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 RegionServer, 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.

The HBase Thrift service is light-weight, and can be run on any node in the cluster.

### *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. In CDH 5.7.0 and higher, you do not need to use `hbase.zookeeper.property.clientPort`. Instead, you can specify the port along with the hostname for each ZooKeeper host:

```
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>zk1.example.com:2181,zk2.example.com:20000,zk3.example.com:31111</value>
</property>
```

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 307, [Using MapReduce with HBase](#) on page 308 and [Troubleshooting HBase](#) on page 308. 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
```

### HBase Shell Overview

- To get help and to see all available commands, use the `help` command.
- To get help on a specific command, use `help "command"`. For example:

```
hbase> help "create"
```

- To remove an attribute from a table or column family or reset it to its default value, set its value to `nil`. For example, use the following command to remove the `KEEP_DELETED_CELLS` attribute from the `f1` column of the `users` table:

```
hbase> alter 'users', { NAME => 'f1', KEEP_DELETED_CELLS => nil }
```

- To exit the HBase Shell, type `quit`.

### Setting Virtual Machine Options for HBase Shell

HBase in CDH 5.2 and higher 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

CDH 5.2 and higher include 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.

CDH 5.7 and higher include the `get_splits` command, which returns the split points for a given table:

```
hbase> get_splits 't2'  
Total number of splits = 5  
=> [ "", "10", "20", "30", "40" ]
```

You can also write Ruby scripts for use with HBase Shell. Example Ruby scripts are included in the `hbase-examples/src/main/ruby/` directory.

## Installation Overview

### 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 310.

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 RegionServer, you will not be able to create a table with LZO compression until you re-start the RegionServer.

#### **Why this happens**

When the RegionServer 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 RegionServer, so the cached results, which pre-date LZO, cause the create to fail.

#### **What to do**

Restart the RegionServer. 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.

## Installation Overview

Location	Purpose	Troubleshooting Notes
/hbase/logs/.oldWALs	Contains HBase WAL files that have already been written to disk. A HBase maintenance thread removes them periodically based on a TTL.	To tune the length of time a WAL stays in the .oldWALs before it is removed, configure the hbase.master.logcleaner.ttl property, which defaults to 60000 milliseconds, or 1 hour.
/hbase/.logs/.corrupt	Contains corrupted HBase WAL files.	Do not remove corrupt WALs manually. If the size of any subdirectory of /hbase/.logs/ 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 is a table and storage management layer for Hadoop that makes the same table information available to Hive, Pig, MapReduce, and Sqoop. Table definitions are maintained in the Hive metastore, which HCatalog requires. WebHCat allows you to access HCatalog using an HTTP (REST style) interface.

This page explains how to install and configure HCatalog and WebHCat. For Sqoop, see [Sqoop-HCatalog Integration](#) in the Sqoop User Guide.

#### Configuring HCatalog Using Cloudera Manager

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

1. Go to the Hive service by clicking **Clusters > Hive**.
2. Select the **Hive Instances** tab.
3. Add a WebHCat server role:
  - a. Click **Add Role Instances**.
  - b. Click **Select hosts** under WebHCat Server.
  - c. Select the host on which you want the WebHCat server; this adds a **WHCS** icon.
  - d. Click **OK**.
4. Click **Continue**.
5. Start the new role type.
  - a. Select the new role type, **WebHCat Server**.
  - b. Select **Actions for Selected > Start**.
  - c. Click **Start and Close**.

#### Configuring HCatalog Using the Command Line

This section applies to unmanaged deployments *without* Cloudera Manager. Use the following sections to install, configure and use HCatalog:

- [Prerequisites](#)
- [Installing and Upgrading the HCatalog RPM or Debian Packages](#) on page 311
- [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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### *Upgrading HCatalog from CDH 4 to CDH 5*

To upgrade HCatalog from CDH 4 to CDH 5, proceed as follows.



#### Note:

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#) on page 691, you can skip Step 1 below and proceed with installing the new CDH 5 version of HCatalog.

#### Step 1: Remove the CDH 4 version of HCatalog

##### To remove HCatalog on a RHEL-compatible system:

```
$ sudo yum remove webhcat-server hcatalog
```

##### To remove HCatalog on an Ubuntu or other Debian system:

```
$ sudo apt-get remove webhcat-server hcatalog
```

## Installation Overview

### To remove HCatalog on a SLES system:

```
$ sudo zypper remove webhcatalog-server hcatalog
```

Step 2: Install the new version of WebHCat and HCatalog

Follow instructions under [Installing the WebHCat REST Server](#) on page 312 and [Installing HCatalog for Use with Pig and MapReduce](#) on page 313.



#### 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.

*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 312 and [Installing HCatalog for Use with Pig and MapReduce](#) on page 313.



#### 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 RHEL system:**

```
$ sudo yum install hive-webhcatt-server
```

**To install the WebHCat REST server components on an Ubuntu or other Debian system:**

```
$ sudo apt-get install hive-webhcatt-server
```

**To install the WebHCat REST server components on a SLES system:**

```
$ sudo zypper install hive-webhcatt-server
```

**Note:**

- You can change the default port 50111 by creating or editing the following file and restarting WebHCat:

```
/etc/webhcatt/conf/webhcatt-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-webhcatt-server` and `hive-webhcatt`.

*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 RHEL 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>
```

## Installation Overview

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 webhcatt-server start  
$ sudo service webhcatt-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/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.hive.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": "groupstable"}
```

## 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 401.

Category	Resource Type	Description
General	<a href="#">:version (GET)</a>	Return a list of supported response types.
	<a href="#">status (GET)</a>	Return the WebHCat server status.
	<a href="#">version (GET)</a>	Return a list of supported versions and the current version.
	<a href="#">version/hive (GET)</a>	Return the Hive version being run.
	<a href="#">version/hadoop (GET)</a>	Return the Hadoop version being run.
DDL	<a href="#">ddl (POST)</a>	Perform an HCatalog DDL command.
	<a href="#">ddl/database (GET)</a>	List HCatalog databases.
	<a href="#">ddl/database/:db (GET)</a>	Describe an HCatalog database.
	<a href="#">ddl/database/:db (PUT)</a>	Create an HCatalog database.
	<a href="#">ddl/database/:db (DELETE)</a>	Delete (drop) an HCatalog database.
	<a href="#">ddl/database/:db/table (GET)</a>	List the tables in an HCatalog database.

## Installation Overview

Category	Resource Type	Description
	<a href="#">ddl/database/:db/table/:table (GET)</a>	Describe an HCatalog table.
	<a href="#">ddl/database/:db/table/:table (PUT)</a>	Create a new HCatalog table.
	<a href="#">ddl/database/:db/table/:table (POST)</a>	Rename an HCatalog table.
	<a href="#">ddl/database/:db/table/:table (DELETE)</a>	Delete (drop) an HCatalog table.
	<a href="#">ddl/database/db/table/existingtable/like/newtable (PUT)</a>	Create a new HCatalog table like an existing one.
	<a href="#">ddl/database/:db/table/:table/partition (GET)</a>	List all partitions in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/partition/partition (GET)</a>	Describe a single partition in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/partition/partition (PUT)</a>	Create a partition in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/partition/partition (DELETE)</a>	Delete (drop) a partition in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/column (GET)</a>	List the columns in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/column/column (GET)</a>	Describe a single column in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/column/column (PUT)</a>	Create a column in an HCatalog table.
	<a href="#">ddl/database/:db/table/:table/property (GET)</a>	List table properties.
	<a href="#">ddl/database/:db/table/:table/property/property (GET)</a>	Return the value of a single table property.
	<a href="#">ddl/database/:db/table/:table/property/property (PUT)</a>	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* DataNodes in the cluster.

The Impala package installs these binaries:

- `impalad` - The Impala daemon. Plans and executes queries against HDFS, HBase, and Amazon S3 data. [Run one impalad process](#) on each node in the cluster that has a DataNode.
- `statestored` - 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 an external mechanism such as Hive or by uploading data to the Amazon S3 filesystem.) [Run one instance of this daemon](#) on a node in your cluster, preferably on the same host as the `statestored` 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 DataNodes 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 317 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 4 and CDH 5 platforms. For details, see the [Supported Operating Systems](#) page for [CDH 4](#) or [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 316.

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 DataNodes 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.cloudera.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 2.2 and higher uses the SSSE3 instruction set, which is included in newer processors.



**Note:** This required level of processor is the same as in Impala version 1.x. The Impala 2.0 and 2.1 releases had a stricter requirement for the SSE4.1 instruction set, which has now been relaxed.

- 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 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`, writeable 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 317 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 4 Installation Guide](#) or the [CDH 5 Installation Guide](#).
2. Install the Hive metastore somewhere in your cluster, as described in the Hive Installation topic in the [CDH 4 Installation Guide](#) or 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

## Installation Overview

/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 317. (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.

6. 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.
7. 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
```

- 8.** 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 324 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 [Apache Impala \(incubating\) Incompatible Changes](#) in the *Release Notes* for any changes needed in your source code, startup scripts, and so on.
- Also check [Apache Impala \(incubating\) Known Issues](#) in the *Release Notes* for any issues or limitations that require workarounds.

### *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
```

## Installation Overview

### 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 [Apache Impala \(incubating\) 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 statestored if the service started successfully.

```
$ sudo service impala-state-store start  
$ ps ax | grep [s]tatestored  
6819 ? S1 0:07 /usr/lib/impala/sbin/statestored -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 ? S1 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 ? S1 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 326 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:

In CDH 5.7 / Impala 2.5 and higher, Impala UDFs and UDAs written in C++ are persisted in the metastore database. Java UDFs are also persisted, if they were created with the new `CREATE FUNCTION` syntax for Java UDFs, where the Java function argument and return types are omitted. Java-based UDFs created with the old `CREATE FUNCTION` syntax do not persist across restarts because they are held in the memory of the catalogd daemon. Until you re-create such Java UDFs using the new `CREATE FUNCTION` syntax, you must reload those Java-based UDFs by running the original `CREATE FUNCTION` statements again each time you restart the catalogd daemon. Prior to CDH 5.7 / Impala 2.5, the requirement to reload functions after a restart applied to both C++ and Java functions.

#### *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`, `statestore`, and `catalog` 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 DataNode using a command similar to the following:

```
$ sudo service impala-server start
```



#### Note:

In CDH 5.7 / Impala 2.5 and higher, Impala UDFs and UDAs written in C++ are persisted in the metastore database. Java UDFs are also persisted, if they were created with the new `CREATE FUNCTION` syntax for Java UDFs, where the Java function argument and return types are omitted. Java-based UDFs created with the old `CREATE FUNCTION` syntax do not persist across restarts because they are held in the memory of the catalogd daemon. Until you re-create such Java UDFs using the new `CREATE FUNCTION` syntax, you must reload those Java-based UDFs by running the original `CREATE FUNCTION` statements again each time you restart the catalogd daemon. Prior to CDH 5.7 / Impala 2.5, the requirement to reload functions after a restart applied to both C++ and Java functions.

If any of the services fail to start, review:

- [Reviewing Impala Logs](#)
- [Troubleshooting Impala](#)

## Installation Overview

### 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: **Clusters > 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 on systems not managed by Cloudera Manager, change:

```
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-false}
```

to:

```
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-true}
```

On systems managed by Cloudera Manager, enable the **Enable Core Dump** setting for the Impala service.



### 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.
- On systems managed by Cloudera Manager, the default location for core dumps is on a temporary filesystem, which can lead to out-of-space issues if the core dumps are large, frequent, or not removed promptly. To specify an alternative location for the core dumps, filter the Impala configuration settings to find the `core_dump_dir` option, which is available in Cloudera Manager 5.4.3 and higher. This option lets you specify a different directory for core dumps for each of the Impala-related daemons.

- 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`.

- During troubleshooting, Cloudera Support might direct you to change other values, particularly for `IMPALA_SERVER_ARGS`, to work around issues or gather debugging information.



### 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 `statestored` 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 statestored Daemon

The statestored 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

Using Hive data in HBase is a common task. See [Importing Data Into HBase](#).

For information about Hive on Spark, see [Running Hive on Spark](#).

Use the following sections to install, update, and configure Hive.

### About Hive

Apache Hive is a powerful data warehousing application for Hadoop. It enables you to access your data using Hive QL, a language similar to SQL.

[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. As of CDH 5, Hive supports [HCatalog](#) which must be installed separately.

### 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 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 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.



**Warning:** Because of concurrency and security issues, HiveServer1 and the Hive CLI 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 CLI is not needed if you are using Beeline with HiveServer2.



**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.

## Installation Overview

- 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 337 for complete instructions for each type of supported database.



### Warning:

Make sure you have read and understood all [incompatible changes](#) and [known issues](#) before you upgrade Hive.

## Upgrading Hive from CDH 4 to CDH 5



### Note:

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#), you can skip Step 1 below and proceed with installing the new CDH 5 version of Hive.

### Step 1: Remove Hive



### 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. Exit the Hive console and make sure no Hive scripts are running.
2. Stop any HiveServer processes that are running. If HiveServer is running as a daemon, use the following command to stop it:

```
$ sudo service hive-server stop
```

If HiveServer is running from the command line, stop it with <CTRL>-c.

3. Stop the metastore. If the metastore is running as a daemon, use the following command to stop it:

```
$ sudo service hive-metastore stop
```

If the metastore is running from the command line, stop it with <CTRL>-c.

4. Remove Hive:

```
$ sudo yum remove hive
```

**To remove Hive on SLES systems:**

```
$ sudo zypper remove hive
```

**To remove Hive on Ubuntu and Debian systems:**

```
$ sudo apt-get remove hive
```

## Step 2: Install the new Hive version on all hosts (Hive servers and clients)

See [Installing Hive](#).

### **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: Configure the Hive Metastore

You must configure the Hive metastore and initialize the service before you can use Hive. See [Configuring the Hive Metastore](#) for detailed instructions.

## Step 4: Upgrade the Metastore Schema

### **Important:**

- Cloudera strongly encourages you to make a backup copy of your metastore database before running the upgrade scripts. You will need this backup copy if you run into problems during the upgrade or 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 run a script, you must first cd to the directory that script is in: that is /usr/lib/hive/scripts/metastore/upgrade/<database>.

As of CDH 5, there are now two ways to do this. You could either use Hive's `schematool` or use the schema upgrade scripts provided with the Hive package.

### **Using `schematool` (Recommended):**

The Hive distribution includes an offline 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.

To upgrade the schema, use the `upgradeSchemaFrom` option to specify the version of the schema you are currently using (see table below) and the compulsory `dbType` option to specify the database you are using. The example that follows shows an upgrade from Hive 0.10.0 (CDH 4) for an installation using the Derby database.

```
$ 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>
Upgrade script upgrade-0.10.0-to-0.11.0.derby.sql
Completed upgrade-0.10.0-to-0.11.0.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
```

## Installation Overview

Possible values for the dbType option are mysql, postgres, derby or oracle. The following table lists the Hive versions corresponding to the older CDH releases.

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 4.x	0.10.0
CDH 5.0, 5.1	0.12.0
CDH 5.2	0.13.0

See [Using the Hive Schema Tool](#) for more details on how to use schematool.

### Using Schema Upgrade Scripts:

Run the appropriate schema upgrade script(s); they are in `/usr/lib/hive/scripts/metastore/upgrade/`. Start with the script for your database and Hive version, and run all subsequent scripts.

For example, if you are currently running Hive 0.10 with MySQL, and upgrading to Hive 0.13.1, start with the script for Hive 0.10 to 0.11 for MySQL, then run the script for Hive 0.11 to 0.12 for MySQL, then run the script for Hive 0.12 to 0.13.1.

For more information about upgrading the schema, see the README in `/usr/lib/hive/scripts/metastore/upgrade/`.

### Step 5: Configure HiveServer2

HiveServer2 is an improved version of the original HiveServer (HiveServer1, no longer supported). Some configuration is required before you initialize HiveServer2; see [Configuring HiveServer2](#) for details.

### Step 6: Upgrade Scripts for HiveServer2 (if necessary)

If you have been running HiveServer1, you may need to make some minor modifications to your client-side scripts and applications when you upgrade:

- HiveServer1 does not support concurrent connections, so many customers run a dedicated instance of HiveServer1 for each client. These can now be replaced by a single instance of HiveServer2.
- HiveServer2 uses a different connection URL and driver class for the JDBC driver. If you have existing scripts that use JDBC to communicate with HiveServer1, you can modify these scripts to work with HiveServer2 by changing the JDBC driver URL from `jdbc:hive://hostname:port` to `jdbc:hive2://hostname:port`, and by changing the JDBC driver class name from `org.apache.hive.jdbc.HiveDriver` to `org.apache.hive.jdbc.HiveDriver`.

### Step 7: Start the Metastore, HiveServer2, and Beeline

See:

- [Starting the Metastore](#)
- [Starting HiveServer2](#)
- [Using Beeline](#)

### Step 8: Upgrade the JDBC driver on the clients

The driver used for CDH 4.x does not work with CDH 5.x. Install the new version, following [these instructions](#).

## Upgrading Hive from a Lower 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](#) on page 708.

### 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.
- If you are upgrading Hive from CDH 5.0.5 to CDH 5.4, 5.3 or 5.2 on Debian 7.0, and a Sentry version higher than 5.0.4 and lower than 5.1.0 is installed, you must upgrade Sentry before upgrading Hive; otherwise the upgrade will fail. See [Apache Hive Known Issues](#) for more details.
- CDH 5.2 and higher clients cannot communicate with CDH 5.1 and lower servers. This means that you must upgrade the server before the clients.

To upgrade Hive from a lower 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 335

### Step 3: Verify that the Hive Metastore is Properly Configured

See [Configuring the Hive Metastore](#) on page 337 for detailed instructions.

### Step 4: Upgrade the Metastore Schema



#### Important:

- Cloudera strongly encourages you to make a backup copy of your metastore database before running the upgrade scripts. You will need this backup copy if you run into problems during the upgrade or 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 run a script, you must first cd to the directory that script is in: that is `/usr/lib/hive/scripts/metastore/upgrade/<database>`.

As of CDH 5, there are now two ways to do this. You could either use Hive's `schematool` or use the schema upgrade scripts provided with the Hive package.

#### Using `schematool` (Recommended):

The Hive distribution includes an offline 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.

To upgrade the schema, use the `upgradeSchemaFrom` option to specify the version of the schema you are currently using (see table below) and the compulsory `dbType` option to specify the database you are using. The example that follows shows an upgrade from Hive 0.10.0 (CDH 4) for an installation using the Derby database.

```
$ 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>
Upgrade script upgrade-0.10.0-to-0.11.0.derby.sql
Completed upgrade-0.10.0-to-0.11.0.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
```

Possible values for the `dbType` option are `mysql`, `postgres`, `derby` or `oracle`. The following table lists the Hive versions corresponding to the older CDH releases.

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 4.x	0.10.0
CDH 5.0, 5.1	0.12.0
CDH 5.2	0.13.0

See [Using the Hive Schema Tool](#) for more details on how to use `schematool`.

#### Using Schema Upgrade Scripts:

Run the appropriate schema upgrade script(s); they are in `/usr/lib/hive/scripts/metastore/upgrade/`. Start with the script for your database and Hive version, and run all subsequent scripts.

For example, if you are currently running Hive 0.10 with MySQL, and upgrading to Hive 0.13.1, start with the script for Hive 0.10 to 0.11 for MySQL, then run the script for Hive 0.11 to 0.12 for MySQL, then run the script for Hive 0.12 to 0.13.1.

For more information about upgrading the schema, see the README in `/usr/lib/hive/scripts/metastore/upgrade/`.

#### Step 5: Start the Metastore, HiveServer2, and Beeline

See:

- [Starting the Metastore](#) on page 352
- [Starting, Stopping, and Using HiveServer2](#) on page 352

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	\$ sudo yum install <pkg1> <pkg2> ...
SLES	\$ sudo zypper install <pkg1> <pkg2> ...
Ubuntu or Debian	\$ sudo apt-get install <pkg1> <pkg2> ...

## Installation Overview

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

The section describes how to tune HiveServer2 and Hive metastore memory and garbage collection properties.

#### 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. The objective is to size to reduce impact of Java garbage collection on active processing by the service.)	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



**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 Beeline 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

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 Beeline 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 Beeline 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`.

### *Table Partitions*

**Tip:** Cloudera recommends keeping table partitions below two or three thousand for optimal performance.

When a hive query has to reference more than a few thousand partitions, performance can suffer. Multiple queries must be run against the Hive Metastore database to retrieve and update these partitions and HDFS must move these files around.

For the best performance, design your tables to partition on fewer columns or to have less granular time frames, for example by day instead of hourly. Also, hone your queries to use only a subset of a table's partitions.

### *Configuration for WebHCat*

If you want to use WebHCat, you need to set the `PYTHON_CMD` variable in `/etc/default/hive-webhcats-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 deployment options and provides instructions for setting up a database in a recommended configuration.

### *Metastore Deployment Modes*

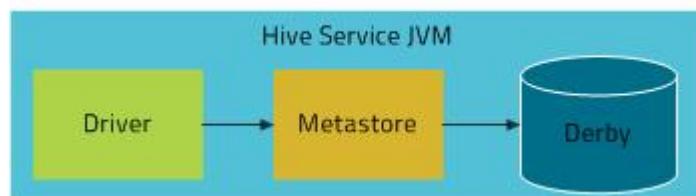


**Note:** On this page, **HiveServer**, refers to HiveServer1 or HiveServer2, whichever you are using.

#### *Embedded Mode*

**Cloudera recommends using this mode for experimental purposes only.**

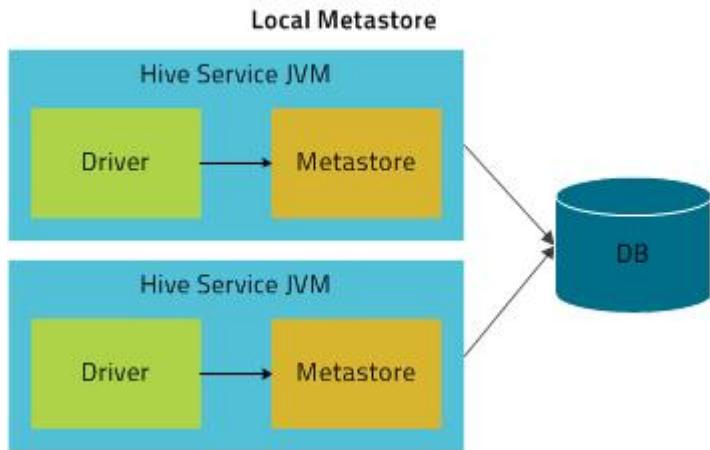
##### **Embedded Metastore**



## Installation Overview

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 are 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.

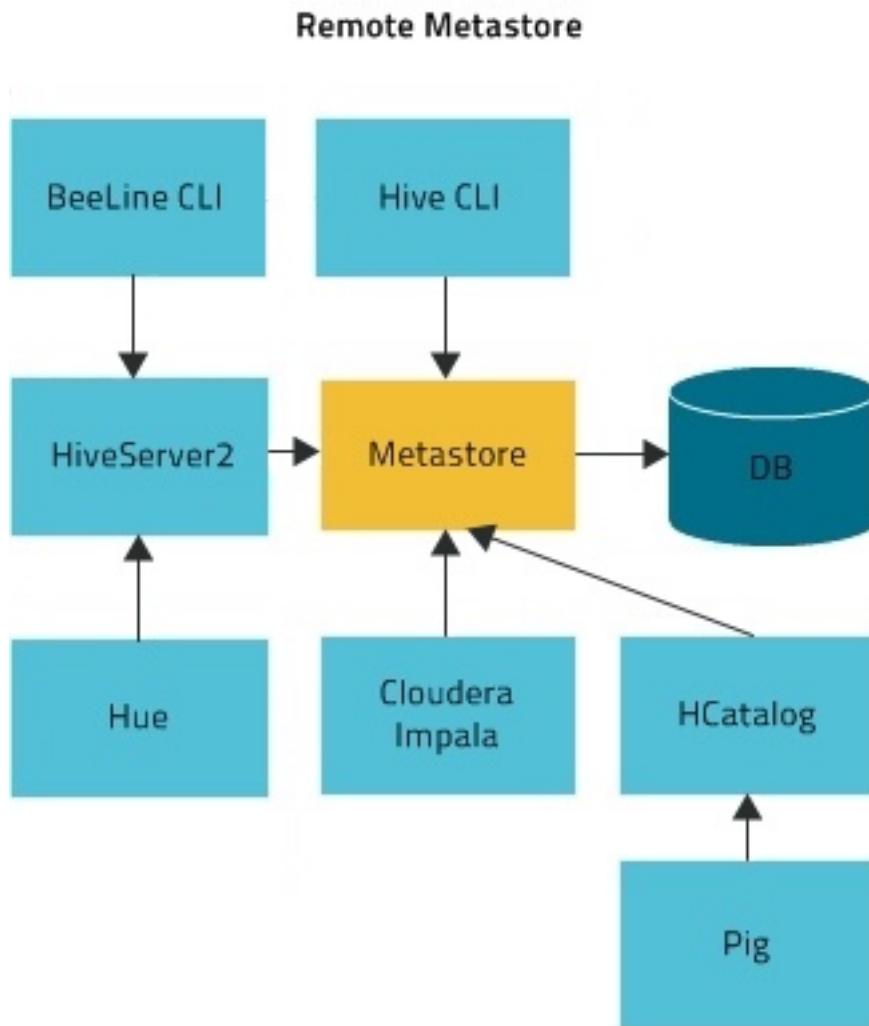
### Local Mode



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*

For up-to-date information, see [Supported Databases](#) on page 32. Cloudera strongly encourages you to use MySQL because it is the most popular with the rest of the Hive user community, and, hence, receives more testing than the other options. For installation information, see:

- [MySQL Database](#) on page 98
- [External PostgreSQL Database](#) on page 86
- [Oracle Database](#) on page 105

## Installation Overview

### 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 and load balancing once you start allocating >12 GB to HiveServer2. The objective is to size to reduce impact of Java garbage collection on active processing by the service.)	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

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 336.

### Configuring the Metastore Database

This section describes how to configure Hive to use a remote database, with examples for [MySQL](#), [PostgreSQL](#), and [Oracle](#).

The configuration properties for the Hive metastore are documented on the [Hive Metastore documentation](#) page, which also includes a pointer to the E/R diagram for the Hive metastore.



**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 RHEL 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 RHEL systems

```
$ sudo service mysqld start
```

## On SLES and Debian/Ubuntu systems

```
$ sudo service mysql start
```

### 2. Configure the MySQL service and connector

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:
```

## Installation Overview

```
| 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 RHEL 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-n.n.n.mysql.sql` file instead; that file is located in the `/usr/lib/hive/scripts/metastore/upgrade/mysql/` directory. (`n.n.n` is the current Hive version, for example 1.1.0.) Proceed as follows if you decide to use `hive-schema-n.n.n.mysql.sql`.

#### Example using `hive-schema-n.n.n.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>
```

## Installation Overview

```
<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 RHEL 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 RHEL 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 RHEL 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 RHEL 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/
```

## Installation Overview



### 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/`. *n.n.n* is the current Hive version, for example 1.1.0:

```
$ 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-n.n.n.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=# \o pg_tables  
metastore=#   WHERE tableowner = CURRENT_USER and schemaname = 'public';  
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 file `ojdbc6.jar` into `/usr/lib/hive/lib/` directory. The driver is available for download [here](#).



**Note:**

These URLs were correct at the time of publication, but the Oracle site is restructured frequently.

```
$ sudo mv ojdbc6.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 (for example `hive-schema-1.1.0.oracle.sql`) in `/usr/lib/hive/scripts/metastore/upgrade/oracle/`:

```
$ sqlplus hiveuser
SQL> @/usr/lib/hive/scripts/metastore/upgrade/oracle/hive-schema-n.n.n.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
 4          ||R.owner||'.'||R.table_name||' to hiveuser';
 5      END LOOP;
 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 and load balancing once you start)	12 GB	12 GB

## Installation Overview

Number of Concurrent Connections	HiveServer2 Heap Size Minimum Recommendation	Hive Metastore Heap Size Minimum Recommendation
allocating >12 GB to HiveServer2. The objective is to size to reduce impact of Java garbage collection on active processing by the service.		
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 336.

### 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>zkl.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

2181 (the default), set `hive.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://&lt;host&gt;:&lt;port&gt;</code>	<code>org.apache.hive.jdbc.HiveDriver</code>
HiveServer1	<code>jdbc:hive://&lt;host&gt;:&lt;port&gt;</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



**Warning:** Because of concurrency and security issues, HiveServer1 and the Hive CLI 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 CLI is not needed if you are using Beeline with HiveServer2.

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>	<i>&lt;Host bindings cannot be specified&gt;</i>

## Installation Overview

### 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:

Cloudera recommends setting permissions on the Hive warehouse directory to 1777, making it accessible to all users, with the sticky bit set. This allows users to create and access their tables, but prevents them from deleting tables they do not own.

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.



#### 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



**Warning:** Because of concurrency and security issues, HiveServer1 and the Hive CLI 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 CLI 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 351.

## Installation Overview

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.verification` configuration property to `false` in `hive-site.xml`.

### Using schematool

The Hive distribution now includes an offline 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 handle upgrading schema from an older version to the current one. The tool will try to find the current schema from the metastore if available. However, this will be applicable only to any future upgrades. In case you are upgrading from existing CDH releases like CDH 4 or CDH 3, you should specify the schema version of the existing metastore as a command line option to the tool.

The `schematool` figures out the SQL scripts 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
```

## Installation Overview

```
Metastore connection User: APP
Starting upgrade metastore schema from version 0.10.0 to <new_version>
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 Driver 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. HiveServer2 has a new JDBC driver that supports both embedded and remote access to HiveServer2. The connection URLs are also different from those in previous versions of Hive.

For more information see the [HiveServer2 Client](#) document.

## Connection URLs

The HiveServer2 connection URL has the following format:

```
jdbc:hive2://<host1>:<port1>,<host2>:<port2>/dbName;sess_var_list?hive_conf_list#hive_var_list
```

where:

- *<host1>:<port1>,<host2>:<port2>* is a server instance or a comma separated list of server instances to connect to (if dynamic service discovery is enabled). If no server is mentioned here, the embedded server will be used.
- *dbName* is the name of the initial database.
- *sess\_var\_list* is a semicolon separated list of key=value pairs of session variables. For example, user=foo;password=bar.
- *hive\_conf\_list* is a semicolon separated list of key=value pairs of Hive configuration variables for this session. For example, hive.server2.transport.mode=http;hive.server2.thrift.http.path=hs2.
- *hive\_var\_list* is a semicolon separated list of key=value pairs of Hive variables for this session.

**Connection URLs for Remote or Embedded Mode:** For remote or embedded access, the JDBC Driver class is `org.apache.hive.jdbc.HiveDriver`.

- For a remote server, the URL format is `jdbc:hive2://<host>:<port>/<db>`. The default HiveServer2 port is 10000).
- For an embedded server, the URL format is `jdbc:hive2://` (no host or port).

## Connection URLs in HTTP Mode:

```
jdbc:hive2://<host>:<port>/<db>?hive.server2.transport.mode=http;hive.server2.thrift.http.path=<http_endpoint>
```

where *<http\_endpoint>* is the corresponding HTTP endpoint configured in `hive-site.xml`. The default value for the endpoint is `cliservice`. The default port for HTTP transport mode is 10001.

## Connection URLs with SSL Enabled:

```
jdbc:hive2://<host>:<port>/<db>;ssl=true;sslTrustStore=<trust_store_path>;trustStorePassword=<trust_store_password>
```

where:

- *<trust\_store\_path>* is the path where client's truststore file is located.
- *<trust\_store\_password>* is the password to access the truststore.

In HTTP mode with SSL enabled, the URL is of the format:

```
jdbc2://<host>:<port>/<db>?ssl=true&sslTrustStore=<trust_store_path>&trustStorePassword=<trust_store_password>&hive.server2.transport.mode=https;hive.server2.thrift.http.path=<https_endpoint>
```

## 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
```

## Installation Overview

- 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 [Configuring Other CDH Components 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



##### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

Use the following sections to install and configure HttpFS:

- [About HttpFS](#)
- [Packaging](#)
- [Prerequisites](#)

- [Installing HttpFS](#)
- [Configuring HttpFS](#)
- [Starting the Server](#)
- [Stopping the Server](#)
- [Using the Server with curl](#)

### 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### To install the HttpFS package on a RHEL-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	/var/lib/hadoop-hftpfs/
Logs	/var/log/hadoop-hftpfs/
temp	/var/tmp/hadoop-hftpfs/
PID file	/var/run/hadoop-hftpfs/

### 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.hftpfs.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hftpfs.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-hftpfs start
```

If you see the message `Server hftpfs started!, status NORMAL` in the `hftpfs.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-hftpfs stop
```

## Installation Overview

### 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

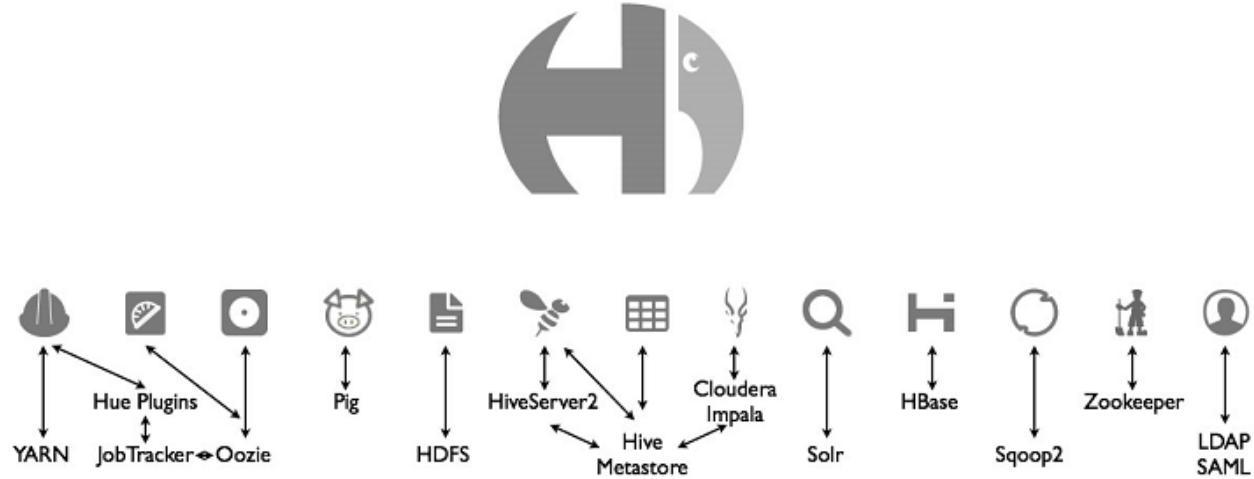
{"Path": "\/user\/babu"}
```

See the [WebHDFS REST API web page](#) for complete documentation of the API.

### Hue Installation

Hue is a web-based tool that lets you interact with CDH components to analyze and manipulate your data. You can also use Hue to build [custom applications](#).

The Hue Server is a "container" web application that sits between your CDH installation and the browser. It hosts a suite of Hue applications and communicates with various servers that interface with CDH components.



The Hue Server uses a dedicated [database](#) to manage session, authentication, and Hue application data. For example, the Job Designer application stores job designs in the Hue database.

The Hue Server runs on a dedicated host, preferably within your CDH cluster, though it can be remote if the firewall is not overly restrictive. For small clusters of less than 10 hosts, the Hue Server can share a host with master daemons such as the NameNode. In a pseudo-distributed installation, the Hue Server runs on the same machine as the rest of your CDH services.

Follow the instructions in the following sections to install, upgrade, and administer Hue.



**Note:** Before installing or upgrading Hue, ensure that CDH is installed and up-to-date. See [Installing the Latest CDH 5 Release](#) on page 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

- [Supported Browsers](#)

- [Installing Hue](#) on page 365
- [Upgrading Hue](#) on page 363
- [Configuring CDH Components for Hue](#) on page 366
- [Hue Configuration](#) on page 370
- [Administering Hue](#) on page 380
- [Hue User Guide](#)

### Supported Browsers for Hue

Hue works with the two most recent versions of the following browsers. Cookies and JavaScript must be on.

- **Chrome**
- **Firefox**
- **Safari** (not supported on Windows)
- **Internet Explorer**

Hue could display in older versions and even other browsers, but you might not have access to all of its features.

### 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 381.

#### 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;
```

## Installation Overview

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 381.

### *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](#) on page 708) 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 381.



### 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 365.



### 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 381.

#### 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 system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6 you will need Python 2.6 to start Hue.

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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

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
```

## Installation Overview

- 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 381.

### 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
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:**

- a. 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>
```

- b. 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.

## Installation Overview

- 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 374 for details on how to configure the Search application for Hue.

## HBase Configuration

See [HBase Configuration](#) on page 374 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.
```

## Installation Overview

```
# 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.

### 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/desktop/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=qpbdxoewsqlkhztybfidtvwekftusgdlofbcfgaswuibcmqp
```



**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 375.

For more information, see the [Hue SDK Documentation](#).

### Configuring the Hue Server for TLS/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 TLS/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 TLS/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 <a href="#">Django</a> .
---------	--	---

## Installation Overview

desktop.auth.backend.AllowAllBackend	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.
desktop.auth.backend.AllowFirstUserDjangoBackend	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.
desktop.auth.backend.LdapBackend	Authenticates users against an LDAP service.
desktop.auth.backend.PamBackend	Authenticates users with PAM (pluggable authentication module). The authentication mode depends on the PAM module used.
desktop.auth.backend.SpnegoDjangoBackend	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.
desktop.auth.backend.RemoteUserDjangoBackend	Authenticating remote users with the Django backend. See the <a href="#">Django documentation</a> for more details.
desktop.auth.backend.OAuthBackend	Delegates authentication to a third-party OAuth server.
libsaml.backend.SAML2Backend	Secure Assertion Markup Language (SAML) single sign-on (SSO) backend. Delegates authentication to the configured Identity Provider. See <a href="#">Configuring Hue for SAML</a> 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:

hive_server_host	The fully-qualified domain name or IP address of the host running HiveServer2.
hive_server_port	The port of the HiveServer2 Thrift server. Default: 10000.
hive_conf_dir	The directory containing <code>hive-site.xml</code> , the HiveServer2 configuration file.

### Impala Query UI Configuration

In the [impala] section of the configuration file, you can optionally specify the following:

server_host	The hostname or IP address of the Impala Server. Default: localhost.
-------------	---

server_port	The port of the Impalad Server. Default: 21050
impersonation_enabled	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 `[librdbsms]`. 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 32.

Database Type	Configuration Properties
SQLite: <code>[ [ [sqlite] ] ]</code>	<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: <code>[ [ [mysql] ] ]</code>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">  <b>Note:</b> 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.
-----------------	---

## Installation Overview

### 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:

share_jobs	Indicate that workflows, coordinators, and bundles should be shared with all users. If set to false, they will be visible only to the owner and administrators.
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 379

### 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)

**HBase Impersonation:** - To enable the HBase app to use impersonation, perform the following steps:

1. Ensure you have a [secure](#) HBase Thrift server.
2. Enable impersonation for the Thrift server by adding the following properties to `hbase-site.xml` on each Thrift gateway:

```
<property>
  <name>hbase.regionserver.thrift.http</name>
  <value>true</value>
</property>
<property>
  <name>hbase.thrift.support.proxyuser</name>
  <value>true</value>
</property>
```

See: [Configure doAs Impersonation for the HBase Thrift Gateway](#).

3. Configure Hue to point to a valid HBase configuration directory. You will find this property under the [hbase] section of the `hue.ini` file.

<code>hbase_conf_dir</code>	HBase configuration directory, where <code>hbase-site.xml</code> is located. Default: <code>/etc/hbase/conf</code>
-----------------------------	---

### User Admin Configuration

In the `[useradmin]` section of the configuration file, you can optionally specify the following:

<code>default_user_group</code>	The name of the group to which a manually created user is automatically assigned. Default: <code>default</code> .
---------------------------------	--

### 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 375.
- 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 376 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>

## Installation Overview

Property	Description	Example
nt_domain	The NT domain to connect to (only for use with Active Directory).	nt_domain=mycompany.com
ldap_url	URL of the LDAP server.	ldap_url=ldap://auth.mycompany.com
ldap_cert	Path to certificate for authentication over TLS (optional).	ldap_cert=/mycertsdir/myTLScert
bind_dn	Distinguished name of the user to bind as – not necessary if the LDAP server supports anonymous searches.	bind_dn="CN=ServiceAccount,DC=mycompany,DC=com"
bind_password	Password of the bind user – not necessary if the LDAP server supports anonymous searches.	bind_password=P@ssw0rd

- Configure the following properties in the [[ [users] ]]] section:

Property	Description	Example
user_filter	Base filter for searching for users.	user_filter="objectclass=*
user_name_attr	The username attribute in the LDAP schema.	user_name_attr=sAMAccountName

- Configure the following properties in the [[ [groups] ]]] section:

Property	Description	Example
group_filter	Base filter for searching for groups.	group_filter="objectclass=*
group_name_attr	The username attribute in the LDAP schema.	group_name_attr=cn



**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.

- In the Hue configuration file, configure the following properties in the [[ ldap ]]] section:

Property	Description	Example
ldap_url	URL of the LDAP server, prefixed by ldap:// or ldaps://	ldap_url=ldap://auth.mycompany.com
search_bind_authentication	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 false. When using search bind semantics, Hue will ignore the following nt_domain and ldap_username_pattern properties.	search_bind_authentication=false
nt_domain	The NT domain over which the user connects (not strictly necessary if using ldap_username_pattern).	nt_domain=mycompany.com
ldap_username_pattern	Pattern for searching for usernames – Use <username> for the username parameter. For use when using LdapBackend for Hue authentication	ldap_username_pattern="uid=<username>,ou=People,dc=mycompany,dc=com"

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
ldap_cert	Path to certificate for authentication over TLS.   <b>Note:</b> If you provide a TLS certificate, it must be signed by a Certificate Authority that is trusted by the LDAP server.	ldap_cert=/mycertsdir/myTLScert

3. In the [[auth]] sub-section inside [desktop] change the following:

backend	Change the setting of backend from  backend=desktop.auth.backend.AllowFirstUserDjangoBackend 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:

[[[default]]]	The section containing the default settings.
fs_defaultfs	The equivalent of fs.defaultFS (also referred to as fs.default.name) in a Hadoop configuration.
webhdfs_url	The HttpFS URL. The default value is the HTTP port on the NameNode.

## Installation Overview

### 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 [[yarn\_clusters]] sub-section:

[[[default]]]	The section containing the default settings.
resourcemanager_host	The fully-qualified domain name of the host running the ResourceManager.
resourcemanager_port	The port for the ResourceManager IPC service.
submit_to	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.
proxy_api_url	URL of the ProxyServer API. Default: http://localhost:8088
history_server_api_url	URL of the HistoryServer API Default: http://localhost:19888

The following MapReduce cluster properties are defined under the [[mapred\_clusters]] sub-section:

[[[default]]]	The section containing the default settings.
jobtracker_host	The fully-qualified domain name of the host running the JobTracker.
jobtracker_port	The port for the JobTracker IPC service.
submit_to	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 ResourceManager. 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:

security_enabled	Indicate whether Oozie requires clients to perform Kerberos authentication.
remote_deployment_dir	The location in HDFS where the workflows and coordinators are deployed when submitted by a non-owner.
oozie_url	The URL of the Oozie server.

### *Sentry Configuration*

In the [libsentry] section of the configuration file, specify the following:

hostname	Hostname or IP of server. Default: localhost
port	The port where the Sentry service is running. Default: 8038
sentry_conf_dir	Sentry configuration directory, where <code>sentry-site.xml</code> is located. Default: /etc/sentry/conf

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*

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
```

## Installation Overview

```
[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

This page explains how to manage and operate a Hue installation:

### *Hue Superusers and Users*

The Hue User Admin application provides two levels of privileges: superusers and users.

- **Superusers** — The first user who logs into Hue after its installation becomes the first superuser. Superusers can perform administrative functions such as:
  - Add and delete users
  - Add and delete groups
  - Assign permissions to groups
  - Change a user to a superuser
  - Import users and groups from an LDAP server
- **Users** — Can change their name, email address, and password. They can login to Hue and run Hue applications according to their group permissions.

### *Hue UI Customization*

The Hue Web UI has the following customization options in `hue.ini` under **[desktop] > [[custom]]**.

#### Banner

You can add a custom banner to the Hue Web UI by applying HTML directly to the property, `banner_top_html`. For example:

```
banner_top_html=<H4>My company's custom Hue Web UI banner</H4>
```

To configure a banner in Cloudera Manager:

1. Go to the **Hue** service.
2. Click the **Configuration** tab.
3. Select **Scope > Hue Server** and **Category > Advanced**.
4. Locate **Top Banner Custom HTML** and input your desired HTML in the text field.

5. Click **Save Changes** to commit the changes.
6. Select **Actions > Restart** and, when done, click **Close**.
7. Click **Web UI** to view your changes.

## Splash Screen

You can customize a splash screen on the login page by applying HTML directly to the property, `login_splash_html`. For example:

```
login_splash_html=WARNING: You are required to have authorization before you proceed.
```

To configure a splash screen in Cloudera Manager:

1. Go to the **Hue** service.
2. Click the **Configuration** tab.
3. Select **Scope > HUE-1 (Service-Wide)** and **Category > Advanced**.
4. Locate **Hue Service Advanced Configuration Snippet (Safety Valve)** for `hue_safety_valve.ini` and add your HTML in the text field as follows:

```
[dektop]
[[custom]]
login_splash_html=WARNING: You are required to have authorization before you proceed.
```

5. Click **Save Changes** to commit the changes.
6. Select **Actions > Restart** and, when done, click **Close**.
7. Click **Web UI** to view your changes.

## Cache Timeout

You enable Hue UI caching by setting a timeout in milliseconds. The default is 86400000 milliseconds or one day. Set the timeout to 0 to disable caching.

As with the splash screen, set the cache timeout in **Hue Service Advanced Configuration Snippet (Safety Valve)** for `hue_safety_valve.ini`.

```
[dektop]
[[custom]]
cacheable_ttl=86400000
```

## *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 a Firewall for Hue*

Hue currently requires that the machines in your cluster 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.

## Installation Overview

### *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 going to Step 3 of Hue's Quick Start Wizard. Under **Anonymous usage analytics**, uncheck the box for **Check to enable usage analytics**.

### *Managing Hue Processes*

The `supervisor` script manages all Hue processes. The supervisor is a watchdog process; its only purpose is to spawn and monitor other processes. To see active supervised processes, run `ps -f -u hue`.

A standard Hue installation starts and monitors the `runcpserver` process, which provides the core web functionality for Hue. If you installed other applications into your Hue instance, you may see other daemons running under the supervisor as well.

Supervisor automatically restarts these processes if they fail for any reason. If they fail repeatedly in a short period of 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 renewals
<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

The Hue Server logs `INFO` level messages and keeps a small buffer of log messages at all levels in memory. The `DEBUG` level messages can sometimes be helpful in troubleshooting issues.

**Tip:** You can turn on `DEBUG` for all Hue logs by setting `django_debug_mode=true` in `hue.ini` under `[ [desktop] ]`.

### *Using an External Database for Hue Using the Command Line*

The Hue server requires a SQL database to store small amounts of data such as user account information, job submissions, and Hive queries. SQLite is the default embedded database. Hue also supports several types of external databases. This page explains how to configure Hue with a selection of external [Supported Databases](#) on page 32.



**Important:** Cloudera strongly recommends an external database for clusters with multiple Hue users.

#### Embedded Database

By default, Hue is configured to use the embedded database, SQLite, and should require no configuration or management by the administrator.

#### Inspecting the Embedded Hue Database

The default SQLite database is located in `/var/lib/hue/desktop.db`. You can inspect this database from the command line with 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:** Avoid modifying the database directly with `sqlite3`. It is best used as a troubleshooting tool.

#### 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

Cloudera strongly recommends an external database for clusters with multiple Hue users, especially clusters in a production environment. The default embedded database, SQLite, cannot support large data migrations. See [Supported Databases](#) on page 32.

High-level steps to configure Hue with any of the supported external databases are:

1. Stop Hue service.
2. Backup default SQLite database.
3. Install database software and dependencies.
4. Create and configure database and load data.
5. Start Hue service.

If you do not need to migrate a SQLite database, you can skip the steps on dumping the database and editing the JSON objects.

Continue Reading:

- [Configuring the Hue Server to Store Data in MariaDB](#) on page 384
- [Configuring the Hue Server to Store Data in MySQL](#) on page 386
- [Configuring the Hue Server to Store Data in PostgreSQL](#) on page 389
- [Configuring the Hue Server to Store Data in Oracle](#) on page 392

#### Prerequisites

## Installation Overview

Before configuring Hue to use an external database:

- Install all support libraries required by your operating system. See [Development Preferences](#) in the Hue documentation for the full list.
- Ensure the Hue server is running on Python 2.6 or higher.

Configuring the Hue Server to Store Data in MariaDB



**Note:** Cloudera recommends InnoDB over MyISAM as the Hue MySQL Engine. On CDH 5, Hue *requires* InnoDB.

1. Shut down the Hue server if it is running.
2. 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"  
        },  
    },  
    ....
```

3. Start the Hue server.
4. Install the MariaDB client developer package.

OS	Command
RHEL	\$ sudo yum install mariadb-devel
SLES	\$ sudo zypper install mariadb-devel
Ubuntu or Debian	\$ sudo apt-get install libmariadbclient-dev

5. Install the MariaDB connector.

OS	Command
RHEL	\$ sudo yum install mariadb-connector-java
SLES	\$ sudo zypper install mariadb-connector-java
Ubuntu or Debian	\$ sudo apt-get install libmariadb-java

6. Install and start MariaDB.

OS	Command
RHEL	\$ sudo yum install mariadb-server

OS	Command
SLES	\$ sudo zypper install mariadb-server \$ sudo zypper install libmariadbclient18
Ubuntu or Debian	\$ sudo apt-get install mariadb-server

**7.** 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
```

**8.** Start the MariaDB daemon.

```
$ sudo service mariadb start
```

**9.** Configure MariaDB 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!
```

**10** Configure MariaDB to start at boot.

OS	Command
RHEL	\$ sudo /sbin/chkconfig mariadb on \$ sudo /sbin/chkconfig --list mariadb mariadb      0:off    1:off    2:on     3:on     4:on     5:on               6:off
SLES	\$ sudo chkconfig --add mariadb
Ubuntu or Debian	\$ sudo chkconfig mariadb on

**11** 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)
```

**12** Open the Hue configuration file in a text editor.

## Installation Overview

- 13** 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
```

- 14** 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.



**Note:** `HUE_HOME` is a reference to the location of your Hue installation. For package installs, this is usually `/usr/lib/hue`; for parcel installs, this is usually, `/opt/cloudera/parcels/<parcel version>/lib/hue/`.

```
$ 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;
```

- 15 (InnoDB only)** Drop the foreign key.

```
mysql > ALTER TABLE auth_permission DROP FOREIGN KEY content_type_id_refs_id_XXXXXX;
```

- 16** Delete the rows in the `django_content_type` table.

```
mysql > DELETE FROM hue.django_content_type;
```

- 17** Load the data.

```
$ <HUE_HOME>/build/env/bin/hue loaddata <some-temporary-file>.json
```

- 18 (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 MySQL



**Note:** Cloudera recommends InnoDB over MyISAM as the Hue MySQL engine. On CDH 5, Hue *requires* InnoDB.

- 1.** Stop the Hue server, if running.

```
sudo service hue stop
```

- 2.** Backup the existing database:

- a.** Dump the existing database data to a `.json` file.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue dumpdata > <some-temporary-file>.json
```



**Note:** HUE\_HOME refers to the location of your Hue installation. For package installs, this is usually /usr/lib/hue; for parcel installs, this is usually /opt/cloudera/parcels/<parcel version>/lib/hue/.

- b.** Delete all JSON objects with useradmin.userprofile in the model field from <some-temporary-file>.json. For example.

```
{
    "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"
    }
},
....
```

- 3.** Install and configure MySQL as an external database for Hue:

- a.** 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

- b.** 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

- c.** Install the MySQL server.

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

## Installation Overview

- d. Configure /etc/my.cnf 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
```

- e. Start the MySQL daemon.

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

- f. Configure MySQL with a strong password. Press the **Enter** key when you're prompted for the `root` password (as your current `root` password is blank).

```
$ 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!
```

- g. 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

- h. Create the database, hue:

```
mysql> create database hue;
```

- i. Grant privileges:

```
mysql> grant all on hue.* to 'hue'@'localhost' identified by '<secretpassword>';
```

- j. Configure /etc/hue/conf/hue.ini to use MySQL. Modify these options as appropriate and paste below `[[database]]` and `[desktop]`:

```
host=localhost
port=3306
```

```
engine=mysql
user=hue
password=<secretpassword>
name=hue
```

#### 4. Load any backed up data:

- a. Ensure a logs directory exists and is writable at /opt/cloudera/parcels/CDH/lib/hue/logs.
- b. Ensure the logs directory has hue user and group ownership.
- c. Synchronize and migrate the database.

```
$ 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;
```



**Note:** HUE\_HOME refers to the location of your Hue installation. For package installs, this is usually /usr/lib/hue; for parcel installs, this is usually /opt/cloudera/parcels/<parcel version>/lib/hue/.

- d. (InnoDB only) Drop the foreign key:

```
mysql > ALTER TABLE auth_permission DROP FOREIGN KEY content_type_id_refs_id_XXXXXX;
```

- e. Delete the rows in the django\_content\_type table:

```
mysql > DELETE FROM hue.django_content_type;
```

- f. Load the data:

```
$ <HUE_HOME>/build/env/bin/hue loaddata <some-temporary-file>.json
```

- g. (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`);
```

#### 5. Start the Hue server.

```
sudo service hue start
```

### Configuring the Hue Server to Store Data in PostgreSQL



**Warning:** Hue requires PostgreSQL 8.4 or higher.

1. Stop the Hue server, if running.

```
sudo service hue stop
```

2. Backup the existing database:

## Installation Overview

- a. Dump the existing database data to a .json file.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue dumpdata > <some-temporary-file>.json
```



**Note:** HUE\_HOME refers to the location of your Hue installation. For package installs, this is usually /usr/lib/hue; for parcel installs, this is usually, /opt/cloudera/parcels/<parcel version>/lib/hue/.

- b. Delete all JSON objects with useradmin.userprofile in the model field from <some-temporary-file>.json. For example.

```
{  
    "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"  
        },  
        ....  
    }  
}
```

3. Install the PostgreSQL client dev packages.

OS	Commandcd /
RHEL	\$ sudo yum install postgresql-devel gcc python-devel
SLES	\$ sudo zypper install postgresql-devel gcc python-devel
Ubuntu or Debian	\$ sudo apt-get install postgresql-devel gcc python-devel

4. Install the Python modules that provide the PostgreSQL connector.

```
sudo -u hue <HUE_HOME>/build/env/bin/pip install setuptools  
sudo -u hue <HUE_HOME>/build/env/bin/pip install psycopg2
```

5. Install the PostgreSQL server.

OS	Command
RHEL	\$ sudo yum install postgresql-server
SLES	\$ sudo zypper install postgresql-server
Ubuntu or Debian	\$ sudo apt-get install postgresql

6. Initialize the data directories:

```
$ service postgresql initdb
```

7. Configure /var/lib/pgsql/data/pg\_hba.conf for client authentication.

- a. Set the authentication method for local to trust.
- b. Set the authentication method for host to password.
- c. Append the following line to the end of the file.

```
host hue hue 0.0.0.0/0 md5
```

**8.** Start the PostgreSQL server.

```
$ su - postgres
# /usr/bin/postgres -D /var/lib/pgsql/data > logfile 2>&1 &
```

**9.** 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
```

**10** 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
```

**11** Restart the PostgreSQL server.

```
$ sudo service postgresql restart
```

**12** Verify connectivity.

```
psql -h localhost -U hue -d hue
Password for user hue: <secretpassword>
```

**13** 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

**14** Configure `/etc/hue/conf/hue.ini` to use PostgreSQL. Modify these options as appropriate and paste below `[[database]]` and `[desktop]`:

```
host=localhost
port=5432
engine=postgresql_psycopg2
user=hue
password=<secretpassword>
name=hue
```

**15** Load any backed up data:

- a. Ensure a logs directory exists and is writable at `/opt/cloudera/parcels/CDH/lib/hue/logs`.

## Installation Overview

- b. Ensure the logs directory has hue user and group ownership.
- c. Synchronize and migrate the database.

```
$ 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
```

- d. Determine the foreign key ID.

```
bash# su - postgres  
$ psql -h localhost -U hue -d hue  
postgres=# \d auth_permission;
```

- e. Drop the foreign key that you retrieved in the previous step.

```
postgres=# ALTER TABLE auth_permission DROP CONSTRAINT content_type_id_refs_id_<XXXXXX>;
```

- f. Delete the rows in the django\_content\_type table.

```
postgres=# TRUNCATE django_content_type CASCADE;
```

- g. Load the data.

```
$ sudo -u hue <HUE_HOME>/build/env/bin/hue loaddata <some-temporary-file>.json
```

- h. Add the foreign key:

```
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



**Important:** Configure the database for character set AL32UTF8 and national character set UTF8.

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.



**Note:** The Oracle 12 instant client is currently not supported. Hue works with the Oracle 12 database, but only with the Oracle 11 client libraries.

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:



**Note:** HUE\_HOME is a reference to the location of your Hue installation. For package installs, this is usually /usr/lib/hue; for parcel installs, this is usually, /opt/cloudera/parcels/<parcel version>/lib/hue/.

```
$ <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 CREATE <sequence> TO <user>;
GRANT CREATE <session> TO <user>;
GRANT CREATE <table> TO <user>;
GRANT CREATE <view> TO <user>;
GRANT CREATE <procedure> TO <user>;
GRANT CREATE <trigger> TO <user>;
GRANT EXECUTE ON sys.dbms_crypto TO <user>;
GRANT EXECUTE ON SYS.DBMS_LOB TO <user>;
```

**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 you are connected to Oracle as the hue user, then run the following command to delete all data from Oracle tables:

```
SELECT 'DELETE FROM ' || ' . ' || table_name || ';' FROM user_tables;
```

**11** Run the statements generated in the preceding step.

## Installation Overview

12 Commit your changes.

```
commit;
```

13 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 and Upgrade

Hadoop Key Management Server (KMS) is a cryptographic key management server based on the Hadoop **KeyProvider** API. It provides a KeyProvider implementation client that interacts with the KMS using the HTTP REST API. Both the KMS and its client support HTTP SPNEGO Kerberos authentication and TLS/SSL-secured communication. The KMS is a Java-based web application that runs using a preconfigured Tomcat server bundled with the Hadoop distribution.

Cloudera provides two implementations of the Hadoop KMS:

- **Java KeyStore KMS** - The default Hadoop KMS included in CDH that uses a file-based Java KeyStore (JKS) for its backing keystore. For parcel-based installations, no additional action is required to install or upgrade the KMS. For package-based installations, you must install additional packages. For more information, see [Installing and Upgrading Java KeyStore KMS](#) on page 394. Cloudera strongly recommends not using Java Keystore KMS in production environments.
- **Key Trustee KMS** - A custom KMS that uses [Cloudera Navigator Key Trustee Server](#) for its backing keystore instead of the file-based Java KeyStore (JKS) used by the default Hadoop KMS. Cloudera strongly recommends using Key Trustee KMS in production environments to improve the security, durability, and scalability of your cryptographic key management. For more information about the architecture and components involved in encrypting data at rest for production environments, see [Cloudera Navigator Data Encryption Overview](#) and [Data at Rest Encryption Reference Architecture](#). For instructions on installing and upgrading Key Trustee KMS, see:
  - [Installing Key Trustee KMS](#) on page 211
  - [Upgrading Key Trustee KMS](#) on page 520

### Installing and Upgrading Java KeyStore 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### To install or upgrade Java KeyStore KMS on a RHEL-compatible system:

```
$ sudo yum install hadoop-kms hadoop-kms-server
```

#### To install or upgrade Java KeyStore KMS on a SLES system:

```
$ sudo zypper install hadoop-kms hadoop-kms-server
```

#### To install or upgrade Java KeyStore 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
```

## Installation Overview

### Mahout Installation



**Important:** This item is deprecated and will be removed in a future release. Cloudera supports items that are deprecated until they are removed. For more information about deprecated and removed items, see [Deprecated Items](#).

[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 220.

### 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 CDH 4 to CDH 5*

To upgrade Mahout to CDH 5, you must uninstall the CDH 4 version and then install the CDH 5 version. Proceed as follows.

##### Step 1: Remove CDH 4 Mahout

###### To remove Mahout on a Red Hat system:

```
$ sudo yum remove mahout
```

###### To remove Mahout on a SLES system:

```
$ sudo zypper remove mahout
```

###### To remove Mahout on an Ubuntu or Debian system:

```
$ sudo apt-get remove mahout
```

## Step 2: Install CDH 5 Mahout

See [Installing Mahout](#).



### 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](#).

### *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 397.



### 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

## Installation Overview

### 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](http://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.



#### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is 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 CDH 4 to CDH 5*

To upgrade Oozie from CDH 4 to CDH 5, back up the configuration files and database, uninstall the CDH 4 version and then install and configure the CDH 5 version. Proceed as follows.



#### Note:

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#), you can skip Step 1 below and proceed with installing the new CDH 5 version of Oozie.



#### Important: Ubuntu and Debian upgrades

When you uninstall CDH 4 Oozie on Ubuntu and Debian systems, the contents of `/var/lib/oozie` are removed, leaving a bare directory. This can cause the Oozie upgrade to CDH 5 to fail. To prevent this, either copy the database files to another location and restore them after the uninstall, or recreate them after the uninstall. Make sure you do this before starting the re-install.

### Step 1: Remove Oozie

1. 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.
2. Stop the Oozie Server.

#### To stop the Oozie Server:

```
sudo service oozie stop
```

3. Uninstall Oozie.

#### To uninstall Oozie, run the appropriate command on each host:

## Installation Overview

- On RHEL-compatible systems:

```
$ sudo yum remove oozie
```

```
$ sudo yum remove oozie-client
```

- On SLES systems:

```
$ sudo zypper remove oozie
```

```
$ sudo zypper remove oozie-client
```

- On Ubuntu or Debian systems:

```
sudo apt-get remove oozie
```

```
sudo apt-get remove oozie-client
```

### Step 2: Install Oozie

Follow the procedure under [Installing Oozie](#) and then proceed to [Configuring Oozie after Upgrading from CDH 4](#) on page 403. For packaging information, see [Oozie Packaging](#).

#### 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](#).

#### *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](#) on page 708.

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 401 and then proceed to [Configuring Oozie after Upgrading from an Earlier CDH 5 Release](#) on page 406.



#### 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 server package (`oozie`) and a client package (`oozie-client`). Choose the appropriate packages and install them with 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 which Hadoop Version to Use](#) on page 402.



#### 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### 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 [Configuring Oozie](#) on page 402.

### Configuring Oozie

This page explains how to configure Oozie, for new installs and upgrades, in an unmanaged deployment, *without* Cloudera Manager.

**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

#### Configuring which Hadoop Version to Use

The Oozie server works with either MRv1 or YARN, but not both simultaneously. The Oozie client does not interact directly with Hadoop MapReduce and does not require any MapReduce configuration.

To configure the Oozie server to work with YARN or MRv1, and with or without [TLS/SSL](#), use the `alternatives` command (or `update-alternatives`, depending on your operating system).



**Important: Stop the Oozie server before upgrading from MRv1 to YARN or workflows that depend on MRv1 may cause the MRv1 jobs to fail.**

- To use YARN (without TLS/SSL):

```
alternatives --set oozie-tomcat-conf /etc/oozie/tomcat-conf.http
```

- To use YARN (with TLS/SSL):

```
alternatives --set oozie-tomcat-conf /etc/oozie/tomcat-conf.https
```

- To use MRv1 (without TLS/SSL) :

```
alternatives --set oozie-tomcat-conf /etc/oozie/tomcat-conf.http.mrv1
```

- To use MRv1 (with TLS/SSL) :

```
alternatives --set oozie-tomcat-conf /etc/oozie/tomcat-conf.https.mrv1
```



**Important: In CDH 5 Beta 2 and higher, ensure that CATALINA\_BASE in /etc/oozie/conf/oozie-env.sh is set to:**

```
export CATALINA_BASE=/var/lib/oozie/tomcat-deployment
```

## Configuring Oozie after Upgrading from CDH 4



**Note:** If you are installing Oozie for the first time, skip this section and proceed with [Configuring Oozie after a New Installation](#).

### 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 CDH 4 `oozie-site.xml`:



**Important:** *Do not copy over the CDH 4 configuration files into the CDH 5 configuration directory.*

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 that is required when you upgrade Oozie from CDH 4 to CDH 5. 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
Update db.version in OOZIE_SYS table to 2
DONE
Post-upgrade COORD_JOBS new columns default values
DONE
Post-upgrade COORD_JOBS & COORD_ACTIONS status values
```

## Installation Overview

```
DONE
Post-upgrade MISSING_DEPENDENCIES column in Derby
DONE
Table 'WF_ACTIONS' column 'execution_path', length changed to 1024
Table 'WF_ACTIONS', column 'error_message', changed to varchar/varchar2
Table 'COORD_JOB' column 'frequency' changed to varchar/varchar2
DONE
Post-upgrade BUNDLE_JOBS, COORD_JOBS, WF_JOBS to drop AUTH_TOKEN column
DONE
Upgrading to db schema for Oozie 4.0.0-cdh5.0.0
Update db.version in OOZIE_SYS table to 3
DONE
Dropping discriminator column
DONE

Oozie DB has been upgraded to Oozie version '4.0.0-cdh5.0.0'

The SQL commands have been written to: /tmp/ooziedb-3809837539907706.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/oozie/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
Update db.version in OOZIE_SYS table to 2
DONE
Post-upgrade COORD_JOBS new columns default values
DONE
Post-upgrade COORD_JOBS & COORD_ACTIONS status values
DONE
Post-upgrade MISSING_DEPENDENCIES column in Derby
DONE
Table 'WF_ACTIONS' column 'execution_path', length changed to 1024
Table 'WF_ACTIONS', column 'error_message', changed to varchar/varchar2
Table 'COORD_JOB' column 'frequency' changed to varchar/varchar2
DONE
Post-upgrade BUNDLE_JOBS, COORD_JOBS, WF_JOBS to drop AUTH_TOKEN column
DONE
Upgrading to db schema for Oozie 4.0.0-cdh5.0.0
Update db.version in OOZIE_SYS table to 3
DONE
Dropping discriminator column
```

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; CDH 5 Oozie does not work with CDH 4 shared libraries.

CDH 5 Oozie has a new shared library that bundles CDH 5 JAR files for streaming, DistCp and for Pig, Hive, HiveServer 2, Sqoop, and HCatalog.



**Note:** 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`.
- The shared library file for MRv1 is `oozie-sharelib-mr1`.

Proceed as follows to upgrade the shared library.

1. Install the Oozie CDH 5 shared libraries. For example:

```
$ sudo oozie-setup sharelib create -fs FS_URI -locallib /usr/lib/oozie/oozie-sharelib-yarn
```

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` 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](#).

### 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 408.

#### 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/oozie/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 OZIE_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.
- The shared library file for MRv1 is oozie-sharelib-mr1.

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
```

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` 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 401.

### Configuring Oozie after a New Installation



**Note:** If you are upgrading Oozie from CDH 4 or from an earlier CDH 5 release, select: [Configuring Oozie after Upgrading from CDH 4](#) on page 403 or [Configuring Oozie after Upgrading from an Earlier CDH 5 Release](#) on page 406.

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	/usr/lib/oozie/
configuration	/etc/oozie/conf/
documentation	for SLES: /usr/share/doc/packages/oozie/ for other platforms: /usr/share/doc/oozie/
examples TAR.GZ	for SLES: /usr/share/doc/packages/oozie/ for other platforms: /usr/share/doc/oozie/

Type of File	Where Installed
sharelib TAR.GZ	/usr/lib/oozie/
data	/var/lib/oozie/
logs	/var/log/oozie/
temp	/var/tmp/oozie/
PID file	/var/run/oozie/

## Deciding Which Database to Use

Oozie has a built-in Derby database, but Cloudera recommends that you use a [PostgreSQL](#), [MariaDB](#), [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 that do not happen with server-based databases.

See [Supported Databases](#) on page 32 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.UTF-8'
  LC_CTYPE = 'en_US.UTF-8'
  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`.

## Installation Overview

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>
...</pre>
```



**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 MariaDB

Use the procedure that follows to configure Oozie to use MariaDB instead of Apache Derby.

### Install and Start MariaDB

For more information, see [Installing the MariaDB Server](#) on page 92.

### Create the Oozie Database and Oozie MariaDB User

For example, using the MariaDB `mysql` command-line tool:

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

mysql> create database oozie;
Query OK, 1 row affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> exit
Bye
```

## Configure Oozie to Use MariaDB

Edit properties in the `oozie-site.xml` file as follows:

```
...
<property>
    <name>oozie.service.JPAService.jdbc.driver</name>
    <value>org.mysql.jdbc.Driver</value>
</property>
<property>
    <name>oozie.service.JPAService.jdbc.url</name>
    <value>jdbc:mysql://localhost:3306/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 MariaDB is running.

## Add the MariaDB JDBC Driver JAR to Oozie

Cloudera recommends that you use the MySQL JDBC driver for MariaDB. Copy or symbolically link the MySQL JDBC driver JAR to the `/var/lib/oozie/` directory.



**Note:** You must manually download the MySQL JDBC driver JAR file.

## 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;
Query OK, 1 row affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.03 sec)

mysql> exit
Bye
```

## Installation Overview

### Configure Oozie to Use MySQL

Edit properties in the `oozie-site.xml` file as follows:

```
...<property><name>oozie.service.JPAService.jdbc.driver</name><value>com.mysql.jdbc.Driver</value></property><property><name>oozie.service.JPAService.jdbc.url</name><value>jdbc:mysql://localhost:3306/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 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 sure 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.

### 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`.

### Step 3: Configure SPNEGO authentication (in Kerberos clusters only)

The web console shares a port with the Oozie REST API, and the API allows modifications of Oozie jobs (kill, submission, and inspection). SPNEGO authentication ensures that the Kerberos realm trusts the client browser credentials and that configuration of the client web browser passes these credentials. If this configuration is not possible, use the Hue Oozie Dashboard instead of the Oozie Web Console.

See [Using a Web Browser to Access an URL Protected by Kerberos HTTP SPNEGO](#) and [Configuring a Cluster-dedicated MIT KDC with Cross-Realm Trust](#).

## 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`.
- The shared library file for YARN is `oozie-sharelib-yarn`.



**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
```

## Installation Overview

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` 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).

### Configuring Oozie Failover (hot/cold)



**Note:**

The functionality described below is supported in CDH 5, but Cloudera recommends that you use the [new capabilities](#) introduced in CDH 5 instead.

1. Set up your database for High Availability (see the database documentation for details).



**Note:**

Oozie database configuration properties may need special configuration (see the JDBC driver documentation for details).

2. Configure Oozie on two or more servers:
3. These servers should be configured identically
4. Set the `OOZIE_HTTP_HOSTNAME` variable in `oozie-env.sh` to the Load Balancer or Virtual IP address (see step 3)
5. Only one of the Oozie servers should be started (the *hot* server).

## Installation Overview

6. Use either a Virtual IP Address or Load Balancer to direct traffic to the hot server.
7. Access Oozie using the Virtual IP or Load Balancer address.

### Points to note

- The Virtual IP Address or Load Balancer can be used to periodically check the health of the hot server.
- If something is wrong, you can shut down the hot server, start the cold server, and redirect the Virtual IP Address or Load Balancer to the new hot server.
- This can all be automated with a script, but a false positive indicating the hot server is down will cause problems, so test your script carefully.
- There will be no data loss.
- Any running workflows will continue from where they left off.
- It takes only about 15 seconds to start the Oozie server.

See also [Configuring Oozie to Use HDFS HA](#).

### 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 CDH 4 to CDH 5

To upgrade Pig to CDH 5:



#### Note:

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#), you can skip Step 1 below and proceed with installing the new CDH 5 version of Pig.

### Step 1: Remove Pig

1. Exit the Grunt shell and make sure no Pig scripts are running.
2. Remove the CDH 4 version of Pig

#### To remove Pig On RHEL-compatible systems:

```
$ sudo yum remove hadoop-pig
```

**To remove Pig on SLES systems:**

```
$ sudo zypper remove pig
```

**To remove Pig on Ubuntu and other Debian systems:**

```
$ sudo apt-get remove pig
```

**Step 2: Install the new version**

Follow the instructions in the next section, [Installing Pig](#).

**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](#).

***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](#) on page 708.

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 419.

**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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

## Installation Overview

### 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. They were developed by LinkedIn and are now open source under an Apache 2.0 license.

A number of usage examples and other information are available at <https://github.com/linkedin/datafu>.

### To Use DataFu in a Parcel-deployed Cluster

If your cluster uses parcels, DataFu is installed for you. You need to register the JAR file prior to use with the following command.

```
REGISTER /opt/cloudera/parcels/CDH/lib/pig/datafu.jar
```

### To Use DataFu in a Package-deployed Cluster:

1. Install the DataFu package:

## Installation Overview

Operating system	Install command
Red-Hat-compatible	<pre>sudo yum install pig-udf-datafu</pre>
SLES	<pre>sudo zypper install pig-udf-datafu</pre>
Debian or Ubuntu	<pre>sudo apt-get install pig-udf-datafu</pre>

This puts the DataFu JAR file (for example, datafu-0.0.4-cdh5.0.0.jar) 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
```

### 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.

### 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:

- 16 GB is sufficient for some smaller loads or for evaluation.
- 32 GB is sufficient for some production environments.
- 96 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 facetting vary depending on the number of unique terms in the faceted field. Suppose you want to use facetting 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 facetting 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:

## Installation Overview

- 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.
- 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:
    - 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 facetting 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 reshuffle, 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](#). This technique is recommended for reliable and verifiable Search installation.
- Using the manual process described in [Installing Cloudera Search without Cloudera Manager](#). 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 220.



**Note:** This page describes how to install CDH using packages as well as how to install CDH using Cloudera Manager.

You can also install Cloudera Search manually in some situations; for example, if you have an existing installation to which you want to add Search.

- For general information about using repositories to install or upgrade Cloudera software, see [Understanding Custom Installation Solutions](#) on page 170.
- For instructions on installing or upgrading CDH, see [Installing the Latest CDH 5 Release](#) on page 220 and [Upgrading from CDH 4 to CDH 5](#) on page 691.
- For CDH 5 repository locations and client .repo files, which include Cloudera Search, see [Version and Download Information](#).

Cloudera Search includes the following packages:

Package Name	Description
<code>solr</code>	Solr
<code>solr-server</code>	Platform specific service script for starting, stopping, or restart Solr.
<code>solr-doc</code>	Cloudera Search documentation.
<code>solr-mapreduce</code>	Tools to index documents using MapReduce.
<code>solr-crunch</code>	Tools to index documents using Crunch.
<code>search</code>	Examples, Contrib, and Utility code and data.

### Before You Begin

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 instead of 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 causing unexpected results. If you install CDH from packages, service is installed as part of the Linux Standard Base (LSB).
- **Install the Cloudera repository:** Before using the instructions in this guide to install or upgrade Cloudera Search from packages, install the Cloudera 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 431.

#### 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 426.

#### 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
```

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](#).
- Without Cloudera Manager as described in [Installing Cloudera Search without Cloudera Manager](#).

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](#) on page 422.

## 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](#).

### 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/CDH-*/*/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=<zkhost1>:2181,<zkhost2>:2181,<zkhost3>: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/CDH-*/*/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 used 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/CDH-*/*/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`.

## Installation Overview

### Configuring Solr to Use Secure HDFS

- For information on setting up a secure CDH cluster for CDH 4, see the [CDH 4 Security Guide](#).
- For information on setting up a secure CDH cluster for CDH 5, 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/CDH-*/*/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 browse to

`http://myhost.example.com:8983/solr/collection1/select?q=*&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`.

For more information on completing additional collection management tasks, see [Managing Solr Using solrctl](#).

## Installation Overview

### *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, `solr-crunch` must be installed on hosts where you want to submit a batch indexing job.

By default, this tool is installed when Cloudera Search is installed using parcels, such as in a Cloudera Manager deployment. If you are using a package installation and this tool does not exist on your system, you can install this tool using the commands described in this topic.

#### **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 Reference \(CDH 5.2 and higher only\)](#).

### *Installing MapReduce Tools for use with Cloudera Search*

Cloudera Search provides the ability to batch index documents using MapReduce jobs. To use the MapReduce tools, `solr-mapreduce` must be installed on hosts where you want to submit a batch indexing job.

By default, this tool is installed when Cloudera Search is installed using parcels, such as in a Cloudera Manager deployment. If you are using a package installation and this tool does not exist on your system, you can install this tool using the commands described in this topic.

#### **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.

By default, this tool is installed when Cloudera Search is installed using parcels, such as in a Cloudera Manager deployment. If you are using a package installation and this tool does not exist on your system, you can install this tool using the commands described in this topic.

**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 1.x, which works with CDH 4, there is a separate parcel for Search. 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 [Upgrading to CDH 5](#).

**Upgrading Manually without Cloudera Manager**

The update process is different for Search 1.x and Search for CDH 5. With Search 1.x, Search is a separate package from CDH. Therefore, to upgrade from Search 1.x, you must upgrade to CDH 5, which includes Search as part of the CDH 5 repository.



**Important:** Before upgrading, make backup copies of the following configuration files:

- /etc/default/solr or /opt/cloudera/parcels/CDH-\*/etc/default/solr
- All collection configurations

Make sure you copy every host that is part of the SolrCloud.

- If you are running CDH 4 and want to upgrade to Search for CDH 5, see [Upgrading Search 1.x to Search for CDH 5](#) on page 431.
- 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](#) on page 708.

***Upgrading Search 1.x to Search for CDH 5***

If you are running Cloudera Manager, you must upgrade to Cloudera Manager 5 to run CDH 5. Because Search 1.x is in a separate repository from CDH 4, you must remove the Search 1.x packages and the Search .repo or .list file before upgrading CDH. This is true whether or not you are upgrading through Cloudera Manager.

1. Remove the Search packages.

The list of packages you may need to remove are:

- solr
- solr-doc

## Installation Overview

- solr-mapreduce
- hbase-solr
- hbase-solr-doc
- search

- 1.** 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
```

- 2.** Remove the packages using the appropriate remove command for your OS. For example:

```
sudo yum remove solr solr-doc solr-mapreduce hbase-solr \
hbase-solr-doc search      # RHEL, Oracle Linux, CentOS
```

- 2.** Remove the Cloudera Search .repo or .list file:

Operating System	File to remove:
RHEL	/etc/yum.repos.d/cloudera-search.repo
SLES	/etc/zypp/repos.d/cloudera-search.repo
Ubuntu or Debian	/etc/apt/sources.list.d/cloudera.list

- 3.** Upgrade from CDH 4 to CDH 5:

- To upgrade using Cloudera Manager, see [Upgrading CDH 4 to CDH 5](#). This assumes you have upgraded to Cloudera Manager 5.
- To upgrade without using Cloudera Manager, see [Upgrading from CDH 4 to CDH 5](#).

- 4.** If you upgraded to CDH 5 without using Cloudera Manager, you need to install the new version of Search:

Operating System	Command
RHEL	sudo yum install solr-server
SLES	sudo zypper install solr-server
Ubuntu or Debian	sudo apt-get install solr-server

### Installing Hue Search

You must install and configure Hue before you can use Search with Hue.

1. Follow the instructions for [Installing Hue](#).
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
<p><b>1.</b> Connect to Cloudera Manager.</p> <p><b>2.</b> Select the <b>Hue</b> service.</p> <p><b>3.</b> Click the <b>Configuration</b> tab.</p> <p><b>4.</b> Search for the word "safety".</p> <p><b>5.</b> Add information about your Solr host to <b>Hue Server Advanced Configuration Snippet (Safety Valve) for hue_safety_valve_server.ini</b>. For example, if your hostname is SOLR_HOST, you might add the following:</p> <pre>[search] # URL of the Solr Server solr_url=http://SOLR_HOST:8983/solr</pre> <p><b>6.</b> (Optional) To enable Hue in environments where Kerberos authentication is required, update the <b>security_enabled</b> property as follows:</p> <pre># Requires FQDN in solr_url if enabled security_enabled=true</pre>	<p>Update configuration information in /etc/hue/hue.ini.</p> <p><b>1.</b> Specify the Solr URL. For example, to use localhost as your Solr host, you would add the following:</p> <pre>[search] # URL of the Solr Server, replace 'localhost' if Solr is running on another host solr_url=http://localhost:8983/solr/</pre> <p><b>2.</b> (Optional) To enable Hue in environments where Kerberos authentication is required, update the <b>security_enabled</b> property as follows:</p> <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 and higher, secure impersonation for Hue is automatically configured. To review secure impersonation settings in the Cloudera Manager home page:
  - Go to the HDFS service.
  - Click the **Configuration** tab.
  - Select **Scope > All**.
  - Select **Category > All**.
  - Type `hue proxy` in the Search box.
  - 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` or `/opt/cloudera/parcels/CDH-*/etc/default/solr`. The changes you make may vary according to

## Installation Overview

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](#).

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

OS	Command
Ubuntu or Debian	\$ sudo apt-get update; \$ sudo apt-get install sentry

## Upgrading Sentry

### Upgrading from CDH 4 to CDH 5

If you are upgrading Sentry from CDH 4 to CDH 5, you must uninstall the old version and install the new version.



**Note:** If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#) on page 691, you can skip this step and proceed with [installing the latest version of Sentry](#).

1. Remove the CDH 4 Version of Sentry:

OS	Command
RHEL	\$ sudo yum remove sentry
SLES	\$ sudo zypper remove sentry
Ubuntu or Debian	\$ sudo apt-get remove sentry

2. [Install the new version of 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. Upgrade Sentry Database Schema Using `schematool`

- **From a release earlier than CDH 5.2 to CDH 5.4:**

Use the Sentry `schematool` to upgrade the database schema as follows:

```
bin/sentry --command schema-tool --confFile <sentry-site.xml> --dbType <db-type>
--upgradeSchema
```

Where `<db-type>` should be either `mysql`, `postgres` or `oracle`.

- **For CDH 5.5 and higher:** The newer releases include password encryption which means you can no longer run `schematool` as it requires a plaintext password. Password encryption is an important part of security

## Installation Overview

and Sentry defaults to using the CredentialProvider API to accomplish this. Cloudera recommends you use Cloudera Manager to upgrade the Sentry database instead.

- However, if using Cloudera Manager is not an option, and `schematool` is required, to work around the default encryption, obtain the password in plaintext from the API, open `sentry-site.xml` and manually set the `sentry.store.jdbc.password` property to use the plaintext password, and remove the `hadoop.security.credential.provider.path` property and its value. You should now be able to run `schematool`.

### 5. Start the Sentry Service

- a. Set the `SENTRY_HOME` and `HADOOP_HOME` parameters.
- b. Run the following command:

```
bin/sentry --command service --conffile <sentry-site.xml>
```

## Snappy Installation

[Snappy](#) is a compression/decompression library. It optimizes for very high-speed compression and decompression, and moderate compression instead of maximum compression or compatibility with other compression libraries.

### 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.

### Upgrading Snappy

To upgrade Snappy, simply install the `hadoop` package if you haven't already done so. This applies whether you are upgrading from CDH 4 or 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](#).

## Spark Installation

Spark is a fast, general engine for large-scale data processing.

See also the [Apache Spark Documentation](#).

### Spark Packages

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 220)

## 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

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 and Changes in CDH 5](#)
- [Apache Spark Incompatible Changes and Limitations](#)
- [Apache Spark Known Issues](#)
- **RHEL-compatible system:**
  - To install all Spark packages:

```
$ sudo yum install spark-core spark-master spark-worker spark-history-server spark-python
```

- To install only the packages needed to run Spark on YARN:

```
$ sudo yum install spark-core spark-history-server spark-python
```

- **SLES system:**

- To install all Spark packages:

```
$ sudo zypper install spark-core spark-master spark-worker spark-history-server  
spark-python
```

- To install only the packages needed to run Spark on YARN:

```
$ sudo zypper install spark-core spark-history-server spark-python
```

- **Ubuntu or Debian system:**

- To install all Spark packages:

```
$ sudo apt-get install spark-core spark-master spark-worker spark-history-server  
spark-python
```

- To install only the packages needed to run Spark on YARN:

```
$ sudo apt-get install spark-core spark-history-server spark-python
```

You are now ready to configure and start Spark. See [Managing Spark Standalone Using the Command Line](#).

**Note:**

If you uploaded the Spark JAR file as described under [Optimizing YARN Mode in Unmanaged CDH Deployments](#), use the same instructions to upload the new version of the file each time you upgrade to a new minor release of CDH (for example, any CDH 5.4.x release, including 5.4.0).

### 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 439
- [Packaging](#)
- [Prerequisites](#)
- [Installing Packages](#)
- [Installing a Tarball](#)
- [Installing the JDBC Drivers](#)
- [Setting HADOOP\\_MAPRED\\_HOME](#) on page 442
- [Apache Sqoop 1 Documentation](#)

Also see [Feature Differences - Sqoop 1 and Sqoop 2](#) on page 449 for major feature differences between Sqoop 1 and Sqoop 2.

#### Upgrading Sqoop 1 from CDH 4 to CDH 5

To upgrade Sqoop 1 from CDH 4 to CDH 5, proceed as follows.

**Note:**

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#) on page 691, you can skip Step 1 below and proceed with installing the new CDH 5 version of Sqoop 1.

#### Step 1: Remove the CDH 4 version of Sqoop 1

**To remove Sqoop 1 on a Red Hat-compatible system:**

```
$ sudo yum remove sqoop
```

**To remove Sqoop 1 on an Ubuntu or other Debian system:**

```
$ sudo apt-get remove sqoop
```

**To remove Sqoop 1 on a SLES system:**

```
$ sudo zypper remove sqoop
```

## Step 2: Install the new version of Sqoop 1

Follow instructions under [Installing the Sqoop 1 RPM Packages](#) or [Installing the Sqoop 1 Tarball](#).



### 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.

### 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](#) on page 708.

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 440 or [Installing the Sqoop 1 Tarball](#) on page 440



### 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

Sqoop 1 requires the following:

- An [operating system supported by CDH 5](#).
- [Oracle JDK](#).
- Services that you want to use with Sqoop, such as HBase, Hive HCatalog, and Accumulo. When you run Sqoop, it checks to see if these services are installed and configured. It logs warnings for services it does not find. These warnings, shown below, are harmless. You can suppress these error messages by setting the variables \$HBASE\_HOME, \$HCAT\_HOME and \$ACCUMULO\_HOME to any existing directory.

```
> Warning: /usr/lib/sqoop/.../hbase does not exist! HBase imports will fail.
> Please set $HBASE_HOME to the root of your HBase installation.
```

## Installation Overview

```
> 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### 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 [Package and Tarball Binaries](#) on page 220 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`.

### 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.

## Installation Overview

### *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 [user guides](#).

### Sqoop 2 Installation

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.



#### Note:

Sqoop 2 lacks some of the features of Sqoop 1. Cloudera recommends you use Sqoop 1. Use Sqoop 2 only if it contains all the features required for your use case.

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`)

These topics describe the steps to install Sqoop 2.

#### Upgrading Sqoop 2 from CDH 4 to CDH 5

To upgrade Sqoop 2 from CDH 4 to CDH 5, proceed as follows.



##### Note:

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#) on page 691, you can skip Step 1 below and proceed with installing the new CDH 5 version of Sqoop 2.

For more detailed instructions for upgrading Sqoop 2, see the [Apache Sqoop Upgrade page](#).

#### Step 1: Remove the CDH 4 version of Sqoop 2

##### To remove Sqoop 2 on a Red Hat-compatible system:

```
$ sudo yum remove sqoop2-server sqoop2-client
```

##### To remove Sqoop 2 on an Ubuntu or other Debian system:

```
$ sudo apt-get remove sqoop2-server sqoop2-client
```

##### To remove Sqoop 2 on a SLES system:

```
$ sudo zypper remove sqoop2-server sqoop2-client
```

#### Step 2: Install the new version of Sqoop 2

1. Install the new version of Sqoop 2 following directions under [Installing Sqoop 2](#) on page 444.

2. If you have been running MRv1 on CDH 4 and will continue to run it on CDH 5:
  - a. Update `/etc/default/sqoop2-server` to point to MRv1:

```
mv /etc/default/sqoop2-server.rpmnew /etc/default/sqoop2-server
```

- b. Update alternatives:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.mrl
```

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](#).

The upgrade is now complete.

#### Upgrading Sqoop 2 from an Earlier CDH 5 Release

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](#) on page 708.

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 444.
2. *If you are running MRv1 on CDH 5 Beta 1 and will continue to run it after upgrading:*

- a. Update /etc/default/sqoop2-server to point to MR1:

```
mv /etc/default/sqoop2-server.rpmnew /etc/default/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



**Note:** Sqoop 2 lacks some of the features of Sqoop 1. Cloudera recommends you use Sqoop 1. Use Sqoop 2 only if it contains all the features required for your use case.

## 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 that you want to use with Sqoop, such as HBase, Hive HCatalog, and Accumulo. Sqoop checks for these services when you run it, and finds services that 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 acts as a client. A Sqoop 2 client always connects 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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

```
$ 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
```

## Installation Overview

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.

#### 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.mrl
```



**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/default/sqoop2-server` file. This does not work as of CDH 5 Beta 2, and in fact could cause problems. **Check your `/etc/default/sqoop2-server` file and make sure `CATALINA_BASE` is not set.**

#### Configuring Sqoop 2 to Use PostgreSQL instead of Apache Derby

##### Deciding which Database to Use

Sqoop 2 has a built-in Derby database, but Cloudera recommends that you use a PostgreSQL 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 that do not happen with server-based databases.

See [Supported Databases](#) on page 32 for tested database versions.

**Note:**

There is currently no recommended way to migrate data from an existing Derby database into the new PostgreSQL database.

Use the procedure that follows to configure Sqoop 2 to use PostgreSQL instead of Apache Derby.

**Install PostgreSQL 8.4.x or 9.0.x**

See [External PostgreSQL Database](#) on page 86.

**Create the Sqoop User and Sqoop Database**

For example, using the PostgreSQL `psql` command-line tool:

```
$ psql -U postgres
Password for user postgres: *****

postgres=# CREATE ROLE sqoop LOGIN ENCRYPTED PASSWORD 'sqoop'
  NOSUPERUSER INHERIT CREATEDB NOCREATEROLE;
CREATE ROLE

postgres=# CREATE DATABASE "sqoop" WITH OWNER = sqoop
  ENCODING = 'UTF8'
  TABLESPACE = pg_default
  LC_COLLATE = 'en_US.UTF8'
  LC_CTYPE = 'en_US.UTF8'
  CONNECTION LIMIT = -1;
CREATE DATABASE

postgres=# \q
```

**Stop the Sqoop 2 Server**

```
$ sudo /sbin/service sqoop2-server stop
```

**Configure Sqoop 2 to use PostgreSQL**

Edit the `sqoop.properties` file (normally `/etc/sqoop2/conf`) as follows:

```
org.apache.sqoop.repository.jdbc.handler=org.apache.sqoop.repository.postgresql.PostgresqlRepositoryHandler
org.apache.sqoop.repository.jdbc.transaction.isolation=isolation level
org.apache.sqoop.repository.jdbc.maximum.connections=max connections
org.apache.sqoop.repository.jdbc.url=jdbc URL
org.apache.sqoop.repository.jdbc.driver=org.postgresql.Driver
org.apache.sqoop.repository.jdbc.user=username
org.apache.sqoop.repository.jdbc.password=password
org.apache.sqoop.repository.jdbc.properties.property=value
```

**Note:**

- Replace *isolation level* with a value such as `READ_COMMITTED`.
- Replace *max connections* with a value such as `10`.
- Replace *jdbc URL* with the hostname on which you installed PostgreSQL.
- Replace *username* with (in this example) `sqoop`
- Replace *password* with (in this example) `sqoop`
- Use `org.apache.sqoop.repository.jdbc.properties.property` to set each additional property you want to configure; see <https://jdbc.postgresql.org/documentation/head/connect.html> for details. For example, replace *property* with `loglevel` and *value* with `3`

## Installation Overview

### Restart the Sqoop 2 Server

```
$ sudo /sbin/service sqoop2-server start
```

### 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 must sign up for an account if you do not already have one, then log in before you can download the driver. 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 might change 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 might 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:

**Moving from Sqoop 1 to Sqoop 2:** Sqoop 2 is essentially the future of the Apache Sqoop project. However, since Sqoop 2 currently lacks some of the features of Sqoop 1, Cloudera recommends you use Sqoop 2 only if it contains all the features required for your use case; otherwise, continue to use Sqoop 1.

Feature	Sqoop 1	Sqoop 2
Connectors for all major RDBMS	Supported.	Not supported.

## Installation Overview

Feature	Sqoop 1	Sqoop 2
		<p><b>Workaround:</b> Use the generic JDBC Connector which has been tested on the following databases: Microsoft SQL Server, PostgreSQL, MySQL and Oracle.</p> <p>This connector should work on any other JDBC compliant database. However, performance might not be comparable to that of specialized connectors in Sqoop.</p>
Kerberos Security Integration	Supported.	Supported.
Data transfer from RDBMS to Hive or HBase	Supported.	<p>Not supported.</p> <p><b>Workaround:</b> Follow this two-step approach.</p> <ol style="list-style-type: none"><li>1. Import data from RDBMS into HDFS</li><li>2. Load data into Hive or HBase manually using appropriate tools and commands such as the <code>LOAD DATA</code> statement in Hive</li></ol>
Data transfer from Hive or HBase to RDBMS	<p>Not supported.</p> <p><b>Workaround:</b> Follow this two-step approach.</p> <ol style="list-style-type: none"><li>1. Extract data from Hive or HBase into HDFS (either as a text or Avro file)</li><li>2. Use Sqoop to export output of previous step to RDBMS</li></ol>	<p>Not supported.</p> <p>Follow the same workaround as for Sqoop 1.</p>

## Whirr Installation



**Important:** This item is deprecated and will be removed in a future release. Cloudera supports items that are deprecated until they are removed. For more information about deprecated and removed items, see [Deprecated Items](#).

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 CDH 4 to CDH 5*

To upgrade Whirr from CDH 4, uninstall the CDH 4 version, modify the properties file, and install the CDH 5 version. Proceed as follows.



### Note:

If you have already performed the steps to uninstall CDH 4 and all components, as described under [Upgrading from CDH 4 to CDH 5](#), you can skip Step 1 below and proceed with installing the new CDH 5 version of Whirr.

#### Step 1: Remove Whirr

1. Stop the Whirr proxy. Kill the `hadoop-proxy.sh` process by pressing Control-C.
2. Destroy the Cluster. Whirr clusters are normally short-lived. If you have a running cluster, destroy it: see [Destroying a cluster](#).
3. Uninstall the CDH 4 version of Whirr:

##### On Red Hat-compatible systems:

```
$ sudo yum remove whirr
```

##### On SLES systems:

```
$ sudo zypper remove whirr
```

##### On Ubuntu and Debian systems:

```
sudo apt-get remove whirr
```

#### 4. Update the Properties File.

Edit the configuration file, called `hadoop.properties` in these instructions, and save it.

- For Hadoop, configure the following properties as shown:
  - For MRv1:

```
whirr.env.repo=cdh5
whirr.hadoop.install-function=install_cdh_hadoop
whirr.hadoop.configure-function=configure_cdh_hadoop
```

- For YARN: see [Defining a Cluster](#).

- For HBase, configure the following properties as shown:

```
whirr.env.repo=cdh5
whirr.hadoop.install-function=install_cdh_hadoop
whirr.hadoop.configure-function=configure_cdh_hadoop
whirr.hbase.install-function=install_cdh_hbase
whirr.hbase.configure-function=configure_cdh_hbase
whirr.zookeeper.install-function=install_cdh_zookeeper
whirr.zookeeper.configure-function=configure_cdh_zookeeper
```

## Installation Overview

- For ZooKeeper, configure the following properties as shown:

```
whirr.env.repo=cdh5
whirr.zookeeper.install-function=install_cdh_zookeeper
whirr.zookeeper.configure-function=configure_cdh_zookeeper
```

See [Defining a Whirr Cluster](#) for a sample file.



### Important:

If you are upgrading from Whirr version 0.3.0, and are using an explicit image (AMI), make sure it comes from one of the supplied Whirr recipe files.

## Step 2: Install the new Version

See the next section, [Installing Whirr](#).

The upgrade is now complete. For more information, see [Defining a Whirr Cluster](#), [Launching a Cluster](#), and [Viewing the Whirr Documentation](#).

*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.

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 455.

Step 3: Install the New Version of Whirr

See [Installing Whirr](#) on page 452.

The upgrade is now complete. For more information, see [Managing a Cluster with Whirr](#) on page 454, and [Viewing the Whirr Documentation](#) on page 455.

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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

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
```

## Installation Overview

### *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-resourcemanager+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**

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is 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 CDH 4 to CDH 5](#) on page 456
- [Upgrading ZooKeeper from an Earlier CDH 5 Release](#) on page 457
- [Installing the ZooKeeper Packages](#) on page 458
- [Maintaining a ZooKeeper Server](#) on page 460
- [Viewing the ZooKeeper Documentation](#) on page 461

#### Upgrading ZooKeeper from CDH 4 to CDH 5

To upgrade ZooKeeper from CDH 4 to CDH 5, uninstall the CDH 4 version (if you have not already done so) and then install the CDH 5 version. Do the following on each server.



**Note:** If you have already performed the steps to uninstall CDH 4 described under [Upgrading from CDH 4 to CDH 5](#), you can skip Step 1 below and proceed with [Step 2](#).

#### Step 1: Remove ZooKeeper

##### 1. Stop the ZooKeeper server:

```
$ sudo service zookeeper-server stop
```

or

```
$ sudo service zookeeper stop
```

depending on the platform and release.

##### 2. Remove CDH 4 ZooKeeper

###### To remove ZooKeeper on Red Hat-compatible systems:

```
$ sudo yum remove zookeeper-server
```

###### To remove ZooKeeper on Ubuntu and Debian systems:

```
$ sudo apt-get remove zookeeper-server
```

**To remove ZooKeeper on SLES systems:**

```
$ sudo zypper remove zookeeper-server
```

**Step 2: Install the ZooKeeper Base Package**

See [Installing the ZooKeeper Base Package](#).

**Step 3: Install the ZooKeeper Server Package**

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.

**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](#) on page 708.

***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 220 and [Upgrading Unmanaged CDH Using the Command Line](#) on page 690.

#### Installing the ZooKeeper Base Package

##### To install ZooKeeper On RHEL-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 459 for more information.

#### **To install the ZooKeeper Server On RHEL-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*.

## Installation Overview

1. Use the commands under [Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server](#) on page 459 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 zoo1:2181
```

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/>.

### 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 RHEL 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,SOURCE,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/redhat/5/x86_64/cdh/5/SRPMs/),

[https://archive.cloudera.com/cdh5/sles/11/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/](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.

## Installation Overview

### 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	yum remove	
Debian and Ubuntu	apt-get remove or apt-get purge	apt-get 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	zypper remove	

### Uninstalling from Red Hat, CentOS, and Similar Systems

Component to Remove	Command
Flume	\$ sudo yum remove flume
Hadoop core packages	\$ sudo yum remove hadoop
Hadoop repository packages	\$ sudo yum remove cloudera-cdh5
HBase	\$ sudo yum remove hadoop-hbase
HDFS HA Journal Node	\$ sudo yum remove hadoop-hdfs-hadoop-hdfs-journalnode
Hive	\$ sudo yum remove hive hive-metastore hive-server hive-server2
HttpFS	\$ sudo yum remove hadoop-httpfs
Hue	\$ sudo yum remove hue
Mahout	\$ sudo yum remove mahout
Pig	\$ sudo yum remove pig

Component to Remove	Command
Search	\$ sudo yum remove solr hbase-solr search solr-mapreduce solr-doc search-crunch
Sentry	\$ sudo yum remove sentry
Spark	\$ sudo yum remove spark-core spark-master spark-worker spark-history-server spark-python
Sqoop 1	\$ sudo yum remove sqoop
Sqoop 2	\$ sudo yum remove sqoop2-server sqoop2-client
Oozie client	\$ sudo yum remove oozie-client
Oozie server	\$ sudo yum remove oozie
Whirr	\$ sudo yum remove whirr
ZooKeeper server	\$ sudo yum remove hadoop-zookeeper-server
ZooKeeper client	\$ sudo yum remove hadoop-zookeeper
ZooKeeper Failover Controller (ZKFC)	\$ sudo yum remove hadoop-hdfs-zkfc

### 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
Flume	\$ sudo apt-get remove flume
Hadoop core packages	\$ sudo apt-get remove hadoop
Hadoop repository packages	\$ sudo apt-get remove cdhn-repository
HBase	\$ sudo apt-get remove hadoop-hbase
HDFS HA Journal Node	\$ apt-get remove hadoop-hdfs-hadoop-hdfs-journalnode
Hive	\$ sudo apt-get remove hive hive-metastore hive-server hive-server2
HttpFS	\$ sudo apt-get remove hadoop-httpfs
Hue	\$ sudo apt-get remove hue
Oozie client	\$ sudo apt-get remove oozie-client
Oozie server	\$ sudo apt-get remove oozie
Pig	\$ sudo apt-get remove pig

## Installation Overview

Component to Remove	Command
Search	\$ sudo apt-get remove solr hbase-solr search solr-mapreduce solr-doc search-crunch
Sentry	\$ sudo apt-get remove sentry
Spark	\$ sudo apt-get remove spark-core spark-master spark-worker spark-history-server spark-python
Sqoop 1	\$ sudo apt-get remove sqoop
Sqoop 2	\$ sudo apt-get remove sqoop2-server sqoop2-client
Whirr	\$ sudo apt-get remove whirr
ZooKeeper client	\$ sudo apt-get remove hadoop-zookeeper
ZooKeeper Failover Controller (ZKFC)	\$ sudo apt-get remove hadoop-hdfs-zkfc
ZooKeeper server	\$ sudo apt-get remove hadoop-zookeeper-server

### Uninstalling from SLES

Component to Remove	Command
Flume	\$ sudo zypper remove flume
Hadoop core packages	\$ sudo zypper remove hadoop
Hadoop repository packages	\$ sudo zypper remove cloudera-cdh
HBase	\$ sudo zypper remove hadoop-hbase
HDFS HA Journal Node	\$ sudo zypper remove hadoop-hdfs-hadoop-hdfs-journalnode
Hive	\$ sudo zypper remove hive hive-metastore hive-server hive-server2
HttpFS	\$ sudo zypper remove hadoop-httfs
Hue	\$ sudo zypper remove hue
Oozie client	\$ sudo zypper remove oozie-client
Oozie server	\$ sudo zypper remove oozie
Pig	\$ sudo zypper remove pig
Search	\$ sudo zypper remove solr hbase-solr search solr-mapreduce solr-doc search-crunch
Sentry	\$ sudo zypper remove sentry
Spark	\$ sudo zypper remove spark-core spark-master spark-worker spark-history-server spark-python
Sqoop 1	\$ sudo zypper remove sqoop
Sqoop 2	\$ sudo zypper remove sqoop2-server sqoop2-client

Component to Remove	Command
Whirr	\$ sudo zypper remove whirr
ZooKeeper client	\$ sudo zypper remove hadoop-zookeeper
ZooKeeper Failover Controller (ZKFC)	\$ sudo zypper remove hadoop-hdfs-zkfc
ZooKeeper server	\$ sudo zypper remove hadoop-zookeeper-server

#### 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
- Hue, Oozie, and Sqoop 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>.

# Upgrade

This section describes how to upgrade Cloudera Manager, Cloudera Navigator, CDH, and the JDK.

The process for upgrading CDH and the JDK varies depending on whether you have a [Cloudera Manager Deployment](#) on page 74 or an [Unmanaged Deployment](#) on page 77.



**Warning:** You can use Cloudera Manager to roll back an upgrade from CDH 4 to CDH 5 as long as you backup certain configuration files, databases, and other artifacts before beginning an upgrade. However, after you have [finalized the HDFS upgrade](#) you can no longer roll back the CDH upgrade. See [Rolling Back a CDH 4-to-CDH 5 Upgrade](#) on page 750 for the backup and rollback procedures.

## Upgrading Cloudera Manager

Upgrading Cloudera Manager enables new features of the latest product versions while preserving existing data and settings. Some new settings are added, and some additional steps may be required, but no existing configuration is removed.



**Note:** When an upgraded Cloudera Manager adds support for a new feature (for example, Sqoop 2, WebHCat, and so on), it does not install the software on which the new feature depends. If you install CDH and managed services from packages, you must add the packages to your managed hosts before adding a service or role that supports the new feature.

### Understanding Upgrades

The process for upgrading Cloudera Manager varies depending on the starting point. Categories of tasks to be completed include the following:

- Install databases required for the release. In Cloudera Manager 5, the Host Monitor and Service Monitor roles use an internal database that provides greater capacity and flexibility. You do not need to configure an external database for these roles. If you are upgrading from Cloudera Manager 4, this transition is handled automatically. If you are upgrading a Free Edition installation and you are running a MapReduce service, you are asked to configure an additional database for the Activity Monitor that is part of Cloudera Express.
- Upgrade the Cloudera Manager Server.
- Upgrade the Cloudera Manager Agent. You can use an upgrade wizard that is invoked when you connect to the Admin Console or manually install the Cloudera Manager Agent packages.

## Upgrading Cloudera Manager

You can upgrade from any version of Cloudera Manager 4 running CDH 4 to Cloudera Manager 5, or from Cloudera Manager 5 to a later version of Cloudera Manager 5.



**Note:** If necessary, you can roll back an upgrade from CDH 4 to CDH 5 and, optionally, downgrade to Cloudera Manager 4.x. See [Rolling Back a CDH 4-to-CDH 5 Upgrade](#) on page 750.

To upgrade Cloudera Manager, see the instructions at:

- [Upgrading Cloudera Manager 5 to the Latest Cloudera Manager](#) on page 469:
  - The database schema is upgraded to reflect the current version.
  - The Cloudera Manager Server and all supporting services are updated.
  - Client configurations are redeployed to ensure client services have the most current configuration.

- [Upgrading Cloudera Manager 4 to Cloudera Manager 5](#) on page 481 and [Upgrading Cloudera Manager 3.7.x](#) on page 500:

- The database schema is upgraded to reflect the current version. Data from the existing Host and Service Monitor databases is migrated.
- The Cloudera Manager Server and all supporting services are updated.
- Client configurations are redeployed to ensure client services have the most current configuration.
- Cloudera Manager 5 continues to support a CDH 4 cluster with an existing high availability deployment using NFS shared edits directories. However, if you disable high availability in Cloudera Manager 5, you can re-enable high availability only by using Quorum-based Storage. CDH 5 does not support enabling NFS shared edits directories with high availability.

## Upgrading CDH

Cloudera Manager 5 can manage both CDH 4 and CDH 5, so upgrading existing CDH 4 installations is not required. However, to benefit from the most current CDH features, you must upgrade CDH. For more information on upgrading CDH, see [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524.

## Database Considerations for Cloudera Manager Upgrades

Cloudera Manager uses databases to store information about system configurations and tasks. Before upgrading, complete the pre-upgrade database tasks that apply in your environment.



### Note:

Cloudera Manager 4.5 added support for Hive, which includes the Hive Metastore Server role type. This role manages the metastore process when Hive is configured with a remote metastore.

When upgrading from Cloudera Manager versions before 4.5, Cloudera Manager automatically creates new Hive services to capture the previous implicit Hive dependency from Hue and Impala. Your previous services continue to function without impact. If Hue was using a Hive metastore backed by a Derby database, the newly created Hive Metastore Server also uses Derby. Because Derby does not allow concurrent connections, Hue continues to work, but the new Hive Metastore Server does not run. The failure is harmless (because nothing uses this new Hive Metastore Server at this point) and intentional, to preserve the set of cluster functionality as it was before upgrade. Cloudera discourages the use of a Derby-backed Hive metastore due to its limitations and recommends switching to a different supported database.

After you have completed these steps, the upgrade processes automatically complete any additional updates to database schema and service data stored. You do not need to complete any data migration.

### Backing up Databases

Before beginning the upgrade process, shut down the services that are using databases. This includes the Cloudera Manager Management Service roles, the Hive Metastore Server, and Cloudera Navigator, if it is in use. Cloudera strongly recommends that you then back up all databases, however backing up the Activity Monitor database is optional. For information on backing up databases see [Backing Up Databases](#) on page 117.

### Creating New Databases

If any additional databases will be required as a result of the upgrade, complete any required preparatory work to install and configure those databases. The upgrade instructions assume all required databases have been prepared. For more information on required databases, see [Cloudera Manager and Managed Service Datastores](#) on page 79.

### Modifying Databases to Support UTF-8

Cloudera Manager 4 adds support for UTF-8 character sets. Update any existing databases in your environment that are not configured to support UTF-8.

## Upgrade

### Modifying MySQL to Support UTF-8

To modify a MySQL database to support UTF-8, the default character set must be changed and then you must restart the `mysql` service. Use the following commands to complete these tasks:

```
mysql> alter database default character set utf8;
mysql> quit
$ sudo service mysql restart
```

### Modifying PostgreSQL to Support UTF-8

There is no single command available to modify an existing PostgreSQL database to support UTF-8. As a result, you must complete the following process:

1. Use `pg_dump` to export the database to a file. This creates a backup of the database that you will import into a new, empty database that supports UTF-8.
2. Drop the existing database. This deletes the existing database.
3. Create a new database that supports Unicode encoding and that has the same name as the old database. Use a command of the following form, replacing the database name and user name with values that match your environment:

```
CREATE DATABASE scm_database WITH OWNER scm_user ENCODING 'UTF8'
```

4. Review the contents of the exported database for non-standard characters. If you find unexpected characters, modify these so the database backup file contains the expected data.
5. Import the database backup to the newly created database.

### Modifying Oracle to Support UTF-8

Work with your Oracle database administrator to ensure any Oracle databases support UTF-8.

### Modifying Databases to Support Appropriate Maximum Connections

Check existing databases configurations to ensure the proper maximum number of connections is supported. Update the maximum configuration values, as required.

#### Modifying the Maximum Number of MySQL Connections

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.

#### Modifying the Maximum Number of PostgreSQL Connections

Update the `max_connection` parameter in the `/etc/postgresql.conf` file.

You may have to increase the system resources available to PostgreSQL, as described at <http://www.postgresql.org/docs/9.1/static/kernel-resources.html>.

#### 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;
```

## Next Steps

After you have completed any required database preparatory tasks, continue to [Upgrading Cloudera Manager 4 to Cloudera Manager 5](#) on page 481 or [Upgrading Cloudera Manager 5 to the Latest Cloudera Manager](#) on page 469.

## Upgrading Cloudera Manager 5 to the Latest Cloudera Manager

### Minimum Required Role: [Full Administrator](#)

This process applies to upgrading all versions of Cloudera Manager 5.

In most cases it is possible to complete the following upgrade without shutting down most CDH services, although you may need to stop some dependent services. CDH daemons can continue running, unaffected, while Cloudera Manager is upgraded. The upgrade process does not affect your CDH installation. After upgrading Cloudera Manager you may also want to upgrade CDH 4 clusters to CDH 5.

Upgrading Cloudera Manager 5 to the latest version of Cloudera Manager involves the following steps.

### Review Warnings



#### Warning:

- **Cloudera Management Service TLS/SSL configuration**

If you have enabled TLS security for the Cloudera Manager Admin Console, as of Cloudera Manager 5.1, Cloudera Management Service roles try to communicate with Cloudera Manager using TLS, and fail to start until TLS/SSL properties have been configured.

- **Navigator**

If you have enabled auditing with Cloudera Navigator, during the upgrade to Cloudera Manager 5, auditing is suspended and is only restarted when you restart the roles of audited services.

- **JDK upgrade**

If you upgrade the JDK during the installation of the Cloudera Manager Agent, you must *restart all services*. Additionally, if you have enabled TLS/SSL, you must reinstall CA certificates to your truststores. See [Creating Truststores](#).

## Upgrade

### Before You Begin

Before upgrading the Cloudera Manager, perform the following steps:

- **Obtain host credentials** - For Cloudera Manager to upgrade the Agent packages, you must have SSH access and be able to log in using a root account or an account that has password-less sudo permission. See [Cloudera Manager 5 Requirements and Supported Versions](#) on page 10 for more information.
- **Prepare databases** - See [Database Considerations for Cloudera Manager Upgrades](#) on page 467.
- If upgrading from Cloudera Manager 5.4 to Cloudera Manager 5.5 or higher, perform pre-upgrade steps required for the associated Cloudera Navigator upgrade:
  1. Stop the Navigator Metadata Server role.
  2. Back up the [Navigator Metadata Server storage directory](#).
  3. [Make sure that the Navigator Metadata Server has sufficient memory](#) to complete the upgrade.
  4. If you are using an Oracle database, in SQL\*Plus, ensure that the following additional privileges are set:

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

where `nav` is the user of the Navigator Audit Server database.

For further information, see [Upgrading the Cloudera Navigator Data Management Component](#) on page 503.

### Stop Selected Services and Roles

If your cluster meets any of the conditions listed in the following table, you must stop the indicated services or roles.

Condition	Procedure
Running a version of Cloudera Manager that has the Cloudera Management Service	Stop the Cloudera Management Service.
Running the embedded PostgreSQL database	Stop all services that are using the embedded database: <ul style="list-style-type: none"><li>• Hive service and all services such as Impala and Hue that use the Hive metastore</li><li>• Oozie</li><li>• Sentry</li></ul>
Running Cloudera Navigator data management component and the following services are enabled for auditing: <ul style="list-style-type: none"><li>• <b>HDFS</b></li><li>• <b>HBase</b></li><li>• <b>Hive</b></li><li>• <b>Hue</b></li></ul>	Stop the following roles: <ul style="list-style-type: none"><li>• <b>HDFS</b> - NameNode</li><li>• <b>HBase</b> - Master and RegionServers</li><li>• <b>Hive</b> - HiveServer2</li><li>• <b>Hue</b> - Beeswax Server</li></ul> Stopping these roles renders any service depending on these roles unavailable. For the <b>HDFS</b> - NameNode case this implies most of the services in the cluster will be unavailable until the upgrade is finished.

### Remove Kafka 1.2 CSD

If you have previously installed Kafka 1.2, remove the Kafka CSD:

1. Determine the location of the CSD directory:
  - a. Select **Administration > Settings**.
  - b. Click the **Custom Service Descriptors** category.
  - c. Retrieve the directory from the **Local Descriptor Repository Path** property.
2. Delete the Kafka CSD from the directory.

## Stop Cloudera Manager Server, Database, and Agent

**1.** Use the Admin Console to stop any running commands. These include commands a user runs and commands Cloudera Manager automatically triggers in response to a state change or a schedule. You can either wait for commands to complete or abort any running commands. For more information on viewing and aborting running commands, see [Viewing Running and Recent Commands](#). If you do not stop all commands, the Cloudera Manager Server will fail to start after upgrade.

**2.** On the host running the Cloudera Manager Server, stop the Cloudera Manager Server:

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

**3.** If you are using the embedded PostgreSQL database for Cloudera Manager, stop the database:

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



**Important:** If you are *not* running the embedded database service and you attempt to stop it, you receive a message indicating that the service cannot be found. If instead you get a message that the shutdown failed, the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

- RHEL-compatible 7 and higher:

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

- All other Linux distributions:

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

**4.** If the Cloudera Manager host is also running the Cloudera Manager Agent, stop the Cloudera Manager Agent:

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

## Upgrade the JDK on Cloudera Manager Server and Agent Hosts

If you are using JDK 1.6, you *must* upgrade to JDK 1.7 or 1.8. See [Java Development Kit Installation](#) on page 78.

## Upgrade Cloudera Manager Software

Choose a procedure based on how you installed Cloudera Manager:

### Upgrade Cloudera Manager Server (Packages)

**1.** To upgrade the Cloudera Manager Server packages, you can upgrade from the Cloudera repository at <https://archive.cloudera.com/cm5/>, or you can create your own repository, as described in [Understanding Custom Installation Solutions](#) on page 170. You must create your own repository if you are upgrading a cluster that does not have Internet access.

**a.** Find the Cloudera repo file for your distribution by starting at <https://archive.cloudera.com/cm5/> and navigating to the directory that matches your operating system.

For example, for Red Hat or CentOS 6, you would go to [https://archive.cloudera.com/cm5/redhat/6/x86\\_64/cm/](https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/). In that directory, find the repo file that

## Upgrade

contains information including the repository base URL and GPG key. The contents of the `cloudera-manager.repo` are similar to the following:

```
[cloudera-manager]
# Packages for Cloudera Manager, Version 5, on RHEL or CentOS 6 x86_64
name=Cloudera Manager
baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/
gpgkey = https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

For Ubuntu or Debian systems, go to the appropriate release directory, for example, <https://archive.cloudera.com/cm4/debian/wheezy/amd64/cm>. The repo file, in this case, `cloudera.list`, is similar to the following:

```
# Packages for Cloudera Manager, Version 5, on Debian 7.0 x86_64
deb https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm wheezy-cm5 contrib
deb-src https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm wheezy-cm5 contrib
```

- b.** Replace the repo file in the configuration location for the package management software for your system.

Operating System	Commands
RHEL	Copy <code>cloudera-manager.repo</code> to <code>/etc/yum.repos.d/</code> .
SLES	Copy <code>cloudera-manager.repo</code> to <code>/etc/zypp/repos.d/</code> .
Ubuntu or Debian	Copy <code>cloudera.list</code> to <code>/etc/apt/sources.list.d/</code> .

- c.** Run the following commands:

Operating System	Commands
RHEL	\$ sudo yum clean all \$ sudo yum upgrade cloudera-manager-server cloudera-manager-daemons cloudera-manager-server-db-2 cloudera-manager-agent
SLES	\$ sudo zypper clean --all \$ sudo zypper up -r <a href="https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/">https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/</a>  To download from your own repository:  \$ sudo zypper clean --all \$ sudo zypper rr cm \$ sudo zypper ar -t rpm-md <a href="http://myhost.example.com/path_to_cm_repo/cm">http://myhost.example.com/path_to_cm_repo/cm</a> \$ sudo zypper up -r <a href="http://myhost.example.com/path_to_cm_repo">http://myhost.example.com/path_to_cm_repo</a>
Ubuntu or Debian	The following commands clean cached repository information and update Cloudera Manager components:  \$ sudo apt-get clean \$ sudo apt-get update \$ sudo apt-get dist-upgrade \$ sudo apt-get install cloudera-manager-server cloudera-manager-daemons cloudera-manager-server-db-2 cloudera-manager-agent



**Note:**

- `yum clean all` cleans `yum` cache directories, ensuring that you download and install the latest versions of the packages.
- If your system is not up to date, any underlying system components must be upgraded before `yum update` can succeed. `yum` indicates which components must be upgraded.

Operating System	Commands
	<p>During this process, you may be prompted about your configuration file version:</p> <pre>Configuration file `/etc/cloudera-scm-agent/config.ini' ==&gt; Modified (by you or by a script) since installation. ==&gt; Package distributor has shipped an updated version. What would you like to do about it ? Your options are: Y or I : install the package maintainer's version N or O : keep your currently-installed version D : show the differences between the versions Z : start a shell to examine the situation The default action is to keep your current version.</pre> <p>You will receive a similar prompt for /etc/cloudera-scm-server/db.properties. Answer <b>N</b> to both prompts.</p>

- If you customized /etc/cloudera-scm-agent/config.ini, your customized file is moved to a file with the extension .rpmsave or .dpkg-old. Merge any customizations into /etc/cloudera-scm-agent/config.ini installed by the package manager.

You should now have the following packages, corresponding to the version of Cloudera Manager you installed, on the host that will be the Cloudera Manager Server host.

OS	Packages
RPM-based distributions	<pre>\$ rpm -qa 'cloudera-manager-*' cloudera-manager-repository-5.0-1.noarch cloudera-manager-server-5.7.2-0.cm572.p0.16.el6.x86_64 cloudera-manager-server-db-2-5.7.2-0.cm572.p0.16.el6.x86_64 cloudera-manager-agent-5.7.2-0.cm572.p0.16.el6.x86_64 cloudera-manager-daemons-5.7.2-0.cm572.p0.16.el6.x86_64</pre>
Ubuntu or Debian	<pre>~# dpkg-query -l 'cloudera-manager-*' Desired=Unknown/Install/Remove/Purge/Hold   Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend  / Err?=(none)/Reinst-required (Status,Err: uppercase=bad)   / Name                           Version        Description +++ ii  cloudera-manager-agent 5.7.2-0.cm572.p0.16~sq The Cloudera Manager Agent ii  cloudera-manager-daemo 5.7.2-0.cm572.p0.16~sq Provides daemons for monitoring Hadoop and related tools. ii  cloudera-manager-serve 5.7.2-0.cm572.p0.16~sq The Cloudera Manager Server</pre>

You may also see an entry for the cloudera-manager-server-db-2 if you are using the embedded database, and additional packages for plug-ins, depending on what was previously installed on the server host. If the cloudera-manager-server-db-2 package is installed, and you do not plan to use the embedded database, you can remove this package.

#### Install Cloudera Manager Server and Agent Software (Tarballs)

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
```

Extract the contents of the tarball, to this directory. 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
```

## Upgrade

The files are extracted to a subdirectory named according to the Cloudera Manager version being extracted. For example, files could be extracted to `/opt/cloudera-manager/cm-5.0/`. This full path is needed later and is referred to as `tarball_root` directory.

### 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 installation has a `var` subdirectory where state is stored. In a non-tarball installation, state is stored in `/var`. Cloudera recommends that you reconfigure the tarball installation 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 installation 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 do not reuse the state directory between different tarball installations, duplicate Cloudera Manager Agent entries can occur in the Cloudera Manager database.

### Start Cloudera Manager Server

Choose a procedure based on how you installed Cloudera Manager:

#### Start the Cloudera Manager Server (Packages)

On the Cloudera Manager Server host (the system on which you installed the `cloudera-manager-server` package) do the following:

- If you are using the embedded PostgreSQL database for Cloudera Manager, start the database:

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

- Start the Cloudera Manager Server:

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

You should see the following:

```
Starting cloudera-scm-server: [ OK ]
```

#### Start the Cloudera Manager Server (Tarball)

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 this case, you have the 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`.

## Upgrade and Start Cloudera Manager Agents

Choose a procedure based on how you installed Cloudera Manager:

### Upgrade and Start Cloudera Manager Agent (Packages)



**Important:** All hosts in the cluster must have access to the Internet if you plan to use `archive.cloudera.com` as the source for installation files. If you do not have Internet access, create a custom repository.

1. Log in to the Cloudera Manager Admin Console.
2. Upgrade hosts using one of the following methods:

- **Cloudera Manager installs Agent software**

1. Select **Yes, I would like to upgrade the Cloudera Manager Agent packages now** and click **Continue**.

2. Select the release of the Cloudera Manager Agent to install. Normally, this is the **Matched Release for this Cloudera Manager Server**. However, if you used a custom repository (instead of archive.cloudera.com) for the Cloudera Manager server, select **Custom Repository** and provide the required information. The custom repository allows you to use an alternative location, but that location must contain the matched Agent version.
3. Click **Continue**. The JDK Installation Options page displays.
  - Leave **Install Oracle Java SE Development Kit (JDK)** checked to allow Cloudera Manager to install the JDK on each cluster host, or uncheck if you plan to install it yourself.
  - If local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, check the **Install Java Unlimited Strength Encryption Policy Files** checkbox.

Click **Continue**.

4. Specify credentials and initiate Agent installation:
  - Select **root** or enter the username for an account that has password-less sudo permission.
  - Select an authentication method:
    - If you choose password authentication, enter and confirm the password.
    - If you choose public-key authentication, provide a passphrase and path to the required key files.
  - You can 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.
5. Click **Continue**. The Cloudera Manager Agent packages and optionally the JDK are installed.
6. Click **Continue**. The Host Inspector runs to inspect your managed hosts for correct versions and configurations. If there are problems, you can make changes and then rerun the inspector. When you are satisfied with the inspection results, click **Continue**.

- **Manually install Agent software**

1. On all cluster hosts except the Cloudera Manager Server host, stop the Agent:

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

2. In the Cloudera Admin Console, select **No, I would like to skip the agent upgrade now** and click **Continue**.
3. Copy the appropriate repo file as described in [Upgrade Cloudera Manager Server \(Packages\)](#) on page 471.
4. Run the following commands:

Operating System	Commands
RHEL	<pre>\$ sudo yum clean all \$ sudo yum upgrade cloudera-manager-server cloudera-manager-daemons cloudera-manager-server-db-2 cloudera-manager-agent</pre> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <b>Note:</b><ul style="list-style-type: none"><li>• <code>yum clean all</code> cleans <code>yum</code> cache directories, ensuring that you download and install the latest versions of the packages.</li><li>• If your system is not up to date, any underlying system components must be upgraded before <code>yum update</code> can succeed. <code>yum</code> indicates which components must be upgraded.</li></ul></div>
SLES	<pre>\$ sudo zypper clean --all \$ sudo zypper up -r https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/</pre> <p>To download from your own repository:</p>

Operating System	Commands
	<pre>\$ sudo zypper clean --all \$ sudo zypper rr cm \$ sudo zypper ar -t rpm-md http://myhost.example.com/path_to_cm_repo/cm \$ sudo zypper up -r http://myhost.example.com/path_to_cm_repo</pre>
<b>Ubuntu or Debian</b>	<p>Use the following commands to clean cached repository information and update Cloudera Manager components:</p> <pre>\$ sudo apt-get clean \$ sudo apt-get update \$ sudo apt-get dist-upgrade \$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons</pre> <p>During this process, you may be prompted about your configuration file version:</p> <pre>Configuration file '/etc/cloudera-scm-agent/config.ini' ==&gt; Modified (by you or by a script) since installation. ==&gt; Package distributor has shipped an updated version. What would you like to do about it ? Your options are: Y or I : install the package maintainer's version N or O : keep your currently-installed version D : show the differences between the versions Z : start a shell to examine the situation The default action is to keep your current version.</pre> <p>You will receive a similar prompt for /etc/cloudera-scm-server/db.properties. Answer <b>N</b> to both prompts.</p>

5. If you customized /etc/cloudera-scm-agent/config.ini, your customized file is moved to a file with the extension .rpmsave or .dpkg-old. Merge any customizations into /etc/cloudera-scm-agent/config.ini installed by the package manager.
6. On all cluster hosts, start the Agent:

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

7. Click **Continue**. The Host Inspector runs to inspect your managed hosts for correct versions and configurations. If there are problems, you can make changes and then rerun the inspector. When you are satisfied with the inspection results, click **Continue**.
3. Click **Finish**.
4. If you are upgrading from Cloudera Manager 5.0 and are using an external database for Cloudera Navigator, the Database Setup page displays. Configure database settings:
  - 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.)
5. The Review Changes page displays. Review the configuration changes to be applied and click **Continue**. The Upgrade wizard displays a dialog box allowing you to choose whether to restart the Cloudera Management Service.
6. Click **Continue**. If you kept the default selection, the Upgrade wizard restarts the Cloudera Management Service.
7. Click **Finish**. The **Home > Status** tab displays.

All services (except for the services you stopped in [Stop Selected Services and Roles](#) on page 470) should be running.

## Upgrade

### Restart Cloudera Manager Agents (Tarballs)

#### Stop Cloudera Manager Agents (Tarballs)

- To stop the Cloudera Manager Agent, run this command on each Agent host:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent stop
```

- If you are running [single user mode](#), stop Cloudera Manager Agent using the user account you chose. For example, if you are running the Cloudera Manager Agent as `cloudera-scm`, you have the following options:

- Run the following command:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent stop
```

- Edit the configuration files so the script internally changes the user, and then run the script as root:

- Remove the following line from `tarball_root/etc/default/cloudera-scm-agent`:

```
export CMF_SUDO_CMD= "
```

- 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
```

- Run the Agent script as root:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent stop
```

### Start Cloudera Manager Agents (Tarballs)

Start the Cloudera Manager Agent according to the 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, to run the Cloudera Manager Agent as `cloudera-scm`, you have the 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, and then run the script as root:

- Remove the following line from `tarball_root/etc/default/cloudera-scm-agent`:

```
export CMF_SUDO_CMD= "
```

- 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`.

### Verify the Upgrade Succeeded

If the commands to update and start the Cloudera Manager Server complete without errors, you can assume the upgrade has completed successfully. To verify, you can check that the server versions have been updated.

1. In the Cloudera Manager Admin Console, click the **Hosts** tab.
2. Click **Host Inspector**. On large clusters, the host inspector may take some time to finish running. You must wait for the process to complete before proceeding to the next step.
3. Click **Show Inspector Results**. All results from the host inspector process are displayed, including the currently installed versions. If this includes listings of current component versions, the installation completed as expected.

### (Optional) Configure TLS/SSL for Cloudera Management Service

If you have enabled TLS security for the Cloudera Manager Admin Console, as of Cloudera Manager 5.1, Cloudera Management Service roles try to communicate with Cloudera Manager using TLS, and fail to start until TLS/SSL properties have been configured. Configure Cloudera Management Service roles to communicate with Cloudera Manager over TLS/SSL as follows:

1. Do one of the following:
  - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
  - On the **Home > Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select **Scope > Cloudera Management Service (Service-Wide)**.
4. Select **Category > Security**.
5. Edit the following TLS/SSL properties according to your cluster configuration.

Property	Description
<b>TLS/SSL Client Truststore File Location</b>	Path to the client truststore file used in HTTPS communication. The contents of this truststore can be modified without restarting the Cloudera Management Service roles. By default, changes to its contents are picked up within ten seconds.
<b>TLS/SSL Client Truststore File Password</b>	Password for the client truststore file.

## Upgrade

6. Click **Save Changes** to commit the changes.
7. Restart the Cloudera Management Service.

For more information, see [HTTPS Communication in Cloudera Manager](#).

### Deploy JDK Upgrade

If you upgraded the JDK when installing the Cloudera Manager Agents, do the following:

1. If the Cloudera Manager Server host is also running a Cloudera Manager Agent, restart the Cloudera Manager Server:

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

If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.

2. Restart all services:

- a. From the **Home > Status** tab click



next to the cluster name and select **Restart**.

- b. In the confirmation dialog box that displays, click **Restart**.

### Disable Kafka Monitoring

If you have a Kafka 1.2 service, disable Kafka monitoring:

1. Go to the Kafka service.
2. Click the **Configuration** tab.
3. Type enable in the Search box.
4. Deselect the **Enable Kafka monitoring** checkbox.
5. Click **Save Changes** to commit the changes.

### Start Selected Services and Roles

Start the services and roles you shut down in [Stop Selected Services and Roles](#) on page 470 that have not been started in other steps:

1. If you do not plan on upgrading CDH, do the following:

- a. If you are running Cloudera Navigator, start the following roles of audited services whose service's Queue Policy configuration (`navigator.batch.queue_policy`) is set to SHUTDOWN:

- **HDFS** - NameNode
- **HBase** - Master and RegionServers
- **Hive** - HiveServer2
- **Hue** - Beeswax Server

- b. From the **Home > Status** tab click



next to the name of each service you shut down and select **Start**.

- c. In the confirmation dialog box that displays, click **Start**.

2. From the **Home > Status** tab click



next to the Cloudera Management Service and select **Start**.

3. In the confirmation dialog box that displays, click **Start**.

### (Optional) Restart Services and Deploy Updated Client Configurations

When upgrading Cloudera Manager, even across maintenance releases, sometimes Cloudera Manager reports [stale configurations](#) after the upgrade. This can be due to a fix that requires configuring CDH services differently.

You do not need to restart services and redeploy client configurations immediately, but if you do not plan to upgrade CDH, you should plan to apply the configuration change in the near future.

You restart services and update the client configurations as follows:

1. On the **Home > Status** tab, click



next to the cluster name and select **Restart**.

2. In the confirmation dialog box, click **Restart**.

3. On the **Home > Status** tab, click



next to the cluster name and select **Deploy Client Configuration**.

4. In the confirmation dialog box, click **Deploy Client Configuration**.

### Test the Installation

When you have finished the upgrade to Cloudera Manager, you can test the installation to verify that the monitoring features are working as expected; follow the instructions in [Testing the Installation](#) on page 198.

### (Optional) Upgrade CDH

Cloudera Manager 5 can manage both CDH 4 and CDH 5, so upgrading existing CDH 4 and 5 installations is not required, but you may want to upgrade to the latest version. For more information on upgrading CDH, see [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524.

## Upgrading Cloudera Manager 4 to Cloudera Manager 5

### Minimum Required Role: [Full Administrator](#)

This process applies to upgrading all versions of Cloudera Manager 4 to Cloudera Manager 5.

In most cases, you can upgrade without shutting down most CDH services, although you may need to stop some dependent services. CDH daemons can run unaffected while Cloudera Manager is upgraded, and the upgrade process does not affect your CDH installation. However, to use Cloudera Manager 5 features, all services must be restarted after the upgrade. After upgrading Cloudera Manager you may also want to upgrade CDH 4 clusters to CDH 5.

Follow these steps to upgrade Cloudera Manager 4 to the latest version of Cloudera Manager.

### Review Warnings and Notes



#### Warning:

- **Cloudera Management Service databases**

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. For more information, see [Data Storage for Monitoring Data](#) on page 118.

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.

If you have been storing the data in an external database, you can drop those [databases](#) after upgrade completes.

- **Cloudera Management Service TLS/SSL configuration**

If you have enabled TLS security for the Cloudera Manager Admin Console, as of Cloudera Manager 5.1, Cloudera Management Service roles try to communicate with Cloudera Manager using TLS, and fail to start until TLS/SSL properties have been configured.

- **Impala**

Cloudera Manager 5 supports Impala 1.2.1 and higher. If the version of your Impala service is 1.1 and lower, the following upgrade instructions will work, but once the upgrade has completed, you will see a validation warning for your Impala service, and you will not be able to restart your Impala (or Hue) services until you upgrade your Impala service to 1.2.1 and higher. If you want to continue to use Impala 1.1 and lower, *do not* upgrade to Cloudera Manager 5.

- **Navigator**

If you have enabled auditing with Cloudera Navigator, during the upgrade to Cloudera Manager 5, auditing is suspended and is only restarted when you restart the roles of audited services.

- **JDK upgrade**

If you upgrade the JDK during the installation of the Cloudera Manager Agent, you must *restart all services*. Additionally, if you have enabled TLS/SSL, you must reinstall CA certificates to your truststores. See [Creating Truststores](#).

- **Hard Restart of Cloudera Manager Agents**

Certain circumstances require that you hard restart the Cloudera Manager Agent on each host:

- Deploying a fix to an issue where Cloudera Manager did not always correctly restart services
- Using the maximum file descriptor feature
- Enabling HDFS DataNodes to start if you perform the step [\(Optional\) Upgrade CDH](#) on page 499 after upgrading Cloudera Manager

**Important:**

- **Hive**

Cloudera Manager 4.5 added support for Hive, which includes the Hive Metastore Server role type. This role manages the metastore process when Hive is configured with a remote metastore.

When upgrading from Cloudera Manager versions before 4.5, Cloudera Manager automatically creates new Hive services to capture the previous implicit Hive dependency from Hue and Impala. Your previous services continue to function without impact. If Hue was using a Hive metastore backed by a Derby database, the newly created Hive Metastore Server also uses Derby. Because Derby does not allow concurrent connections, Hue continues to work, but the new Hive Metastore Server does not run. The failure is harmless (because nothing uses this new Hive Metastore Server at this point) and intentional, to preserve the set of cluster functionality as it was before upgrade. Cloudera discourages the use of a Derby-backed Hive metastore due to its limitations and recommends switching to a different supported database.

Cloudera Manager provides a Hive configuration option to bypass the Hive Metastore Server. When this configuration is enabled, Hive clients, Hue, and Impala connect directly to the Hive metastore database. Prior to Cloudera Manager 4.5, Hue and Impala connected directly to the Hive metastore database, so the bypass mode is enabled by default when upgrading to Cloudera Manager 4.5 and higher. This ensures that the upgrade does not disrupt your existing setup. You should plan to disable the bypass mode, especially when using CDH 4.2 and higher. Using the Hive Metastore Server is the recommended configuration, and the WebHCat Server role requires the Hive Metastore Server to *not* be bypassed. To disable bypass mode, see [Disabling Bypass Mode](#).

Cloudera Manager 4.5 and higher also supports HiveServer2 with CDH 4.2. In CDH 4, HiveServer2 is not added by default, but can be added as a new role under the Hive service (see [Role Instances](#)). In CDH 5, HiveServer2 is a mandatory role.



**Note:** If you are upgrading from Cloudera Manager Free Edition 4.5 and lower, you are upgraded to Cloudera Express, which includes a number of features that were previously available only with Cloudera Enterprise. Of those features, activity monitoring requires a database. Thus, upon upgrading to Cloudera Manager 5, you must specify Activity Monitor database information. You have the option to use the embedded PostgreSQL database, which Cloudera Manager can set up automatically.

**Before You Begin**

**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.

Perform the following *before* upgrading to Cloudera Manager 5:

- **Upgrade Cloudera Manager 3.7.x to Cloudera Manager 4** - See [Upgrading Cloudera Manager 3.7.x](#) on page 500.
- **Upgrade all CDH 3 clusters to CDH 4** - See [Upgrading CDH 3](#) on page 690. If you attempt to upgrade to Cloudera Manager 5 and Cloudera Manager 4 is managing a CDH 3 cluster, the Cloudera Manager 5 server will not start, and you will be notified that you must downgrade to Cloudera Manager 4. For instructions for downgrading, see [Reverting a Failed Cloudera Manager Upgrade](#) on page 500. After downgrading, you must upgrade your CDH 3 cluster to CDH 4 before you can upgrade Cloudera Manager. See [Upgrading CDH 3](#) on page 690.
- **Obtain host credentials** - For Cloudera Manager to upgrade the Agent packages, you must have SSH access and be able to log in using a root account or an account that has password-less sudo permission. See [Cloudera Manager 5 Requirements and Supported Versions](#) on page 10 for more information.
- **Prepare databases** - See [Database Considerations for Cloudera Manager Upgrades](#) on page 467.

## Upgrade

- Cloudera Manager 5 supports HDFS high availability only with automatic failover. If your cluster has enabled high availability without automatic failover, you must enable automatic failover before upgrading to Cloudera Manager 5. See [Configuring HDFS High Availability](#).

### Stop Selected Services and Roles

If your cluster meets any of the conditions listed in the following table, you must stop the indicated services or roles.

Condition	Procedure
Running a version of Cloudera Manager that has the Cloudera Management Service	Stop the Cloudera Management Service.
Upgrading from Cloudera Manager 4.5 and higher, and using the embedded PostgreSQL database for the Hive metastore	Stop the services that have a dependency on the Hive metastore (Hue, Impala, and Hive). You cannot stop the Cloudera Manager Server database while these services are running. If you attempt to upgrade while the embedded database is running, the upgrade fails. Stop services that depend on the Hive metastore in the following order: <ol style="list-style-type: none"><li>1. Stop the Hue and Impala services.</li><li>2. Stop the Hive service.</li></ol>
Running Cloudera Navigator	Stop any of the following roles whose service's Queue Policy configuration ( <code>navigator.batch.queue_policy</code> ) is set to SHUTDOWN: <ul style="list-style-type: none"><li>• <b>HDFS</b> - NameNode</li><li>• <b>HBase</b> - Master and RegionServers</li><li>• <b>Hive</b> - HiveServer2</li><li>• <b>Hue</b> - Beeswax Server</li></ul> Stopping these roles renders any service depending on these roles unavailable. For <b>HDFS</b> - NameNode, this implies most of the services in the cluster will be unavailable until the upgrade is finished.

### Stop Cloudera Manager Server, Database, and Agent

1. Use the Admin Console to stop any running commands. These include commands a user runs and commands Cloudera Manager automatically triggers in response to a state change or a schedule. You can either wait for commands to complete or abort any running commands. For more information on viewing and aborting running commands, see [Viewing Running and Recent Commands](#). If you do not stop all commands, the Cloudera Manager Server will fail to start after upgrade.
2. On the host running the Cloudera Manager Server, stop the Cloudera Manager Server:

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

3. If you are using the embedded PostgreSQL database for Cloudera Manager, stop the database:

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



**Important:** If you are *not* running the embedded database service and you attempt to stop it, you receive a message indicating that the service cannot be found. If instead you get a message that the shutdown failed, the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

- RHEL-compatible 7 and higher:

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

- All other Linux distributions:

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

#### 4. If the Cloudera Manager host is also running the Cloudera Manager Agent, stop the Cloudera Manager Agent:

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

#### (Optional) Upgrade the JDK on Cloudera Manager Server and Agent Hosts

If you are manually upgrading the Cloudera Manager Agent software in [Upgrade and Start Cloudera Manager Agents \(Packages\)](#) on page 491 or [Install Cloudera Manager Server and Agent Software \(Tarballs\)](#) on page 487, and you are upgrading to CDH 5, install the Oracle JDK on the Agent hosts as described in [Java Development Kit Installation](#) on page 78.

If you are not running Cloudera Manager Server on the same host as a Cloudera Manager Agent, and you want all hosts to run the same JDK version, optionally install the Oracle JDK on that host.

#### Upgrade Cloudera Manager Software

Choose a procedure based on how you installed Cloudera Manager:

##### Upgrade Cloudera Manager Server (Packages)

1. To upgrade the Cloudera Manager Server packages, you can upgrade from the Cloudera repository at <https://archive.cloudera.com/cm5/>, or you can create your own repository, as described in [Understanding Custom Installation Solutions](#) on page 170. You must create your own repository if you are upgrading a cluster that does not have Internet access.

- a. Find the Cloudera repo file for your distribution by starting at <https://archive.cloudera.com/cm5/> and navigating to the directory that matches your operating system.

For example, for Red Hat or CentOS 6, you would go to

[https://archive.cloudera.com/cm5/redhat/6/x86\\_64/cm/](https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/). In that directory, find the repo file that contains information including the repository base URL and GPG key. The contents of the `cloudera-manager.repo` are similar to the following:

```
[cloudera-manager]
# Packages for Cloudera Manager, Version 5, on RHEL or CentOS 6 x86_64
name=Cloudera Manager
baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/
gpgkey = https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

## Upgrade

For Ubuntu or Debian systems, go to the appropriate release directory, for example, <https://archive.cloudera.com/cm4/debian/wheezy/amd64/cm>. The repo file, in this case, cloudera.list, is similar to the following:

```
# Packages for Cloudera Manager, Version 5, on Debian 7.0 x86_64
deb https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm wheezy-cm5 contrib
deb-src https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm wheezy-cm5 contrib
```

- b. Replace the repo file in the configuration location for the package management software for your system.

Operating System	Commands
RHEL	Copy cloudera-manager.repo to /etc/yum.repos.d/.
SLES	Copy cloudera-manager.repo to /etc/zypp/repos.d/.
Ubuntu or Debian	Copy cloudera.list to /etc/apt/sources.list.d/.

- c. Run the following commands:

Operating System	Commands
RHEL	\$ sudo yum clean all \$ sudo yum upgrade cloudera-manager-server cloudera-manager-daemons cloudera-manager-server-db-2 cloudera-manager-agent
SLES	\$ sudo zypper clean --all \$ sudo zypper up -r <a href="https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/">https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/</a> To download from your own repository: \$ sudo zypper clean --all \$ sudo zypper rr cm \$ sudo zypper ar -t rpm-md <a href="http://myhost.example.com/path_to_cm_repo/cm">http://myhost.example.com/path_to_cm_repo/cm</a> \$ sudo zypper up -r <a href="http://myhost.example.com/path_to_cm_repo">http://myhost.example.com/path_to_cm_repo</a>
Ubuntu or Debian	The following commands clean cached repository information and update Cloudera Manager components: \$ sudo apt-get clean \$ sudo apt-get update \$ sudo apt-get dist-upgrade \$ sudo apt-get install cloudera-manager-server cloudera-manager-daemons cloudera-manager-server-db-2 cloudera-manager-agent  During this process, you may be prompted about your configuration file version: Configuration file `/etc/cloudera-scm-agent/config.ini' ==> Modified (by you or by a script) since installation. ==> Package distributor has shipped an updated version. What would you like to do about it? Your options are: Y or I : install the package maintainer's version N or O : keep your currently-installed version D : show the differences between the versions Z : start a shell to examine the situation The default action is to keep your current version.

Operating System	Commands
	You will receive a similar prompt for /etc/cloudera-scm-server/db.properties. Answer <b>N</b> to both prompts.

2. If you customized /etc/cloudera-scm-agent/config.ini, your customized file is moved to a file with the extension .rpmsave or .dpkg-old. Merge any customizations into /etc/cloudera-scm-agent/config.ini installed by the package manager.

You should now have the following packages, corresponding to the version of Cloudera Manager you installed, on the host that will be the Cloudera Manager Server host.

OS	Packages
RPM-based distributions	\$ rpm -qa 'cloudera-manager-*' cloudera-manager-repository-5.0-1.noarch cloudera-manager-server-5.7.2-0.cm572.p0.16.el6.x86_64 cloudera-manager-server-db-2-5.7.2-0.cm572.p0.16.el6.x86_64 cloudera-manager-agent-5.7.2-0.cm572.p0.16.el6.x86_64 cloudera-manager-daemons-5.7.2-0.cm572.p0.16.el6.x86_64
Ubuntu or Debian	~# dpkg-query -l 'cloudera-manager-*' Desired=Unknown/Install/Remove/Purge/Hold   Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend / Err?=(none)/Reinst-required (Status,Err: uppercase=bad)    / Name Version Description ++- ii cloudera-manager-agent 5.7.2-0.cm572.p0.16~sq The Cloudera Manager Agent ii cloudera-manager-daemon 5.7.2-0.cm572.p0.16~sq Provides daemons for monitoring Hadoop and related tools. ii cloudera-manager-server 5.7.2-0.cm572.p0.16~sq The Cloudera Manager Server

You may also see an entry for the cloudera-manager-server-db-2 if you are using the embedded database, and additional packages for plug-ins, depending on what was previously installed on the server host. If the cloudera-manager-server-db-2 package is installed, and you do not plan to use the embedded database, you can remove this package.

#### Install Cloudera Manager Server and Agent Software (Tarballs)

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
```

Extract the contents of the tarball, to this directory. 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 be extracted 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 are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in [Single User Mode Requirements](#) on page 17.

## Upgrade

### Create Users

The Cloudera Manager Server and managed services require 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. This created user, is used automatically after installation is complete.

To create user `cloudera-scm`, use a command such as the following:

```
$ sudo useradd --system --home=/opt/cloudera-manager/cm-5.6.0/run/cloudera-scm-server  
--no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm
```

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-5.6.0/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 installation has a `var` subdirectory where state is stored. In a non-tarball installation, state is stored in `/var`. Cloudera recommends that you reconfigure the tarball installation 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 installation 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 do not reuse the state directory between different tarball installations, duplicate Cloudera Manager Agent entries can occur in the Cloudera Manager database.

### Configuring for a Custom Cloudera Manager User and Custom Directories

You can change the default username and directories used by Cloudera Manager. If you do not change the default, skip to [Cloudera Manager and Managed Service Datastores](#) on page 79. By default, Cloudera Manager creates the following directories in `/var/log` and `/var/lib`:

- `/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 username and custom directories for Cloudera Manager, you must create these directories on the Cloudera Manager Server host and assign ownership of these directories to the custom username. Cloudera Manager installer makes no changes to any directories that already exist. Cloudera Manager cannot write to any existing directories for which it does not have proper permissions, and if you do not change ownership, Cloudera Management Service roles may not perform as expected. To resolve these issues, do one of the following:

- **Change ownership of existing directories:**

1. Use the `chown` command to change ownership of all existing directories to the Cloudera Manager user. If the Cloudera Manager username and group are `cloudera-scm`, to change the ownership of the headlamp log directory, you issue a command similar to the following:

```
$ sudo chown -R cloudera-scm:cloudera-scm /var/log/cloudera-scm-headlamp
```

- **Use alternate directories:**

1. If the directories you plan to use do not exist, create them. For example, to create `/var/cm_logs/cloudera-scm-headlamp` for use by the `cloudera-scm` user, you can use the following commands:

```
mkdir /var/cm_logs/cloudera-scm-headlamp
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.

## Start Cloudera Manager Server

Choose a procedure based on how you installed Cloudera Manager:

### Start the Cloudera Manager Server (Packages)

On the Cloudera Manager Server host (the system on which you installed the `cloudera-manager-server` package) do the following:

1. If you are using the embedded PostgreSQL database for Cloudera Manager, start the database:

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

## Upgrade

### 2. Start the Cloudera Manager Server:

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

You should see the following:

```
[Starting cloudera-scm-server: [ OK ]]
```

If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.

#### Start the Cloudera Manager Server (Tarballs)

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 this case, you have the 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`.

### Upgrade and Start Cloudera Manager Agents

Choose a procedure based on how you installed Cloudera Manager:

#### Upgrade and Start Cloudera Manager Agents (Packages)



**Important:** All hosts in the cluster must have access to the Internet if you use `archive.cloudera.com` as the source for installation files. If you do not have Internet access, create a custom repository.

1. Log in to the Cloudera Manager Admin Console.
2. Upgrade hosts using one of the following methods:

- **Cloudera Manager installs Agent software**

1. Select **Yes, I would like to upgrade the Cloudera Manager Agent packages now** and click **Continue**.
2. Select the release of the Cloudera Manager Agent to install. Normally, this is the **Matched Release for this Cloudera Manager Server**. However, if you used a custom repository (instead of `archive.cloudera.com`) for the Cloudera Manager server, select **Custom Repository** and provide the required information. The custom repository allows you to use an alternative location, but that location must contain the matched Agent version.
3. Click **Continue**. The JDK Installation Options page displays.
  - Leave **Install Oracle Java SE Development Kit (JDK)** checked to allow Cloudera Manager to install the JDK on each cluster host, or uncheck if you plan to install it yourself.
  - If local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, check the **Install Java Unlimited Strength Encryption Policy Files** checkbox.

Click **Continue**.

4. Specify credentials and initiate Agent installation:

- Select **root** or enter the username for an account that has password-less sudo permission.
- Select an authentication method:
  - If you choose password authentication, enter and confirm the password.
  - If you choose public-key authentication, provide a passphrase and path to the required key files.
- You can 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.

5. Click **Continue**. The Cloudera Manager Agent packages are installed.
6. Click **Continue**. The Host Inspector runs to inspect your managed hosts for correct versions and configurations. If there are problems, you can make changes and then rerun the inspector. When you are satisfied with the inspection results, click **Finish**.

- **Manually install Agent software**

1. On all cluster hosts except the Cloudera Manager Server host, stop the Agent:

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

## Upgrade

2. In the Cloudera Admin Console, select **No, I would like to skip the agent upgrade now** and click **Continue**.
3. Copy the appropriate repo file as described in [Upgrade Cloudera Manager Server \(Packages\)](#) on page 485.
4. Run the following commands:

Operating System	Commands
RHEL	<pre>\$ sudo yum clean all \$ sudo yum upgrade cloudera-manager-server cloudera-manager-daemons cloudera-manager-server-db-2 cloudera-manager-agent</pre> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <b>Note:</b><ul style="list-style-type: none"><li>• <code>yum clean all</code> cleans <code>yum</code> cache directories, ensuring that you download and install the latest versions of the packages.</li><li>• If your system is not up to date, any underlying system components must be upgraded before <code>yum update</code> can succeed. <code>yum</code> indicates which components must be upgraded.</li></ul></div>
SLES	<pre>\$ sudo zypper clean --all \$ sudo zypper up -r https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/</pre> <p>To download from your own repository:</p> <pre>\$ sudo zypper clean --all \$ sudo zypper rr cm \$ sudo zypper ar -t rpm-md http://myhost.example.com/path_to_cm_repo/cm \$ sudo zypper up -r http://myhost.example.com/path_to_cm_repo</pre>
Ubuntu or Debian	<p>Use the following commands to clean cached repository information and update Cloudera Manager components:</p> <pre>\$ sudo apt-get clean \$ sudo apt-get update \$ sudo apt-get dist-upgrade \$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons</pre> <p>During this process, you may be prompted about your configuration file version:</p> <pre>Configuration file `/etc/cloudera-scm-agent/config.ini' ==&gt; Modified (by you or by a script) since installation. ==&gt; Package distributor has shipped an updated version. What would you like to do about it? Your options are: Y or I : install the package maintainer's version N or O : keep your currently-installed version D : show the differences between the versions Z : start a shell to examine the situation The default action is to keep your current version.</pre> <p>You will receive a similar prompt for <code>/etc/cloudera-scm-server/db.properties</code>. Answer <b>N</b> to both prompts.</p>

5. If you customized `/etc/cloudera-scm-agent/config.ini`, your customized file is moved to a file with the extension `.rpmsave` or `.dpkg-old`. Merge any customizations into `/etc/cloudera-scm-agent/config.ini` installed by the package manager.
6. On all cluster hosts, start the Agent:

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

3. If you are upgrading from a free version of Cloudera Manager prior to 4.6:

a. Click **Continue** to assign the Cloudera Management Services roles to hosts.

b. If you are upgrading to Cloudera Enterprise, specify required databases:

a. Choose the database type:

- Keep the default setting of **Use Embedded Database** to have Cloudera Manager create and configure required databases. Record 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:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

✓ Skipped. Cloudera Manager will create this database in a later step.

Database Name :

hive

Username:

hive

Password:

t56iwbdk4F

✓ Successful

##### Reports Manager

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

rman

Username:

rman

Password:

Y6S4IWvIno

✓ Successful

##### Navigator Audit Server

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

nav

Username:

nav

Password:

QLR2B0qqQ9

✓ Successful

##### Navigator Metadata Server

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

Database Name :

navms

Username:

navms

Password:

imo07jxOen

✓ Successful

##### Oozie Server

Currently assigned to run on tcdn2-1.ent.cloudera.com.

Database Host Name:

tcdn2-1.ent.cloudera.com:7432

Database Type:

PostgreSQL

✓ Skipped. Cloudera Manager will create this database in a later step.

Database Name :

oozie\_oozie\_se

Username:

oozie\_oozie\_se

Password:

NTF1KNdP1

Test Connection

- Select **Use Custom Databases** to specify external database host, enter the database type, database name, username, and password for the database that you created when you set up the database.
- If you are adding the Oozie service, you can change your Oozie configuration to control when data is purged in order to improve performance, cut down on database disk usage, improve upgrade performance, or to keep the history for a longer period of time. See [Configuring Oozie Data Purge Settings Using Cloudera Manager](#).

- 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 **Cluster Setup Review Changes** screen displays.

#### 4. Click Finish.

5. If you are upgrading from Cloudera Manager prior to 4.5:

- Select the host for the Hive Metastore Server role.
- Review the configuration values and click **Accept** to continue.



### Note:

- If Hue was using a Hive metastore backed by a Derby database, the newly created Hive Metastore Server also uses Derby. Because Derby does not allow concurrent connections, Hue continues to work, but the new Hive Metastore Server does not run. The failure is harmless (because nothing uses this new Hive Metastore Server at this point) and intentional, to preserve the set of cluster functionality as it was before upgrade. Cloudera discourages the use of a Derby-backed Hive metastore due to its limitations and recommends switching to a different supported database.
- Prior to Cloudera Manager 4.5, Hue and Impala connected directly to the Hive metastore database, so the bypass mode is enabled by default when upgrading to Cloudera Manager 4.5 and higher. This ensures that the upgrade does not disrupt your existing setup. You should plan to disable the bypass mode, especially when using CDH 4.2 and higher. Using the Hive Metastore Server is the recommended configuration, and the WebHCat Server role requires the Hive Metastore Server to *not* be bypassed. To disable bypass mode, see [Disabling Bypass Mode](#). After changing this configuration, you must redeploy your client configurations, restart Hive, and restart any Hue or Impala services configured to use that Hive.
- If you are using CDH 4.0 or CDH 4.1, see known issues related to Hive in [Known Issues and Workarounds in Cloudera Manager 5](#).

6. If you are upgrading from Cloudera Manager 4.5 and lower, correct the Hive home directory permissions (/user/hive) as follows:

```
sudo -u hdfs hdfs dfs -chown hive:hive /user/hive/
sudo -u hdfs hdfs dfs -chmod 1775 /user/hive/
```

7. If you are upgrading from Cloudera Manager prior to 4.8 and have an Impala service, assign the Impala Catalog Server role to a host.
8. Review the configuration changes to be applied.
9. Click **Finish**.

All services (except for the services you stopped in [Stop Selected Services and Roles](#) on page 484) should be running.

[Restart Cloudera Manager Agents \(Tarballs\)](#)

[Stop Cloudera Manager Agents \(Tarballs\)](#)

- To stop the Cloudera Manager Agent, run this command on each Agent host:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent stop
```

- If you are running [single user mode](#), stop Cloudera Manager Agent using the user account you chose. For example, if you are running the Cloudera Manager Agent as `cloudera-scm`, you have the following options:

- Run the following command:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent stop
```

- Edit the configuration files so the script internally changes the user, and 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 stop
```

### *Start Cloudera Manager Agents (Tarballs)*

Start the Cloudera Manager Agent according to the 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, to run the Cloudera Manager Agent as `cloudera-scm`, you have the 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, and 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`.

## Upgrade

### Verify the Upgrade Succeeded

If the commands to update and start the Cloudera Manager Server complete without errors, you can assume the upgrade has completed successfully. To verify, you can check that the server versions have been updated.

1. In the Cloudera Manager Admin Console, click the **Hosts** tab.
2. Click **Host Inspector**. On large clusters, the host inspector may take some time to finish running. You must wait for the process to complete before proceeding to the next step.
3. Click **Show Inspector Results**. All results from the host inspector process are displayed, including the currently installed versions. If this includes listings of current component versions, the installation completed as expected.

### Add Hive Gateway Roles

If you are upgrading from a release prior to Cloudera Manager 4.5, add Hive gateway roles to any hosts where Hive clients should run.

1. Go to the Hive service.
2. Click the **Instances** tab.
3. Click the **Add Role Instances** button.
4. Select the hosts on which you want a Hive gateway role to run. This ensures that the Hive client configurations are deployed on these hosts.

### Configure Cluster Version for Package Installs

If you have installed CDH as a package, after an installation or upgrade, make sure that the cluster CDH version matches the package CDH version, using the procedure in [Configuring the CDH Version of a Cluster](#) on page 524. If the cluster CDH version does not match the package CDH version, Cloudera Manager incorrectly enables and disables service features based on the cluster's configured CDH version.

### Upgrade Impala

If your version of Impala is 1.1 and lower, upgrade to Impala 1.2.1 and higher.

### (Optional) Configure TLS/SSL for Cloudera Management Service

If you have enabled TLS security for the Cloudera Manager Admin Console, as of Cloudera Manager 5.1, Cloudera Management Service roles try to communicate with Cloudera Manager using TLS, and fail to start until TLS/SSL properties have been configured. Configure Cloudera Management Service roles to communicate with Cloudera Manager over TLS/SSL as follows:

1. Do one of the following:
  - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
  - On the **Home > Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select **Scope > Cloudera Management Service (Service-Wide)**.
4. Select **Category > Security**.
5. Edit the following TLS/SSL properties according to your cluster configuration.

Property	Description
<b>TLS/SSL Client Truststore File Location</b>	Path to the client truststore file used in HTTPS communication. The contents of this truststore can be modified without restarting the Cloudera Management Service roles. By default, changes to its contents are picked up within ten seconds.
<b>TLS/SSL Client Truststore File Password</b>	Password for the client truststore file.

6. Click **Save Changes** to commit the changes.

## 7. Restart the Cloudera Management Service.

For more information, see [HTTPS Communication in Cloudera Manager](#).

### (Optional) Deploy a Cloudera Manager Agent Upgrade

Several conditions require you to upgrade the Cloudera Manager Agent:

- Deploying a fix to an issue where Cloudera Manager did not always correctly restart services
- Using the maximum file descriptor feature
- Enabling HDFS DataNodes to start if you perform the step [\(Optional\) Upgrade CDH](#) on page 499 after upgrading Cloudera Manager

To deploy the Cloudera Manager Agent upgrade, perform the following steps:

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 `hard_restart` command by reading [Hard Stopping and Restarting Agents](#).
  - Packages
    - RHEL-compatible 7 and higher:

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

- All other Linux distributions:

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

- Tarballs

- To stop the Cloudera Manager Agent, run this command on each Agent host:
  - RHEL-compatible 7 and higher:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard
$ sudo tarball_root/etc/init.d/cloudera-scm-agent restart
```

- All other Linux distributions:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent hard_restart
```

- If you are running [single user mode](#), start Cloudera Manager Agent using the user account you chose. For example to run the Cloudera Manager Agent as `cloudera-scm`, you have the following options:

- Run the following command:

- RHEL-compatible 7 and higher:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent restart
```

- All other Linux distributions:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent hard_restart
```

- Edit the configuration files so the script internally changes the user, and then run the script as root:

1. Remove the following line from `tarball_root/etc/default/cloudera-scm-agent`:

```
export CMF_SUDO_CMD= " "
```

## Upgrade

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:

- RHEL-compatible 7 and higher:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard  
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent restart
```

- All other Linux distributions:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent hard_restart
```

3. Start all services.

### Deploy a JDK Upgrade

If you upgraded the JDK when installing the Cloudera Manager Agents, do the following:

1. If the Cloudera Manager Server host is also running a Cloudera Manager Agent, restart the Cloudera Manager Server:

- Packages

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

- Tarballs

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent restart
```

If the Cloudera Manager Server does not start, see [Troubleshooting Installation and Upgrade Problems](#) on page 742.

2. If you have not restarted services in previous steps, restart all services:

- a. On the **Home > Status** tab, click



next to the cluster name and select **Restart**.

- b. In the confirmation dialog box, click **Restart**.

### (Optional) Deploy Monitoring Upgrade

Cloudera Manager 5 provides monitoring support for all roles. However, the Cloudera Manager Agent does not send monitoring data for these roles until:

1. The Cloudera Manager Agent has been upgraded and restarted.
2. The monitored roles have been restarted.

Until you restart the roles, some data is not in the monitoring charts and health tests. To enable monitoring for all roles, if you have not restarted services in previous steps, perform the following steps:

1. On the **Home > Status** tab, click



next to the cluster name and select **Restart**.

2. In the confirmation dialog box that displays, click **Restart**.

#### [Start Selected Services and Roles](#)

Start services and roles you shut down in [Stop Selected Services and Roles](#) on page 484 that have not been started in other steps:

1. If you are not upgrading CDH, do the following:

- a. If you are running Cloudera Navigator, start the following roles of audited services:

- **HDFS** - NameNode
- **HBase** - Master and RegionServers
- **Hive** - HiveServer2
- **Hue** - Beeswax Server

- b. On the **Home > Status** tab, click



next to the name of each service you shut down and select **Start**.

- c. In the confirmation dialog box, click **Start**.

2. On the **Home > Status** tab, click



next to the Cloudera Management Service and select **Start**.

3. In the confirmation dialog box, click **Start**.

#### [Deploy Updated Client Configurations](#)

The services for which client configurations require redeployment are indicated with the icon on the **Home > Status** tab. To ensure clients have current information about resources, update the client configuration:

1. On the **Home > Status** tab, click



next to the cluster name and select **Deploy Client Configuration**.

2. In the confirmation dialog box, click **Deploy Client Configuration**.

#### [Test the Installation](#)

When you have finished the upgrade to Cloudera Manager, you can test the installation to verify that the monitoring features are working as expected; follow the instructions in [Testing the Installation](#) on page 198.

#### [\(Optional\) Upgrade CDH](#)

Cloudera Manager 5 can manage both CDH 4 and CDH 5, so upgrading existing CDH 4 installations is not required, but you may want to upgrade to the latest version. For more information on upgrading CDH, see [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524.

### Upgrading Cloudera Manager 3.7.x



#### Warning:

- Cloudera Manager 4 and CDH 4 have reached End of Maintenance (EOM) on August 9, 2015. Cloudera does not support or provide updates for Cloudera Manager 4 and CDH 4 releases.
- Cloudera Manager 3 and CDH 3 have reached End of Maintenance (EOM) on June 20, 2013. Cloudera does not support or provide updates for Cloudera Manager 3 and CDH 3 releases.

You cannot upgrade directly from Cloudera Manager 3.7.x to Cloudera Manager 5; you must upgrade to Cloudera Manager 4 first before upgrading to Cloudera Manager 5. Follow the instructions for upgrading Cloudera Manager 3.7.x to Cloudera Manager 4 in [Upgrade Cloudera Manager 3.7.x to the Latest Cloudera Manager](#).

The last step in the Cloudera Manager upgrade process is an optional step to upgrade CDH. If you are running CDH 3, this step is *not optional*. Cloudera Manager 5 does not support CDH 3 and will not allow you to complete the upgrade if it detects a CDH 3 cluster. *You must upgrade to CDH 4 before you can upgrade to Cloudera Manager 5.* Follow the steps in [Upgrading CDH 3 to CDH 4 in a Cloudera Manager Deployment](#) before you attempt to upgrade to Cloudera Manager 5.

### Re-Running the Cloudera Manager Upgrade Wizard

#### Minimum Required Role: [Full Administrator](#)

The first time you log in to the Cloudera Manager server after upgrading your Cloudera Manager software, the upgrade wizard runs. If you did not complete the wizard at that time, or if you had hosts that were unavailable at that time and still need to be upgraded, you can re-run the upgrade wizard:

1. Click the **Hosts** tab.
2. Click **Re-run Upgrade Wizard**. This takes you back through the installation wizard to upgrade Cloudera Manager Agents on your hosts as necessary.
3. Select the release of the Cloudera Manager Agent to install. Normally, this is the **Matched Release for this Cloudera Manager Server**. However, if you used a custom repository (instead of archive.cloudera.com) for the Cloudera Manager server, select **Custom Repository** and provide the required information. The custom repository allows you to use an alternative location, but that location must contain the matched Agent version.
4. Specify credentials and initiate Agent installation:
  - Select **root** or enter the username for an account that has password-less sudo permission.
  - Select an authentication method:
    - If you choose password authentication, enter and confirm the password.
    - If you choose public-key authentication, provide a passphrase and path to the required key files.
  - You can 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.

When you click **Continue** the Cloudera Manager Agent is upgraded on all the currently managed hosts. You cannot search for new hosts through this process. To add hosts to your cluster, click the **Add New Hosts to Cluster** button.

### Reverting a Failed Cloudera Manager Upgrade

If you have a CDH 3 cluster running under Cloudera Manager 4, you cannot upgrade to Cloudera Manager 5 because it does not support CDH 3. Likewise, an upgrade from Cloudera Manager 3 to Cloudera Manager 5 is not supported. In either case, the Cloudera Manager 5 server will not start, and you must now downgrade your Cloudera Manager server, back to the version you were using prior to attempting the upgrade.



**Important:** The following instructions assume that a Cloudera Manager upgrade failed, and that the upgraded server never started, so that the remaining steps of the upgrade process were not performed. The steps below are not sufficient to revert from a running Cloudera Manager 5 deployment.

## Reinstall the Cloudera Manager Server Packages

In this step, you install the Cloudera Manager Server packages to the version you were running previously. You must reinstall the same version of Cloudera Manager you were using previously, so that the version of your Cloudera Manager Agents match the server.

The steps below assume that the Cloudera Manager Server is already stopped (as it failed to start after the attempted upgrade).

1. If you are using the embedded PostgreSQL database for Cloudera Manager, stop the database on the Cloudera Manager Server host:

- RHEL-compatible 7 and higher:

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

- All other Linux distributions:

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

2. Reinstall the same Cloudera Manager Server version that you were previously running. You can reinstall from the Cloudera repository at <https://archive.cloudera.com/cm4/> or <https://archive.cloudera.com/cm5/> or alternately, you can create your own repository, as described in [Understanding Custom Installation Solutions](#) on page 170.

- a. Find the Cloudera repo file for your distribution by starting at <https://archive.cloudera.com/cm4/> or <https://archive.cloudera.com/cm5/> and navigating to the directory that matches your operating system.

For example, for RHEL or CentOS 6, you would go to [https://archive.cloudera.com/cm5/redhat/6/x86\\_64/cm/](https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/). Within that directory, find the repo file that contains information including the repository's base URL and GPG key. On CentOS 6, the contents of the `cloudera-manager.repo` file might appear as follows:

```
[cloudera-manager]
# Packages for Cloudera Manager, Version 5, on RHEL or CentOS 6 x86_64
name=Cloudera Manager
baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/
gpgkey = https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

For Ubuntu or Debian systems, the repo file can be found by navigating to the appropriate directory, for example,

<https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm> <https://archive.cloudera.com/cm4/debian/squeeze/amd64/cm>

The repo file, in this case, `cloudera.list`, may appear as follows:

```
# Packages for Cloudera's Distribution for Hadoop, Version 4, on Debian 7.0 x86_64
deb https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm wheezy-cm5 contrib
deb-src https://archive.cloudera.com/cm5/debian/wheezy/amd64/cm wheezy-cm5 contrib
```

You must edit the file if it exist and modify the URL to reflect the exact version of Cloudera Manager you are using (unless you want the downgrade to also upgrade to the latest version of Cloudera Manager 4). The possible versions are shown in the directory on archive. Setting the URL (an example):

## Upgrade

OS	Command
RHEL	Replace baseurl=https://archive.cloudera.com/cm5/redhat/5/x86_64/cm/5/ with baseurl=https://archive.cloudera.com/cm5/redhat/5/x86_64/cm/5.0.5/
Ubuntu or Debian	Replace deb https://archive.cloudera.com/cm5/debian/squeeze/amd64/cm squeeze-cm5 contrib with deb https://archive.cloudera.com/cm5/debian/squeeze/amd64/cm squeeze-cm5.0.5 contrib

b. Copy the repo file to the configuration location for the package management software for your system:

Operating System	Commands
RHEL	Copy cloudera-manager.repo to /etc/yum.repos.d/.
SLES	Copy cloudera-manager.repo to /etc/zypp/repos.d/.
Ubuntu or Debian	Copy cloudera.list to /etc/apt/sources.list.d/.

c. Run the following commands:

Operating System	Commands
RHEL	\$ sudo yum downgrade 'cloudera-*'
SLES	\$ sudo zypper clean --all \$ sudo zypper dup -r https://archive.cloudera.com/cm4/sles/11/x86_64/cm/4/  To download from your own repository:  \$ sudo zypper clean --all \$ sudo zypper dup -r http://myhost.example.com/path_to_cm_repo
Ubuntu or Debian	There's no action that will downgrade to the version currently in the repository. Read <a href="#">DowngradeHowto</a> , download the script described therein, run it, and then run apt-get install for the name=version pairs that it provides for Cloudera Manager.

At the end of this process you should have the following packages, corresponding to the version of Cloudera Manager you installed, on the Cloudera Manager Server host. For example, for CentOS,

```
$ rpm -qa 'cloudera-manager-*'  
cloudera-manager-daemons-5.0.5-1.cm505.p0.163.el6.x86_64  
cloudera-manager-server-5.0.5-1.cm505.p0.163.el6.x86_64  
cloudera-manager-agent-5.0.5-1.cm505.p0.163.el6.x86_64
```

For Ubuntu or Debian, you should have packages similar to those shown below.

```
~# dpkg-query -l 'cloudera-manager-*'  
Desired=Unknown/Install/Remove/Purge/Hold  
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend  
/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)  
||/ Name Version Description  
=====  
ii  cloudera-manager-agent 5.0.5-1.cm505.p0.163~sq The Cloudera Manager Agent  
ii  cloudera-manager-daemon 5.0.5-1.cm505.p0.163~sq Provides daemons for monitoring Hadoop  
and related tools.  
ii  cloudera-manager-server 5.0.5-1.cm505.p0.163~sq The Cloudera Manager Server
```

You may also see an entry for the cloudera-manager-server-db if you are using the embedded database, and additional packages for plug-ins, depending on what was previously installed on the server host. If the commands to

update the server complete without errors, you can assume the upgrade has completed as desired. For additional assurance, you will have the option to check that the server versions have been updated after you start the server.

### Start the Server

On the Cloudera Manager Server host (the system on which you installed the `cloudera-manager-server` package) do the following:

1. If you are using the embedded PostgreSQL database for Cloudera Manager, start the database:

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

2. Start the server:

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

You should see the following:

```
Starting cloudera-scm-server: [ OK ]
```



**Note:** If you have problems starting the server, such as database permissions problems, you can use the server's log `/var/log/cloudera-scm-server/cloudera-scm-server.log` to troubleshoot the problem.

## Upgrading the Cloudera Navigator Data Management Component

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

### Upgrade Notes

- 2.4-2.6
  - Cloudera Navigator no longer supports JDK 1.6. If you are using JDK 1.6, you *must* upgrade to JDK 1.7 or 1.8. See [Java Development Kit Installation](#) on page 78.
  - If upgrading from Navigator 2.3 or lower, the Cloudera Navigator Metadata Server requires an upgrade of data in the storage directory. See [Upgrade Procedure](#) on page 504.
- 2.2 - Policies created with Cloudera Navigator 2.1 (containing the Beta version policy engine) are not retained when upgrading to Cloudera Navigator 2.2.
- 2.1 - When you upgrade to Cloudera Navigator 2.1 or higher from Navigator 2.0 or lower, the upgrade wizard adds a database for the Navigator Metadata Server. This is a different database than the currently existing Navigator Audit Server database.
- 2.0 - Cloudera does not provide an upgrade path from the Navigator Metadata Server which was a beta release in Cloudera Navigator 1.2 to the Cloudera Navigator 2 release. If you are upgrading from Cloudera Navigator 1.2 (included with Cloudera Manager 5.0), you must perform a clean install of Cloudera Navigator 2. Therefore, if you have Cloudera Navigator roles from a 1.2 release:
  1. Delete the Navigator Metadata Server role.
  2. Remove the contents of the Navigator Metadata Server storage directory.
  3. Add the Navigator Metadata Server role according to the process described in [Adding the Navigator Metadata Server](#).
  4. Clear the cache of any browser that had used the 1.2 release of the Navigator Metadata component. Otherwise, you may observe errors in the Navigator Metadata UI.

## Upgrade

### Upgrade Procedure

To upgrade the Cloudera Navigator data management component:

1. If upgrading from Navigator 2.3 or lower, do the following pre-upgrade steps:
  - a. Stop the Navigator Metadata Server role.
  - b. Back up the [Navigator Metadata Server storage directory](#).
  - c. [Make sure that the Navigator Metadata Server has sufficient memory](#) to complete the upgrade.
  - d. If you are using an Oracle database, in SQL\*Plus, ensure that the following additional privileges are set:

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

where `nav` is the user of the Navigator Audit Server database.

2. Upgrade Cloudera Manager following the instructions in [Upgrading Cloudera Manager](#) on page 466. For information on compatible Cloudera Navigator and Cloudera Manager versions, see the [Product Compatibility Matrix for Cloudera Navigator](#) product compatibility matrix.
3. If upgrading from Navigator 2.3 or lower, [start and log into the Cloudera Navigator data management component UI](#). The Upgrading Navigator page displays. Depending on the amount of data in the Navigator Metadata Server storage directory, the upgrade process can take up to *three to four hours*. When the upgrade is complete, the **Continue** button displays. Click **Continue**. The Cloudera Navigator landing page displays.

### Related Information

- [Cloudera Navigator 2 Overview](#)
- [Installing the Cloudera Navigator Data Management Component](#) on page 204
- [Upgrading Cloudera Navigator Key Trustee Server](#) on page 504
- [Upgrading Cloudera Navigator Key HSM](#) on page 519
- [Upgrading Cloudera Navigator Encrypt](#) on page 521
- [Cloudera Navigator Data Management Component Administration](#)
- [Cloudera Data Management](#)
- [Configuring Encryption](#)
- [Configuring Authentication in the Cloudera Navigator Data Management Component](#)
- [Configuring TLS/SSL for the Cloudera Navigator Data Management Component](#)
- [Cloudera Navigator Data Management Component User Roles](#)

## Upgrading Cloudera Navigator Key Trustee Server

Navigator Key Trustee Server 5.4.x is the first release that supports installation using Cloudera Manager. If you are using Cloudera Manager, you must upgrade Key Trustee Server to 5.4 or higher using the command line or the [ktupgrade script](#) before you can migrate Key Trustee Server to Cloudera Manager control.

To upgrade Key Trustee Server from 3.8 to 5.5 or higher, use the [ktupgrade script](#) to simplify the upgrade process.

### Upgrading Cloudera Navigator Key Trustee Server 3.x to 5.4.x

Navigator Key Trustee Server 5.4.x is the first release that supports installation using Cloudera Manager. If you are using Cloudera Manager, you must upgrade Key Trustee Server using the command line before you can migrate Key Trustee Server to Cloudera Manager control.

To upgrade Key Trustee Server to 5.5 or higher, see [Upgrading Cloudera Navigator Key Trustee Server 3.8 to 5.5 Using the ktupgrade Script](#) on page 509.



**Note:** Before upgrading Key Trustee Server, back up the Key Trustee Server database and configuration directory. See [Backing Up Key Trustee Server Manually](#) for instructions.

## Upgrading Key Trustee Server 3.x to 5.4.x Using the Command Line

The following instructions apply to both standalone and high availability Key Trustee Servers. For standalone Key Trustee Server, follow the instructions that refer to the *active* Key Trustee Server. For high availability Key Trustee Servers, follow the instructions on all Key Trustee Servers, unless otherwise indicated.

### Upgrade Key Trustee Server

1. Stop the `httpd` service:

```
$ sudo service httpd stop
```

### 2. Install the EPEL Repository

Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the `epel-release` package:

1. Copy the URL for the `epel-release-<version>.noarch` located at the bottom of the [EPEL 6](#) page.
2. Run the following commands to install the EPEL repository:

```
$ sudo wget <epel_rpm_url>
$ sudo yum install epel-release-<version>.noarch.rpm
```

Replace `<version>` with the version number of the downloaded RPM (for example, 6-8).

If the `epel-release` package is already installed, you see a message similar to the following:

```
Examining /var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
/var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: does not update installed package.
Error: Nothing to do
```

Confirm that the EPEL repository is installed:

```
$ sudo yum repolist | grep -i epel
```

### 3. Install the Cloudera Repository

Create or edit the `/etc/yum.repos.d/gazzang.repo` file (for example, `sudo vi /etc/yum.repos.d/gazzang.repo`) and add the following text. Replace `USER` and `PASSWD` with the username and password provided by Cloudera. If you do not know your username or password, contact your Cloudera account team.

```
[gazzang_stable]
name=RHEL $releasever - gazzang.com - base
baseurl=https://USER:PASSWD@archive.gazzang.com/redhat/stable/$releasever
enabled=1
gpgcheck=1
gpgkey=http://archive.gazzang.com/gpg_gazzang.asc
```



**Important:** If you are using CentOS, add the following line to the CentOS base repository:

```
exclude=python-psycopg2*
```

By default, the base repository is located at `/etc/yum.repos.d/CentOS-Base.repo`. If you have an internal mirror of the base repository, update the correct file for your environment.

## Upgrade

Import the GPG key by running the following command:

```
$ sudo rpm --import http://archive.gazzang.com/gpg_gazzang.asc
```

**4. Upgrade Key Trustee Server using yum:**

```
$ sudo yum update keytrustee-server python-keytrustee
```

**5. Start the httpd service:**

```
$ sudo service httpd start
```

### Migrate Apache Web Server to CherryPy



**Note:** Confirm that all ports listed in [Network Requirements](#) are open before proceeding.

For versions 5.4.0 and higher, Key Trustee Server uses CherryPy for the front end web interface; lower versions use the Apache web server. The CherryPy service is managed using the `keytrusteed` service. The Apache web server is managed using the `httpd` service. Run the following commands to migrate the web server from Apache to CherryPy.

**1. On the active Key Trustee Server, run the `ktadmin db --configure` command as follows:**

```
$ sudo -u keytrustee ktadmin db --configure --port 11381 --pg-rootdir /var/lib/keytrustee/db --slave keytrustee02.example.com
```

Replace `keytrustee02.example.com` with the hostname of the passive Key Trustee Server. For standalone Key Trustee Server, omit the `--slave keytrustee02.example.com` portion of the command.

**2. Export the active Key Trustee Server database. Run the following commands on the active Key Trustee Server:**

```
$ sudo -u postgres pg_dump keytrustee > /var/lib/keytrustee/ktdbexport.psql  
$ chown keytrustee:keytrustee /var/lib/keytrustee/ktdbexport.psql
```

**3. Start the Key Trustee Server database and import `ktdbexport.psql`:**

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db start --log /var/lib/keytrustee/db/pg_ctl.log  
$ sudo -u keytrustee /usr/pgsql-9.3/bin/createdb --host /tmp --port 11381 -O keytrustee keytrustee  
$ sudo -u keytrustee psql -d keytrustee -h /tmp -p 11381 < /var/lib/keytrustee/ktdbexport.psql
```



**Note:** The `/etc/init.d/postgresql` script does not work when the PostgreSQL database is started by Key Trustee Server, and cannot be used to monitor the status of the database. Use `/etc/init.d/keytrustee-db` instead.

**4. (High Availability Key Trustee Servers Only) Start the passive Key Trustee Server. Run the following commands on the passive Key Trustee Server:**

```
$ sudo -u keytrustee ktadmin --confdir /var/lib/keytrustee/.keytrustee init-slave --master keytrustee01.example.com --pg-rootdir /var/lib/keytrustee/db --no-import-key --master-host-port 11381 --logdir /var/lib/keytrustee/.keytrustee/logs --postgres-config=local --no-start  
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db start
```

5. Edit `/var/lib/keytrustee/.keytrustee/keytrustee.conf` on the active and passive Key Trustee Servers to reference the new database and port. Set the `DB_CONNECT` parameter as follows:

```
"DB_CONNECT": "postgresql://localhost:11381/keytrustee?host=/tmp",
```

6. Restart the Apache web server. Run this command on all Key Trustee Servers:

```
$ sudo service httpd restart
```

7. Start the Key Trustee daemon (which starts the CherryPy web server). Run this command on all Key Trustee Servers:

```
$ sudo /etc/init.d/keytrusted start
```

8. After verifying that the Key Trustee daemon and CherryPy web server are running, stop the Apache web server and original database and prevent them from starting after reboots. Run these commands on all Key Trustee Servers:

```
$ sudo service httpd stop
$ sudo -u postgres /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/pgsql/9.3/keytrustee stop
$ sudo chkconfig httpd off
$ sudo chkconfig postgresql-9.3 off
```

#### (High Availability Key Trustee Servers Only) Enable Synchronous Replication

Run the following command on the active Key Trustee Server to enable synchronous replication after upgrading:

```
$ sudo -u keytrustee ktadmin enable-synchronous-replication --pg-rootdir
/var/lib/keytrustee/db
```

#### Validating Key Operations

Verify that the upgrade was successful by running the following command on all Key Trustee Servers. The output should be similar to the following. If high availability is enabled, the output should be identical on all Key Trustee Servers:

```
$ curl -k https://keytrustee.example.com:11371/?a=fingerprint
4096R/4EDC46882386C827E20DEEA2D850ACA33BEDB0D1
```

Replace `keytrustee.example.com` with the fully qualified domain name (FQDN) of each Key Trustee Server you are validating.

If you are using Key Trustee Server as the backing key store for [HDFS Transparent Encryption](#), run the following commands to verify that Hadoop key operations are successful:

```
$ hadoop key create hadoop_test_key
$ hadoop key list
$ hadoop key delete hadoop_test_key
```

#### Migrating Unmanaged Key Trustee Server to Cloudera Manager



**Important:** If you are upgrading to Key Trustee Server 5.5 or higher without the [ktupgrade script](#), skip this step and continue to [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x to the Latest Release](#) on page 514.

For simplified and centralized administration, perform the following steps to move Key Trustee Server under Cloudera Manager control after upgrading Key Trustee Server:

1. Download the Key Trustee Server CSD from the following location:

```
https://archive.gazzang.com/parcels/cloudera/keytrustee-server/5.4.9/stable/latest/csd/
```

## Upgrade

When prompted, enter your credentials. If you do not know your credentials, contact your Cloudera account team.

2. Install the CSD into Cloudera Manager as instructed in [Custom Service Descriptor Files](#). The CSD can only be installed on parcel-deployed clusters.
3. Add the following parcel repository to Cloudera Manager following the instructions in [Configuring Cloudera Manager Server Parcel Settings](#) on page 63:

```
https://<username>:<password>@archive.gazzang.com/parcels/cloudera/keytrustee-server/5.4.9/stable/latest
```

Replace `<username>` and `<password>` with your credentials. If you do not know your credentials, contact your Cloudera account team.

4. **(Recommended)** Create a new cluster in Cloudera Manager containing only the hosts the Key Trustee Server will be installed on. Cloudera strongly recommends installing Key Trustee Server in a dedicated cluster to enable multiple clusters to share the same Key Trustee Server and to avoid restarting the Key Trustee Server when restarting a cluster. See [Adding and Deleting Clusters](#) for instructions on how to create a new cluster in Cloudera Manager.
5. Download, distribute, and activate the Key Trustee Server parcel, following the instructions in [Managing Parcels](#) on page 57. After you activate the Key Trustee Server parcel, Cloudera Manager prompts you to restart the cluster. Click the **Close** button to ignore this prompt. You *do not* need to restart the cluster after installing Key Trustee Server.
6. Stop the active and passive Key Trustee Server web servers using the command that corresponds to your backing web server. See [Migrate Apache Web Server to CherryPy](#) on page 506 for more information.

For Apache web servers:

```
$ sudo service httpd stop
```

For CherryPy web servers:

```
$ sudo service keytrusteed stop
```

7. Stop the active Key Trustee Server database. Run the following command on the active Key Trustee Server:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db stop
```



**Warning:** *Do not* stop the passive Key Trustee Server database. If it is stopped, start it before proceeding by running the following command on the passive Key Trustee Server:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db start
```

8. Add the Key Trustee Server service to your cluster, following the instructions in [Adding a Service](#). When customizing role assignments, assign the Active Key Trustee Server and Active Database roles to the active Key Trustee Server host, and the Passive Key Trustee Server and Passive Database roles to the passive Key Trustee Server host.
9. Stop the passive Key Trustee Server database. Run the following command on the passive Key Trustee Server:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db stop
```

10. Restart the Key Trustee Server service ([Key Trustee Server service > Actions > Restart](#)).



**Important:** Starting or restarting the Key Trustee Server service attempts to start the Active Database and Passive Database roles. If the Active Database is not running when the Passive Database attempts to start, the Passive Database fails to start. If this occurs, manually restart the Passive Database role after confirming that the Active Database role is running.

- 11 (High Availability Key Trustee Servers Only)** Enable synchronous replication. Run the following command on the active Key Trustee Server:

```
$ sudo -u keytrustee ktadmin enable-synchronous-replication --pg-rootdir /var/lib/keytrustee/db
```

### Updating Key Trustee Server Clients

After upgrading Key Trustee Server to 5.4 or higher, you must configure Key Trustee Server clients (namely Key Trustee KMS and Cloudera Navigator Encrypt) to communicate with Key Trustee Server over the new ports:

- **Key Trustee KMS**

Add the following entries to the Key Trustee KMS advanced configuration snippet (**Key Trustee KMS service > Configuration > Advanced > Key Management Server Advanced Configuration Snippet (Safety Valve) for kms-site.xml**):

```
<property>
    <name>cloudera.trustee.keyprovider.hkpport</name>
    <value>hkp_port_number</value>
    <description>
        Indicates the HTTP port on which Key Trustee Server clients should request public keys.
        On Key Trustee Server 3.8 (Apache webserver-based) servers, this is usually port 80 (unencrypted).
        On Key Trustee Server 5.4 and higher (CherryPy-based) servers, this is usually port 11371 (SSL-encrypted).
    </description>
</property>
<property>
    <name>cloudera.trustee.keyprovider.ktsport</name>
    <value>kts_port_number</value>
    <description>
        Indicates the HTTPS port on which the client sends and receives Key Trustee Server protocol messages.
        On Key Trustee Server 3.8 (Apache webserver-based) servers, this is usually port 443 (SSL-encrypted).
        On Key Trustee Server 5.4 and higher (CherryPy-based) servers, this is usually port 11371 (SSL-encrypted).
    </description>
</property>
<property>
    <name>cloudera.trustee.keyprovider.hkpssl</name>
    <value>boolean</value>
    <description>
        Indicates whether the client should communicate with the HKP server over an SSL-encrypted (true) or unencrypted (false) channel.
        On Key Trustee Server 3.8 (Apache webserver-based) servers, this is usually false (unencrypted).
        On Key Trustee Server 5.4 and higher (CherryPy-based) servers, this is usually true (SSL-encrypted).
    </description>
</property>
```

- **Cloudera Navigator Encrypt**

See [Updating Key Trustee Server Ports](#) for instructions on updating Cloudera Navigator Encrypt to use the new ports.

## Upgrading Cloudera Navigator Key Trustee Server 3.8 to 5.5 Using the ktupgrade Script

Cloudera provides a Python script (`ktupgrade`) to simplify upgrading Key Trustee Server 3.8 to 5.5. The script upgrades package-based Key Trustee Server 3.8 to package-based Key Trustee Server 5.5 and switches the web server from Apache to CherryPy. After the upgrade completes, you must manually migrate Key Trustee Server to use parcels and be managed by Cloudera Manager.

To upgrade from 3.x to 5.5 manually, you must first upgrade to 5.4, and then upgrade to 5.5:

## Upgrade

- [Upgrading Cloudera Navigator Key Trustee Server 3.x to 5.4.x](#) on page 504
- [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x to the Latest Release](#) on page 514

### Prerequisites



**Important:** The `ktupgrade` script supports upgrading from version 3.8.0 or 3.8.1 to version 5.5.0 or 5.5.2 only. To upgrade to a version higher than 5.5.2, use the `ktupgrade` script to upgrade to 5.5.2, and then follow the instructions in [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x to the Latest Release](#) on page 514 to upgrade to the version you want.

- Before upgrading Key Trustee Server, upgrade Cloudera Manager and CDH. See [Upgrading Cloudera Manager](#) on page 466 and [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524. If you are upgrading Key Trustee Server to a version higher than 5.5.2, you can upgrade Cloudera Manager and CDH directly to the version you want before continuing; you do not need to upgrade Cloudera Manager and CDH to 5.5 and complete the Key Trustee Server upgrade before upgrading Cloudera Manager and CDH to a higher version. The Cloudera Manager version must be equal to or higher than the Key Trustee Server version. See [Product Compatibility Matrix for Cloudera Navigator Encryption](#) for more information.
- If you are using [HDFS Transparent Encryption](#) with Key Trustee Server, upgrade Key Trustee KMS. See [Upgrading Key Trustee KMS](#) on page 520 for instructions.
- You must run the `ktupgrade` script as `root`.
- The `ktupgrade` script uses `yum` to upgrade Key Trustee Server. If the Key Trustee Server host does not have Internet access, you must download the Key Trustee Server dependencies from a host with Internet access and copy them to the Key Trustee Server host:

1. Create a temporary directory to store the packages:

```
$ mkdir tmp-keytrustee
```

2. Download the `bigtop-utils` package from the CDH repository:

```
$ sudo wget -P tmp-keytrustee <url>
```

Replace `<url>` with the URL corresponding to the Key Trustee Server version to which you are upgrading:

**Table 26: URL for `bigtop-utils` Package**

Key Trustee Server Version	URL
5.5.2	<a href="http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.5.2/RPM/noarch/bigtop-utils-0.7.0+cdh5.5.2+0-1.cdh5.5.2.p0.10.el6.noarch.rpm">http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.5.2/RPM/noarch/bigtop-utils-0.7.0+cdh5.5.2+0-1.cdh5.5.2.p0.10.el6.noarch.rpm</a>
5.5.0	<a href="http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.5.0/RPM/noarch/bigtop-utils-0.7.0+cdh5.5.0+0-1.cdh5.5.0.p0.15.el6.noarch.rpm">http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.5.0/RPM/noarch/bigtop-utils-0.7.0+cdh5.5.0+0-1.cdh5.5.0.p0.15.el6.noarch.rpm</a>

3. Download the `python-paste` and `python-cherrypy` packages:

```
$ sudo yum install yum-downloadonly
$ sudo yum install --downloadonly --downloaddir=tmp-keytrustee/ python-paste
python-cherrypy
```

4. Copy the packages to the Key Trustee Server host:

```
$ sudo scp tmp-keytrustee/*.rpm <username>@kts01.example.com:/path/to/tmp-keytrustee
```

Replace `kts01.example.com` with the hostname of the active Key Trustee Server, and `/path/to/tmp-keytrustee` with the path to a directory to which you have access.

## Download the ktupgrade Script and Repository Tarball

1. Download the ktupgrade script on the active Key Trustee Server host:

```
$ sudo wget http://archive.gazzang.com/keytrustee/ktupgrade
```

If the Key Trustee Server host does not have Internet access, run the command on an Internet-connected host, and then copy the file to the active Key Trustee Server host.

2. Download the repository tarball for Key Trustee Server [5.5.0](#) or [5.5.2](#):

- Select **Packages** from the **SELECT DOWNLOAD TYPE** drop-down menu.
- Select your operating system from the **SELECT AN OS** drop-down menu.
- Click **DOWNLOAD NOW**.
- Copy the downloaded file to the active Key Trustee Server host. Make sure you put the repository tarball and ktupgrade script in the same directory.

## Run the ktupgrade Script



**Important:** You must run the ktupgrade script as the `root` user. By default, the script upgrades the active Key Trustee Server, and then connects to the passive Key Trustee Server host as `root` over SSH (if you are using Key Trustee Server high availability) to upgrade it. You are prompted twice for the `root` password (first to copy the files, and then for the SSH connection).

If your environment does not allow `root` to log in over SSH, contact [Cloudera Support](#) for assistance.

## Upgrade the Active Key Trustee Server Using the ktupgrade Script

1. On the active Key Trustee Server host, change to the directory that contains the ktupgrade script and the repository tarball:

```
# cd /path/to/tmp-keytrustee
```

If the host does not have Internet access, make sure that the dependency files you downloaded in [Prerequisites](#) on page 510 are in the same directory as the script and tarball.

2. Make sure the script is executable:

```
# chmod a+x ktupgrade
```

3. Run the ktupgrade script as follows:

```
# ./ktupgrade upgrade-active-kts key-trustee-server-5.5.2-el6.tar.gz
```

Replace `key-trustee-server-5.5.2-el6.tar.gz` with the file name of the repository tarball you downloaded in [Download the ktupgrade Script and Repository Tarball](#) on page 511.

## Downgrade Key Trustee Server Using the ktupgrade Script

If you experience any problems upgrading Key Trustee Server, you can use the script to downgrade to your previous version. Run the following command on the active Key Trustee Server:

```
# cd /path/to/tmp-keytrustee
# ./ktupgrade downgrade-active-kts
```

## Migrate Key Trustee Server to Cloudera Manager

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

## Upgrade

Before continuing, you must create an internal repository for the Key Trustee Server parcel. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172.

After creating the internal Key Trustee Server parcel repository, do the following:

1. Create a new cluster in Cloudera Manager containing only the Key Trustee Server hosts. This enables multiple clusters to share the same Key Trustee Server and avoids restarting Key Trustee Server when restarting a cluster. See [Adding and Deleting Clusters](#) for instructions on how to create a new cluster in Cloudera Manager.
2. Download, distribute, and activate the Key Trustee Server parcel, following the instructions in [Managing Parcels](#) on page 57. After you activate the Key Trustee Server parcel, Cloudera Manager prompts you to restart the cluster. Click the **Close** button to ignore this prompt. You *do not* need to restart the cluster after installing Key Trustee Server.
3. Stop the active and passive Key Trustee Server web servers by running the following command on all Key Trustee Server hosts:

```
$ sudo -u keytrustee service keytrusteed stop
```

4. Stop the active Key Trustee Server database by running the following command on the active Key Trustee Server:

```
$ sudo -u keytrustee service keytrustee-db stop
```



**Warning:** *Do not* stop the passive Key Trustee Server database. If it is stopped, start it before proceeding by running the following command on the passive Key Trustee Server:

```
$ sudo -u keytrustee service keytrustee-db start
```

5. Add the Key Trustee Server service to your cluster, following the instructions in [Adding a Service](#). When customizing role assignments, assign the Active Key Trustee Server and Active Database roles to the active Key Trustee Server host, and the Passive Key Trustee Server and Passive Database roles to the passive Key Trustee Server host.
6. Stop the passive Key Trustee Server database. Run the following command on the passive Key Trustee Server:

```
$ sudo -u keytrustee service keytrustee-db stop
```

7. Restart the Key Trustee Server service (**Key Trustee Server service > Actions > Restart**).



**Important:** Starting or restarting the Key Trustee Server service attempts to start the Active Database and Passive Database roles. If the Active Database is not running when the Passive Database attempts to start, the Passive Database fails to start. If this occurs, manually restart the Passive Database role after confirming that the Active Database role is running.

8. **(High Availability Key Trustee Servers Only)** Enable synchronous replication. Run the following command on the active Key Trustee Server:

```
$ sudo -u keytrustee ktadmin enable-synchronous-replication --pg-rootdir /var/lib/keytrustee/db
```

### Validate Key Operations

Verify that the upgrade was successful by running the following command on all Key Trustee Servers. The output should be similar to the following. If high availability is enabled, the output should be identical on all Key Trustee Servers:

```
$ curl -k https://keytrustee.example.com:11371/?a=fingerprint  
4096R/4EDC46882386C827E20DEEA2D850ACA33BEDB0D1
```

Replace `keytrustee.example.com` with the fully qualified domain name (FQDN) of each Key Trustee Server you are validating.

If you are using Key Trustee Server as the backing key store for [HDFS Transparent Encryption](#), run the following commands to verify that Hadoop key operations are successful:

```
$ hadoop key create hadoop_test_key
$ hadoop key list
$ hadoop key delete hadoop_test_key
```

### Updating Key Trustee Server Clients

After upgrading Key Trustee Server to 5.4 or higher, you must configure Key Trustee Server clients (namely Key Trustee KMS and Cloudera Navigator Encrypt) to communicate with Key Trustee Server over the new ports:

- **Key Trustee KMS**

Add the following entries to the Key Trustee KMS advanced configuration snippet (**Key Trustee KMS service > Configuration > Advanced > Key Management Server Advanced Configuration Snippet (Safety Valve) for kms-site.xml**):

```
<property>
    <name>cloudera.trustee.keyprovider.hkpport</name>
    <value>hkp_port_number</value>
    <description>
        Indicates the HTTP port on which Key Trustee Server clients should request public keys.
        On Key Trustee Server 3.8 (Apache webserver-based) servers, this is usually port 80 (unencrypted).
        On Key Trustee Server 5.4 and higher (CherryPy-based) servers, this is usually port 11371 (SSL-encrypted).
    </description>
</property>
<property>
    <name>cloudera.trustee.keyprovider.ktsport</name>
    <value>kts_port_number</value>
    <description>
        Indicates the HTTPS port on which the client sends and receives Key Trustee Server protocol messages.
        On Key Trustee Server 3.8 (Apache webserver-based) servers, this is usually port 443 (SSL-encrypted).
        On Key Trustee Server 5.4 and higher (CherryPy-based) servers, this is usually port 11371 (SSL-encrypted).
    </description>
</property>
<property>
    <name>cloudera.trustee.keyprovider.hkpssl</name>
    <value>boolean</value>
    <description>
        Indicates whether the client should communicate with the HKP server over an SSL-encrypted (true) or unencrypted (false) channel.
        On Key Trustee Server 3.8 (Apache webserver-based) servers, this is usually false (unencrypted).
        On Key Trustee Server 5.4 and higher (CherryPy-based) servers, this is usually true (SSL-encrypted).
    </description>
</property>
```

- **Cloudera Navigator Encrypt**

See [Updating Key Trustee Server Ports](#) for instructions on updating Cloudera Navigator Encrypt to use the new ports.

### (Optional) Upgrade to a Higher Release

If you are upgrading Key Trustee Server to a version higher than 5.5.2, continue to [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x to the Latest Release](#) on page 514.

### Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x to the Latest Release

If you are upgrading Key Trustee Server from 3.8 to 5.5 or higher, see [Upgrading Cloudera Navigator Key Trustee Server 3.8 to 5.5 Using the ktupgrade Script](#) on page 509.



**Note:** Before upgrading Key Trustee Server, back up the Key Trustee Server. See [Backing Up and Restoring Key Trustee Server and Clients](#) for instructions.

#### Setting Up an Internal Repository

You must create an internal repository to install or upgrade the Cloudera Navigator data encryption components. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see the following topics:

- [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172
- [Creating and Using a Package Repository for Cloudera Manager](#) on page 174

#### Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x Using Cloudera Manager

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



**Note:** These instructions apply to using Cloudera Manager only. To upgrade Key Trustee Server using the command line, skip to the [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x Using the Command Line \(CherryPy Web Server\)](#) on page 514 or [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x Using the Command Line \(Apache Web Server\)](#) on page 516 section.

1. Add your internal parcel repository to Cloudera Manager following the instructions in [Configuring Cloudera Manager Server Parcel Settings](#) on page 63.
2. Download, distribute, and activate the latest Key Trustee Server parcel on the cluster containing the Key Trustee Server host, following the instructions in [Managing Parcels](#) on page 57.



**Important:** The KEYTRUSTEE parcel in Cloudera Manager is *not* the Key Trustee Server parcel; it is the Key Trustee KMS parcel. The parcel name for Key Trustee Server is KEYTRUSTEE\_SERVER.

After you activate the Key Trustee Server parcel, Cloudera Manager prompts you to restart the cluster. Click the **Close** button to ignore this prompt. You *do not* need to restart the cluster after installing Key Trustee Server.

3. **(High Availability Key Trustee Servers Only)** Enable synchronous replication. On the active Key Trustee Server, run the following command:

```
$ sudo ktadmin enable-synchronous-replication --pg-rootdir /var/lib/keytrustee/db
```

#### Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x Using the Command Line (CherryPy Web Server)



**Important:** Use these instructions only if you have previously [migrated Key Trustee Server](#) to use the CherryPy web server instead of the Apache web server. Otherwise, skip to [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x Using the Command Line \(Apache Web Server\)](#) on page 516.

The following instructions apply to both standalone and high availability Key Trustee Servers. For standalone Key Trustee Server, follow the instructions that refer to the *active* Key Trustee Server. For high availability Key Trustee Servers, follow the instructions on all Key Trustee Servers, unless otherwise indicated.

## Upgrade Key Trustee Server

### 1. Stop the keytrusteed service:

```
$ sudo service keytrusteed stop
```

### 2. Install the EPEL Repository

Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the `epel-release` package:

1. Copy the URL for the `epel-release-<version>.noarch` located at the bottom of the [EPEL 6](#) page.
2. Run the following commands to install the EPEL repository:

```
$ sudo wget <epel_rpm_url>
$ sudo yum install epel-release-<version>.noarch.rpm
```

Replace `<version>` with the version number of the downloaded RPM (for example, 6-8).

If the `epel-release` package is already installed, you see a message similar to the following:

```
Examining /var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
/var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: does not update installed package.
Error: Nothing to do
```

Confirm that the EPEL repository is installed:

```
$ sudo yum repolist | grep -i epel
```

### 3. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
```

### 4. Install the CDH Repository

Key Trustee Server and Key HSM depend on the `bigtop-utils` package, which is included in the CDH repository. For instructions on adding the CDH repository, see [To add the CDH repository](#). To create a local CDH repository, see [Creating a Local Yum Repository](#) on page 219 for instructions.

### 5. Upgrade Key Trustee Server:

```
$ sudo yum update keytrustee-server python-keytrustee
```

### 6. Start the keytrusteed service:

```
$ sudo service keytrusteed start
```

### (High Availability Key Trustee Servers Only) Enable Synchronous Replication

Run the following command on the active Key Trustee Server to enable synchronous replication after upgrading:

```
$ sudo ktadmin enable-synchronous-replication --pg-rootdir /var/lib/keytrustee/db
```

## Upgrade

### Migrate Key Trustee Server to Cloudera Manager

Skip to [Migrating Unmanaged Key Trustee Server to Cloudera Manager](#) on page 518 for instructions on migrating Key Trustee Server to Cloudera Manager control if you have not already done so during a previous upgrade.

### Upgrading Cloudera Navigator Key Trustee Server 5.4.x or 5.5.x Using the Command Line (Apache Web Server)



**Important:** Use these instructions only if you have *not yet* [migrated Key Trustee Server](#) to use the CherryPy web server instead of the Apache web server. The Apache web server is not supported in versions 5.5 and higher.

The following instructions apply to both standalone and high availability Key Trustee Servers. For standalone Key Trustee Server, follow the instructions that refer to the *active* Key Trustee Server. For high availability Key Trustee Servers, follow the instructions on all Key Trustee Servers, unless otherwise indicated.

### Migrate Apache Web Server to CherryPy



**Note:** Confirm that all ports listed in [Network Requirements](#) are open before proceeding.

For versions 5.4.0 and higher, Key Trustee Server uses CherryPy for the front end web interface; lower versions use the Apache web server. The Apache web server is not supported in versions 5.5 and higher. The CherryPy service is managed using the `keytrustee` service. The Apache web server is managed using the `httpd` service. Before upgrading, run the following commands to migrate the web server from Apache to CherryPy.

1. On the active Key Trustee Server, run the `ktadmin db --configure` command as follows:

```
$ sudo ktadmin db --configure --port 11381 --pg-rootdir /var/lib/keytrustee/db --slave keytrustee02.example.com
```

Replace `keytrustee02.example.com` with the hostname of the passive Key Trustee Server. For standalone Key Trustee Server, omit the `--slave keytrustee02.example.com` portion of the command.

If you use a database directory other than `/var/lib/keytrustee/db`, create or edit the `/etc/sysconfig/keytrustee-db` file and add the following:

```
ARGS="--pg-rootdir /path/to/db"
```

2. Export the Key Trustee Server database. Run the following commands on the active Key Trustee Server:

```
$ sudo -u postgres pg_dump keytrustee > /var/lib/keytrustee/ktedbexport.sql  
$ chown keytrustee:keytrustee /var/lib/keytrustee/ktedbexport.sql
```

3. Start the Key Trustee Server database and import `ktedbexport.sql`:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db start --log /var/lib/keytrustee/db/pg_ctl.log  
$ sudo -u keytrustee /usr/pgsql-9.3/bin/createdb --host /tmp --port 11381 -O keytrustee keytrustee  
$ sudo -u keytrustee psql -d keytrustee -h /tmp -p 11381 < /var/lib/keytrustee/ktedbexport.sql
```



**Note:** The `/etc/init.d/postgresql` script does not work when the PostgreSQL database is started by Key Trustee Server, and cannot be used to monitor the status of the database. Use `/etc/init.d/keytrustee-db` instead.

- 4. (High Availability Key Trustee Servers Only)** Start the passive Key Trustee Server. Run the following commands on the passive Key Trustee Server:

```
$ sudo ktadmin --confdir /var/lib/keytrustee/.keytrustee init-slave --master keytrustee01.example.com --pg-rootdir /var/lib/keytrustee/db --no-import-key --master-host-port 11381 --logdir /var/lib/keytrustee/.keytrustee/logs --postgres-config=local --no-start
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db start
```

If you use a database directory other than `/var/lib/keytrustee/db`, create or edit the `/etc/sysconfig/keytrustee-db` file and add the following:

```
ARGS="--pg-rootdir /path/to/db"
```

- 5.** Edit `/var/lib/keytrustee/.keytrustee/keytrustee.conf` on all Key Trustee Servers to reference the new database and port. Set the `DB_CONNECT` parameter as follows:

```
"DB_CONNECT": "postgresql://localhost:11381/keytrustee?host=/tmp",
```

- 6.** Restart the Apache web server. Run this command on all Key Trustee Servers:

```
$ sudo service httpd restart
```

- 7.** Start the Key Trustee daemon (which starts the CherryPy web server). Run this command on all Key Trustee Servers:

```
$ sudo service keytrusteed start
```

- 8.** After verifying that the Key Trustee daemon and CherryPy web server are running, stop the Apache web server and original database and prevent them from starting after reboots. Run these commands on all Key Trustee Servers:

```
$ sudo service httpd stop
$ sudo -u postgres /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/pgsql/9.3/keytrustee stop
$ sudo chkconfig httpd off
$ sudo chkconfig postgresql-9.3 off
```

## Upgrade Key Trustee Server

- 1.** Stop the `httpd` service:

```
$ sudo service httpd stop
```

## 2. Install the EPEL Repository

Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the `epel-release` package:

- 1.** Copy the URL for the `epel-release-<version>.noarch` located at the bottom of the [EPEL 6](#) page.
- 2.** Run the following commands to install the EPEL repository:

```
$ sudo wget <epel_rpm_url>
$ sudo yum install epel-release-<version>.noarch.rpm
```

Replace `<version>` with the version number of the downloaded RPM (for example, 6-8).

If the `epel-release` package is already installed, you see a message similar to the following:

```
Examining /var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
/var/tmp/yum-root-jmZhl0/epel-release-6-8.noarch.rpm: does not update installed package.
Error: Nothing to do
```

## Upgrade

Confirm that the EPEL repository is installed:

```
$ sudo yum repolist | grep -i epel
```

### 3. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
```

### 4. Upgrade Key Trustee Server:

```
$ sudo yum update keytrustee-server python-keytrustee
```

### 5. Start the httpd service:

```
$ sudo service httpd start
```

#### (High Availability Key Trustee Servers Only) Enable Synchronous Replication

Run the following command on the active Key Trustee Server to enable synchronous replication after upgrading:

```
$ sudo ktadmin enable-synchronous-replication --pg-rootdir /var/lib/keytrustee/db
```

#### Migrate Key Trustee Server to Cloudera Manager

Continue to [Migrating Unmanaged Key Trustee Server to Cloudera Manager](#) on page 518 for instructions on migrating Key Trustee Server to Cloudera Manager control if you have not already done so during a previous upgrade.

#### Migrating Unmanaged Key Trustee Server to Cloudera Manager

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

For simplified and centralized administration, perform the following steps to move Key Trustee Server under Cloudera Manager control after upgrading Key Trustee Server:

1. **(Recommended)** Create a new cluster in Cloudera Manager containing only the hosts the Key Trustee Server will be installed on. Cloudera strongly recommends installing Key Trustee Server in a dedicated cluster to enable multiple clusters to share the same Key Trustee Server and to avoid restarting the Key Trustee Server when restarting a cluster. See [Adding and Deleting Clusters](#) for instructions on how to create a new cluster in Cloudera Manager.
2. Download, distribute, and activate the Key Trustee Server parcel, following the instructions in [Managing Parcels](#) on page 57. After you activate the Key Trustee Server parcel, Cloudera Manager prompts you to restart the cluster. Click the **Close** button to ignore this prompt. You *do not* need to restart the cluster after installing Key Trustee Server.
3. Stop the active and passive Key Trustee Server web servers using the command that corresponds to your backing web server. See [Migrate Apache Web Server to CherryPy](#) on page 516 for more information.

For Apache web servers:

```
$ sudo service httpd stop
```

For CherryPy web servers:

```
$ sudo service keytrusteed stop
```

- Stop the active Key Trustee Server database. Run the following command on the active Key Trustee Server:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db stop
```



**Warning:** Do not stop the passive Key Trustee Server database. If it is stopped, start it before proceeding by running the following command on the passive Key Trustee Server:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db start
```

- Add the Key Trustee Server service to your cluster, following the instructions in [Adding a Service](#). When customizing role assignments, assign the Active Key Trustee Server and Active Database roles to the active Key Trustee Server host, and the Passive Key Trustee Server and Passive Database roles to the passive Key Trustee Server host.
- Stop the passive Key Trustee Server database. Run the following command on the passive Key Trustee Server:

```
$ sudo -u keytrustee /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/keytrustee/db stop
```

- Restart the Key Trustee Server service ([Key Trustee Server service > Actions > Restart](#)).



**Important:** Starting or restarting the Key Trustee Server service attempts to start the Active Database and Passive Database roles. If the Active Database is not running when the Passive Database attempts to start, the Passive Database fails to start. If this occurs, manually restart the Passive Database role after confirming that the Active Database role is running.

- (High Availability Key Trustee Servers Only) Enable synchronous replication. Run the following command on the active Key Trustee Server:

```
$ sudo ktadmin enable-synchronous-replication --pg-rootdir /var/lib/keytrustee/db
```

## Upgrading Cloudera Navigator Key HSM

### Setting Up an Internal Repository

You must create an internal repository to install or upgrade Cloudera Navigator Key HSM. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see [Creating and Using a Package Repository for Cloudera Manager](#) on page 174.

### Upgrading Key HSM



**Important:** If you have implemented Key Trustee Server high availability, upgrade Key HSM on each Key Trustee Server.

#### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
```

#### 2. Install the CDH Repository

## Upgrade

Key Trustee Server and Key HSM depend on the `bigtop-utils` package, which is included in the CDH repository. For instructions on adding the CDH repository, see [To add the CDH repository](#). To create a local CDH repository, see [Creating a Local Yum Repository](#) on page 219 for instructions.

### 3. Stop the Key HSM Service

Stop the Key HSM service before upgrading:

```
$ sudo service keyhsm shutdown
```

### 4. Upgrade Navigator Key HSM

Upgrade the Navigator Key HSM package using `yum`:

```
$ sudo yum update keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the `/usr/share/keytrustee-server-keyhsm` directory by default.

### 5. Start the Key HSM Service

Start the Key HSM service:

```
$ sudo service keyhsm start
```

## Upgrading Key Trustee KMS



**Important:** Following these instructions upgrades the software for the Key Trustee KMS service; this enables you to use Cloudera Navigator Key Trustee Server as the underlying keystore for [HDFS Transparent Encryption](#). This *does not* upgrade Key Trustee Server. See [Upgrading Cloudera Navigator Key Trustee Server](#) on page 504 for instructions on upgrading Key Trustee Server.

Key Trustee KMS is supported only in Cloudera Manager deployments. You can install the software using parcels or packages, but running Key Trustee KMS outside of Cloudera Manager is not supported.

## Setting Up an Internal Repository

You must create an internal repository to upgrade Key Trustee KMS. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172 if you are using parcels, or [Creating and Using a Package Repository for Cloudera Manager](#) on page 174 if you are using packages.

## Upgrading Key Trustee KMS Using Parcels



**Important:** Back up Key Trustee KMS before upgrading. See [Backing Up and Restoring Key Trustee Server and Clients](#) for instructions.

1. Go to **Hosts > Parcels**.
2. Click **Configuration** and add your internal repository to the **Remote Parcel Repository URLs** section. See [Configuring the Cloudera Manager Server to Use the Parcel URL](#) on page 174 for more information.
3. Click **Save Changes**.
4. Download, distribute, and activate the KEYTRUSTEE parcel for the version to which you are upgrading. See [Parcels](#) on page 55 for detailed instructions on using parcels to install or upgrade components.
5. Restart the Key Trustee KMS service (**Key Trustee KMS service > Actions > Restart**).

## Upgrading Key Trustee KMS Using Packages

- After [Setting Up an Internal Repository](#) on page 520, configure the Key Trustee KMS host to use the repository. See [Modifying Clients to Find the Repository](#) on page 176 for more information.
- Add the CDH repository. See [To add the CDH repository](#) for instructions. If you want to create an internal CDH repository, see [Creating a Local Yum Repository](#) on page 219.
- Upgrade 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
```

4. Restart the Key Trustee KMS service (**Key Trustee KMS service > Actions > Restart**).

## Upgrading Cloudera Navigator Encrypt

### Setting Up an Internal Repository

You must create an internal repository to install or upgrade the Cloudera Navigator data encryption components. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see the following topics:

- [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172
- [Creating and Using a Package Repository for Cloudera Manager](#) on page 174

### Upgrading Navigator Encrypt (RHEL-Compatible)



**Important:** Cloudera supports RHEL 7 with the following limitations:

- Only RHEL 7.2 and 7.1 are supported. RHEL 7.0 is not supported.
- Only new installations of RHEL 7.2 and 7.1 are supported by Cloudera. For upgrades to RHEL 7.1 or 7.2, contact your OS vendor and see [Does Red Hat support upgrades between major versions of Red Hat Enterprise Linux?](#)

#### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

#### 2. Stop Navigator Encrypt

Stop the Navigator Encrypt service:

```
$ sudo service navencrypt-mount stop
```

## Upgrade

For RHEL 7, use `systemctl` instead:

```
$ sudo systemctl stop navencrypt-mount
```

### 3. Upgrade Navigator Encrypt

Upgrade the Navigator Encrypt client using `yum`:

```
$ sudo yum update navencrypt
```

### 4. Start Navigator Encrypt

Start the Navigator Encrypt service:

```
$ sudo service navencrypt-mount start
```

For RHEL 7, use `systemctl` instead:

```
$ sudo systemctl start navencrypt-mount
```

## Upgrading Navigator Encrypt (SLES)

### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

### 2. Stop Navigator Encrypt

Stop the Navigator Encrypt service:

```
$ sudo service navencrypt-mount stop
```

### 3. Upgrade the Navigator Encrypt Client

Upgrade Navigator Encrypt:

```
$ sudo zypper update navencrypt
```

### 4. Enable Unsupported Modules

Edit `/etc/modprobe.d/unsupported-modules` and set `allow_unsupported_modules` to 1. For example:

```
#  
# Every kernel module has a flag 'supported'. If this flag is not set loading  
# this module will taint your kernel. You will not get much help with a kernel  
# problem if your kernel is marked as tainted. In this case you firstly have  
# to avoid loading of unsupported modules.  
#  
# Setting allow_unsupported_modules 1 enables loading of unsupported modules  
# by modprobe, setting allow_unsupported_modules 0 disables it. This can  
# be overridden using the --allow-unsupported-modules command line switch.  
allow_unsupported_modules 1
```

### 5. Start Navigator Encrypt

Start the Navigator Encrypt service:

```
$ sudo service navencrypt-mount start
```

## Upgrading Navigator Encrypt (Debian or Ubuntu)

### 1. Install the Cloudera Repository

Add the internal repository you created. See [Modifying Clients to Find the Repository](#) on page 176 for more information.

- **Ubuntu**

```
$ echo "deb http://repo.example.com/path/to/ubuntu/stable $DISTRIB_CODENAME main" | sudo tee -a /etc/apt/sources.list
```

- **Debian**

```
$ echo "deb http://repo.example.com/path/to/debian/stable $DISTRIB_CODENAME main" | sudo tee -a /etc/apt/sources.list
```

Import the GPG key by running the following command:

```
$ wget -O - http://repo.example.com/path/to/gpg_gazzang.asc | apt-key add -
```

Update the repository index with `apt-get update`.

### 2. Stop Navigator Encrypt

Stop the Navigator Encrypt service:

```
$ sudo service navencrypt-mount stop
```

### 3. Upgrade the Navigator Encrypt Client

Upgrade Navigator Encrypt:

```
$ sudo apt-get install navencrypt
```

### 4. Start Navigator Encrypt

Start the Navigator Encrypt service:

```
$ sudo service navencrypt-mount start
```

## Best Practices for Upgrading Navigator Encrypt Hosts

The following lists best practices for upgrading operating systems (OS) and kernels on hosts that have Navigator Encrypt installed:

- Make sure that the version you are upgrading to is supported by Navigator Encrypt. See [Product Compatibility Matrix for Cloudera Navigator Encryption](#) for more information.
- Always test upgrades in a development or testing environment before upgrading production hosts.
- If possible, upgrade the entire operating system instead of only upgrading the kernel.
- If you need to upgrade the kernel only, make sure that your OS version supports the kernel version to which you are upgrading.
- Always back up the `/etc/navencrypt` directory before upgrading. If you have problems accessing encrypted data after upgrading the OS or kernel, restore `/etc/navencrypt` from your backup and try again.

## Upgrading CDH and Managed Services Using Cloudera Manager



**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.

Cloudera Manager 5 supports clusters running CDH 4 and CDH 5. To ensure the highest level of functionality and stability, consider upgrading to the most recent version of CDH.



**Warning:** You can use Cloudera Manager to roll back an upgrade from CDH 4 to CDH 5 as long as you backup certain configuration files, databases, and other artifacts before beginning an upgrade. However, after you have [finalized the HDFS upgrade](#) you can no longer roll back the CDH upgrade. See [Rolling Back a CDH 4-to-CDH 5 Upgrade](#) on page 750 for the backup and rollback procedures.

The Cloudera Manager minor version must always be *equal to or greater than* the CDH minor version because older versions of Cloudera Manager may not support features in newer versions of CDH. For example, if you want to upgrade to CDH 5.4.8 you must first upgrade to Cloudera Manager 5.4 or higher.

Cloudera Manager 5.3 introduces an enhanced CDH upgrade wizard that supports major (CDH 4 to CDH 5), minor (CDH 5.x to 5.y), and maintenance upgrades (CDH a.b.x to CDH a.b.y). Both [parcels](#) and package installations are supported, but packages must be manually installed, whereas parcels are installed by Cloudera Manager. For an easier upgrade experience, consider switching from packages to parcels so Cloudera Manager can automate more of the process.

Depending on the nature of the changes in CDH between the old and new versions, the enhanced upgrade wizard performs service-specific upgrades that in the past you would have had to perform manually. When you start the wizard, notices regarding steps you must perform before upgrading to safeguard existing data that will be upgraded by the wizard.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

See the following topics for details on upgrading to the specific versions of CDH 4 or CDH 5.

### Configuring the CDH Version of a Cluster

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

Cloudera Manager has version-specific features based on the minor and maintenance versions. For example, the Sqoop 2 service and the HiveServer2 and WebHCat roles are only available for CDH 4.2.0 and higher. These behaviors are controlled by what is configured in Cloudera Manager, not what is actually installed on the hosts. The versions should be in sync, and is the case if parcels are used.

In package-based clusters, you can manually upgrade CDH packages. For example, you can upgrade the packages from CDH 4.1.0 to CDH 4.2.1. However, in previous releases Cloudera Manager did detect this change and behaved as if the cluster was 4.1.0. In such cases, it would not display Sqoop 2 as a service. You would have to set the CDH version manually using the cluster Configure CDH Version action.

Cloudera Manager now sets the CDH version correctly. However, if you had an older Cloudera Manager and forgot to set the version, and then upgraded to latest Cloudera Manager, you would need to set the version.

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

Cloudera Manager has version-specific features based on minor and maintenance versions. For example, the Sqoop 2 service and the HiveServer2 and WebHCat roles are only available for CDH 4.2.0 and higher. These behaviors are controlled by what is configured in Cloudera Manager, not what is actually installed on the hosts. The versions should be in sync, and is the case if parcels are used.

In package-based clusters, you can manually upgrade CDH packages. For example, you can upgrade the packages from CDH 4.1.0 to CDH 4.2.1. However, in previous releases, Cloudera Manager did not detect this change and behaved as

if the cluster was 4.1.0. In such cases, it would not display Sqoop 2 as a service. You would have to set the CDH version manually using the cluster Configure CDH Version action.

Cloudera Manager now sets the CDH version correctly. However, if you have an older Cloudera Manager and forgot to set the version, and then upgraded to latest Cloudera Manager, you need to set the version.

To inform Cloudera Manager of the CDH version, select **ClusterName > Configure CDH Version**. In the dialog box, Cloudera Manager displays the installed CDH version, and asks for confirmation to configure itself with the new version. The dialog box will also detect if a major upgrade was done, and direct you to use the major upgrade flow documented in [Upgrading from CDH 4 Packages to CDH 5 Packages](#) on page 672.

## Performing a Rolling Upgrade on a CDH 5 Cluster

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

### Performing a Rolling Upgrade to CDH 5.7

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.7 are as follows:

#### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Upgrade

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

### Back up HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Back up HDFS metadata using the following command:

```
hdfs dfsadmin -fetchImage local directory
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.
9. Choose the type of upgrade and restart:

- **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
  1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
  2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
- **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
  1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
  2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

  1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Continue**.

**11** Click **Finish** to return to the Home page.

#### Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 528. If any of the steps in the **Upgrade Cluster Command** screen fails, complete the steps as described in that section before proceeding.

#### Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- **Not including Impala and Search**

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

- **Including Impala and Search**

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

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:

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

## Upgrade

### Finalize HDFS Rolling Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.

3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

#### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

#### Start Cluster Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
  2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.
- When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

#### Deploy Client Configuration Files

1. On the Home page, click  to the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

#### Performing a Rolling Upgrade to CDH 5.6

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

## Upgrade

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.6 are as follows:

### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

### Back up HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Back up HDFS metadata using the following command:

```
hdfs dfsadmin -fetchImage local directory
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.

4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.

5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.

6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.

7. The selected parcels are downloaded and distributed. Click **Continue**.

8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.

9. Choose the type of upgrade and restart:

- **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
  1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
  2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
- **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
  1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
  2. Click **Continue**. The wizard reports the result of the upgrade.
- **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
  1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Continue**.

**11** Click **Finish** to return to the Home page.

#### Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 532. If any of the steps in the **Upgrade Cluster Command** screen fails, complete the steps as described in that section before proceeding.

#### Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:
  - **Not including Impala and Search**

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

- **Including Impala and Search**

Operating System	Command
RHEL	\$ sudo yum remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc
SLES	\$ sudo zypper remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc

## Upgrade

Operating System	Command
Ubuntu or Debian	\$ sudo apt-get purge 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc

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

### Finalize HDFS Rolling Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

#### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.

3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

#### [Upgrade the Hive Metastore Database](#)

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### [Upgrade the Oozie ShareLib](#)

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### [Upgrade Sqoop](#)

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### [Upgrade the Sentry Database](#)

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

#### [Upgrade Spark](#)

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

#### [Start Cluster Services](#)

1. On the **Home > Status** tab, click



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

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

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

## Upgrade

### Deploy Client Configuration Files

1. On the Home page, click



ot

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Performing a Rolling Upgrade to CDH 5.5

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.5 are as follows:

#### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

#### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

#### Back up HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Back up HDFS metadata using the following command:

```
hdfs dfsadmin -fetchImage local directory
```

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.
9. Choose the type of upgrade and restart:
  - **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
    1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
    2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
  - **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
    1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
    2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Continue**.

**11** Click **Finish** to return to the Home page.

## Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 537. If any of the steps in the **Upgrade Cluster Command** screen fails, complete the steps as described in that section before proceeding.

## Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

## Upgrade

- Not including Impala and Search

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

- Including Impala and Search

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

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:

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

### Finalize HDFS Rolling Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

## Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

## Upgrade

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Performing a Rolling Upgrade to CDH 5.4

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.4 are as follows:

#### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

## Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

## Back up HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Back up HDFS metadata using the following command:

```
hdfs dfsadmin -fetchImage local directory
```

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.
9. Choose the type of upgrade and restart:

- **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
  1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
  2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
- **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
  1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
  2. Click **Continue**. The wizard reports the result of the upgrade.
- **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
  1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

## Upgrade

**10** Click **Continue**.

**11** Click **Finish** to return to the Home page.

### Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 541. If any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:
  - Not including Impala and Search

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

- Including Impala and Search

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

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

### Finalize HDFS Rolling Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher

- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

#### Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

##### [Back up Metastore Databases](#)

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

#### [Upgrade HDFS Metadata](#)

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

#### [Upgrade the Hive Metastore Database](#)

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### [Upgrade the Oozie ShareLib](#)

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

## Upgrade

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



to the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Performing a Rolling Upgrade to CDH 5.3

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.3 are as follows:

### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

### Back up HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Back up HDFS metadata using the following command:

```
hdfs dfsadmin -fetchImage local directory
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.

4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the [Modify the Remote Parcel Repository URLs](#) link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.

5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.

6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.

7. The selected parcels are downloaded and distributed. Click **Continue**.

8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.

9. Choose the type of upgrade and restart:

## Upgrade

- **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
  1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
  2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
- **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
  1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
  2. Click **Continue**. The wizard reports the result of the upgrade.
- **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
  1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Continue**.

**11** Click **Finish** to return to the Home page.

### Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 628. If any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:
  - **Not including Impala and Search**

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

- **Including Impala and Search**

Operating System	Command
RHEL	\$ sudo yum remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc
SLES	\$ sudo zypper remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc

Operating System	Command
Ubuntu or Debian	\$ sudo apt-get purge 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc

- 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.

### Finalize HDFS Rolling Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

- Go to the HDFS service.
- Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

#### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

- For each affected service:
  - If not already stopped, stop the service.
  - Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

- Start the ZooKeeper service.
- Go to the HDFS service.

## Upgrade

3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.  
When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

## Deploy Client Configuration Files

1. On the Home page, click



ot

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Performing a Rolling Upgrade to CDH 5.2

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.2 are as follows:

### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

### Back up HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Back up HDFS metadata using the following command:

```
hdfs dfsadmin -fetchImage local directory
```

## Upgrade

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.
9. Choose the type of upgrade and restart:
  - **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
    1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
    2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
  - **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
    1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
    2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Continue**.

**11** Click **Finish** to return to the Home page.

### Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 550. If any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- Not including Impala and Search

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

- Including Impala and Search

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

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.

### Finalize HDFS Rolling Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

## Upgrade

### Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

#### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

## Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

## Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

## Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Performing a Rolling Upgrade to CDH 5.1

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between maintenance and minor versions of CDH 5, *except Beta versions*. For rolling upgrade between CDH 4 versions, see [Performing a Rolling Upgrade on a CDH 4 Cluster](#) on page 554.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

The steps to perform a rolling upgrade of a cluster to CDH 5.1 are as follows:

### Before You Begin

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).

## Upgrade

### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**. The Choose Upgrade Procedure displays.
9. Choose the type of upgrade and restart:
  - **Rolling Restart (Default)** - Cloudera Manager upgrades services and performs a rolling restart.
    1. Choose the batch size (default 1) described in [Performing a Cluster-Level Rolling Restart](#) and click **Advanced Options** to specify additional rolling restart options. Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.
    2. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down services, activates the new parcel, upgrades services as necessary, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.
  - **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster, during which services are not available.
    1. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services.
    2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual Upgrade** - Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.
10. Click **Continue**.
11. Click **Finish** to return to the Home page.

### Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 553. If any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

## Remove CDH 5 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- Not including Impala and Search

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

- Including Impala and Search

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

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:

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

## Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

### Back up Metastore Databases

Back up the Sqoop metastore database.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

## Upgrade

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Restart All Services

1. [Restart the cluster](#).

### Deploy Client Configuration Files

1. On the Home page, click  or the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Performing a Rolling Upgrade on a CDH 4 Cluster

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



**Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

The rolling upgrade feature takes advantage of parcels and the HDFS high availability to enable you to upgrade your cluster software and restart the upgraded services without taking the entire cluster down. You must have HDFS high availability enabled to perform a rolling upgrade.

This page described how to perform a rolling upgrade between minor versions of CDH 4. For rolling upgrade between CDH 5 versions, see [Performing a Rolling Upgrade on a CDH 5 Cluster](#) on page 525.

*It is not possible to perform a rolling upgrade from CDH 4 to CDH 5 because of incompatibilities between the two major versions. Instead, follow the instructions for a full upgrade at [Upgrading from CDH 4 to CDH 5 Parcels](#) on page 664.*

A rolling upgrade involves two steps:

1. Download, distribute, and activate the parcel for the new software you want to install.
2. Perform a [rolling restart](#) to restart the services in your cluster. You can do a rolling restart of individual services, or if you have high availability enabled, you can perform a restart of the entire cluster. Cloudera Manager will manually fail over your NameNode at the appropriate point in the process so that your cluster will not be without a functional NameNode.

To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration

validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

The steps to perform a rolling upgrade of a cluster are as follows:

#### Ensure High Availability is Enabled

To enable high availability, see [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you wish. Automatic failover does not affect the rolling restart operation. If you have JobTracker high availability configured, Cloudera Manager will fail over the JobTracker during the rolling restart, but this is not a requirement for performing a rolling upgrade.

#### Download, Distribute, and Activate Parcels

1. In the Cloudera Manager Admin Console, click the Parcels indicator in the top navigation bar ( or ) to go to the Parcels page.
2. In the parcels page, click **Download** for the version(s) you want to download. 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 parcel repos available through the [parcel settings](#) page. If your Cloudera Manager server does not have Internet access, you can obtain the required parcel file(s) and put them into the local repository. See [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172 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**. You are asked if you want to restart the cluster. *Do not restart the cluster at this time.*
6. Click **Close**.

#### Upgrade the Hive Metastore Database

Required if you are upgrading from an earlier version of CDH 4 to CDH 4.2 or higher.

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.

## Upgrade

3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Restart the Cluster

1. On the **Home > Status** tab, click



to the right of the cluster name and click **Rolling Restart** to proceed with a rolling restart. Rolling restart is available only if high availability is enabled. Click **Restart** to perform a normal restart. Services that do not support rolling restart will undergo a normal restart, and will not be available during the restart process.

2. For a rolling restart, a pop-up allows you to chose which services you want to restart, and presents caveats to be aware of for those services that can undergo a rolling restart.



**Note:** If you have just upgraded your Cloudera Manager deployment to 4.6, and are now doing a rolling upgrade of your cluster, you must ensure that MapReduce is restarted *before* the rest of your services, or the restart may fail. This is necessary to ensure the MapReduce configuration changes are propagated.

Further, if you are upgrading from CDH 4.1 with Impala to CDH 4.2 or 4.3, you must restart MapReduce before Impala restarts (by default Impala is restarted before MapReduce).

The workaround is to perform a restart of MapReduce alone as the first step, then perform a cluster restart of the remaining services.

3. Click **Confirm** to start the rolling restart.

### Remove CDH 4 Packages

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

1. If Hue is configured to use SQLite as its database:

- a. Stop the Hue service.
- b. Back up the desktop .db to a temporary location before deleting the old Hue Common package. The location of the database can be found in the Hue service **Configuration** tab under **Service > Database > Hue's Database Directory**.



**Important:** Removing the Hue Common package will remove your Hue database; if you do not back it up you may lose all your Hue user account information.

2. Uninstall the CDH packages on each host:

- **Not including Impala and Search**

Operating System	Command
RHEL	\$ sudo yum remove hadoop hue-common bigtop-jsvc bigtop-tomcat
SLES	\$ sudo zypper remove hadoop hue-common bigtop-jsvc bigtop-tomcat

Operating System	Command
Ubuntu or Debian	\$ sudo apt-get purge hadoop hue-common bigtop-jsvc bigtop-tomcat

- Including Impala and Search

Operating System	Command
RHEL	\$ sudo yum remove hadoop hue-common impala-shell solr-server 'bigtop-*'
SLES	\$ sudo zypper remove hadoop hue-common impala-shell solr-server 'bigtop-*'
Ubuntu or Debian	\$ sudo apt-get purge hadoop hue-common impala-shell solr-server 'bigtop-*'

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
```

#### Restore Hue Database

If you [removed CDH 4 packages](#), restore the Hue database back up.

1. Go to the Hue service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Copy the backup from the temporary location to the newly created Hue database directory:  
`/opt/cloudera/parcels/CDH-4.x.0-x.cdh4.x.0.p0.xx/share/hue/desktop`.
4. Restart the Hue service.

## Upgrading to CDH Maintenance Releases

Use the instructions in this section to upgrade to a CDH maintenance release, that is from CDH *a.b.x* to CDH *a.b.y*. For example, CDH 4.7.0 to CDH 4.7.1 or CDH 5.1.0 to 5.1.4.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.

## Upgrade

- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Upgrading to CDH Maintenance Releases Using Parcels

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

Use the instructions in this section to upgrade to a CDH maintenance release, that is from CDH *a.b.x* to CDH *a.b.y*. For example, CDH 4.7.0 to CDH 4.7.1 or CDH 5.1.0 to 5.1.4.

You can upgrade your cluster to another maintenance version using parcels from within the Cloudera Manager Admin Console. Your current CDH cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Back up Metastore Databases

Back up the Sqoop metastore database.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### ② Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds    [Retry](#)

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

- 10 Click **Finish** to return to the Home page.

## Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 559. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### [Upgrade Wizard Actions](#)

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure.

## Upgrade

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Restart the Services

1. On the Home page, click  the right of the cluster name and select **Restart**.
2. Click the **Restart** button in the confirmation pop-up that appears. The **Command Details** window shows the progress of starting services.

or

### Deploy Client Configuration Files

1. On the Home page, click  the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

or

### Upgrading to CDH Maintenance Releases Using Packages

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

Use the instructions in this section to upgrade to a CDH maintenance release, that is from CDH *a.b.x* to CDH *a.b.y*. For example, CDH 4.7.0 to CDH 4.7.1 or CDH 5.1.0 to 5.1.4.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

## Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

## Back up Metastore Databases

Back up the Sqoop metastore database.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

## Upgrade Managed Components

1. Download and save the repo file.

- On Red Hat-compatible systems:

Click the entry in the table below that matches your Red Hat or CentOS system, go 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	<a href="#">Red Hat/CentOS/Oracle 5 link</a>

## Upgrade

For OS Version	Click this Link
Red Hat/CentOS 6 (64-bit)	<a href="#">Red Hat/CentOS 6 link</a>

- On SLES systems:

1. Run the following command:

```
$ sudo zypper addrepo -f  
http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

- On Ubuntu and Debian systems:

Create a new file `/etc/apt/sources.list.d/cloudera.list` with the following contents:

- For Ubuntu systems:

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib deb-src http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib
```

- For Debian systems:

```
deb http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib deb-src  
http://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`.

2. Edit the repo file to point to the release you want to install or upgrade to.

- On Red Hat-compatible systems:

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=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/  
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera  
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

`baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/` to

`baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/`

In this example, the resulting file should look like this:

```
[cloudera-cdh5]  
name=Cloudera's Distribution for Hadoop, Version 5  
baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/  
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera  
gpgcheck = 1
```

- On SLES systems:

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=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

baseurl=http://archive.cloudera.com/cdh5/sles/11/x86\_64/cdh/5/ to

baseurl= http://archive.cloudera.com/cdh5/sles/11/x86\_64/cdh/5.1.0/

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.1.0/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

- On Ubuntu and Debian systems:

Replace -cdh5 near the end of each line (before contrib) with the CDH release you need to install. Here are examples using CDH 5.1.0:

#### **For 64-bit Ubuntu Precise:**

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh
precise-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh precise-cdh5.1.0
contrib
```

#### **For Debian Wheezy:**

```
deb http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
```

### **3. (Optionally) add a repository key:**

- Red Hat compatible
  - Red Hat/CentOS/Oracle 5

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

#### **– Red Hat/CentOS/Oracle 6**

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

- SLES

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

- Ubuntu and Debian

## Upgrade

### – Ubuntu Precise

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

### – Debian Wheezy

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

#### 4. Install the CDH packages:

- Red Hat compatible

```
$ sudo yum clean all  
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

- SLES

```
$ sudo zypper clean --all  
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

- Ubuntu and Debian

```
$ sudo apt-get update  
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs  
hadoop-kms hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr  
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

#### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. In the **Choose Method** field, select the **Use Packages** option.
4. In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.
7. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.

**8.** Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.
1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### ✖ Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds    [Retry](#)

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**9.** Click **Finish** to return to the Home page.

### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 565. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

## Upgrade

### Restart the Services

1. On the Home page, click



ot

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

2. Click the **Restart** button in the confirmation pop-up that appears. The **Command Details** window shows the progress of starting services.

### Deploy Client Configuration Files

1. On the Home page, click



ot

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading to CDH 5.7

Use the instructions in this section to upgrade from an earlier version of CDH 5 to CDH 5.7.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

- Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
- Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME", "TBLS". "TBL_NAME", "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Upgrading to CDH 5.7 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.7 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in [CDH 5.2](#), Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

## Upgrade

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
    INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
    INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
        AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
        AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Stop Cluster Services

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. On the **Home > Status** tab, click



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

2. 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.

### Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

- On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

- For each affected service:
  - If not already stopped, stop the service.
  - Back up the database. See [Backing Up Databases](#) on page 117.

### Run the Upgrade Wizard

- Log into the Cloudera Manager Admin console.

- From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

- If the option to pick between packages and parcels displays, select the **Use Parcels** option.
- In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
- Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
- Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
- The selected parcels are downloaded and distributed. Click **Continue**.
- The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
- Choose the type of upgrade and restart:
  - Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

- Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### ⌚ Upgrade Cluster Command

Status: Failed

Context: Cluster 1

Start Time: Sep 17, 6:29:40 PM

Duration: 113.01 seconds

Retry

## Upgrade

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

10 Click **Finish** to return to the Home page.

### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 571. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- **Not including Impala and Search**

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

- **Including Impala and Search**

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

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:

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

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.

## Upgrade

2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Upgrading to CDH 5.7 Using Packages

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

If you originally used Cloudera Manager to install CDH 5 using packages, you can upgrade to CDH 5.7 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

## Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME" , "TBLS". "TBL_NAME" , "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
  AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
  AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

## Upgrade

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

### Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

### Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn  
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./  
./current/  
./current/fsimage  
./current/fstime  
./current/VERSION  
./current/edits  
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### Back up Metastore Databases

Back up the Hive and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade Managed Components

Use *one* of the following strategies to upgrade CDH 5:

- Use the Cloudera "1-click Install" package. This is the simplest way to upgrade only the Cloudera packages.

1. Check whether you have the CDH 5 "1-click" repository installed.

- **Red Hat/CentOS-compatible and SLES**

```
rpm -q CDH 5-repository
```

If you are upgrading from CDH 5 Beta 1 or higher, and you used the "1-click" package for the previous CDH 5 release, you should see:

```
CDH5-repository-1-0
```

In this case, skip to installing the CDH 5 packages. If instead you see:

```
package CDH 5-repository is not installed
```

proceed with installing the 1-click package.

- **Ubuntu and Debian**

```
dpkg -l | grep CDH 5-repository
```

If the repository is installed, skip to installing the CDH 5 packages; otherwise proceed with installing the "1-click" package.

2. If the CDH 5 "1-click" repository is not already installed on each host in the cluster, follow the instructions below for that host's operating system.

- **Red Hat compatible**

1. Download and install the "1-click Install" package.

- a. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

- b. Install the RPM for all RHEL versions:

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

2. (Optionally) add a repository key:

- **Red Hat/CentOS/Oracle 5**

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

## Upgrade

- Red Hat/CentOS/Oracle 6

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

- SLES

1. Download and install the "1-click Install" package:

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

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

- b. Install the RPM:

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

- c. Update your system package index by running:

```
$ sudo zypper refresh
```

2. (Optional) add a repository key:

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

- Ubuntu and Debian

1. Download and install the "1-click Install" package:

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

- b. 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
```

2. (Optional) add a repository key:

- Ubuntu Trusty

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

- Ubuntu Precise

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

- **Debian Wheezy**

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

**3. Install the CDH packages:**

- **Red Hat compatible**

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- **SLES**

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- **Ubuntu and Debian**

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs
  hadoop-kms hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr
  solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

- Use your operating system's package management tools to update all packages to the latest version using standard repositories. This approach works well because it minimizes the amount of configuration required and uses the simplest commands. Be aware that this can take a considerable amount of time if you have not upgraded the system recently. To update all packages on your system, use the following command:

Operating System	Command
RHEL	\$ sudo yum update
SLES	\$ sudo zypper up
Ubuntu or Debian	\$ sudo apt-get upgrade

#### Update Symlinks for the Newly Installed Components

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
```

## Upgrade

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. In the **Choose Method** field, select the **Use Packages** option.
4. In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.
7. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
8. Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### ✖ Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds

**Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

9. Click **Finish** to return to the Home page.

### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 579. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

#### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

#### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

#### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

## Upgrade

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



ot

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading to CDH 5.6

Use the instructions in this section to upgrade from an earlier version of CDH 5 to CDH 5.6.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
    AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
    AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Upgrading to CDH 5.6 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.6 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

## Upgrade

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH **5.2**, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

- Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
- Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME", "TBLS". "TBL_NAME", "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Stop Cluster Services

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. In the Search field, search for "NameNode Data Directories" and note the value.
4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

## Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click  next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.
3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Choose the type of upgrade and restart:
  - **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

## Upgrade

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### ④ Upgrade Cluster Command

Status: **Failed** Context: [Cluster 1](#) ↗ Start Time: Sep 17, 6:29:40 PM Duration: 113.01 seconds **Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

10 Click **Finish** to return to the Home page.

#### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 585. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- **Not including Impala and Search**

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

- **Including Impala and Search**

Operating System	Command
RHEL	\$ sudo yum remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc
SLES	\$ sudo zypper remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc

Operating System	Command
Ubuntu or Debian	\$ sudo apt-get purge 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc

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:

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

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

## Upgrade

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.3 to 5.4 or higher
1. Go to the Hive service.
  2. Select **Actions > Stop** and click **Stop** to confirm.
  3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
  4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.

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

### Deploy Client Configuration Files

1. On the Home page, click  to the right of the cluster name and select **Deploy Client Configuration**.

ot

- Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading to CDH 5.6 Using Packages

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

If you originally used Cloudera Manager to install CDH 5 using packages, you can upgrade to CDH 5.6 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME" , "TBLS". "TBL_NAME" , "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
  AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
  AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the

## Upgrade

upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.

- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

### Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

### Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. In the Search field, search for "NameNode Data Directories" and note the value.
4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn  
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./  
./current/  
./current/fsimage  
./current/fstime  
./current/VERSION  
./current/edits  
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### [Back up Metastore Databases](#)

Back up the Hive and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### [Upgrade Managed Components](#)

Use *one* of the following strategies to upgrade CDH 5:

- Use the Cloudera "1-click Install" package. This is the simplest way to upgrade only the Cloudera packages.

1. Check whether you have the CDH 5 "1-click" repository installed.

- **Red Hat/CentOS-compatible and SLES**

```
rpm -q CDH 5-repository
```

If you are upgrading from CDH 5 Beta 1 or higher, and you used the "1-click" package for the previous CDH 5 release, you should see:

```
CDH5-repository-1-0
```

In this case, skip to installing the CDH 5 packages. If instead you see:

```
package CDH 5-repository is not installed
```

proceed with installing the 1-click package.

- **Ubuntu and Debian**

```
dpkg -l | grep CDH 5-repository
```

If the repository is installed, skip to installing the CDH 5 packages; otherwise proceed with installing the "1-click" package.

2. If the CDH 5 "1-click" repository is not already installed on each host in the cluster, follow the instructions below for that host's operating system.

- **Red Hat compatible**

1. Download and install the "1-click Install" package.

- a. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

## Upgrade

### b. Install the RPM for all RHEL versions:

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

#### 2. (Optional) add a repository key:

- Red Hat/CentOS/Oracle 5

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

- Red Hat/CentOS/Oracle 6

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

#### • SLES

##### 1. Download and install the "1-click Install" package:

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

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

- b. Install the RPM:

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

- c. Update your system package index by running:

```
$ sudo zypper refresh
```

#### 2. (Optional) add a repository key:

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

#### • Ubuntu and Debian

##### 1. Download and install the "1-click Install" package:

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

- b. 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
```

**2.** (Optionally) add a repository key:

- **Ubuntu Trusty**

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

- **Ubuntu Precise**

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

- **Debian Wheezy**

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

**3.** Install the CDH packages:

- **Red Hat compatible**

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- **SLES**

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- **Ubuntu and Debian**

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs
  hadoop-kms hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr
  solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

- Use your operating system's package management tools to update all packages to the latest version using standard repositories. This approach works well because it minimizes the amount of configuration required and uses the simplest commands. Be aware that this can take a considerable amount of time if you have not upgraded the system recently. To update all packages on your system, use the following command:

Operating System	Command
RHEL	\$ sudo yum update
SLES	\$ sudo zypper up

## Upgrade

Operating System	Command
Ubuntu or Debian	\$ sudo apt-get upgrade

### Update Symlinks for the Newly Installed Components

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
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. In the **Choose Method** field, select the **Use Packages** option.
4. In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.
7. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
8. Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### ✖ Upgrade Cluster Command

Status: Failed    Context: Cluster 1    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds

**Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

9. Click **Finish** to return to the Home page.

## Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 593. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.

## Upgrade

2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Upgrading to CDH 5.5

Use the instructions in this section to upgrade from an earlier version of CDH 5 to CDH 5.5.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

## Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME", "TBLS". "TBL_NAME", "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
  AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
  AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Upgrading to CDH 5.5 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.5 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

## Upgrade

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME", "TBLS". "TBL_NAME", "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
    INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
    INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
    INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
    INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
        AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
        AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Stop Cluster Services

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

## Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

## Upgrade

- From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

- If the option to pick between packages and parcels displays, select the **Use Parcels** option.
- In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
- Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
- Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
- The selected parcels are downloaded and distributed. Click **Continue**.
- The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
- Choose the type of upgrade and restart:
  - Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

- Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### ✖ Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds    [Retry](#)

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

- Click **Continue**. The wizard reports the result of the upgrade.
  - Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    - Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

- Click **Finish** to return to the Home page.

#### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 600. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

- 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.
- Uninstall the CDH packages on each host:
  - Not including Impala and Search**

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

- Including Impala and Search

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

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.

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

## Upgrade

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.2 or 5.3 to 5.4 or higher
1. Start the ZooKeeper service.
  2. Go to the HDFS service.
  3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.3 to 5.4 or higher
1. Go to the Hive service.
  2. Select **Actions > Stop** and click **Stop** to confirm.
  3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
  4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
  - CDH 5.2 to 5.3 or higher
  - CDH 5.4 to 5.5 or higher
1. Go to the Sentry service.
  2. Select **Actions > Stop** and click **Stop** to confirm.
  3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

## Start Cluster Services

1. On the Home > Status tab, click



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

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

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

## Deploy Client Configuration Files

1. On the Home page, click



or

to the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading to CDH 5.5 Using Packages

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

If you originally used Cloudera Manager to install CDH 5 using packages, you can upgrade to CDH 5.5 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

## Upgrade

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
    INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
    INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
        AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
        AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

### Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

### Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.2 or 5.3 to 5.4 or higher
1. Go to the HDFS service.

2. Click the **Configuration** tab.
3. In the Search field, search for "NameNode Data Directories" and note the value.
4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### Back up Metastore Databases

Back up the Hive and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade Managed Components

Use *one* of the following strategies to upgrade CDH 5:

- Use the Cloudera "1-click Install" package. This is the simplest way to upgrade only the Cloudera packages.
  1. Check whether you have the CDH 5 "1-click" repository installed.
    - Red Hat/CentOS-compatible and SLES

```
rpm -q CDH 5-repository
```

If you are upgrading from CDH 5 Beta 1 or higher, and you used the "1-click" package for the previous CDH 5 release, you should see:

```
CDH5-repository-1-0
```

In this case, skip to installing the CDH 5 packages. If instead you see:

```
package CDH 5-repository is not installed
```

proceed with installing the 1-click package.

- Ubuntu and Debian

```
dpkg -l | grep CDH 5-repository
```

If the repository is installed, skip to installing the CDH 5 packages; otherwise proceed with installing the "1-click" package.

## Upgrade

2. If the CDH 5 "1-click" repository is not already installed on each host in the cluster, follow the instructions below for that host's operating system.

- Red Hat compatible

1. Download and install the "1-click Install" package.

- a. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

- b. Install the RPM for all RHEL versions:

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

2. (Optionally) add a repository key:

- Red Hat/CentOS/Oracle 5

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

- Red Hat/CentOS/Oracle 6

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

- SLES

1. Download and install the "1-click Install" package:

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

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

- b. Install the RPM:

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

- c. Update your system package index by running:

```
$ sudo zypper refresh
```

2. (Optionally) add a repository key:

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

- **Ubuntu and Debian**

1. Download and install the "1-click Install" package:

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

- b. 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
```

2. (Optionally) add a repository key:

- **Ubuntu Trusty**

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

- **Ubuntu Precise**

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

- **Debian Wheezy**

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

3. Install the CDH packages:

- **Red Hat compatible**

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- **SLES**

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

## Upgrade

- **Ubuntu and Debian**

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httputfs
hadoop-kms hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

- Use your operating system's package management tools to update all packages to the latest version using standard repositories. This approach works well because it minimizes the amount of configuration required and uses the simplest commands. Be aware that this can take a considerable amount of time if you have not upgraded the system recently. To update all packages on your system, use the following command:

Operating System	Command
RHEL	\$ sudo yum update
SLES	\$ sudo zypper up
Ubuntu or Debian	\$ sudo apt-get upgrade

### Update Symlinks for the Newly Installed Components

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
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. In the **Choose Method** field, select the **Use Packages** option.
  4. In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.
  5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
  6. Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.
  7. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
  8. Choose the type of upgrade and restart:
    - **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.
1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### Upgrade Cluster Command

Status: **Failed** Context: [Cluster 1](#) Start Time: Sep 17, 6:29:40 PM Duration: 113.01 seconds [Retry](#)

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

9. Click **Finish** to return to the Home page.

### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 607. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.

## Upgrade

2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.

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

## Deploy Client Configuration Files

- On the Home page, click



ot

the right of the cluster name and select **Deploy Client Configuration**.

- Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading to CDH 5.4

Use the instructions in this section to upgrade from an earlier version of CDH 5 to CDH 5.4.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME", "TBLS". "TBL_NAME", "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.

## Upgrade

- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Upgrading to CDH 5.4 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.4 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME", "TBLS". "TBL_NAME", "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Stop Cluster Services

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is `/data/dfs/nm`, do the following as root:

```
# cd /data/dfs/nm
# tar -cvf /root/nm_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
```

## Upgrade

```
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.
  1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### ④ Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds    [Retry](#)

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

- 10 Click **Finish** to return to the Home page.

## Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 614. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- Not including Impala and Search

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

- Including Impala and Search

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

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

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

## Upgrade

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

#### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

#### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

## Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

## Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

## Start Cluster Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.  
When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

## Deploy Client Configuration Files

1. On the Home page, click  to the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading to CDH 5.4 Using Packages

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

If you originally used Cloudera Manager to install CDH 5 using packages, you can upgrade to CDH 5.4 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739 or [Upgrading to Oracle JDK 1.8](#) on page 740, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.

## Upgrade

- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH **5.2**, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

- Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
- Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME" , "TBLS". "TBL_NAME" , "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

## Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

## Back up Metastore Databases

Back up the Hive and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

## Upgrade Managed Components

Use *one* of the following strategies to upgrade CDH 5:

- Use the Cloudera "1-click Install" package. This is the simplest way to upgrade only the Cloudera packages.

1. Check whether you have the CDH 5 "1-click" repository installed.

- **Red Hat/CentOS-compatible and SLES**

```
rpm -q CDH 5-repository
```

## Upgrade

If you are upgrading from CDH 5 Beta 1 or higher, and you used the "1-click" package for the previous CDH 5 release, you should see:

```
CDH5-repository-1-0
```

In this case, skip to installing the CDH 5 packages. If instead you see:

```
package CDH 5-repository is not installed
```

proceed with installing the 1-click package.

- **Ubuntu and Debian**

```
dpkg -l | grep CDH 5-repository
```

If the repository is installed, skip to installing the CDH 5 packages; otherwise proceed with installing the "1-click" package.

2. If the CDH 5 "1-click" repository is not already installed on each host in the cluster, follow the instructions below for that host's operating system.

- **Red Hat compatible**

1. Download and install the "1-click Install" package.

- a. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

- b. Install the RPM for all RHEL versions:

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

2. (Optionally) add a repository key:

- **Red Hat/CentOS/Oracle 5**

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

- **Red Hat/CentOS/Oracle 6**

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

- **SLES**

1. Download and install the "1-click Install" package:

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

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

- Install the RPM:

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

- Update your system package index by running:

```
$ sudo zypper refresh
```

- (Optionally) add a repository key:

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

- Ubuntu and Debian**

- Download and install the "1-click Install" package:

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

- 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
```

- (Optionally) add a repository key:

- Ubuntu Trusty**

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

- Ubuntu Precise**

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

- Debian Wheezy**

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

- Install the CDH packages:

## Upgrade

- Red Hat compatible

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- SLES

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms
  hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce
  spark-python sqoop sqoop2 whirr zookeeper
```

- Ubuntu and Debian

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs
  hadoop-kms hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr
  solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

- Use your operating system's package management tools to update all packages to the latest version using standard repositories. This approach works well because it minimizes the amount of configuration required and uses the simplest commands. Be aware that this can take a considerable amount of time if you have not upgraded the system recently. To update all packages on your system, use the following command:

Operating System	Command
RHEL	\$ sudo yum update
SLES	\$ sudo zypper up
Ubuntu or Debian	\$ sudo apt-get upgrade

### Update Symlinks for the Newly Installed Components

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
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. In the **Choose Method** field, select the **Use Packages** option.

4. In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.
7. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
8. Choose the type of upgrade and restart:
  - **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### Upgrade Cluster Command

Status: Failed    Context: Cluster 1    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds    **Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.
    1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

9. Click **Finish** to return to the Home page.

### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 622. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.

## Upgrade

2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

#### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.2 or 5.3 to 5.4 or higher
1. Start the ZooKeeper service.
  2. Go to the HDFS service.
  3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

#### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.3 to 5.4 or higher
1. Go to the Hive service.
  2. Select **Actions > Stop** and click **Stop** to confirm.
  3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
  4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
  - CDH 5.2 to 5.3 or higher
  - CDH 5.4 to 5.5 or higher
1. Go to the Sentry service.
  2. Select **Actions > Stop** and click **Stop** to confirm.
  3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

#### Upgrade Spark

1. Go to the Spark service.

2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

#### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

#### Deploy Client Configuration Files

1. On the Home page, click



ot

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Upgrading to CDH 5.3

Use the instructions in this section to upgrade from an earlier version of CDH 5 to CDH 5.3.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

- Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
- Store dates in partition columns as strings or integers.

## Upgrade

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
    INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
    INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
        AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
        AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

### Upgrading to CDH 5.3 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.3 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

- Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
- Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
    AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
    AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.
- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Stop Cluster Services

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

## Upgrade

- On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn  
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./  
./current/  
./current/fsimage  
./current/fstime  
./current/VERSION  
./current/edits  
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

- For each affected service:
  - If not already stopped, stop the service.
  - Back up the database. See [Backing Up Databases](#) on page 117.

### Run the Upgrade Wizard

- Log into the Cloudera Manager Admin console.

- From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

- If the option to pick between packages and parcels displays, select the **Use Parcels** option.
- In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
- Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
- Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
- The selected parcels are downloaded and distributed. Click **Continue**.
- The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
- Choose the type of upgrade and restart:
  - Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

- Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### Upgrade Cluster Command

Status: Failed

Context: Cluster 1

Start Time: Sep 17, 6:29:40 PM

Duration: 113.01 seconds

Retry

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Finish** to return to the Home page.

#### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 628. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

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:

- **Not including Impala and Search**

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

- **Including Impala and Search**

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

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:

## Upgrade

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

### (Optional) Install KMS (Navigator Key Trustee)

If you want to use Navigator Key Trustee Server as the underlying key store for [HDFS Transparent Encryption](#), see [Installing Key Trustee KMS](#) on page 211 for instructions on installing the Key Trustee parcel.

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
  - CDH 5.3 to 5.4 or higher
1. Go to the Hive service.
  2. Select **Actions > Stop** and click **Stop** to confirm.
  3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.

4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

#### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

#### Start Cluster Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
  2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.
- When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

#### Deploy Client Configuration Files

1. On the Home page, click  to the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

#### Upgrading to CDH 5.3 Using Packages

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

## Upgrade

If you originally used Cloudera Manager to install CDH 5 using packages, you can upgrade to CDH 5.3 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME" , "TBLS". "TBL_NAME" , "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

- Hue validates CA certificates and needs a truststore. To create one, follow the instructions in [Hue as a TLS/SSL Client](#).

## Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

## Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

## Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:

## Upgrade

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### Upgrade Managed Components

#### 1. Download and save the repo file.

- On Red Hat-compatible systems:

Click the entry in the table below that matches your Red Hat or CentOS system, go 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	<a href="#">Red Hat/CentOS/Oracle 5 link</a>
Red Hat/CentOS 6 (64-bit)	<a href="#">Red Hat/CentOS 6 link</a>

- On SLES systems:

##### 1. Run the following command:

```
$ sudo zypper addrepo -f  
http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

##### 2. Update your system package index by running:

```
$ sudo zypper refresh
```

- On Ubuntu and Debian systems:

Create a new file `/etc/apt/sources.list.d/cloudera.list` with the following contents:

##### – For Ubuntu systems:

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib deb-src http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib
```

##### – For Debian systems:

```
deb http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib deb-src  
http://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`.

#### 2. Edit the repo file to point to the release you want to install or upgrade to.

- On Red Hat-compatible systems:

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=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/  
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera  
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

```
baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/ to
```

```
baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/
```

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

- On SLES systems:

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=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

```
baseurl=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/ to
```

```
baseurl= http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.1.0/
```

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.1.0/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

- On Ubuntu and Debian systems:

Replace -cdh5 near the end of each line (before contrib) with the CDH release you need to install. Here are examples using CDH 5.1.0:

#### **For 64-bit Ubuntu Precise:**

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh
precise-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh precise-cdh5.1.0
contrib
```

#### **For Debian Wheezy:**

```
deb http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
```

### 3. (Optionally) add a repository key:

- Red Hat compatible

- Red Hat/CentOS/Oracle 5

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

## Upgrade

### – Red Hat/CentOS/Oracle 6

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

- SLES

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

- Ubuntu and Debian

- Ubuntu Precise

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

- Debian Wheezy

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

## 4. Install the CDH packages:

- Red Hat compatible

```
$ sudo yum clean all  
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

- SLES

```
$ sudo zypper clean --all  
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

- Ubuntu and Debian

```
$ sudo apt-get update  
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs  
hadoop-kms hbase hbase-solr hive-hbase hive-webhcatt 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 parquet pig pig-udf-datafu search sentry solr  
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

## Update Symlinks for the Newly Installed Components

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
```

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. In the **Choose Method** field, select the **Use Packages** option.

4. In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.

5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.

6. Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.

7. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.

8. Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds

**Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

1. Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

9. Click **Finish** to return to the Home page.

## Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 636. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

## Upgrade

### Finalize the HDFS Metadata Upgrade

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful.

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.

3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

#### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

#### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

#### Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Upgrading to CDH 5.2

Use the instructions in this section to upgrade from an earlier version of CDH 5 to CDH 5.2.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

## Upgrade

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
    INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
    INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
    INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
        AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
        AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Upgrading to CDH 5.2 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.2 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH 5.2, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:
  - Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
  - Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS"."NAME", "TBLS"."TBL_NAME", "PARTITION_KEY_VALS"."PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS"."PART_ID" = "PARTITIONS"."PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS"."TBL_ID" = "PARTITIONS"."TBL_ID"
INNER JOIN "TBLS" ON "TBLS"."TBL_ID" = "PARTITIONS"."TBL_ID"
INNER JOIN "DBS" ON "DBS"."DB_ID" = "TBLS"."DB_ID"
    AND "PARTITION_KEYS"."INTEGER_IDX" = "PARTITION_KEY_VALS"."INTEGER_IDX"
    AND "PARTITION_KEYS"."PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Stop Cluster Services

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is `/data/dfs/nn`, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

## Upgrade

You should see output like this:

```
./  
./current/  
./current/fsimage  
./current/fstime  
./current/VERSION  
./current/edits  
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

### Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click  next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.
3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Choose the type of upgrade and restart:
  - **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds

**Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.
  - **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

- Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Finish** to return to the Home page.

#### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 642. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

- 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.
- Uninstall the CDH packages on each host:

- Not including Impala and Search

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

- Including Impala and Search

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

- 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
```

- 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.

## Upgrade

### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.

2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### [Upgrade the Sentry Database](#)

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### [Upgrade Spark](#)

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### [Start Cluster Services](#)

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Start**.
  2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.
- When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

### [Deploy Client Configuration Files](#)

1. On the Home page, click  to the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### [Upgrading to CDH 5.2 Using Packages](#)

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

If you originally used Cloudera Manager to install CDH 5 using packages, you can upgrade to CDH 5.2 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using packages, the steps are as follows.

#### [Before You Begin](#)

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

## Upgrade

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- **Date partition columns:** as of Hive version 13, implemented in CDH **5.2**, Hive validates the format of dates in partition columns, if they are stored as dates. A partition column with a date in invalid form can neither be used nor dropped once you upgrade to CDH 5.2 or higher. To avoid this problem, do one of the following:

- Fix any invalid dates before you upgrade. Hive expects dates in partition columns to be in the form YYYY-MM-DD.
- Store dates in partition columns as strings or integers.

You can use the following SQL query to find any partition-column values stored as dates:

```
SELECT "DBS". "NAME" , "TBLS". "TBL_NAME" , "PARTITION_KEY_VALS". "PART_KEY_VAL"
FROM "PARTITION_KEY_VALS"
INNER JOIN "PARTITIONS" ON "PARTITION_KEY_VALS". "PART_ID" = "PARTITIONS". "PART_ID"
INNER JOIN "PARTITION_KEYS" ON "PARTITION_KEYS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "TBLS" ON "TBLS". "TBL_ID" = "PARTITIONS". "TBL_ID"
INNER JOIN "DBS" ON "DBS". "DB_ID" = "TBLS". "DB_ID"
    AND "PARTITION_KEYS". "INTEGER_IDX" = "PARTITION_KEY_VALS". "INTEGER_IDX"
    AND "PARTITION_KEYS". "PKEY_TYPE" = 'date';
```

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run hdfs fsck / and hdfs dfsadmin -report and fix every issue.
- Run hbase hbck.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

## Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

## Back up the HDFS Metadata on the NameNode

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Go to the HDFS service.

2. Click the **Configuration** tab.

3. In the Search field, search for "NameNode Data Directories" and note the value.

4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

## Back up Metastore Databases

Back up the Hive, Sentry, and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

## Upgrade Managed Components

1. Download and save the repo file.

- On Red Hat-compatible systems:

Click the entry in the table below that matches your Red Hat or CentOS system, go 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	<a href="#">Red Hat/CentOS/Oracle 5 link</a>

## Upgrade

For OS Version	Click this Link
Red Hat/CentOS 6 (64-bit)	<a href="#">Red Hat/CentOS 6 link</a>

- On SLES systems:

1. Run the following command:

```
$ sudo zypper addrepo -f  
http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

- On Ubuntu and Debian systems:

Create a new file `/etc/apt/sources.list.d/cloudera.list` with the following contents:

- For Ubuntu systems:

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib deb-src http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib
```

- For Debian systems:

```
deb http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib deb-src  
http://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`.

2. Edit the repo file to point to the release you want to install or upgrade to.

- On Red Hat-compatible systems:

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=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/  
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera  
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

`baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/` to

`baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/`

In this example, the resulting file should look like this:

```
[cloudera-cdh5]  
name=Cloudera's Distribution for Hadoop, Version 5  
baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/  
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera  
gpgcheck = 1
```

- On SLES systems:

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=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

baseurl=http://archive.cloudera.com/cdh5/sles/11/x86\_64/cdh/5/ to

baseurl= http://archive.cloudera.com/cdh5/sles/11/x86\_64/cdh/5.1.0/

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.1.0/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

- On Ubuntu and Debian systems:

Replace -cdh5 near the end of each line (before contrib) with the CDH release you need to install. Here are examples using CDH 5.1.0:

#### **For 64-bit Ubuntu Precise:**

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh
precise-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh precise-cdh5.1.0
contrib
```

#### **For Debian Wheezy:**

```
deb http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
```

### **3. (Optionally) add a repository key:**

- Red Hat compatible
  - Red Hat/CentOS/Oracle 5

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

#### **– Red Hat/CentOS/Oracle 6**

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

- SLES

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

- Ubuntu and Debian

## Upgrade

### – Ubuntu Precise

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

### – Debian Wheezy

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

#### 4. Install the CDH packages:

##### • Red Hat compatible

```
$ sudo yum clean all  
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

##### • SLES

```
$ sudo zypper clean --all  
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

##### • Ubuntu and Debian

```
$ sudo apt-get update  
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs  
hadoop-kms hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr  
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

#### Update Symlinks for the Newly Installed Components

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
```

#### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 649. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Finalize the HDFS Metadata Upgrade

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher

- CDH 5.2 or 5.3 to 5.4 or higher

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful. Once you have finalized the upgrade, it is not possible to roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

To finalize the metadata upgrade:

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

#### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

#### Upgrade HDFS Metadata

If upgrading from:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

#### Upgrade the Hive Metastore Database

Required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.3 to 5.4 or higher

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

## Upgrade

### Upgrade the Sentry Database

Required for the following upgrades:

- CDH 5.1 to 5.2 or higher
- CDH 5.2 to 5.3 or higher
- CDH 5.4 to 5.5 or higher

1. Go to the Sentry service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sentry Database Tables** and click **Upgrade Sentry Database Tables** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



to the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

or

## Upgrading to CDH 5.1

Use the instructions in this section to upgrade an earlier version of CDH 5 to CDH 5.1.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade from CDH 4 to CDH 5, use the instructions under [Upgrading CDH 4 to CDH 5](#) on page 660.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x

Target CDH Version	Minimum Cloudera Manager Version
5.1.4	5.1.x
5.4.1	5.4.x

- Make sure there are no Oozie workflows in RUNNING or SUSPENDED status; otherwise the Oozie database upgrade will fail and you will have to reinstall CDH 4 to complete or kill those running workflows.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Upgrading to CDH 5.1 Using Parcels

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

You can upgrade your CDH 5 cluster to CDH 5.1 using parcels from within the Cloudera Manager Admin Console. Your current CDH 5 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

To upgrade CDH using parcels, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Make sure there are no Oozie workflows in RUNNING or SUSPENDED status; otherwise the Oozie database upgrade will fail and you will have to reinstall CDH 4 to complete or kill those running workflows.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.

## Upgrade

- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Back up Metastore Databases

Back up the Sqoop metastore database.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the [Modify the Remote Parcel Repository URLs](#) link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Choose the type of upgrade and restart:

- **Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

1. Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

### Upgrade CDH running on Cluster 1

#### Upgrade Cluster Command

Status: **Failed**   Context: [Cluster 1](#)   Start Time: Sep 17, 6:29:40 PM   Duration: 113.01 seconds

**Retry**

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

2. Click **Continue**. The wizard reports the result of the upgrade.

- **Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

- Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

**10** Click **Finish** to return to the Home page.

#### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 654. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

- 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.
- Uninstall the CDH packages on each host:

- Not including Impala and Search

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

- Including Impala and Search

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

- 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
```

- 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.

## Upgrade

### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

#### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

#### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

#### Restart All Services

1. [Restart the cluster.](#)

#### Deploy Client Configuration Files

1. On the Home page, click  the right of the cluster name and select **Deploy Client Configuration**.  
or  
the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

#### Upgrading to CDH 5.1 Using Packages

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

If you installed or upgraded to CDH 5 using packages, you can upgrade to CDH 5.1 using either packages or parcels. Using parcels is recommended, because the upgrade wizard for parcels handles the upgrade almost completely automatically.

To upgrade CDH using packages, the steps are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x

Target CDH Version	Minimum Cloudera Manager Version
5.4.1	5.4.x

- Make sure there are no Oozie workflows in RUNNING or SUSPENDED status; otherwise the Oozie database upgrade will fail and you will have to reinstall CDH 4 to complete or kill those running workflows.
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Upgrade Unmanaged Components

Upgrade unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Mahout
- Pig
- Whirr

For information on upgrading these unmanaged components, see [Upgrading Mahout](#) on page 396, [Upgrading Pig](#) on page 418, and [Upgrading Whirr](#) on page 451.

## Stop Cluster Services

1. On the **Home > Status** tab, click



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

2. 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.

## Back up Metastore Databases

Back up the Sqoop metastore database.

1. For each affected service:
  - a. If not already stopped, stop the service.
  - b. Back up the database. See [Backing Up Databases](#) on page 117.

## Upgrade Managed Components

1. Download and save the repo file.

- On Red Hat-compatible systems:

## Upgrade

Click the entry in the table below that matches your Red Hat or CentOS system, go 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	<a href="#">Red Hat/CentOS/Oracle 5 link</a>
Red Hat/CentOS 6 (64-bit)	<a href="#">Red Hat/CentOS 6 link</a>

- On SLES systems:

1. Run the following command:

```
$ sudo zypper addrepo -f  
http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

- On Ubuntu and Debian systems:

Create a new file `/etc/apt/sources.list.d/cloudera.list` with the following contents:

- For Ubuntu systems:

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib deb-src http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5  
contrib
```

- For Debian systems:

```
deb http://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib deb-src  
http://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`.

2. Edit the repo file to point to the release you want to install or upgrade to.

- On Red Hat-compatible systems:

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=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/  
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera  
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

`baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/` to

`baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/`

In this example, the resulting file should look like this:

```
[cloudera-cdh5]  
name=Cloudera's Distribution for Hadoop, Version 5
```

```
baseurl=http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.1.0/
gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

- On SLES systems:

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=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

For example, if you want to install CDH 5.1.0, change

```
baseurl=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/ to
```

```
baseurl= http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.1.0/
```

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.1.0/
gpgkey = http://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

- On Ubuntu and Debian systems:

Replace `-cdh5` near the end of each line (before `contrib`) with the CDH release you need to install. Here are examples using CDH 5.1.0:

#### **For 64-bit Ubuntu Precise:**

```
deb [arch=amd64] http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh
precise-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh precise-cdh5.1.0
contrib
```

#### **For Debian Wheezy:**

```
deb http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
deb-src http://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.1.0 contrib
```

### 3. (Optionally) add a repository key:

- Red Hat compatible

- Red Hat/CentOS/Oracle 5

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

- Red Hat/CentOS/Oracle 6

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

## Upgrade

- SLES

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

- Ubuntu and Debian

- Ubuntu Precise

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

- Debian Wheezy

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

### 4. Install the CDH packages:

- Red Hat compatible

```
$ sudo yum clean all  
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

- SLES

```
$ sudo zypper clean --all  
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs hadoop-kms  
hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr solr-mapreduce  
spark-python sqoop sqoop2 whirr zookeeper
```

- Ubuntu and Debian

```
$ sudo apt-get update  
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-httpfs  
hadoop-kms hbase hbase-solr hive-hbase hive-webhcate 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 parquet pig pig-udf-datafu search sentry solr  
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



**Note:** Installing these packages will also install all the other CDH packages that are needed for a full CDH 5 installation.

### Update Symlinks for the Newly Installed Components

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
```

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.

- From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

- In the **Choose Method** field, select the **Use Packages** option.
- In the **Choose CDH Version (Packages)** field, specify the CDH version of the packages you have installed on your cluster. Click **Continue**.
- Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
- Cloudera Manager checks that hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the packages and click **Check Again**. When there are no errors, click **Continue**.
- The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
- Choose the type of upgrade and restart:

- Cloudera Manager upgrade** - Cloudera Manager performs all service upgrades and restarts the cluster.

- Click **Continue**. The **Command Progress** screen displays the result of the commands run by the wizard as it shuts down all services, activates the new parcel, upgrades services as necessary, deploys client configuration files, and restarts services. If any of the steps fails or you click the **Abort** button the **Retry** button at the top right is enabled.

## Upgrade CDH running on Cluster 1

### ④ Upgrade Cluster Command

Status: **Failed**    Context: [Cluster 1](#) ↗    Start Time: Sep 17, 6:29:40 PM    Duration: 113.01 seconds    [Retry](#)

You can click **Retry** to retry the step and continue the wizard or click the Cloudera Manager logo to return to the **Home > Status** tab and manually perform the failed step and all following steps.

- Click **Continue**. The wizard reports the result of the upgrade.

- Manual upgrade** - Select the **Let me upgrade the cluster** checkbox. Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually doing the upgrade is difficult and is for advanced users only.

- Click **Continue**. Cloudera Manager displays links to documentation describing the required upgrade steps.

- Click **Finish** to return to the Home page.

### Perform Manual Upgrade or Recover from Failed Steps

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 659. If you chose manual upgrade or any of the steps in the **Command Progress** screen fails, complete the steps as described in that section before proceeding.

#### Upgrade Wizard Actions

Do the steps in this section only if you chose a manual upgrade or the upgrade wizard reports a failure and you choose not to retry.

#### Upgrade the Oozie ShareLib

- Go to the Oozie service.
- Select **Actions > Start** and click **Start** to confirm.
- Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

#### Upgrade Sqoop

- Go to the Sqoop service.

## Upgrade

2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Upgrade Spark

1. Go to the Spark service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Install Spark JAR** and click **Install Spark JAR** to confirm.
4. Select **Actions > Create Spark History Log Dir** and click **Create Spark History Log Dir** to confirm.

### Restart All Services

1. [Restart the cluster.](#)

### Deploy Client Configuration Files

1. On the Home page, click  the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

or

## Upgrading CDH 4 to CDH 5

The instructions in this topic describe how to upgrade from a CDH 4 to CDH 5 release. You can upgrade to CDH 5 within the Cloudera Manager Admin Console using parcels or packages. Using parcels vastly simplifies the upgrade process. Electing to upgrade using packages means that all future upgrades must be done manually.

**Important:**

- You cannot perform a rolling upgrade from CDH 4 to CDH 5. There are incompatibilities between the major versions, so a rolling restart is not possible. Rolling upgrade is also *not* supported from CDH 5 Beta 2 to CDH 5.
- If you have just upgraded to Cloudera Manager 5, you must hard restart the Cloudera Manager Agents as described in the [\(Optional\) Deploy a Cloudera Manager Agent Upgrade](#) on page 497.
- **HBase** - After you upgrade you must recompile all HBase coprocessor and custom JARs.
- **Impala**
  - If you upgrade to CDH 5.1, Impala will be upgraded to 1.4.1. See [New Features in Impala](#) for information about Impala 1.4.x features.
  - If you upgrade to CDH 5.0, Impala will be upgraded to 1.3.2. If you have CDH 4 installed with Impala 1.4.0, Impala will be *downgraded* to Impala 1.3.2. See [New Features in Impala](#) for information about Impala 1.3 features.
- **Hive and Parquet**
  - When upgrading from CDH 4 to CDH 5, upgrade scripts may modify your schemas. For example, if you used Parquet with CDH 4, CDH 5 changes the input and output formats. The upgrade script changes `parquet.hive.DeprecatedParquetInputFormat` and `parquet.hive.DeprecatedParquetOutputFormat` to `org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat` and `org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat` in the schema. This may cause errors such as `Table already exists` but schema doesn't match and you may need to modify MapReduce jobs to use the newer Parquet Serdes.
- **MapReduce and YARN**
  - In a Cloudera Manager deployment of a CDH 5 cluster, the YARN service is the default MapReduce computation framework. In CDH 5, the MapReduce service has been deprecated. However, the MapReduce service is fully supported for backward compatibility through the CDH 5 lifecycle.
  - In a Cloudera Manager deployment of a CDH 4 cluster, the MapReduce service is the default MapReduce computation framework. You can create a YARN service in a CDH 4 cluster, but it is not considered production ready.
  - For production uses, Cloudera recommends that *only one* MapReduce framework should be running at any given time. If development needs or other use case requires switching between MapReduce and YARN, both services can be configured at the same time, but only one should be running (to fully optimize the hardware resources available).

For information on migrating from MapReduce to YARN, see [Managing YARN \(MRv2\)](#) and [MapReduce \(MRv1\)](#).



**Warning:** You can use Cloudera Manager to roll back an upgrade from CDH 4 to CDH 5 as long as you backup certain configuration files, databases, and other artifacts before beginning an upgrade. However, after you have [finalized the HDFS upgrade](#) you can no longer roll back the CDH upgrade. See [Rolling Back a CDH 4-to-CDH 5 Upgrade](#) on page 750 for the backup and rollback procedures.

**Before You Begin**

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Upgrade to Cloudera Manager 5 *before* upgrading to CDH 5.

## Upgrade

- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Make sure there are no Oozie workflows in RUNNING or SUSPENDED status; otherwise the Oozie database upgrade will fail and you will have to reinstall CDH 4 to complete or kill those running workflows.
- Delete Symbolic Links in HDFS

If there are symbolic links in HDFS when you upgrade from CDH 4 to CDH 5, the upgrade will fail and you will have to downgrade to CDH 4, delete the symbolic links, and start over. To prevent this, proceed as follows.

1. cd to the directory on the NameNode that contains the latest `fsimage`. The location of this directory is specified as the value of `dfs.namenode.name.dir` (or `dfs.name.dir`) in `hdfs-site.xml`.
2. Use a command such as the following to write out the path names in the `fsimage`:

```
$ hdfs oiv -i FSIMAGE -o /tmp/YYYY-MM-DD_FSIMAGE.txt
```

3. Use a command such as the following to find the path names of any symbolic links listed in `/tmp/YYYY-MM-DD_FSIMAGE.txt` and write them out to the file `/tmp/symlinks.txt`:

```
$ grep -- ">" /tmp/YYYY-MM-DD_FSIMAGE.txt > /tmp/symlinks.txt
```

4. Delete any symbolic links listed in `/tmp/symlinks.txt`.

- When upgrading from CDH 4 to CDH 5, Oozie upgrade can take a very long time. For upgrades from CDH 4.3 and higher, you can reduce this time by reducing the amount of history Oozie retains. To reduce Oozie history:

1. Go to the Oozie service.
2. Click the Configuration tab.
3. Click Category > Advanced.
4. In **Oozie Server Advanced Configuration Snippet (Safety Valve)** for `oozie-site.xml`, enter the following

```
<property>
<name>oozie.service.PurgeService.older.than</name>
<value>7</value>
</property>
<property>
<name>oozie.service.PurgeService.purge.limit</name>
<value>1000</value>
</property>
```

5. For CDH lower than 5.2, enable DEBUG level logging:

- a. Click **Category > Logs**.
- b. Set **Oozie Server Logging Threshold** to **DEBUG**.

6. Click **Save Changes** to commit the changes.
7. Restart the Oozie Server role.

8. Wait for the purge service to run and finish. By default, the service runs every hour. The purge service emits the following messages in the Oozie server log:

```
STARTED Purge to purge Workflow Jobs older than [7] days, Coordinator Jobs older than [7] days, and Bundlejobs older than [7] days.
ENDED Purge deleted [x] workflows, [y] coordinatorActions, [z] coordinators, [w] bundles
```

9. Revert the purge service and log level settings to the default.

- When upgrading from CDH 4 to CDH 5, Hue upgrade can take a very long time if the beeswax\_queryhistory, beeswax\_savedquery, and oozie\_job tables are larger than 1000 records. You can reduce the upgrade time by running a script to reduce the size of the Hue database:

- Stop the Hue service.
- Back up the Hue database.
- Download the [history cleanup script](#) to the host running the Hue Server.
- Run the following as root:

- parcel installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"
/opt/cloudera/parcels/CDH/share/hue/build/env/bin/hue shell
```

- package installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"
/usr/share/hue/build/env/bin/hue shell
```

5. Run the downloaded script in the Hue shell.

- 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:

- Open the `my.cnf` file for MySQL, search for "default-storage-engine" and note its value.
- Connect to MySQL and run the following commands:

```
use hue;
show create table auth_user;
```

- 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`). 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.

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- If using HBase:
  - Run `hbase hbck`.
  - Before you can upgrade HBase from CDH 4 to CDH 5, your HFiles must be upgraded from HFile v1 format to HFile v2, because CDH 5 no longer supports HFile v1. The upgrade procedure itself is different if you are using Cloudera Manager or the command line, but has the same results. The first step is to check for instances of

## Upgrade

HFile v1 in the HFiles and mark them to be upgraded to HFile v2, and to check for and report about corrupted files or files with unknown versions, which need to be removed manually. The next step is to rewrite the HFiles during the next major compaction. After the HFiles are upgraded, you can continue the upgrade. After the upgrade is complete, you must recompile custom coprocessors and JARs. To check and upgrade the files:

1. In the Cloudera Admin Console, go to the HBase service and run **Actions > Check HFile Version**.
2. Check the output of the command in the stderr log.

Your output should be similar to the following:

```
Tables Processed:  
hdfs://localhost:41020/myHBase/.META.  
hdfs://localhost:41020/myHBase/usertable  
hdfs://localhost:41020/myHBase/TestTable  
hdfs://localhost:41020/myHBase/t  
  
Count of HFileV1: 2  
HFileV1:  
hdfs://localhost:41020/myHBase/usertable  
/fa02dac1f38d03577bd0f7e666f12812/family/249450144068442524  
hdfs://localhost:41020/myHBase/usertable  
/ecdd3eae2d2fcf8184ac025555bb2af/family/249450144068442512  
  
Count of corrupted files: 1  
Corrupted Files:  
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812/family/1  
Count of Regions with HFileV1: 2  
Regions to Major Compact:  
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812  
hdfs://localhost:41020/myHBase/usertable/ecdd3eae2d2fcf8184ac025555bb2af
```

In the example above, you can see that the script has detected two HFile v1 files, one corrupt file and the regions to major compact.

3. Trigger a major compaction on each of the reported regions. This major compaction rewrites the files from HFile v1 to HFile v2 format. To run the major compaction, start HBase Shell and issue the `major_compact` command.

```
$ /usr/lib/hbase/bin/hbase shell  
hbase> major_compact 'usertable'
```

You can also do this in a single step by using the `echo` shell built-in command.

```
$ echo "major_compact 'usertable'" | /usr/lib/hbase/bin/hbase shell
```

- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Upgrading from CDH 4 to CDH 5 Parcels

#### Minimum Required Role: [Full Administrator](#)

This topic covers upgrading a CDH 4 cluster to a CDH 5 cluster using the upgrade wizard, which will install CDH 5 parcels. Your CDH 4 cluster can be using *either parcels or packages*; you can use the cluster upgrade wizard to upgrade using parcels in either case.

If you want to upgrade using CDH 5 packages, you can do so using a manual process. See [Upgrading from CDH 4 Packages to CDH 5 Packages](#) on page 672.

The steps to upgrade a CDH installation managed by Cloudera Manager using parcels are as follows.

### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Upgrade to Cloudera Manager 5 *before* upgrading to CDH 5.
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Make sure there are no Oozie workflows in RUNNING or SUSPENDED status; otherwise the Oozie database upgrade will fail and you will have to reinstall CDH 4 to complete or kill those running workflows.
- Delete Symbolic Links in HDFS

If there are symbolic links in HDFS when you upgrade from CDH 4 to CDH 5, the upgrade will fail and you will have to downgrade to CDH 4, delete the symbolic links, and start over. To prevent this, proceed as follows.

1. cd to the directory on the NameNode that contains the latest `fsimage`. The location of this directory is specified as the value of `dfs.namenode.name.dir` (or `dfs.name.dir`) in `hdfs-site.xml`.
2. Use a command such as the following to write out the path names in the `fsimage`:

```
$ hdfs civ -i FSIMAGE -o /tmp/YYYY-MM-DD_FSIMAGE.txt
```

3. Use a command such as the following to find the path names of any symbolic links listed in `/tmp/YYYY-MM-DD_FSIMAGE.txt` and write them out to the file `/tmp/symlinks.txt`:

```
$ grep -- "->" /tmp/YYYY-MM-DD_FSIMAGE.txt > /tmp/symlinks.txt
```

4. Delete any symbolic links listed in `/tmp/symlinks.txt`.

- When upgrading from CDH 4 to CDH 5, Oozie upgrade can take a very long time. For upgrades from CDH 4.3 and higher, you can reduce this time by reducing the amount of history Oozie retains. To reduce Oozie history:

1. Go to the Oozie service.
2. Click the **Configuration** tab.
3. Click **Category > Advanced**.
4. In **Oozie Server Advanced Configuration Snippet (Safety Valve) for oozie-site.xml**, enter the following

```
<property>
<name>oozie.service.PurgeService.older.than</name>
<value>7</value>
</property>
<property>
<name>oozie.service.PurgeService.purge.limit</name>
<value>1000</value>
</property>
```

5. For CDH lower than 5.2, enable DEBUG level logging:

## Upgrade

- a. Click **Category > Logs**.
- b. Set **Oozie Server Logging Threshold** to **DEBUG**.
6. Click **Save Changes** to commit the changes.
7. Restart the Oozie Server role.
8. Wait for the purge service to run and finish. By default, the service runs every hour. The purge service emits the following messages in the Oozie server log:

```
STARTED Purge to purge Workflow Jobs older than [7] days, Coordinator Jobs older than [7] days, and Bundlejobs older than [7] days.  
ENDED Purge deleted [x] workflows, [y] coordinatorActions, [z] coordinators, [w] bundles
```

9. Revert the purge service and log level settings to the default.
- When upgrading from CDH 4 to CDH 5, Hue upgrade can take a very long time if the beeswax\_queryhistory, beeswax\_savedquery, and oozie\_job tables are larger than 1000 records. You can reduce the upgrade time by running a script to reduce the size of the Hue database:
  1. Stop the Hue service.
  2. Back up the Hue database.
  3. Download the [history cleanup script](#) to the host running the Hue Server.
  4. Run the following as root:
    - **parcel installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"  
/opt/cloudera/parcels/CDH/share/hue/build/env/bin/hue shell
```

- **package installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"  
/usr/share/hue/build/env/bin/hue shell
```

5. Run the downloaded script in the Hue shell.
- **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`). 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.

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.

- If using HBase:

- Run `hbase hbck`.
- Before you can upgrade HBase from CDH 4 to CDH 5, your HFiles must be upgraded from HFile v1 format to HFile v2, because CDH 5 no longer supports HFile v1. The upgrade procedure itself is different if you are using Cloudera Manager or the command line, but has the same results. The first step is to check for instances of HFile v1 in the HFiles and mark them to be upgraded to HFile v2, and to check for and report about corrupted files or files with unknown versions, which need to be removed manually. The next step is to rewrite the HFiles during the next major compaction. After the HFiles are upgraded, you can continue the upgrade. After the upgrade is complete, you must recompile custom coprocessors and JARs. To check and upgrade the files:

1. In the Cloudera Admin Console, go to the HBase service and run **Actions > Check HFile Version**.
2. Check the output of the command in the `stderr` log.

Your output should be similar to the following:

```
Tables Processed:
hdfs://localhost:41020/myHBase/.META.
hdfs://localhost:41020/myHBase/usertable
hdfs://localhost:41020/myHBase/TestTable
hdfs://localhost:41020/myHBase/t

Count of HFileV1: 2
HFileV1:
hdfs://localhost:41020/myHBase/usertable
/fa02dac1f38d03577bd0f7e666f12812/family/249450144068442524
hdfs://localhost:41020/myHBase/usertable
/ecdd3eaeee2d2fcf8184ac025555bb2af/family/249450144068442512

Count of corrupted files: 1
Corrupted Files:
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812/family/1
Count of Regions with HFileV1: 2
Regions to Major Compact:
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812
hdfs://localhost:41020/myHBase/usertable/ecdd3eaeee2d2fcf8184ac025555bb2af
```

In the example above, you can see that the script has detected two HFile v1 files, one corrupt file and the regions to major compact.

3. Trigger a major compaction on each of the reported regions. This major compaction rewrites the files from HFile v1 to HFile v2 format. To run the major compaction, start HBase Shell and issue the `major_compact` command.

```
$ /usr/lib/hbase/bin/hbase shell
hbase> major_compact 'usertable'
```

You can also do this in a single step by using the `echo` shell built-in command.

```
$ echo "major_compact 'usertable'" | /usr/lib/hbase/bin/hbase shell
```

- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Upgrade

### Stop All Services

#### 1. Stop the cluster.

- On the **Home > Status** tab, click



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

- 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.

#### 2. Stop the Cloudera Management Service:

- Do one of the following:

- 1. Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
  2. Select **Actions > Stop**.
- 1. On the **Home > Status** tab, click



to the right of **Cloudera Management Service** and select **Stop**.

- Click **Stop** to confirm. The **Command Details** window shows the progress of stopping the roles.

- When **Command completed with n/n successful subcommands** appears, the task is complete. Click **Close**.

### Perform Service-Specific Prerequisite Actions

- **Accumulo** - if you have installed the Accumulo parcel, deactivate it following the instructions in [Managing Parcels](#) on page 57.
- **HDFS** - Back up HDFS metadata on the NameNode:
  1. Go to the HDFS service.
  2. Click the **Configuration** tab.
  3. In the Search field, search for "NameNode Data Directories" and note the value.
  4. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn  
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./  
./current/  
./current/fsimage  
./current/fstime  
./current/VERSION  
./current/edits  
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

- Back up the Hive and Sqoop metastore databases.

1. For each affected service:

- a. If not already stopped, stop the service.
- b. Back up the database. See [Backing Up Databases](#) on page 117.

### Remove CDH Packages

If your previous installation of CDH was done using *packages*, you must remove those packages on all hosts in the cluster being upgraded. This will definitely be the case if you are running a version of CDH prior to CDH 4.1.3, since parcels were not available with those releases.

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:

Operating System	Command
RHEL	\$ sudo yum remove bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc
SLES	\$ sudo zypper remove bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc
Ubuntu or Debian	\$ sudo apt-get purge bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc

3. Restart all the Cloudera Manager Agents to force an update of the installed binaries reported by the Agent. On each host:

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

4. Run the Host Inspector to verify that the packages have been removed:
  - a. Click **Hosts** tab and then click the **Host Inspector** button.
  - b. When the command completes, click **Show Inspector Results**.
5. If your Hue service uses the embedded SQLite DB, restore the DB you backed up:
  - a. Stop the Hue service.
  - b. Copy the backup from the temporary location to the newly created Hue database directory, `/var/lib/hue`.
  - c. Start the Hue service.

### Deactivate and Remove the GPL Extras Parcel

If you are using LZO, deactivate and remove the CDH 4 GPL Extras parcel.

### Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click  next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.
3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 page where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.

## Upgrade

7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Click **Continue**. Cloudera Manager performs all service upgrades and restarts the cluster.
- 10 The wizard reports the result of the upgrade. Choose one of the following:
  - Leave **OK, set up YARN and import existing configuration from my MapReduce service** checked.
    1. Click **Continue** to proceed. Cloudera Manager stops the YARN service (if running) and its dependencies.
    2. Click **Continue** to proceed. The next page indicates some additional configuration required by YARN.
    3. Verify or modify the configurations and click **Continue**. The Switch Cluster to MR2 step proceeds.
    4. When all steps have completed, click **Continue**.
  - Deselect **OK, set up YARN and import existing configuration from my MapReduce service**.
- 11 Click **Finish** to return to the **Home > Status** tab.
- 12 (Optional) Remove the MapReduce service.
  - a. In the MapReduce row, right-click
  - and select **Delete**. Click **Delete** to confirm.

### Recover from Failed Steps



**Note:** If you encounter errors during these steps:

- If the converting configuration parameters step fails, Cloudera Manager rolls back all configurations to CDH 4. Fix any reported problems and retry the upgrade.
- If the upgrade command fails at any point after the convert configuration step, there is no retry support in Cloudera Manager. You must first correct the error, then manually re-run the individual commands. You can view the remaining commands in the Recent Commands page.
- If the HDFS upgrade metadata step fails, you cannot revert back to CDH 4 unless you restore a backup of Cloudera Manager.

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 671. If any of the steps in the **Command Progress** screen fails, complete the step as described in that section before proceeding.

### Upgrade the GPL Extras Parcel

If you are using LZO:

1. Install the CDH 5 GPL Extras parcel. See [Installing the GPL Extras Parcel](#) on page 169.
2. Reconfigure and restart services that use the parcel. See [Configuring Services to Use the GPL Extras Parcel](#).

### Restart the Reports Manager Role

1. Do one of the following:
  - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
  - On the **Home > Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Instances** tab.
3. Check the checkbox next to **Reports Manager**.
4. Select **Actions for Selected > Restart** and then **Restart** to confirm.

## Recompile JARs

- **MapReduce and YARN** - Recompile JARs used in MapReduce applications. For further information, see [For MapReduce Programmers: Writing and Running Jobs](#) on page 234.
- **HBase** - Recompile coprocessor and custom JARs used by HBase applications.

## Finalize the HDFS Metadata Upgrade

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful.

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

## Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

### Upgrade HDFS Metadata

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade HBase

1. Go to the HBase service.
2. Select **Actions > Upgrade HBase** and click **Upgrade HBase** to confirm.

### Upgrade the Hive Metastore Database

1. Go to the Hive service.
2. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
3. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade Oozie

1. Go to the Oozie service.
2. Select **Actions > Upgrade Database** and click **Upgrade Database** to confirm.
3. Start the Oozie service.
4. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

## Start Cluster Services

1. On the **Home > Status** tab, click



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

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

## Upgrade

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

### Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Upgrading from CDH 4 Packages to CDH 5 Packages

#### Minimum Required Role: [Full Administrator](#)

If you originally used Cloudera Manager to install CDH using packages, you can upgrade to CDH 5 either using packages or parcels. Parcels is the preferred and recommended way to upgrade, as the upgrade wizard provided for parcels handles the upgrade process almost completely automatically.

The steps to upgrade a CDH installation managed by Cloudera Manager using packages are as follows.

#### Before You Begin

- Read the [CDH 5 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Upgrade to Cloudera Manager 5 *before* upgrading to CDH 5.
- Ensure Java 1.7 is installed across the cluster. For installation instructions and recommendations, see [Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment](#) on page 739, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Make sure there are no Oozie workflows in RUNNING or SUSPENDED status; otherwise the Oozie database upgrade will fail and you will have to reinstall CDH 4 to complete or kill those running workflows.
- Delete Symbolic Links in HDFS

If there are symbolic links in HDFS when you upgrade from CDH 4 to CDH 5, the upgrade will fail and you will have to downgrade to CDH 4, delete the symbolic links, and start over. To prevent this, proceed as follows.

1. cd to the directory on the NameNode that contains the latest `fsimage`. The location of this directory is specified as the value of `dfs.namenode.name.dir` (or `dfs.name.dir`) in `hdfs-site.xml`.
2. Use a command such as the following to write out the path names in the `fsimage`:

```
$ hdfs oiv -i FSIMAGE -o /tmp/YYYY-MM-DD_FSIMAGE.txt
```

3. Use a command such as the following to find the path names of any symbolic links listed in `/tmp/YYYY-MM-DD_FSIMAGE.txt` and write them out to the file `/tmp/symlinks.txt`:

```
$ grep -- "->" /tmp/YYYY-MM-DD_FSIMAGE.txt > /tmp/symlinks.txt
```

4. Delete any symbolic links listed in `/tmp/symlinks.txt`.

- When upgrading from CDH 4 to CDH 5, Oozie upgrade can take a very long time. For upgrades from CDH 4.3 and higher, you can reduce this time by reducing the amount of history Oozie retains. To reduce Oozie history:
  - Go to the Oozie service.
  - Click the **Configuration** tab.
  - Click **Category > Advanced**.
  - In **Oozie Server Advanced Configuration Snippet (Safety Valve) for oozie-site.xml**, enter the following

```
<property>
<name>oozie.service.PurgeService.older.than</name>
<value>7</value>
</property>
<property>
<name>oozie.service.PurgeService.purge.limit</name>
<value>1000</value>
</property>
```

- For CDH lower than 5.2, enable DEBUG level logging:
  - Click **Category > Logs**.
  - Set **Oozie Server Logging Threshold** to **DEBUG**.
- Click **Save Changes** to commit the changes.
- Restart the Oozie Server role.
- Wait for the purge service to run and finish. By default, the service runs every hour. The purge service emits the following messages in the Oozie server log:

```
STARTED Purge to purge Workflow Jobs older than [7] days, Coordinator Jobs older than [7] days, and Bundlejobs older than [7] days.
ENDED Purge deleted [x] workflows, [y] coordinatorActions, [z] coordinators, [w] bundles
```

- Revert the purge service and log level settings to the default.
- When upgrading from CDH 4 to CDH 5, Hue upgrade can take a very long time if the beeswax\_queryhistory, beeswax\_savedquery, and oozie\_job tables are larger than 1000 records. You can reduce the upgrade time by running a script to reduce the size of the Hue database:

- Stop the Hue service.
- Back up the Hue database.
- Download the [history cleanup script](#) to the host running the Hue Server.
- Run the following as root:
  - parcel installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"
/opt/cloudera/parcels/CDH/share/hue/build/env/bin/hue shell
```

- package installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"
/usr/share/hue/build/env/bin/hue shell
```

- Run the downloaded script in the Hue shell.
- 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:
  - Open the `my.cnf` file for MySQL, search for "default-storage-engine" and note its value.

## Upgrade

### 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.

- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- If using HBase:
  - Run `hbase hbck`.
  - Before you can upgrade HBase from CDH 4 to CDH 5, your HFiles must be upgraded from HFile v1 format to HFile v2, because CDH 5 no longer supports HFile v1. The upgrade procedure itself is different if you are using Cloudera Manager or the command line, but has the same results. The first step is to check for instances of HFile v1 in the HFiles and mark them to be upgraded to HFile v2, and to check for and report about corrupted files or files with unknown versions, which need to be removed manually. The next step is to rewrite the HFiles during the next major compaction. After the HFiles are upgraded, you can continue the upgrade. After the upgrade is complete, you must recompile custom coprocessors and JARs. To check and upgrade the files:
    1. In the Cloudera Admin Console, go to the HBase service and run **Actions > Check HFile Version**.
    2. Check the output of the command in the `stderr` log.

Your output should be similar to the following:

```
Tables Processed:
hdfs://localhost:41020/myHBase/.META.
hdfs://localhost:41020/myHBase/usertable
hdfs://localhost:41020/myHBase/TestTable
hdfs://localhost:41020/myHBase/t

Count of HFileV1: 2
HFileV1:
hdfs://localhost:41020/myHBase/usertable
/fa02dac1f38d03577bd0f7e666f12812/family/249450144068442524
hdfs://localhost:41020/myHBase/usertable
/ecdd3eaeee2d2fcf8184ac025555bb2af/family/249450144068442512

Count of corrupted files: 1
Corrupted Files:
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812/family/1

Count of Regions with HFileV1: 2
Regions to Major Compact:
hdfs://localhost:41020/myHBase/usertable/fa02dac1f38d03577bd0f7e666f12812
hdfs://localhost:41020/myHBase/usertable/ecdd3eaeee2d2fcf8184ac025555bb2af
```

In the example above, you can see that the script has detected two HFile v1 files, one corrupt file and the regions to major compact.

3. Trigger a major compaction on each of the reported regions. This major compaction rewrites the files from HFile v1 to HFile v2 format. To run the major compaction, start HBase Shell and issue the `major_compact` command.

```
$ /usr/lib/hbase/bin/hbase shell
hbase> major_compact 'usertable'
```

You can also do this in a single step by using the `echo` shell built-in command.

```
$ echo "major_compact 'usertable'" | /usr/lib/hbase/bin/hbase shell
```

- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Stop All Services

1. Stop the cluster.

- a. On the **Home > Status** tab, 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.

2. Stop the Cloudera Management Service:

- a. Do one of the following:

- 1. Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
- 2. Select **Actions > Stop**.
- 1. On the **Home > Status** tab, click



to the right of **Cloudera Management Service** and select **Stop**.

- b. Click **Stop** to confirm. The **Command Details** window shows the progress of stopping the roles.

- c. When **Command completed with n/n successful subcommands** appears, the task is complete. Click **Close**.

### Perform Service-Specific Prerequisite Actions

- **HDFS** - Back up HDFS metadata on the NameNode:

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. In the Search field, search for "NameNode Data Directories" and note the value.

## Upgrade

- On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, since each directory is a complete copy. For example, if the NameNode data directory is /data/dfs/nn, do the following as root:

```
# cd /data/dfs/nn  
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
./  
./current/  
./current/fsimage  
./current/fstime  
./current/VERSION  
./current/edits  
./image/  
./image/fsimage
```

If there is a file with the extension *lock* in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, starting by shutting down the NameNode role.

- Back up the Hive and Sqoop metastore databases.
  - For each affected service:
    - If not already stopped, stop the service.
    - Back up the database. See [Backing Up Databases](#) on page 117.

### Uninstall CDH 4

Uninstall CDH 4 on each host as follows:

Operating System	Command
RHEL	\$ sudo yum remove bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc
SLES	\$ sudo zypper remove bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc
Ubuntu or Debian	\$ sudo apt-get purge bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc

### Remove CDH 4 Repository Files

Remove all Cloudera CDH 4 repository files. For example, on a Red Hat or similar system, remove all files in /etc/yum.repos.d that have cloudera as part of the name.



#### Important:

- Before removing the files, make sure you have not added any custom entries that you want to preserve. (To preserve custom entries, back up the files before removing them.)
- Make sure you remove Impala and Search repository files, as well as the CDH repository file.

### Install CDH 5 Components

- Red Hat

- Download and install the "1-click Install" package.
  - Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

**b.** Install the RPM for all RHEL versions:

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

**2.** (Optional) add a repository key:

- **Red Hat/CentOS/Oracle 5**

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

- **Red Hat/CentOS/Oracle 6**

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

**3.** Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

• **SLES**

**1.** Download and install the "1-click Install" package.

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

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

**b.** Install the RPM:

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

**c.** Update your system package index by running:

```
$ sudo zypper refresh
```

## Upgrade

### 2. (Optional) add a repository key:

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

### 3. Install the CDH packages:

```
$ 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 hive-webhcate 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-core spark-master spark-worker spark-history-server spark-python  
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

#### • Ubuntu and Debian

##### 1. Download and install the "1-click Install" package

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

###### b. 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
```

##### 2. Optionally add a repository key:

###### • Debian Wheezy

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

###### • Ubuntu Precise

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

### 3. Install the CDH packages:

```
$ 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-webhcate 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-core spark-master spark-worker spark-history-server spark-python
sqoop sqoop2 whirr
```



**Note:** Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.

## Run the Upgrade Wizard

1. Log into the Cloudera Manager Admin console.
2. From the **Home > Status** tab, click



next to the cluster name and select **Upgrade Cluster**. The Upgrade Wizard starts.

3. If the option to pick between packages and parcels displays, select the **Use Parcels** option.
4. In the **Choose CDH Version (Parcels)** field, select the CDH version. If there are no qualifying parcels, click the **Modify the Remote Parcel Repository URLs** link to go to the [Parcel Configuration Settings](#) on page 63 where you can add the locations of parcel repositories. Click **Continue**.
5. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**.
6. Cloudera Manager checks that hosts have the correct software installed. Click **Continue**.
7. The selected parcels are downloaded and distributed. Click **Continue**.
8. The Host Inspector runs and displays the CDH version on the hosts. Click **Continue**.
9. Click **Continue**. Cloudera Manager performs all service upgrades and restarts the cluster.
10. The wizard reports the result of the upgrade. Choose one of the following:

- Leave **OK, set up YARN and import existing configuration from my MapReduce service** checked.
  1. Click **Continue** to proceed. Cloudera Manager stops the YARN service (if running) and its dependencies.
  2. Click **Continue** to proceed. The next page indicates some additional configuration required by YARN.
  3. Verify or modify the configurations and click **Continue**. The Switch Cluster to MR2 step proceeds.
  4. When all steps have completed, click **Continue**.
- Deselect **OK, set up YARN and import existing configuration from my MapReduce service**.

11. Click **Finish** to return to the **Home > Status** tab.

12. (Optional) Remove the MapReduce service.

- a. In the MapReduce row, right-click



and select **Delete**. Click **Delete** to confirm.

## Recover from Failed Steps



**Note:** If you encounter errors during these steps:

- If the converting configuration parameters step fails, Cloudera Manager rolls back all configurations to CDH 4. Fix any reported problems and retry the upgrade.
- If the upgrade command fails at any point after the convert configuration step, there is no retry support in Cloudera Manager. You must first correct the error, then manually re-run the individual commands. You can view the remaining commands in the Recent Commands page.
- If the HDFS upgrade metadata step fails, you cannot revert back to CDH 4 unless you restore a backup of Cloudera Manager.

## Upgrade

The actions performed by the upgrade wizard are listed in [Upgrade Wizard Actions](#) on page 680. If any of the steps in the **Command Progress** screen fails, complete the step as described in that section before proceeding.

### Restart the Reports Manager Role

1. Do one of the following:
  - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
  - On the **Home > Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Instances** tab.
3. Check the checkbox next to **Reports Manager**.
4. Select **Actions for Selected > Restart** and then **Restart** to confirm.

### Recompile JARs

- **MapReduce and YARN** - Recompile JARs used in MapReduce applications. For further information, see [For MapReduce Programmers: Writing and Running Jobs](#) on page 234.
- **HBase** - Recompile coprocessor and custom JARs used by HBase applications.

### Finalize the HDFS Metadata Upgrade

Finalize the HDFS metadata upgrade. To determine when finalization is warranted, run important workloads and ensure they are successful.

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **NameNode** instance.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Upgrade Wizard Actions

Do the steps in this section only if the upgrade wizard reports a failure.

### Upgrade HDFS Metadata

1. Start the ZooKeeper service.
2. Go to the HDFS service.
3. Select **Actions > Upgrade HDFS Metadata** and click **Upgrade HDFS Metadata** to confirm.

### Upgrade HBase

1. Go to the HBase service.
2. Select **Actions > Upgrade HBase** and click **Upgrade HBase** to confirm.

### Upgrade the Hive Metastore Database

1. Go to the Hive service.
2. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
3. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade Oozie

1. Go to the Oozie service.
2. Select **Actions > Upgrade Database** and click **Upgrade Database** to confirm.
3. Start the Oozie service.

4. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Start Cluster Services

1. On the **Home > Status** tab, click



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

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

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

### Deploy Client Configuration Files

1. On the Home page, click



or

the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading CDH 4

Use the instructions in this section to upgrade to a higher CDH 4 minor release, that is from CDH 4.a.x to CDH 4. b.y. For example, CDH 4.6.0 to CDH 4.7.1.

You can upgrade to CDH 4.1.3 (or higher) within the Cloudera Manager Admin Console, using parcels and an upgrade wizard. This vastly simplifies the upgrade process. Electing to upgrade using packages means that future upgrades will still need to be done manually. Upgrading to a CDH 4 release prior to CDH 4.1.3 is possible using packages, though upgrading to a more current release is strongly recommended.

If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.



**Important:** The following instructions describe how to upgrade from a CDH 4 release to a newer CDH 4 release in a Cloudera Manager deployment. If you are running CDH 3, you must upgrade to CDH 4 using the instructions at [Upgrading CDH 3 to CDH 4 in a Cloudera Managed Deployment](#).

To upgrade from CDH 4 to CDH 5, see [Upgrading CDH 4 to CDH 5](#) on page 660.

### Before You Begin

- Before upgrading, be sure to read about the latest Incompatible Changes and Known Issues and Workarounds in the [CDH 4 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x

## Upgrade

Target CDH Version	Minimum Cloudera Manager Version
5.4.1	5.4.x

- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

### Upgrade Procedures

**Important:**

- **Impala** - If you have CDH 4.1.x with Cloudera Impala installed, and you plan to upgrade to CDH 4.2 or higher, you must also upgrade Impala to version 1.2.1 or higher. With a parcel installation you can download and activate both parcels before you proceed to restart the cluster. You will need to change the remote parcel repo URL to point to the location of the released product as described in the upgrade procedures referenced below.
- **HBase** - In CDH 4.1.x, an HBase table could have an owner that had full administrative permissions on the table. The owner construct was removed as of CDH 4.2.0, and the code now relies exclusively on entries in the ACL table. Since table owners do not have an entry in this table, their permissions are removed on upgrade from CDH 4.1.x to CDH 4.2.0 or higher. If you are upgrading from CDH 4.1.x to CDH 4.2 or higher, and using HBase, you must add permissions for HBase owner users to the HBase ACL table before you perform the upgrade. See the Known Issues in the CDH 4 Release Notes, specifically the item "Must explicitly add permissions for owner users before upgrading from 4.1.x" in the [Known Issues in Apache HBase](#) section.
- **Hive** - Hive has undergone major version changes from CDH 4.0 to 4.1 and between CDH 4.1 and 4.2. (CDH 4.0 had Hive 0.8.0, CDH 4.1 used Hive 0.9.0, and 4.2 or higher has 0.10.0). This requires you to manually back up and upgrade the Hive metastore database when upgrading between major Hive versions. If you are upgrading from a version of CDH 4 prior to CDH 4.2 to a newer CDH 4 version, you must follow the steps for upgrading the metastore included in the upgrade procedures referenced below.

### Upgrading CDH 4 Using Parcels

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

You can upgrade your CDH 4 cluster to a higher minor version of CDH 4 using parcels from within the Cloudera Manager Admin Console. Your current CDH 4 cluster can have been installed with either parcels or packages. The new version will use parcels.

The following procedure requires cluster downtime. If you use parcels, have a Cloudera Enterprise license, and have enabled HDFS high availability, you can perform a [rolling upgrade](#) that lets you avoid cluster downtime.

**Important:**

- **Impala** - If you have CDH 4.1.x with Cloudera Impala installed, and you plan to upgrade to CDH 4.2 or higher, you must also upgrade Impala to version 1.2.1 or higher. With a parcel installation you can download and activate both parcels before you proceed to restart the cluster. You will need to change the remote parcel repo URL to point to the location of the released product as described in the upgrade procedures referenced below.
- **HBase** - In CDH 4.1.x, an HBase table could have an owner that had full administrative permissions on the table. The owner construct was removed as of CDH 4.2.0, and the code now relies exclusively on entries in the ACL table. Since table owners do not have an entry in this table, their permissions are removed on upgrade from CDH 4.1.x to CDH 4.2.0 or higher. If you are upgrading from CDH 4.1.x to CDH 4.2 or higher, and using HBase, you must add permissions for HBase owner users to the HBase ACL table before you perform the upgrade. See the Known Issues in the CDH 4 Release Notes, specifically the item "Must explicitly add permissions for owner users before upgrading from 4.1.x" in the [Known Issues in Apache HBase](#) section.
- **Hive** - Hive has undergone major version changes from CDH 4.0 to 4.1 and between CDH 4.1 and 4.2. (CDH 4.0 had Hive 0.8.0, CDH 4.1 used Hive 0.9.0, and 4.2 or higher has 0.10.0). This requires you to manually back up and upgrade the Hive metastore database when upgrading between major Hive versions. If you are upgrading from a version of CDH 4 prior to CDH 4.2 to a newer CDH 4 version, you must follow the steps for upgrading the metastore included in the upgrade procedures referenced below.

To upgrade your version of CDH using parcels, the steps are as follows.

#### Before You Begin

- Before upgrading, be sure to read about the latest Incompatible Changes and Known Issues and Workarounds in the [CDH 4 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [Incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Upgrade

### Download, Distribute, and Activate Parcels

1. In the Cloudera Manager Admin Console, click the Parcels indicator in the top navigation bar (or ) to go to the Parcels page.
2. In the parcels page, click **Download** for the version(s) you want to download. 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 parcel repos available through the [parcel settings](#) page. If your Cloudera Manager server does not have Internet access, you can obtain the required parcel file(s) and put them into the local repository. See [Creating and Using a Remote Parcel Repository for Cloudera Manager](#) on page 172 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**. You are asked if you want to restart the cluster. *Do not restart the cluster at this time.*
6. Click **Close**.

### Upgrade the Hive Metastore Database

Required if you are upgrading from an earlier version of CDH 4 to CDH 4.2 or higher.

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

### Restart the Services

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Restart**.
2. Click the **Restart** button in the confirmation pop-up that appears. The **Command Details** window shows the progress of starting services.

### Deploy Client Configuration Files

1. On the **Home > Status** tab, click  to the right of the cluster name and select **Deploy Client Configuration**.
2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

### Remove the Previous CDH Version Packages and Refresh Symlinks

If your previous installation of CDH was done using packages, remove those packages on all hosts on which you installed the parcels and refresh the symlinks so that clients will run the new software versions. *Skip this step if your previous installation was using parcels.*

**1.** If Hue is configured to use SQLite as its database:

- a. Stop the Hue service.
- b. Back up the desktop.db to a temporary location before deleting the old Hue Common package. The location of the database can be found in the Hue service **Configuration** tab under **Service > Database > Hue's Database Directory**.



**Important:** Removing the Hue Common package will remove your Hue database; if you do not back it up you may lose all your Hue user account information.

**2.** Uninstall the CDH packages on each host:

- **Not including Impala and Search**

Operating System	Command
RHEL	\$ sudo yum remove hadoop hue-common bigtop-jsvc bigtop-tomcat
SLES	\$ sudo zypper remove hadoop hue-common bigtop-jsvc bigtop-tomcat
Ubuntu or Debian	\$ sudo apt-get purge hadoop hue-common bigtop-jsvc bigtop-tomcat

- **Including Impala and Search**

Operating System	Command
RHEL	\$ sudo yum remove hadoop hue-common impala-shell solr-server 'bigtop-*'
SLES	\$ sudo zypper remove hadoop hue-common impala-shell solr-server 'bigtop-*'
Ubuntu or Debian	\$ sudo apt-get purge hadoop hue-common impala-shell solr-server 'bigtop-*'

**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
```

#### Restore Backed up Hue Database

Restore the backup you created in [Remove the Previous CDH Version Packages and Refresh Symlinks](#) on page 684.

1. Go to the Hue service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Copy the backup from the temporary location to the newly created Hue database directory:  
`/opt/cloudera/parcels/CDH-4.x.0-x.cdh4.x.0.p0.xx/share/hue/desktop`.
4. Restart the Hue service.

#### Upgrading CDH 4 Using Packages

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

If you originally used Cloudera Manager to install your CDH service using packages, you can upgrade to a higher minor version of CDH 4 either using packages or parcels. Parcels is the preferred and recommended way to upgrade, as the upgrade wizard provided for parcels handles the upgrade process almost completely automatically.

**Important:**

- **Impala** - If you have CDH 4.1.x with Cloudera Impala installed, and you plan to upgrade to CDH 4.2 or higher, you must also upgrade Impala to version 1.2.1 or higher. With a parcel installation you can download and activate both parcels before you proceed to restart the cluster. You will need to change the remote parcel repo URL to point to the location of the released product as described in the upgrade procedures referenced below.
- **HBase** - In CDH 4.1.x, an HBase table could have an owner that had full administrative permissions on the table. The owner construct was removed as of CDH 4.2.0, and the code now relies exclusively on entries in the ACL table. Since table owners do not have an entry in this table, their permissions are removed on upgrade from CDH 4.1.x to CDH 4.2.0 or higher. If you are upgrading from CDH 4.1.x to CDH 4.2 or higher, and using HBase, you must add permissions for HBase owner users to the HBase ACL table before you perform the upgrade. See the Known Issues in the CDH 4 Release Notes, specifically the item "Must explicitly add permissions for owner users before upgrading from 4.1.x" in the [Known Issues in Apache HBase](#) section.
- **Hive** - Hive has undergone major version changes from CDH 4.0 to 4.1 and between CDH 4.1 and 4.2. (CDH 4.0 had Hive 0.8.0, CDH 4.1 used Hive 0.9.0, and 4.2 or higher has 0.10.0). This requires you to manually back up and upgrade the Hive metastore database when upgrading between major Hive versions. If you are upgrading from a version of CDH 4 prior to CDH 4.2 to a newer CDH 4 version, you must follow the steps for upgrading the metastore included in the upgrade procedures referenced below.

To upgrade your version of CDH using packages, the steps are as follows.

### Before You Begin

- Before upgrading, be sure to read about the latest Incompatible Changes and Known Issues and Workarounds in the [CDH 4 Release Notes](#).
- Read the [Cloudera Manager 5 Release Notes](#).
- Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

Target CDH Version	Minimum Cloudera Manager Version
5.0.5	5.0.x
5.1.4	5.1.x
5.4.1	5.4.x

- Run the [Host Inspector](#) and fix every issue.
- If using security, run the [Security Inspector](#).
- Whenever upgrading Impala, whether in CDH or a standalone parcel or package, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions or in case of any problems, check against the full list of [Impala keywords](#).
- Run `hdfs fsck /` and `hdfs dfsadmin -report` and fix every issue.
- Run `hbase hbck`.
- Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
- To avoid lots of alerts during the upgrade process, you can enable [maintenance mode](#) on your cluster before you start the upgrade. This will stop email alerts and SNMP traps from being sent, but will not stop checks and configuration validations from being made. Be sure to exit maintenance mode when you have finished the upgrade in order to re-enable Cloudera Manager alerts.

## Upgrade Unmanaged Components

Upgrading unmanaged components is a process that is separate from upgrading managed components. Upgrade the unmanaged components before proceeding to upgrade managed components. Components that you might have installed that are not managed by Cloudera Manager include:

- Pig
- Whirr
- Mahout

For information on upgrading these unmanaged components, see [CDH 4 Installation Guide](#).

## Upgrade Managed Components

Use one of the following strategies to upgrade CDH 4:

- Use your operating system's package management tools to update all packages to the latest version using standard repositories. This approach works well because it minimizes the amount of configuration required and uses the simplest commands. Be aware that this can take a considerable amount of time if you have not upgraded the system recently. To update all packages on your system, use the following command:

Operating System	Command
RHEL	\$ sudo yum update
SLES	\$ sudo zypper up
Ubuntu or Debian	\$ sudo apt-get upgrade

- Use the `cloudera.com` repository that is added during a typical installation, only updating Cloudera components. This limits the scope of updates to be completed, so the process takes less time, however this process will not work if you created and used a custom repository. To install the new version, you can upgrade from the Cloudera repository by adding an entry to your operating system's package management configuration file. The repository location varies by operating system:

Operating System	Configuration File Repository Entry
Red Hat	<a href="https://archive.cloudera.com/cdh4/redhat/6/x86_64/cdh/4/">https://archive.cloudera.com/cdh4/redhat/6/x86_64/cdh/4/</a>
SLES	<a href="https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/4/">https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/4/</a>
Debian Squeeze	[arch=amd64] <a href="https://archive.cloudera.com/cdh4/debian/squeeze">https://archive.cloudera.com/cdh4/debian/squeeze</a> squeeze-cdh4 contrib
Ubuntu Lucid	[arch=amd64] <a href="https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh">https://archive.cloudera.com/cdh4/ubuntu/lucid/amd64/cdh</a> lucid-cdh4 contrib
Ubuntu Precise	[arch=amd64] <a href="https://archive.cloudera.com/cdh4/ubuntu/precise/amd64/cdh">https://archive.cloudera.com/cdh4/ubuntu/precise/amd64/cdh</a> precise-cdh4 contrib

For example, under Red Hat, to upgrade from the Cloudera repository you can run commands such as the following on the CDH host to update only CDH:

```
$ sudo yum clean all
$ sudo yum update 'cloudera-*'
```



### Note:

- cloudera-cdh4 is the name of the repository on your system; the name is usually in square brackets on the first line of the repo file, in this example  
`/etc/yum.repos.d/cloudera-cdh4.repo:`

```
[chris@ca727 yum.repos.d]$ more cloudera-cdh4.repo
[cloudera-cdh4]
...
```

- `yum clean all` cleans up yum's cache directories, ensuring that you download and install the latest versions of the packages. – If your system is not up to date, and any underlying system components need to be upgraded before this yum update can succeed, yum will tell you what those are.

On a SLES system, use commands like this to clean cached repository information and then update only the CDH components. For example:

```
$ sudo zypper clean --all
$ sudo zypper up -r https://archive.cloudera.com/cdh4/sles/11/x86_64/cdh/4
```

To verify the URL, open the Cloudera repo file in `/etc/zypp/repos.d` on your system (for example `/etc/zypp/repos.d/cloudera-cdh4.repo`) and look at the line beginning

```
baseurl=
```

Use that URL in your `sudo zypper up -r` command.

On a Debian/Ubuntu system, use commands like this to clean cached repository information and then update only the CDH components. First:

```
$ sudo apt-get clean
```

After cleaning the cache, use one of the following upgrade commands to upgrade CDH.

Precise:

```
$ sudo apt-get upgrade -t precise-cdh4
```

Lucid:

```
$ sudo apt-get upgrade -t lucid-cdh4
```

Squeeze:

```
$ sudo apt-get upgrade -t squeeze-cdh4
```

- Use a custom repository. This process can be more complicated, but enables updating CDH components for hosts that are not connected to the Internet. You can create your own repository, as described in [Understanding Custom Installation Solutions](#) on page 170. Creating your own repository is necessary if you are upgrading a cluster that does not have access to the Internet.

If you used a custom repository to complete the installation of your current files and now you want to update using a custom repository, the details of the steps to complete the process are variable. In general, begin by updating any existing custom repository that you will use with the installation files you wish to use. This can be completed in a variety of ways. For example, you might use `wget` to copy the necessary installation files. Once the installation files have been updated, use the custom repository you established for the initial installation to update CDH.

OS	Command
RHEL	<p>Ensure you have a custom repo that is configured to use your internal repository. For example, if you could have custom repo file in /etc/yum.conf.d/ called cdh_custom.repo in which you specified a local repository. In such a case, you might use the following commands:</p> <pre>\$ sudo yum clean all \$ sudo yum update 'cloudera-*'</pre>
SLES	<p>Use commands such as the following to clean cached repository information and then update only the CDH components:</p> <pre>\$ sudo zypper clean --all \$ sudo zypper up -r http://internalserver.example.com/path_to_cdh_repo</pre>
Ubuntu or Debian	<p>Use a command that targets upgrade of your CDH distribution using the custom repository specified in your apt configuration files. These files are typically either the /etc/apt/apt.conf file or in various files in the /etc/apt/apt.conf.d/ directory. Information about your custom repository must be included in the repo files. The general form of entries in Debian/Ubuntu is:</p> <pre>deb http://server.example.com/directory/ dist-name pool</pre> <p>For example, the entry for the default repo is:</p> <pre>deb http://us.archive.ubuntu.com/ubuntu/ precise universe</pre> <p>On a Debian/Ubuntu system, use commands such as the following to clean cached repository information and then update only the CDH components:</p> <pre>\$ sudo apt-get clean \$ sudo apt-get upgrade -t your_cdh_repo</pre>

### Upgrade the Hive Metastore Database

Required if you are upgrading from an earlier version of CDH 4 to CDH 4.2 or higher.

1. Go to the Hive service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Hive Metastore Database Schema** and click **Upgrade Hive Metastore Database Schema** to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.

### Upgrade the Oozie ShareLib

1. Go to the Oozie service.
2. Select **Actions > Start** and click **Start** to confirm.
3. Select **Actions > Install Oozie ShareLib** and click **Install Oozie ShareLib** to confirm.

### Upgrade Sqoop

1. Go to the Sqoop service.
2. Select **Actions > Stop** and click **Stop** to confirm.
3. Select **Actions > Upgrade Sqoop** and click **Upgrade Sqoop** to confirm.

## Upgrade

### Restart the Services

1. On the **Home > Status** tab, click



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

2. Click the **Restart** button in the confirmation pop-up that appears. The **Command Details** window shows the progress of starting services.

### Configure Cluster CDH Version for Package Installs

If you have installed CDH as a package, after an installation or upgrade, make sure that the cluster CDH version matches the package CDH version, using the procedure in [Configuring the CDH Version of a Cluster](#) on page 524. If the cluster CDH version does not match the package CDH version, Cloudera Manager incorrectly enables and disables service features based on the cluster's configured CDH version.

### Deploy Client Configuration Files

1. On the **Home > Status** tab, click



to the right of the cluster name and select **Deploy Client Configuration**.

2. Click the **Deploy Client Configuration** button in the confirmation pop-up that appears.

## Upgrading CDH 3



#### Warning:

- Cloudera Manager 4 and CDH 4 have reached End of Maintenance (EOM) on August 9, 2015. Cloudera does not support or provide updates for Cloudera Manager 4 and CDH 4 releases.
- Cloudera Manager 3 and CDH 3 have reached End of Maintenance (EOM) on June 20, 2013. Cloudera does not support or provide updates for Cloudera Manager 3 and CDH 3 releases.

To upgrade CDH 3 to CDH 4 with Cloudera Manager 4, follow the instructions at [Upgrading CDH 3 to CDH 4 in a Cloudera Manager Deployment](#).

## Upgrading Unmanaged CDH Using the Command Line

This section provides instructions for upgrading CDH to the latest release, using the command line rather than Cloudera Manager. (A cluster you are not managing by means of Cloudera Manager is referred to as an **unmanaged** cluster.)



#### Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

To proceed with the upgrade, choose one of the following sections:

## Upgrading from CDH 4 to CDH 5



**Note:** If you are using Cloudera Manager to manage CDH, *do not* use the instructions in this section.

- If you are running Cloudera Manager 4, you must upgrade to Cloudera Manager 5 first, as Cloudera Manager 4 cannot manage CDH 5; see [Upgrading Cloudera Manager](#) on page 466.
- Follow directions in [Upgrading CDH 4 to CDH 5](#) on page 660 to upgrade CDH 4 to CDH 5 in a Cloudera Manager deployment.

Use the following information and instructions to upgrade to the latest CDH 5 release from a CDH 4 release:

- [Before You Begin Upgrading to CDH 5 Using the Command Line](#) on page 691
- [Upgrading to CDH 5](#) on page 693



**Important:** This involves uninstalling the CDH 4 packages and installing the CDH 5 packages.



**Note:**

If you are migrating from MapReduce v1 (MRv1) to MapReduce v2 (MRv2, YARN), see [Migrating from MapReduce \(MRv1\) to MapReduce \(MRv2\)](#) on page 234 for important information and instructions.

### Before You Begin Upgrading to CDH 5 Using the Command Line

Before upgrading, be sure to read about the latest [Incompatible Changes](#) and [Known Issues in CDH 5](#) in the [CDH 5 Release Notes](#). If you are currently running MRv1, you should read [CDH 5 and MapReduce](#) on page 233 before proceeding.



**Warning:**

It's particularly important that you read the [Install and Upgrade Known Issues](#).

### Plan Downtime

If you are upgrading a cluster that is part of a production system, be sure to plan ahead. As with any operational work, be sure to reserve a maintenance window with enough extra time allotted in case of complications. The Hadoop upgrade process is well understood, but it is best to be cautious. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.

### Install Java 1.7

CDH 5 [requires Java 1.7](#) or higher. See [Upgrading to Oracle JDK 1.7](#) on page 738, and make sure you have read the [Install and Upgrade Known Issues](#) before you proceed with the upgrade.

### Delete Symbolic Links in HDFS

If there are symbolic links in HDFS when you upgrade from CDH 4 to CDH 5, the upgrade will fail and you will have to downgrade to CDH 4, delete the symbolic links, and start over. To prevent this, proceed as follows.

To check for symbolic links in CDH 4 HDFS:

1. cd to the directory on the NameNode that contains the latest `fsimage` The location of this directory is specified as the value of `dfs.namenode.name.dir` (or `dfs.name.dir`) in `hdfs-site.xml`.
2. Use a command such as the following to write out the path names in the `fsimage`:

```
$ hdfs oiv -i FSIMAGE -o /tmp/YYYY-MM-DD_FSIMAGE.txt
```

## Upgrade

3. Use a command such as the following to find the path names of any symbolic links listed in `/tmp/YYYY-MM-DD_FSIMAGE.txt` and write them out to the file `/tmp/symlinks.txt`:

```
$ grep -- "->" /tmp/YYYY-MM-DD_FSIMAGE.txt > /tmp/symlinks.txt
```

4. Delete any symbolic links listed in `/tmp/symlinks.txt`.

### Check Hue Table Sizes and Cleanup if Necessary

When upgrading from CDH 4 to CDH 5, Hue upgrade can take a very long time if the beeswax\_queryhistory, beeswax\_savedquery, and oozie\_job tables are larger than 1000 records. You can reduce the upgrade time by running a script to reduce the size of the Hue database:

1. Stop the Hue service.
2. Back up the Hue database.
3. Download the [history cleanup script](#) to the host running the Hue Server.
4. Run the following as root:
  - **parcel installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"/opt/cloudera/parcels/CDH/share/hue/build/env/bin/hue shell
```

- **package installation**

```
export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process | grep HUE| sort -n | tail -1 `"/usr/share/hue/build/env/bin/hue shell
```

5. Run the downloaded script in the Hue shell.

### Considerations for Secure Clusters

If you are upgrading a cluster that has Kerberos security enabled, you must do the following:

- Before starting the upgrade, make sure your installation is properly configured according to the instructions in the installation and configuration sections of the [Cloudera Security](#) guide.
- Before shutting down Hadoop services, put the NameNode into safe mode and perform a saveNamespace operation; see the instructions on [backing up the metadata](#).

### High Availability

In CDH 5 you can configure high availability both for the NameNode and the JobTracker or Resource Manager.

- For more information and instructions on setting up a new HA configuration, see [High Availability](#).



#### Important:

If you decide to configure [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`. After completing the [HDFS HA software configuration](#), follow the installation instructions under [Deploying HDFS High Availability](#).

- To upgrade an existing configuration, follow the instructions under [Upgrading to CDH 5](#) on page 693.

## Upgrading to CDH 5



### Note: Are you on the right page?

Use the instructions on this page only to upgrade from CDH 4.

To upgrade from an earlier CDH 5 release to the latest version, use the appropriate instructions:

- [Upgrading from a Release Earlier than CDH 5.4.0 to the Latest Release](#) on page 723



### Important:

1. To upgrade from CDH 4, you must uninstall CDH 4, and then install CDH 5. Make sure you allow sufficient time for this, and do the necessary backup and preparation as described below.
2. If you have configured HDFS HA with NFS shared storage, do not proceed. This configuration is not supported on CDH 5; Quorum-based storage is the only supported HDFS HA configuration on CDH 5. [Unconfigure](#) your NFS shared storage configuration before you attempt to upgrade.



### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

## Back Up Configuration Data and Stop Services

1. Put the NameNode into safe mode and save the `fsimage`:

- a. Put the NameNode (or active NameNode in an HA configuration) into safe mode:

```
$ sudo -u hdfs hdfs dfsadmin -safemode enter
```

- b. Perform a `saveNamespace` operation:

```
$ sudo -u hdfs hdfs dfsadmin -saveNamespace
```

This will result in a new `fsimage` being written out with no edit log entries.

- c. With the NameNode still in safe mode, shut down all services as instructed below.

2. For each component you are using, back up configuration data, databases, and other important files.

3. Shut down the Hadoop services across your entire cluster:

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

4. Check each host to make sure that there are no processes running as the `hdfs` or `mapred` users from root:

```
# ps -afe | grep java
```

## Upgrade

### Back up the HDFS Metadata



#### Important:

Do this step when you are sure that all Hadoop services have been shut down. **It is particularly important that the NameNode service is not running so that you can make a consistent backup.**

#### To back up the HDFS metadata on the NameNode machine:



#### Note:

- Cloudera recommends backing up HDFS metadata on a regular basis, as well as before a major upgrade.
- `dfs.name.dir` is deprecated but still works; `dfs.namenode.name.dir` is preferred. This example uses `dfs.name.dir`.

#### 1. Find the location of your `dfs.name.dir` (or `dfs.namenode.name.dir`); for example:

```
$ grep -C1 dfs.name.dir /etc/hadoop/conf/hdfs-site.xml
```

You should see something like this:

```
<property>
<name>dfs.name.dir</name>
<value>/mnt/hadoop/hdfs/name</value>
```

#### 2. Back up the directory. The path inside the `<value>` XML element is the path to your HDFS metadata. If you see a comma-separated list of paths, there is no need to back up all of them; they store the same data. Back up the first directory, for example, by using the following commands:

```
$ cd /mnt/hadoop/hdfs/name
# tar -cvf /root/nn_backup_data.tar .
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```



**Warning:** If you see a file containing the word `lock`, the NameNode is probably still running.  
Repeat the preceding steps, starting by shutting down the Hadoop services.

### Uninstall the CDH 4 Version of Hadoop



**Warning:** Do not proceed before you have backed up the HDFS metadata, and the files and databases for the individual components, as instructed in the previous steps.

#### To uninstall Hadoop:

Run this command on each host:

#### On Red Hat-compatible systems:

```
$ sudo yum remove bigtop-utils bigtop-jsvc bigtop-tomcat sqoop2-client hue-common solr
```

**On SLES systems:**

```
$ sudo zypper remove bigtop-utils bigtop-jsvc bigtop-tomcat sqoop2-client hue-common
solr
```

**On Ubuntu systems:**

```
sudo apt-get remove bigtop-utils bigtop-jsvc bigtop-tomcat sqoop2-client hue-common solr
```

**Remove CDH 4 Repository Files**

Remove all Cloudera CDH 4 repository files. For example, on a Red Hat or similar system, remove all files in /etc/yum.repos.d that have cloudera as part of the name.

**Important:**

- Before removing the files, make sure you have not added any custom entries that you want to preserve. (To preserve custom entries, back up the files before removing them.)
- Make sure you remove Impala and Search repository files, as well as the CDH repository file.

**Download the Latest Version of CDH 5****Note:**

For instructions on how to add a CDH 5 yum repository or build your own CDH 5 yum repository, see [Installing the Latest CDH 5 Release](#) on page 220.

**On Red Hat-compatible systems:****1. Download the CDH 5 "1-click Install" package (or RPM).**

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

**2. Install the RPM for all RHEL versions:**

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

**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

Now (optionally) add a repository key:

## Upgrade

- 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
```

### On SLES systems:

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

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

2. Install the RPM:

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

3. Update your system package index by running:

```
$ sudo zypper refresh
```



#### Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

### Now (optionally) add a repository key:

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

### On Ubuntu and Debian systems:

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

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: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo apt-get update
```

Now (optionally) add a repository key:

- **For Ubuntu Precise systems:**

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

- **For Debian Wheezy systems:**

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

### Install CDH 5 with YARN



**Note:** Skip this step and go to [Install CDH 5 with MRv1](#) on page 698 if you intend to use *only* MRv1.

#### 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 or JobTracker.

Follow instructions under [ZooKeeper Installation](#).

#### 2. Install each type of daemon package on the appropriate systems(s), as follows.

Where to install	Install commands
<b>Resource Manager host</b> (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>
<b>NameNode host(s)</b> 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 update; sudo apt-get install hadoop-hdfs-namenode</code>
<b>Secondary NameNode host</b> (if used) running:	

## Upgrade

Where to install	Install commands
<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 update; sudo apt-get install hadoop-hdfs-secondarynamenode</code>
<b>All cluster hosts except the Resource Manager running:</b>	
<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 clean --all; sudo zypper install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get update; sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
<b>One host in the cluster running:</b>	
<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 update; sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<b>All client hosts, running:</b>	
<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 update; 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.

### Install CDH 5 with MRv1



#### Note:

Skip this step if you intend to use *only* YARN. If you are installing both YARN and MRv1, you can skip any packages you have already installed in [Step 6a](#).

### To install CDH 5 with MRv1:



**Note:** If you are also installing YARN, you can skip any packages you have already installed in [Step 6a](#).

## 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 or JobTracker.

Follow instructions under [ZooKeeper Installation](#).

## 2. Install each type of daemon package on the appropriate system(s), as follows.

Where to install	Install commands
<b>JobTracker host running:</b>	
<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>
<b>NameNode host(s) running:</b>	
<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 update; sudo apt-get install hadoop-hdfs-namenode</code>
<b>Secondary NameNode host (if used) running:</b>	
<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 update; sudo apt-get install hadoop-hdfs-secondarynamenode</code>
<b>All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts, running:</b>	
<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>

Where to install	Install commands
<i>Ubuntu or Debian</i>	<code>sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
<b>All client hosts</b> , 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 update; sudo apt-get install hadoop-client</code>

### Copy the CDH 5 Logging File

Copy over the `log4j.properties` file to your custom directory on each node in the cluster; for example:

```
$ cp /etc/hadoop/conf.empty/log4j.properties /etc/hadoop/conf.my_cluster/log4j.properties
```

### In an HA Deployment, Upgrade and Start the Journal Nodes

1. Install the JournalNode daemons on each of the machines where they will run.

#### To install JournalNode on Red Hat-compatible systems:

```
$ sudo yum install hadoop-hdfs-journalnode
```

#### To install JournalNode on Ubuntu and Debian systems:

```
$ sudo apt-get install hadoop-hdfs-journalnode
```

#### To install JournalNode on SLES systems:

```
$ sudo zypper install hadoop-hdfs-journalnode
```

2. Start the JournalNode daemons on each of the machines where they will run:

```
sudo service hadoop-hdfs-journalnode start
```

Wait for the daemons to start before proceeding to the next step.



**Important:** The JournalNodes must be up and running CDH 5 before you proceed.

## Upgrade the HDFS Metadata



### Note:

What you do in this step differs depending on whether you are upgrading an HDFS HA deployment using Quorum-based storage, or a non-HA deployment using a secondary NameNode. (If you have an HDFS HA deployment using NFS storage, do not proceed; you cannot upgrade that configuration to CDH 5. [Unconfigure](#) your NFS shared storage configuration before you attempt to upgrade.)

- For an HA deployment, do sub-steps 1, 2, and 3 below.
- For a non-HA deployment, do sub-steps 1, 3, and 4 below.

1. To upgrade the HDFS metadata, run the following command on the NameNode. If HA is enabled, do this on the *active NameNode only*, and make sure the JournalNodes have been upgraded to CDH 5 and are up and running before you run the command.

```
$ sudo service hadoop-hdfs-namenode upgrade
```



**Important:** In an HDFS HA deployment, it is critically important that you do this on only one NameNode.

You can watch the progress of the upgrade by running:

```
$ sudo tail -f /var/log/hadoop-hdfs/hadoop-hdfs-namenode-<hostname>.log
```

Look for a line that confirms the upgrade is complete, such as:

```
/var/lib/hadoop-hdfs/cache/hadoop/dfs/<name> is complete
```



**Note:** The NameNode upgrade process can take a while depending on how many files you have.

2. Do this step only in an HA configuration. Otherwise skip to starting up the DataNodes.

Wait for NameNode to exit safe mode, and then re-start the standby NameNode.

- If Kerberos is enabled:

```
$ kinit -kt /path/to/hdfs.keytab hdfs/<fully.qualified.domain.name@YOUR-REALM.COM> &&
hdfs namenode -bootstrapStandby
```

```
$ sudo service hadoop-hdfs-namenode start
```

- If Kerberos is not enabled:

```
$ sudo -u hdfs hdfs namenode -bootstrapStandby
$ sudo service hadoop-hdfs-namenode start
```

For more information about the `haadmin -failover` command, see [Administering an HDFS High Availability Cluster](#).

3. Start up the DataNodes:

On each DataNode:

```
$ sudo service hadoop-hdfs-datanode start
```

4. Do this step only in a non-HA configuration. Otherwise skip to starting YARN or MRv1.

## Upgrade

Wait for NameNode to exit safe mode, and then start the Secondary NameNode.

- a. To check that the NameNode has exited safe mode, look for messages in the log file, or the NameNode's web interface, that say "...no longer in safe mode."
- b. To start the Secondary NameNode (if used), enter the following command on the Secondary NameNode host:

```
$ sudo service hadoop-hdfs-secondarynamenode start
```

- c. To complete the cluster upgrade, follow the remaining steps below.

### Start YARN or MapReduce MRv1

You are now ready to start and test MRv1 or YARN.

For YARN	or For MRv1
<a href="#">Start YARN and the MapReduce JobHistory Server</a>	<a href="#">Start MRv1</a>
<a href="#">Verify basic cluster operation</a>	<a href="#">Verify basic cluster operation</a>

### Start MapReduce with YARN



**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. Steps 10a and 10b are mutually exclusive.

After you have verified HDFS is operating correctly, you are ready to start YARN. First, create directories and set the correct permissions.



**Note:** For more information see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).

Create a history directory and set permissions; for example:

```
sudo -u hdfs hadoop fs -mkdir /user/history
sudo -u hdfs hadoop fs -chmod -R 1777 /user/history
sudo -u hdfs hadoop fs -chown yarn /user/history
```

Create the /var/log/hadoop-yarn directory and set ownership:

```
$ sudo -u hdfs hadoop fs -mkdir /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 the yarn-site.xml.

Verify the directory structure, ownership, and permissions:

```
$ 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  - yarn supergroup          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
```

### To start YARN, start the ResourceManager and NodeManager services:



**Note:** Make sure you always start ResourceManager before starting NodeManager services.

On the ResourceManager 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
```

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
```

### Verify basic cluster operation for YARN.

At this point your cluster is upgraded and ready to run jobs. Before running your production jobs, verify basic cluster operation by running an example from the Apache Hadoop web site.



**Note:**

For important configuration information, see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).

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

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

Do the following steps as the user joe.

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

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

3. Set `HADOOP_MAPRED_HOME` for user joe:

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

## Upgrade

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

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

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

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

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

- List the output files.

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

- Read the results in the output file.

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

You have now confirmed your cluster is successfully running CDH 5.



**Important:** If you have client hosts, make sure you also update them to CDH 5, and upgrade the [components](#) running on those clients as well.

### Start MapReduce (MRv1)



**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. Steps 9a and 9b are mutually exclusive.

After you have verified HDFS is operating correctly, you are ready to start MapReduce. On each TaskTracker system:

```
$ sudo service hadoop-0.20-mapreduce-tasktracker start
```

On the JobTracker system:

```
$ sudo service hadoop-0.20-mapreduce-jobtracker start
```

Verify that the JobTracker and TaskTracker started properly.

```
$ sudo jps | grep Tracker
```

If the permissions of directories are not configured correctly, the JobTracker and TaskTracker processes start and immediately fail. If this happens, check the JobTracker and TaskTracker logs and set the permissions correctly.



**Important:** 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
```

### Verify basic cluster operation for MRv1.

At this point your cluster is upgraded and ready to run jobs. Before running your production jobs, verify basic cluster operation by running an example from the Apache Hadoop web site.

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

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

Do the following steps as the user joe.

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

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

3. Set HADOOP\_MAPRED\_HOME for user joe:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce/
```

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

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

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

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

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

6. List the output files.

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

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

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

## Upgrade

```
1     dfs.replication
1     dfs.safemode.extension
1     dfs.safemode.min.datanodes
```

You have now confirmed your cluster is successfully running CDH 5.



### Important:

If you have client hosts, make sure you also update them to CDH 5, and upgrade the [components](#) running on those clients as well.

### Set the Sticky Bit

For security reasons Cloudera strongly recommends you set the sticky bit on directories if you have not already done so.

The sticky bit prevents anyone except the superuser, directory owner, or file owner from deleting or moving the files within a directory. (Setting the sticky bit for a file has no effect.) Do this for directories such as `/tmp`. (For instructions on creating `/tmp` and setting its permissions, see [these instructions](#)).

### Re-Install CDH 5 Components

When upgrading CDH, Cloudera strongly recommends that you upgrade your client jars to match. For help on finding matching artifacts, refer to [Using the CDH 5 Maven Repository](#).

#### *CDH 5 Components*

Use the following sections to install or upgrade CDH 5 components:

- [Crunch Installation](#) on page 270
- [Flume Installation](#) on page 272
- [HBase Installation](#) on page 281
- [HCatalog Installation](#) on page 310
- [Hive Installation](#) on page 329
- [HttpFS Installation](#) on page 358
- [Hue Installation](#) on page 362
- [Impala Installation](#) on page 316
- [KMS Installation and Upgrade](#) on page 394
- [Mahout Installation](#) on page 396
- [Oozie Installation](#) on page 398
- [Pig Installation](#) on page 418
- [Search Installation](#) on page 422
- [Sentry Installation](#) on page 434
- [Snappy Installation](#) on page 436
- [Spark Installation](#) on page 436
- [Sqoop 1 Installation](#) on page 438
- [Sqoop 2 Installation](#) on page 442
- [Whirr Installation](#) on page 450
- [ZooKeeper Installation](#)

See also the instructions for [installing or updating LZO](#).

## Apply Configuration File Changes



### Important:

During uninstall, the package manager renames any configuration files you have modified from <file> to <file>.rpmsave. During re-install, the package manager creates a new <file> with applicable defaults. You are responsible for applying any changes captured in the original CDH 4 configuration file to the new CDH 5 configuration file. In the case of Ubuntu and Debian upgrades, a file will not be installed if there is already a version of that file on the system, and you will be prompted to resolve conflicts; for details, see [Automatic handling of configuration files by dpkg](#).

For example, if you have modified your CDH 4 `zoo.cfg` configuration file (`/etc/zookeeper.dist/zoo.cfg`), RPM uninstall and re-install (using `yum remove`) renames and preserves a copy of your modified `zoo.cfg` as `/etc/zookeeper.dist/zoo.cfg.rpmsave`. You should compare this to the new `/etc/zookeeper/conf/zoo.cfg` and resolve any differences that should be carried forward (typically where you have changed property value defaults). Do this for each component you upgrade to CDH 5.

## Finalize the HDFS Metadata Upgrade

To finalize the HDFS metadata upgrade you began earlier in this procedure, proceed as follows:

1. Make sure you are satisfied that the CDH 5 upgrade has succeeded and everything is running smoothly. To determine when finalization is warranted, run important workloads and ensure they are successful.



**Warning:** Do not proceed until you are sure you are satisfied with the new deployment. Once you have finalized the HDFS metadata, you cannot revert to an earlier version of HDFS.



### Note:

- If you need to restart the NameNode during this period (after having begun the upgrade process, but before you've run `finalizeUpgrade`), restart your NameNode without the `-upgrade` option.
- Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:
  - Deleting files does not free up disk space.
  - Using the balancer causes all moved replicas to be duplicated.
  - All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

2. Finalize the HDFS metadata upgrade: use one of the following commands, depending on whether Kerberos is enabled (see [Configuring Hadoop Security in CDH 5](#)).



**Important:** In an HDFS HA deployment, make sure that both the NameNodes and all of the JournalNodes are up and functioning normally before you proceed.

- If Kerberos is enabled:

```
$ kinit -kt /path/to/hdfs.keytab hdfs/<fully.qualified.domain.name@YOUR-REALM.COM> &&
hdfs dfsadmin -finalizeUpgrade
```

- If Kerberos is not enabled:

```
$ sudo -u hdfs hdfs dfsadmin -finalizeUpgrade
```



**Note:** After the metadata upgrade completes, the `previous/` and `blocksBeingWritten/` directories in the DataNodes' data directories aren't cleared until the DataNodes are restarted.

### Upgrading from an Earlier CDH 5 Release to the Latest Release



#### Important:

- If you are using Cloudera Manager to manage CDH, *do not* use the instructions in this section. Follow the directions under [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524 to upgrade to the latest version of CDH 5 in a Cloudera Manager deployment.
- The instructions in this section describe how to upgrade to the latest CDH 5 release from an earlier CDH 5 release. If you are upgrading from a CDH 4 release, use the instructions under [Upgrading from CDH 4 to CDH 5](#) on page 691 instead.
- **MapReduce v1 (MRv1) and MapReduce v2 (YARN):** the sections that follow cover upgrade for MapReduce v1 (MRv1) and MapReduce v2 (YARN). 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.



#### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, rather than running scripts in `/etc/init.d` directly. The `service` command creates a predictable environment by setting the current working directory to `/` and removing most environment variables (passing only `LANG` and `TERM`). With `/etc/init.d`, existing environment variables remain in force and can produce unpredictable results. When you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).

### Important Tasks

- **Upgrading from any release earlier than CDH 5.4.0 to CDH 5.4.0 or later requires an HDFS metadata upgrade.**
- **Upgrading from a release earlier than 5.2.0 requires all of the following:**
  - Upgrade HDFS metadata
  - Upgrade the Sentry database
  - Upgrade the Hive database
  - Upgrade the Sqoop 2 database

Make sure you also do the following tasks that are required for every upgrade:

- Upgrade the Oozie database and shared library
- If you have uploaded the Spark assembly JAR file to HDFS, upload the new version of the file

Each of these tasks is described in context as you proceed through the upgrade. The following sections provide the information and instructions you need:

- [Before Upgrading to the Latest Release of CDH](#) on page 709
- [Upgrading from CDH 5.4.0 or Higher to the Latest Release](#) on page 710
- [Upgrading from a Release Earlier than CDH 5.4.0 to the Latest Release](#) on page 723

## Before Upgrading to the Latest Release of CDH



### Note:

- Before upgrading, read about the latest [Incompatible Changes](#) and [Known Issues and Workarounds in CDH 5](#) in the [CDH 5 Release Notes](#).



**Warning:** It's particularly important that you read the [Install and Upgrade Known Issues](#).

- If you are upgrading a cluster that is part of a production system, plan ahead. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.

- The instructions in this section assume you are upgrading a multi-node cluster. If you are running a pseudo-distributed (single-machine) cluster, Cloudera recommends that you copy your data off the cluster, remove the old CDH release, install Hadoop from CDH 5, and then restore your data.
- If you have a multi-node cluster running an earlier version of CDH 5, use the appropriate instructions to upgrade your cluster to the latest version:
  - [Upgrading from CDH 5.4.0 or Higher to the Latest Release](#) on page 710
  - [Upgrading from a Release Earlier than CDH 5.4.0 to the Latest Release](#) on page 723

## 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
```

## Upgrade

### 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
```

## Upgrading from CDH 5.4.0 or Higher to the Latest Release

Use the instructions that follow to upgrade from CDH 5.4.0 or higher to the latest version of CDH 5.



#### Important: Are you on the right page?

Use the instructions on this page *only* to upgrade from CDH 5.4.0 or higher. Upgrades from a release earlier than CDH 5.4.0 require an HDFS metadata upgrade.

If you are *not* currently running CDH 5.4.0 or higher:

- Use [these instructions](#) to upgrade from a CDH 5 release earlier than CDH 5.4.0;
- Use [these instructions](#) to upgrade from a CDH 4 release.

### Step 1: Prepare the cluster for the upgrade

#### 1. Put the NameNode into safe mode and save thefsimage

a. Put the NameNode (or active NameNode in an HA configuration) into safe mode:

```
$ sudo -u hdfs hdfs dfsadmin -safemode enter
```

b. Perform a saveNamespace operation:

```
$ sudo -u hdfs hdfs dfsadmin -saveNamespace
```

This will result in a new fsimage being written out with no edit log entries.

c. With the NameNode still in safe mode, shut down all services as instructed below.

#### 2. Shut down Hadoop services across your entire cluster by running the following command on every host in your cluster:

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

3. Check each host to make sure that there are no processes running as the `hdfs`, `yarn`, `mapred` or `httpfs` users from root:

```
# ps -afe | grep java
```



**Important:**

When you are sure that all Hadoop services have been shut down, do the following step. **It is particularly important that the NameNode service is not running so that you can make a consistent backup.**

4. Back up the HDFS metadata on the NameNode machine, as follows.



**Note:**

- Cloudera recommends backing up HDFS metadata on a regular basis, as well as before a major upgrade.
- `dfs.name.dir` is deprecated but still works; `dfs.namenode.name.dir` is preferred. This example uses `dfs.name.dir`.

- a. Find the location of your `dfs.name.dir` (or `dfs.namenode.name.dir`); for example:

```
$ grep -C1 dfs.name.dir /etc/hadoop/conf/hdfs-site.xml
<property> <name>dfs.name.dir</name> <value>/mnt/hadoop/hdfs/name</value>
</property>
```

- b. Back up the directory. The path inside the `<value>` XML element is the path to your HDFS metadata. If you see a comma-separated list of paths, there is no need to back up all of them; they store the same data. Back up the first directory, for example, by using the following commands:

```
$ cd /mnt/hadoop/hdfs/name
# tar -cvf /root/nn_backup_data.tar .
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```



**Important:**

If you see a file containing the word `lock`, the NameNode is probably still running. Repeat the preceding steps from the beginning; start at Step 1 and shut down the Hadoop services.

## Step 2: If necessary, download the CDH 5 "1-click" package on each host in your cluster

**Before you begin:** Check whether you have the CDH 5 "1-click" repository installed.

- On Red Hat/CentOS-compatible and SLES systems:

```
rpm -q cdh5-repository
```

If you are upgrading from CDH 5 Beta 1 or higher, you should see:

```
cdh5-repository-1-0
```

## Upgrade

In this case, skip to [Step 3](#). If instead you see:

```
package cdh5-repository is not installed
```

proceed with [this step](#).

- On Ubuntu and Debian systems:

```
dpkg -l | grep cdh5-repository
```

If the repository is installed, skip to [Step 3](#); otherwise proceed with [this step](#).

If the CDH 5 "1-click" repository is not already installed on each host in the cluster, follow the instructions below for that host's operating system:

[Instructions for Red Hat-compatible systems](#)

[Instructions for SLES systems](#)

[Instructions for Ubuntu and Debian systems](#)

On Red Hat-compatible systems:

1. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

2. Install the RPM for all RHEL versions:

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



**Note:**

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



**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

On SLES systems:

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

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

**2. Install the RPM:**

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

**3. Update your system package index by running:**

```
$ sudo zypper refresh
```

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



**Note:**

For instructions on how to add a repository or build your own repository, see [Installing CDH 5 on SLES Systems](#).



**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

**On Ubuntu and Debian systems:**

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

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

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

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



**Note:**

For instructions on how to add a repository or build your own repository, see [Installing CDH 5 on Ubuntu Systems](#).



**Note: Clean repository cache.**

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo apt-get update
```

## Upgrade

### Step 3: Upgrade the Packages on the Appropriate Hosts

Upgrade [MRv1](#), [YARN](#), or both, depending on what you intend to use.



#### Note:

- Remember that you can install and configure both MRv1 and YARN, but you should not run them both on the same set of nodes at the same time.
- If you are using [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`.

**Before installing MRv1 or YARN:** (Optional) add a repository key on each system in the cluster, if you have not already done so. 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 Precise systems:**

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

- **For Debian Wheezy systems:**

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

### Step 3a: If you are using MRv1, upgrade the MRv1 packages on the appropriate hosts.

Skip this step if you are using [YARN](#) exclusively. Otherwise upgrade each type of daemon package on the appropriate hosts as follows:

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 or JobTracker.

Follow instructions under [ZooKeeper Installation](#).

2. Install each type of daemon package on the appropriate systems(s), as follows.

Where to install	Install commands
JobTracker host running:	

Where to install	Install commands
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-jobtracker
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-jobtracker
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-jobtracker
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-namenode
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode
All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts, running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-client
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-client
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-client

## Upgrade

Step 3b: If you are using YARN, upgrade the YARN packages on the appropriate hosts.

Skip this step if you are using [MRv1](#) exclusively. Otherwise upgrade each type of daemon package on the appropriate hosts as follows:

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 or JobTracker.

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>	\$ sudo yum clean all; sudo yum install hadoop-yarn-resourcemanager
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-resourcemanager
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-yarn-resourcemanager
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-namenode
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode
All cluster hosts except the Resource Manager (analogous to MRv1 TaskTrackers) running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce

Where to install	Install commands
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce
One host in the cluster running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-client
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-client
<i>Ubuntu or Debian</i>	sudo apt-get update; sudo apt-get install hadoop-client

**Note:**

The hadoop-yarn and hadoop-hdfs packages are installed on each system automatically as dependencies of the other packages.

#### Step 4: In an HA Deployment, Upgrade and Start the JournalNodes

1. Install the JournalNode daemons on each of the machines where they will run.

**To install JournalNode on Red Hat-compatible systems:**

```
$ sudo yum install hadoop-hdfs-journalnode
```

**To install JournalNode on Ubuntu and Debian systems:**

```
$ sudo apt-get install hadoop-hdfs-journalnode
```

**To install JournalNode on SLES systems:**

```
$ sudo zypper install hadoop-hdfs-journalnode
```

## Upgrade

2. Start the JournalNode daemons on each of the machines where they will run:

```
sudo service hadoop-hdfs-journalnode start
```

Wait for the daemons to start before proceeding to the next step.



**Important:**

The JournalNodes must be up and running CDH 5 before you proceed.

### Step 5: Start HDFS

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

### Step 6: Start MapReduce (MRv1) or YARN

You are now ready to start and test MRv1 or YARN.

For MRv1	For YARN
<a href="#">Start MRv1</a>	<a href="#">Start YARN and the MapReduce JobHistory Server</a>
<a href="#">Verify basic cluster operation</a>	<a href="#">Verify basic cluster operation</a>

### Step 6a: Start MapReduce (MRv1)



**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 MapReduce cluster deployment. Steps 6a and 6b are mutually exclusive.

After you have verified HDFS is operating correctly, you are ready to start MapReduce. On each TaskTracker system:

```
$ sudo service hadoop-0.20-mapreduce-tasktracker start
```

On the JobTracker system:

```
$ sudo service hadoop-0.20-mapreduce-jobtracker start
```

Verify that the JobTracker and TaskTracker started properly.

```
$ sudo jps | grep Tracker
```

If the permissions of directories are not configured correctly, the JobTracker and TaskTracker processes start and immediately fail. If this happens, check the JobTracker and TaskTracker logs and set the permissions correctly.

#### Verify basic cluster operation for MRv1

At this point your cluster is upgraded and ready to run jobs. Before running your production jobs, verify basic cluster operation by running an example from the Apache Hadoop web site.



**Note:**

For important configuration information, see [Deploying MapReduce v1 \(MRv1\) on a Cluster](#).

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

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

Do the following steps as the user joe.

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

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

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

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

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

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

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

**5.** List the output files.

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

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

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

You have now confirmed your cluster is successfully running CDH 5.



**Important:**

If you have client hosts, make sure you also update them to CDH 5, and upgrade the [components](#) running on those clients as well.

## Upgrade

### Step 6b: Start MapReduce with YARN



#### 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 your performance and may result in an unstable MapReduce cluster deployment. Steps 6a and 6b are mutually exclusive.

After you have verified HDFS is operating correctly, you are ready to start YARN. First, if you have not already done so, create directories and set the correct permissions.



**Note:** For more information see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).

Create a history directory and set permissions; for example:

```
$ 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 yarn /user/history
```

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 the `yarn-site.xml`.

Verify the directory structure, ownership, and permissions:

```
$ 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 - yarn supergroup 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
```

To start YARN, start the ResourceManager and NodeManager services:



#### Note:

Make sure you always start ResourceManager before starting NodeManager services.

On the ResourceManager 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
```

For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop 1 in a YARN installation, set the HADOOP\_MAPRED\_HOME environment variable as follows:

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

#### Verify basic cluster operation for YARN.

At this point your cluster is upgraded and ready to run jobs. Before running your production jobs, verify basic cluster operation by running an example from the Apache Hadoop web site.



##### Note:

For important configuration information, see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).

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

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

Do the following steps as the user joe.

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

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

3. Set HADOOP\_MAPRED\_HOME for user joe:

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

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

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

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

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

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

6. List the output files:

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

## Upgrade

### 7. Read the results in the output file:

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

You have now confirmed your cluster is successfully running CDH 5.



#### Important:

If you have client hosts, make sure you also update them to CDH 5, and upgrade the [components](#) running on those clients as well.

### Step 7: Set the Sticky Bit

For security reasons Cloudera strongly recommends you set the sticky bit on directories if you have not already done so.

The sticky bit prevents anyone except the superuser, directory owner, or file owner from deleting or moving the files within a directory. (Setting the sticky bit for a file has no effect.) Do this for directories such as `/tmp`. (For instructions on creating `/tmp` and setting its permissions, see [these instructions](#)).

### Step 8: Upgrade Components



#### Note:

- For important information on new and changed components, see the [CDH 5 Release Notes](#). To see whether there is a new version of a particular component in CDH 5, check the [CDH Version and Packaging Information](#).
- Cloudera recommends that you regularly update the software on each system in the cluster (for example, on a RHEL-compatible system, regularly run `yum update`) to ensure that all the dependencies for any given component are up to date. (If you have not been in the habit of doing this, be aware that the command may take a while to run the first time you use it.)

### CDH 5 Components

Use the following sections to install or upgrade CDH 5 components:

- [Crunch Installation](#) on page 270
- [Flume Installation](#) on page 272
- [HBase Installation](#) on page 281
- [HCatalog Installation](#) on page 310
- [Hive Installation](#) on page 329
- [HttpFS Installation](#) on page 358
- [Hue Installation](#) on page 362
- [Impala Installation](#) on page 316
- [KMS Installation and Upgrade](#) on page 394
- [Mahout Installation](#) on page 396
- [Oozie Installation](#) on page 398
- [Pig Installation](#) on page 418
- [Search Installation](#) on page 422
- [Sentry Installation](#) on page 434

- [Snappy Installation](#) on page 436
- [Spark Installation](#) on page 436
- [Sqoop 1 Installation](#) on page 438
- [Sqoop 2 Installation](#) on page 442
- [Whirr Installation](#) on page 450
- [ZooKeeper Installation](#)

See also the instructions for [installing or updating LZO](#).

#### Step 9: Apply Configuration File Changes if Necessary

**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](#).

For example, if you have modified your `zoo.cfg` configuration file (`/etc/zookeeper/zoo.cfg`), the upgrade renames and preserves a copy of your modified `zoo.cfg` as `/etc/zookeeper/zoo.cfg.rpmsave`. If you have not already done so, you should now compare this to the new `/etc/zookeeper/conf/zoo.cfg`, resolve differences, and make any changes that should be carried forward (typically where you have changed property value defaults). Do this for each component you upgrade.

#### Upgrading from a Release Earlier than CDH 5.4.0 to the Latest Release

**Important:**

Use the instructions on this page to upgrade only from **a CDH 5 release earlier than CDH 5.4.0**.

To upgrade from other releases, see the following topics:

- [Upgrading from CDH 5.4.0 or Higher to the Latest Release](#) on page 710
- [Upgrading from CDH 4 to CDH 5](#) on page 691

#### Step 1: Prepare the Cluster for the Upgrade

**Important:** Before you begin, read the following topics, which contain important upgrade information:

- [Upgrading from an Earlier CDH 5 Release to the Latest Release](#) on page 708
- [Before Upgrading to the Latest Release of CDH](#) on page 709

##### 1. Put the NameNode into safe mode and save thefsimage:

###### a. Put the NameNode (or active NameNode in an HA configuration) into safe mode:

```
$ sudo -u hdfs hdfs dfsadmin -safemode enter
```

###### b. Run a saveNamespace operation:

```
$ sudo -u hdfs hdfs dfsadmin -saveNamespace
```

## Upgrade

This results in a new `fsimage` written with no edit log entries.

c.

2. Shut down Hadoop services across your entire cluster by running the following command on every host in your cluster:

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

3. Check each host to make sure that there are no processes running as the `hdfs`, `yarn`, `mapred` or `httpfs` users from root:

```
# ps -afe | grep java
```

4. Ensure that the NameNode service is not running, and then back up the HDFS metadata on the NameNode machine, as follows.



**Note:** Cloudera recommends backing up HDFS metadata on a regular basis, as well as before a major upgrade.

- a. Find the location of your `dfs.namenode.name.dir`. For example:

```
$ grep -C1 dfs.namenode.name.dir /etc/hadoop/conf/hdfs-site.xml
<property> <name>dfs.namenode.name.dir</name> <value>/mnt/hadoop/hdfs/name</value>
</property>
```

- b. Back up the directory. The path inside the `<value>` XML element is the path to your HDFS metadata. If you see a comma-separated list of paths, you do not need to back up all of them; they store the same data. Back up the first directory by using the following commands:

```
$ cd /mnt/hadoop/hdfs/name
# tar -cvf /root/nn_backup_data.tar .
./
./current/
./current/fsimage
./current/fstime
./current/VERSION
./current/edits
./image/
./image/fsimage
```



**Important:** If you see a file containing the word `lock`, the NameNode is probably still running. Re-run the procedure, beginning at step 1, and make sure that the NameNode service is not running before backing up the HDFS metadata.

### Step 2: If Necessary, Download the CDH 5 "1-click" Package on Each of the Hosts in the Cluster

**Before you begin:** Check whether you have the CDH 5 "1-click" repository installed, and proceed as indicated.

**Table 27: Checking for the 1-click Repository**

Operating System	Command to Run	Results and Actions
RHEL-compatible	<code>rpm -q cdh5-repository</code>	If the command returns, <code>cdh5-repository-1-0</code> , the 1-click repository is installed. Skip to <a href="#">Step 3</a> .
Ubuntu and Debian	<code>dpkg -l   grep cdh5-repository</code>	If the command returns package <code>cdh5-repository</code> is not

Operating System	Command to Run	Results and Actions
		<p>installed, go to the 1-click instructions for your OS:</p> <ul style="list-style-type: none"> <li>• <a href="#">RHEL</a></li> <li>• <a href="#">SLES</a></li> <li>• <a href="#">Ubuntu and Debian</a></li> </ul>

#### On RHEL-compatible systems:

1. Download the CDH 5 "1-click Install" package (or RPM).

Click the appropriate RPM and **Save File** to a directory with write access (for example, your home directory).

OS Version	Link to CDH 5 RPM
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS/Oracle 6	<a href="#">RHEL/CentOS/Oracle 6 link</a>
RHEL/CentOS/Oracle 7	<a href="#">RHEL/CentOS/Oracle 7 link</a>

2. Install the RPM for all RHEL versions:

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

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



#### Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

#### On SLES systems:

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

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

2. Install the RPM:

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

3. Update your system package index by running:

```
$ sudo zypper refresh
```

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

For instructions on how to add a repository or build your own repository, see [Installing CDH 5 on SLES Systems](#).

## Upgrade



### Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

On Ubuntu and Debian systems:

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

OS Version	Package Link
Wheezy	<a href="#">Wheezy package</a>
Precise	<a href="#">Precise package</a>
Trusty	<a href="#">Trusty package</a>

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

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

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

For instructions on how to add a repository or build your own repository, see the instructions on [installing CDH 5 on Ubuntu and Debian systems](#).



### Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo apt-get update
```

## Step 3: Upgrade the Packages on the Appropriate Hosts

Upgrade [MRv1](#), [YARN](#), or both. Although you can install and configure both MRv1 and YARN, you should not run them both on the same set of hosts at the same time.

If you are using [HA for the NameNode](#), do not install hadoop-hdfs-secondarynamenode

**Before upgrading MRv1 or YARN:** (Optionally) add a repository key on each system in the cluster, if you have not already done so. 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 Precise systems:

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

- For Debian Wheezy systems:

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

**Step 3a: If you are using MRv1, upgrade the MRv1 packages on the appropriate hosts.**

Skip this step if you are using [YARN](#) exclusively.

1. Install and deploy ZooKeeper as described in [ZooKeeper Installation](#). Cloudera recommends that you install (or update) and start a ZooKeeper cluster, and ZooKeeper is required if you are deploying high availability (HA) for the NameNode or JobTracker.
2. Install each daemon package on the appropriate systems, as follows.

Where to install	Install commands
JobTracker host running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-jobtracker
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-jobtracker
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-jobtracker
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-namenode
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode

## Upgrade

Where to install	Install commands
All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts, running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-client
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-client
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-client

Step 3b: If you are using YARN, upgrade the YARN packages on the appropriate hosts.

Skip this step if you are using [MRv1](#) exclusively.

1. Install and deploy ZooKeeper as described in [ZooKeeper Installation](#). Cloudera recommends that you install (or update) and start a ZooKeeper cluster, and ZooKeeper is required if you are deploying high availability (HA) for the NameNode or JobTracker.
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>	\$ sudo yum clean all; sudo yum install hadoop-yarn-resourcemanager
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-resourcemanager
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-yarn-resourcemanager
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-namenode

Where to install	Install commands
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode
All cluster hosts except the Resource Manager (analogous to MRv1 TaskTrackers) running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce
One host in the cluster running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver
<i>Ubuntu or Debian</i>	\$ sudo apt-get update; sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	\$ sudo yum clean all; sudo yum install hadoop-client
<i>SLES</i>	\$ sudo zypper clean --all; sudo zypper install hadoop-client
<i>Ubuntu or Debian</i>	sudo apt-get update; sudo apt-get install hadoop-client

**Note:**

The hadoop-yarn and hadoop-hdfs packages are installed on each system automatically as dependencies of the other packages.

## Upgrade

### Step 4: In an HA Deployment, Upgrade and Start the Journal Nodes

1. Install the JournalNode daemons on each of the machines where they will run.

To install JournalNode on Red Hat-compatible systems:

```
$ sudo yum install hadoop-hdfs-journalnode
```

To install JournalNode on Ubuntu and Debian systems:

```
$ sudo apt-get install hadoop-hdfs-journalnode
```

To install JournalNode on SLES systems:

```
$ sudo zypper install hadoop-hdfs-journalnode
```

2. Start the JournalNode daemons on each of the machines where they will run:

```
sudo service hadoop-hdfs-journalnode start
```

Wait for the daemons to start before proceeding to the next step.



#### Important:

In an HA deployment, the JournalNodes must be up and running CDH 5 before you proceed.

### Step 5: Upgrade the HDFS Metadata

The steps for upgrading HDFS metadata differ for HA and non-HA deployments.

#### Section 5a: Upgrade the HDFS Metadata for HA Deployments

1. Make sure that the JournalNodes have been upgraded to CDH 5 and are up and running.
2. Run the following command on the *active NameNode only*:

```
$ sudo service hadoop-hdfs-namenode upgrade
```



#### Warning:

In an HDFS HA deployment, it is critically important that you do this on only one NameNode.

3. Monitor the progress of the metadata upgrade by running the following:

```
$ sudo tail -f /var/log/hadoop-hdfs/hadoop-hdfs-namenode-<hostname>.log
```

Look for a line that confirms the upgrade is complete, such as:

/var/lib/hadoop-hdfs/cache/hadoop/dfs/<name> is complete.

The NameNode upgrade process can take a while, depending on the number of files.

4. Wait for NameNode to exit safe mode, and then restart the standby NameNode.

- If Kerberos is enabled:

```
$ kinit -kt /path/to/hdfs.keytab hdfs/<fully.qualified.domain.name@YOUR-REALM.COM> &&  
hdfs namenode -bootstrapStandby
```

```
$ sudo service hadoop-hdfs-namenode start
```

- If Kerberos is not enabled:

```
$ sudo -u hdfs hdfs namenode -bootstrapStandby
$ sudo service hadoop-hdfs-namenode start
```

### 5. Start the DataNodes by running the following command on each DataNode:

```
$ sudo service hadoop-hdfs-datanode start
```

## Section 5b: Upgrade the HDFS Metadata for Non-HA Deployments

1. Run the following command on the NameNode:

```
$ sudo service hadoop-hdfs-namenode upgrade
```

2. Monitor the progress of the metadata upgrade by running the following:

```
$ sudo tail -f /var/log/hadoop-hdfs/hadoop-hdfs-namenode-<hostname>.log
```

Look for a line that confirms the upgrade is complete, such as:

/var/lib/hadoop-hdfs/cache/hadoop/dfs/<name> is complete.

The NameNode upgrade process can take a while, depending on the number of files.

3. Start the DataNodes by running the following command on each DataNode:

```
$ sudo service hadoop-hdfs-datanode start
```

4. Wait for NameNode to exit safe mode, and then start the secondary NameNode:

1. To check that the NameNode has exited safe mode, look for messages in the log file, or the NameNode's web interface, that say "...no longer in safe mode".
2. To start the secondary NameNode, enter the following command on the secondary NameNode host:

```
$ sudo service hadoop-hdfs-secondarynamenode start
```

## Step 6: Start MapReduce (MRv1) or YARN

You are now ready to start and test [MRv1](#) or [YARN and the MapReduce JobHistory Server](#).



### Important:

Do not run MRv1 and YARN on the same set of nodes at the same time. This degrades performance and can result in an unstable cluster deployment. Steps 6a and 6b are mutually exclusive.

## Step 6a: Start MRv1

1. Start each TaskTracker:

```
$ sudo service hadoop-0.20-mapreduce-tasktracker start
```

2. Start each JobTracker:

```
$ sudo service hadoop-0.20-mapreduce-jobtracker start
```

## Upgrade

### 3. Verify that the JobTracker and TaskTracker started properly:

```
$ sudo jps | grep Tracker
```

If the permissions of directories are not configured correctly, the JobTracker and TaskTracker processes start and immediately fail. If this happens, check the JobTracker and TaskTracker logs and set the permissions correctly.

### 4. Verify basic cluster operation for MRv1.

Before running production jobs, verify basic cluster operation by running an example from the Apache Hadoop web site.



#### Important:

For important cluster configuration information, see [Deploying MapReduce v1 \(MRv1\) on a Cluster](#).

#### a. Create a home directory on HDFS for user joe:

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

Perform steps a through f as user joe.

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

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

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

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

#### d. After the job completes, find the output in the HDFS directory named `output` which you specified to Hadoop:

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

#### e. List the output files:

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

#### f. Read the results in the output file; for example:

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

This confirms your cluster is successfully running CDH 5.



**Important:**

If you have client hosts, make sure you also update them to CDH 5, and upgrade the [components](#) running on those clients.

### Step 6b: Start MapReduce with YARN

1. If you have not already done so, create directories and set the correct permissions.



**Note:** For more information about YARN configuration and permissions, see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).

- a. Create a history directory and set permissions; for example:

```
$ 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 yarn /user/history
```

- b. 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
```

You create this directory because it is the parent of /var/log/hadoop-yarn/apps, which is explicitly configured in the `yarn-site.xml`.

- c. Verify the directory structure, ownership, and permissions:

```
$ 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 - yarn supergroup 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
```

2. Start YARN, and start the ResourceManager and NodeManager services:



**Important:**

Always start ResourceManager before starting NodeManager services.

- a. On the ResourceManager system, run the following command:

```
$ sudo service hadoop-yarn-resourcemanager start
```

- b. On each NodeManager system (typically the same ones where DataNode service runs):

```
$ sudo service hadoop-yarn-nodemanager start
```

3. Start the MapReduce JobHistory Server:

## Upgrade

- a. On the MapReduce JobHistory Server system, run the following command:

```
$ sudo service hadoop-mapreduce-historyserver start
```

- b. For each user who will be submitting MapReduce jobs using YARN, or running Pig, Hive, or Sqoop 1 in a YARN installation, set the HADOOP\_MAPRED\_HOME environment variable as follows:

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

4. Before running production jobs, verify basic cluster operation by running an example from the Apache Hadoop web site.



### Note:

For important configuration information, see [Deploying MapReduce v2 \(YARN\) on a Cluster](#).

- a. Create a home directory for user joe:

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

Perform the remaining steps as the user joe.

- b. Make a directory in HDFS called input and copy XML files to it by running the following commands in pseudo-distributed mode:

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

- c. Set HADOOP\_MAPRED\_HOME for user joe:

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

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

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

- e. After the job completes, find the output in the HDFS directory named output23, which you specified to Hadoop:

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

- f. List the output files:

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

**g.** Read the results in the output file:

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

This confirms that your cluster is successfully running CDH 5.



**Important:**

If you have client hosts, make sure you also update them to CDH 5, and upgrade the [components](#) running on those clients as well.

### Step 7: Set the Sticky Bit

For security reasons, Cloudera strongly recommends you set the sticky bit on directories if you have not already done so.

The sticky bit prevents anyone except the superuser, directory owner, or file owner from deleting or moving the files within a directory. (Setting the sticky bit for a file has no effect.) Do this for directories such as `/tmp`. (For instructions on creating `/tmp` and setting its permissions, see [Create the /tmp directory](#) on page 260).

### Step 8: Upgrade Components

Cloudera recommends that you regularly update the software on each system in the cluster (for example, on a RHEL-compatible system, regularly run `yum update`) to ensure that all the dependencies for any given component are up to date. If you have not been doing this, the command may take a while to run the first time you use it.



**Note:**

- For important information on new and changed components, see the [CDH 5 Release Notes](#). To see whether there is a new version of a particular component in CDH 5, check the [CDH Version and Packaging Information](#).

### CDH 5 Components

Use the following sections to install or upgrade CDH 5 components:

- [Crunch Installation](#) on page 270
- [Flume Installation](#) on page 272
- [HBase Installation](#) on page 281
- [HCatalog Installation](#) on page 310
- [Hive Installation](#) on page 329
- [HttpFS Installation](#) on page 358
- [Hue Installation](#) on page 362
- [Impala Installation](#) on page 316
- [KMS Installation and Upgrade](#) on page 394
- [Mahout Installation](#) on page 396
- [Oozie Installation](#) on page 398
- [Pig Installation](#) on page 418
- [Search Installation](#) on page 422
- [Sentry Installation](#) on page 434
- [Snappy Installation](#) on page 436

## Upgrade

- [Spark Installation](#) on page 436
- [Scoop 1 Installation](#) on page 438
- [Scoop 2 Installation](#) on page 442
- [Whirr Installation](#) on page 450
- [ZooKeeper Installation](#)

See also the instructions for [installing or updating LZO](#).

### Step 9: Apply Configuration File Changes if Required



#### 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](#).

For example, if you have modified your `zoo.cfg` configuration file (`/etc/zookeeper/zoo.cfg`), the upgrade renames and preserves a copy of your modified `zoo.cfg` as `/etc/zookeeper/zoo.cfg.rpmsave`. If you have not already done so, you should now compare this to the new `/etc/zookeeper/conf/zoo.cfg`, resolve differences, and make any changes that should be carried forward (typically where you have changed property value defaults). Do this for each component you upgrade.

### Step 10: Finalize the HDFS Metadata Upgrade



**Important:** Once you have finalized the upgrade, you cannot roll back to a previous version of HDFS. See [Rolling Back a CDH 4-to-CDH 5 Upgrade](#) on page 750.

To finalize the HDFS metadata upgrade, do the following:

1. Make sure that the CDH 5 upgrade has succeeded and everything is running as expected. You can wait days or even weeks to verify a successful upgrade before finalizing it.

Before finalizing, run important workloads and ensure that they are successful. Once you have finalized the upgrade, you cannot roll back to a previous version of HDFS without using backups.



#### Note:

- If you need to restart the NameNode during this period (after having begun the upgrade process, but before you've run `finalizeUpgrade`), restart your NameNode without the `-upgrade` option.
- Verifying that you are ready to finalize the upgrade can take a long time. Make sure you have enough free disk space, keeping in mind the following:
  - Deleting files does not free up disk space.
  - Using the balancer causes all moved replicas to be duplicated.
  - All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

2. Finalize the HDFS metadata upgrade by using one of the following commands, depending on whether Kerberos is enabled (see [Enabling Kerberos Authentication for Hadoop Using the Command Line](#)).



**Important:** In an HDFS HA deployment, make sure that both the NameNodes and all of the JournalNodes are up and functioning normally before you proceed.

- If Kerberos is enabled:

```
$ kinit -kt /path/to/dfs.keytab hdfs/<fully.qualified.domain.name@YOUR-REALM.COM> &&
hdfs dfsadmin -finalizeUpgrade
```

- If Kerberos is not enabled:

```
$ sudo -u hdfs hdfs dfsadmin -finalizeUpgrade
```

After the metadata upgrade completes, the `previous/` and `blocksBeingWritten/` directories in the DataNode data directories are not cleared until the DataNodes are restarted.

#### Troubleshooting: If You Missed the HDFS Metadata Upgrade Steps

If you skipped [Step 5: Upgrade the HDFS Metadata](#) on page 730, HDFS will not start; the metadata upgrade is required for all upgrades to CDH 5.4.0 and higher from any earlier release. You will see errors such as the following:

```
2014-10-16 18:36:29,112 WARN org.apache.hadoop.hdfs.server.namenode.FSNamesystem:
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.FSImage.recoverTransitionRead(FSImage.java:231)
    at
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:994)
    at
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:726)
    at
org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:529)
    at
org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:585)
    at
org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:751)
    at
org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:735)
    at
org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1410)
    at
org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1476)
2014-10-16 18:36:29,126 INFO org.mortbay.log: Stopped
HttpServer2$SelectChannelConnectorWithSafeStartup@0.0.0.0:50070
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.MetricsSystemImpl:
Stopping NameNode metrics system...
2014-10-16 18:36:29,128 INFO org.apache.hadoop.metrics2.impl.MetricsSystemImpl:
NameNode metrics system stopped.
2014-10-16 18:36:29,128 INFO org.apache.hadoop.metrics2.impl.MetricsSystemImpl:
NameNode metrics system shutdown complete.
2014-10-16 18:36:29,128 FATAL org.apache.hadoop.server.namenode.NameNode:
Exception in namenode join
```

## Upgrade

```
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.FSImage.recoverTransitionRead(FSImage.java:231)
      at
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:994)
      at
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:726)
      at
org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:529)
      at
org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:585)
      at
org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:751)
      at
org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:735)
      at
org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1410)
      at
org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1476)
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.server.namenode.NameNode:
SHUTDOWN_MSG:
```

To recover, proceed as follows:

1. Make sure you have completed all the necessary preceding steps ([Step 1: Prepare the Cluster for the Upgrade](#) on page 723 through [Step 4: In an HA Deployment, Upgrade and Start the Journal Nodes](#) on page 730; or [Step 1: Prepare the Cluster for the Upgrade](#) on page 723 through [Step 3: Upgrade the Packages on the Appropriate Hosts](#) on page 726 if this is not an HA deployment).
2. Starting with [Step 5: Upgrade the HDFS Metadata](#) on page 730, complete all the remaining steps through [Step 10: Finalize the HDFS Metadata Upgrade](#) on page 736.

## Upgrading to Oracle JDK 1.7

Oracle JDK 1.7 is [required](#) for CDH 5. Follow the instructions on this page to upgrade to Java 1.7 *before* you upgrade to CDH 5.

The process for upgrading to Oracle JDK 1.7 varies depending on whether you have a [Cloudera Manager Deployment](#) on page 74 or an [Unmanaged Deployment](#) on page 77.



**Important:** If you are upgrading 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](#) on page 741.

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

## Upgrading to Oracle JDK 1.7 in a Cloudera Manager Deployment

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

**1.** Upgrade the JDK using one of the following processes:

- Cloudera Manager installation - allow Cloudera Manager to upgrade the JDK when you [upgrade Cloudera Manager Agent packages](#).
- Manual installation
  - 1.** On each cluster host:
    - a.** Install the same [supported version](#) of JDK 1.7. See [Java Development Kit Installation](#) on page 78 for instructions.
    - b.** Verify that you have set `JAVA_HOME` on each host to the directory where you installed JDK 1.7, as instructed.

**2.** In the Cloudera Manager Admin Console, restart all services:

- a.** On the **Home > Status** tab, click



next to the cluster name and select **Restart**.

- b.** In the confirmation dialog box that displays, click **Restart**.

**3.** If the cluster host is also running Cloudera Management Service roles, restart the Cloudera Management Service.

- a.** Do one of the following:

- **1.** Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
- **2.** Select **Actions > Restart**.

- On the **Home > Status** tab, click



to the right of **Cloudera Management Service** and select **Restart**.

- b.** Click **Restart** to confirm. The **Command Details** window shows the progress of stopping and then starting the roles.

- c.** When **Command completed with n/n successful subcommands** appears, the task is complete. Click **Close**.

## Upgrading to Oracle JDK 1.7 in an Unmanaged Deployment



**Important:**

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

**1.** Shut down the cluster, following directions in the documentation for the CDH 4 release you are currently running.

**2.** Clean up existing JDK versions.

**3.** On each cluster host:

- a.** Install the same [supported version](#) of JDK 1.7. See [Java Development Kit Installation](#) on page 78 for instructions.
- b.** Verify that you have set `JAVA_HOME` on each host to the directory where you installed JDK 1.7, as instructed.

**4.** Start the [CDH upgrade](#).

### Upgrading to Oracle JDK 1.8

Cloudera Manager 5.3 and higher and CDH 5.3 and higher support Oracle JDK 1.8. For other supported versions, see [Cloudera Manager Supported JDK Versions](#) and [CDH 5 Supported JDK Versions](#). In some cases, using JDK 1.8 can cause minor performance degradation compared to JDK 1.7.



#### Warning:

- Cloudera does not support upgrading to JDK 1.8 while upgrading to Cloudera Manager 5.3 or higher. The Cloudera Manager Server must be upgraded to 5.3 or higher before you start.
- Cloudera does not support upgrading to JDK 1.8 while upgrading a cluster to CDH 5.3 or higher. The cluster must be running CDH 5.3 or higher before you start.
- Cloudera does not support a rolling upgrade to JDK 1.8. You must shut down the entire cluster.
- 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](#) on page 741.

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

The process for upgrading to Oracle JDK 1.8 varies depending on whether you have a [Cloudera Manager Deployment](#) on page 74 or an [Unmanaged Deployment](#) on page 77.

### Upgrading to Oracle JDK 1.8 in a Cloudera Manager Deployment

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

1. [Upgrade to Cloudera Manager 5.3 or higher](#) if you have not done so.
2. [Upgrade to CDH 5.3 or higher](#) if you have not done so.
3. Stop the Cloudera Management Service.
4. Stop all clusters.
5. Stop all Cloudera Manager Agents.
6. Stop the Cloudera Manager Server.
7. Clean up existing Java versions.
8. On the Cloudera Manager Server host and each cluster host:
  - a. Install the same [supported version](#) of JDK 1.8. See [Java Development Kit Installation](#) on page 78 for instructions.
9. On the Cloudera Manager Server host, configure the location of the JDK in `/etc/default/cloudera-scm-server`.
10. Start the Cloudera Manager Server.
11. Start all Cloudera Manager Agents.
12. Configure the location of the JDK on cluster hosts as described in [Configuring a Custom Java Home Location](#) on page 176.
13. Start all clusters.
14. Start the Cloudera Management Service.

## Upgrading to Oracle JDK 1.8 in an Unmanaged Deployment



### Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- This information applies specifically to CDH 5.7.x. If you use a lower version of CDH, see the documentation for that version located at [Cloudera Documentation](#).

1. [Upgrade to CDH 5.3 or higher](#) if you have not already done so.
2. [Shut down the cluster](#).
3. Clean up existing Java versions.
4. On each cluster host:
  - a. Install the same [supported version](#) of JDK 1.8. See [Java Development Kit Installation](#) on page 78 for instructions.
  - b. Verify that you have set *JAVA\_HOME* on each host to the directory where you installed JDK 1.8 and created a symbolic link to it, as instructed.
5. [Start the cluster](#).

### If you are Using AES-256 Encryption, install the JCE Policy File

If you are using CentOS/Red Hat Enterprise Linux 5.6 or higher, or Ubuntu, which use AES-256 encryption by default for tickets, you must install the [Java Cryptography Extension \(JCE\) Unlimited Strength Jurisdiction Policy File](#) on all cluster and Hadoop user machines. For JCE Policy File installation instructions, see the `README.txt` file included in the `jce_policy-x.zip` file.

Alternatively, you can configure Kerberos to not use AES-256 by removing `aes256-cts:normal` from the `supported_enctypes` field of the `kdc.conf` or `krb5.conf` file. After changing the `kdc.conf` file, you must restart both the KDC and the kadmin server for those changes to take affect. You may also need to re-create or change the password of the relevant principals, including potentially the Ticket Granting Ticket principal (`krbtgt/REALM@REALM`). If AES-256 is still used after completing steps, the `aes256-cts:normal` setting existed when the Kerberos database was created. To fix this, create a new Kerberos database and then restart both the KDC and the kadmin server.

#### To verify the type of encryption used in your cluster:

1. On the local KDC host, type this command to create a test principal:

```
$ kadmin -q "addprinc test"
```

2. On a cluster host, type this command to start a Kerberos session as test:

```
$ kinit test
```

3. On a cluster host, type this command to view the encryption type in use:

```
$ klist -e
```

If AES is being used, output like the following is displayed after you type the `klist` command; note that AES-256 is included in the output:

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@SCM
Valid starting     Expires            Service principal
05/19/11 13:25:04  05/20/11 13:25:04  krbtgt/SCM@SCM
      Etype (skey, tkt): AES-256 CTS mode with 96-bit SHA-1 HMAC, AES-256 CTS mode with
      96-bit SHA-1 HMAC
```

# 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><a href="#">Downgrade the Cloudera Manager Server</a>, 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> <ul style="list-style-type: none"> <li>• RHEL-compatible 7 and higher:</li> </ul> <pre>service cloudera-scm-server force_next_start service cloudera-scm-server start</pre> <ul style="list-style-type: none"> <li>• All other Linux distributions:</li> </ul> <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 Caused by: <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 <a href="#">Uninstalling Cloudera Manager and Managed Software</a> on page 199.
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&gt; show table status;</code>  For more information, see <a href="#">MySQL Database</a> on page 98.
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.

Symptom	Reason	Solution
Some cluster hosts do not appear when you click <b>Find Hosts</b> in install or update wizard.	You may have network connectivity problems.	<ul style="list-style-type: none"> <li>Make sure all cluster hosts have SSH port 22 open.</li> <li>Check other common causes of loss of connectivity such as firewalls and interference from SELinux.</li> </ul>
"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"> <li>For hostname configuration, see <a href="#">Configuring Network Names</a> (CDH 4) or <a href="#">Configuring Network Names</a> on page 247 (CDH 5).</li> <li>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 <a href="#">configured the database</a>.</li> </ul> <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 <code>binlog_format=mixed</code> in /etc/my.cnf. For more information, see <a href="#">this MySQL bug report</a> . See also <a href="#">Cloudera Manager and Managed Service Datastores</a> on page 79.
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.)

## Troubleshooting Installation and Upgrade Problems

Symptom	Reason	Solution
Cloudera services fail to start.	Java may not be installed or may be installed at a custom location.	See <a href="#">Configuring a Custom Java Home Location</a> on page 176 for more information on resolving this issue.
The Activity Monitor displays a status of <b>BAD</b> in the Cloudera Manager Admin Console. The log file contains the following message:  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.	The MySQL thread stack is too small.	<ol style="list-style-type: none"> <li>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>.</li> <li>Restart the <code>mysql</code> service: <code>\$ sudo service mysql restart</code></li> <li>Restart Activity Monitor.</li> </ol>
The Activity Monitor fails to start. Logs contain the error <code>read-committed isolation not safe for the statement binlog format</code> .	The <code>binlog_format</code> is not set to mixed.	Modify the <code>mysql.cnf</code> file to include the entry for <code>binlog_format</code> as specified in <a href="#">MySQL Database</a> on page 98.
Attempts to reinstall lower versions of CDH or Cloudera Manager using <code>yum</code> fails.	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>Clear information in the yum cache:</p> <ol style="list-style-type: none"> <li>Connect to the CDH host.</li> <li>Execute either of the following commands: <code>\$ yum --enablerepo='*' clean all</code> or <code>\$ rm -rf /var/cache/yum/cloudera*</code></li> <li>After clearing the cache, proceed with installation.</li> </ol>
Hive, Impala, or Hue complains about a missing table in the Hive metastore database.	The Hive Metastore database must be upgraded after a major Hive version change (Hive had a major version change in CDH 4.0, 4.1, 4.2, and 5.0).	Follow the instructions in <a href="#">Upgrading CDH 4</a> on page 681 or <a href="#">Upgrading CDH 4 to CDH 5</a> on page 660 for upgrading the Hive Metastore database schema. Stop all Hive services before performing the upgrade.
The <b>Create Hive Metastore Database Tables</b> command fails due to a problem with an escape string.	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>	As the administrator user, use the following command to turn <code>standard_conforming_strings</code> off:  <code>ALTER DATABASE &lt;hive_db_name&gt; SET standard_conforming_strings = off;</code>

Symptom	Reason	Solution
	property. Versions up to PostgreSQL 9.0 defaulted to off, but starting with version 9.0 the default is on.	
After upgrading to CDH 5, HDFS DataNodes fail to start with exception:	<p>HDFS caching, which is enabled by default in CDH 5, requires new memlock functionality from Cloudera Manager Agents.</p> <pre>Exception in secureMainjava.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>Do the following:</p> <ol style="list-style-type: none"> <li>Stop all CDH and managed services.</li> <li>On all hosts with Cloudera Manager Agents, hard restart the Agents. Before performing this step, ensure you understand the semantics of the hard_restart command by reading <a href="#">Hard Stopping and Restarting Agents</a>.</li> </ol> <ul style="list-style-type: none"> <li>Packages <ul style="list-style-type: none"> <li>RHEL-compatible 7 and higher:</li> </ul> <pre>\$ sudo service cloudera-scm-agent next_stop_hard \$ sudo service cloudera-scm-agent restart</pre> </li> <li>All other Linux distributions:</li> </ul> <pre>\$ sudo service cloudera-scm-agent hard_restart</pre> <ul style="list-style-type: none"> <li>Tarballs <ul style="list-style-type: none"> <li>To stop the Cloudera Manager Agent, run this command on each Agent host: <ul style="list-style-type: none"> <li>RHEL-compatible 7 and higher:</li> </ul> <pre>\$ sudo tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard \$ sudo tarball_root/etc/init.d/cloudera-scm-agent restart</pre> </li> <li>All other Linux distributions:</li> </ul> <pre>\$ sudo tarball_root/etc/init.d/cloudera-scm-agent hard_restart</pre> </li> <li>If you are running <a href="#">single user mode</a>, start Cloudera Manager Agent using the user account you chose. For example to run the Cloudera Manager Agent as</li> </ul>

## Troubleshooting Installation and Upgrade Problems

Symptom	Reason	Solution
		<p>cloudera-scm, you have the following options:</p> <ul style="list-style-type: none"><li>– Run the following command:<ul style="list-style-type: none"><li>– RHEL-compatible 7 and higher:<pre>\$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard \$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent restart</pre></li><li>– All other Linux distributions:<pre>\$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent hard_restart</pre></li></ul></li><li>– Edit the configuration files so the script internally changes the user, and then run the script as root:<ol style="list-style-type: none"><li>1. Remove the following line from <i>tarball_root/etc/init.d/cloudera-scm-agent</i>:<pre>export CMF_SUDO_CMD= " "</pre></li><li>2. Change the user and group in <i>tarball_root/etc/init.d/cloudera-scm-agent</i> to the user you want the Agent to run as. For example, to run as cloudera-scm, change the user and group as follows:<pre>USER=cloudera-scm GROUP=cloudera-scm</pre></li><li>3. Run the Agent script as root:<ul style="list-style-type: none"><li>• RHEL-compatible 7 and higher:<pre>\$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard \$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent restart</pre></li></ul></li></ol></li></ul>

Symptom	Reason	Solution
		<ul style="list-style-type: none"> <li>• All other Linux distributions:</li> </ul> <pre>\$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent hard_restart</pre> <p><b>3.</b> Start all services.</p>
You see the following error in NameNode log:	<pre>2014-10-16 18:36:29,112 WARN org.apache.hadoop.hdfs.server.namenode.FSNamesystem 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.FSImage at 2014-10-16</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 <a href="#">Upgrading to CDH 5.2</a> on page 637.</p>

## Troubleshooting Installation and Upgrade Problems

Symptom	Reason	Solution
<pre> at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat\$OutputCommitter at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat\$OutputCommitter at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat\$OutputCommitter 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.hdfs.server.namenode.NameNode: SHUTDOWN_MSG: </pre>		
If you are using an Oracle database and the Cloudera Navigator Analytics > Audit > Activity tab displays "No data available" and there is an Oracle error about "invalid identifier" with the query containing the reference to dbms_crypto in the log.	You have not granted execute permission to sys.dbms_crypto.	Run GRANT EXECUTE ON sys.dbms_crypto TO <i>nav</i> ;, where <i>nav</i> is the user of the Navigator Audit Server database.

## Rolling Back a CDH 4-to-CDH 5 Upgrade

You can roll back an upgrade from CDH 4 to CDH 5. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations. Any data created after the upgrade is lost.

In a typical upgrade, you first upgrade Cloudera Manager from version 4.x to version 5.x, and then you use the upgraded version of Cloudera Manager 5 to upgrade CDH 4 to CDH 5. (See [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 524.) If you want to roll back this upgrade, follow these steps to roll back your cluster to its state prior to the upgrade.



**Important:** Follow all of the steps in the order presented in this topic. Cloudera recommends that you read through the backup and rollback steps before starting the backup process. You may want to create a detailed plan to help you anticipate potential problems.

### High-Level Steps

1. Before the upgrade, [back up specified directories, files, and databases used in your CDH cluster](#).
2. To roll back your cluster to its state before the upgrade, perform the [rollback steps on the hosts in your CDH cluster](#).
3. Use Cloudera Manager 5 to start and manage the CDH 4 cluster. You can use Cloudera Manager 5 to manage both CDH 4 and CDH 5 clusters. (If required, you can also [roll back Cloudera Manager 5 to Cloudera Manager 4](#).)

### Limitations

The rollback procedure has the following limitations:

- If you have [finalized the HDFS upgrade](#), *you cannot roll back your cluster*.
- Configuration changes, including the addition of new services or roles after the upgrade, are not retained after rolling back Cloudera Manager.
- Cloudera recommends that you not make configuration changes or add new services and roles until you have finalized the HDFS upgrade and no longer require the option to roll back your upgrade.
- If your cluster is configured to use HBase replication, data written to HBase after the upgrade might not be replicated to peers when you start your rollback. This topic does not describe how to determine which, if any, peers have the replicated data and how to roll back that data. For more information about HBase replication, see [HBase Replication](#).



#### Note:

- Hadoop version 2.0, which is included with CDH 4, does not support HDFS rollback for clusters on which high availability is enabled for HDFS. If your CDH 4 cluster has high availability enabled for HDFS, you can temporarily reconfigure your cluster without high availability before proceeding with the rollback. After the rollback, you can re-enable high availability for HDFS. Procedures for these reconfigurations are provided in [the rollback procedures](#).
- Because of an HDFS bug, rollback fails on DataNodes because the DataNode `storageID` format changed between the Hadoop versions used in CDH 4 and CDH 5. The workaround is described in [the rollback procedures](#).

## Backing Up Before Upgrading from CDH 4 to CDH 5

Backing up CDH components before upgrading your Cloudera Manager and CDH software provides a way to roll back the upgrade. This topic provides procedures to back up your cluster so that you can roll back your cluster to its pre-upgrade state.

## Backup Steps

1. [Preparing to Back Up](#) on page 751
2. [Stopping the Cluster](#) on page 751
3. [Backing Up CDH 4 and Cloudera Manager Repository Files](#) on page 751
4. [Backing Up ZooKeeper](#) on page 752
5. [Backing Up HDFS \(With High Availability\)](#) on page 752
6. [Backing Up HDFS \(Without High Availability\)](#) on page 752
7. [Backing Up HBase](#) on page 752
8. [Backing Up Hive](#) on page 752
9. [Backing Up Oozie](#) on page 753
10. [Backing Up Search](#) on page 753
11. [Backing Up Sqoop 2](#) on page 753
12. [Backing Up Hue](#) on page 753
13. [Backing Up Cloudera Manager](#) on page 753
14. [Backing Up Other CDH Components](#) on page 753

### Preparing to Back Up

Because many storage locations are configurable, you may need to use the Cloudera Manager Admin Console to determine the location of files you need to back up. Where applicable, the parameter names that specify these locations are provided in the backup sections for each component. If you have not changed these values for your cluster, use the provided default values. To find these parameter values:

1. Open the Cloudera Manager Admin console.
2. Go to the service where you need to look up a parameter (for example, HDFS, HBase, or ZooKeeper).
3. Click the **Configuration** tab.
4. Enter the name of the parameter in the search box.

The parameter and its value display on the right.

For some services, you back up data stored in relational databases such as Oracle, MariaDB, MySQL, or PostgreSQL. [See the documentation](#) for those products to learn how to back up and restore the databases.



#### Important:

- As you back up the required files described in this topic, record which hosts the backups come from (or back up the files to the same host).
- After starting your backups, do not add additional components or change any configurations until your upgrade is successfully completed or rolled back.
- Complete all backup steps before starting the upgrade to CDH 5.

### Stopping the Cluster

Stop the CDH cluster before performing the backup:

1. Go to the **Home** page.
2. In the drop-down list next to your cluster, select **Stop**.

### Backing Up CDH 4 and Cloudera Manager Repository Files

If your cluster was installed using *packages*, back up the CDH 4 and Cloudera Manager repository files on all hosts from the system repository directory. If your cluster was installed using Cloudera *parcels*, back up only the Cloudera Manager repository file. Following are the typical locations for the repository directories:

## Rolling Back a CDH 4-to-CDH 5 Upgrade

Operating System	Path
RHEL	/etc/yum.repos.d
SLES	/etc/zypp/repos.d
Ubuntu or Debian	/etc/apt/sources.list.d

For example, on a RHEL or similar system, back up the files in `/etc/yum.repos.d` that have `cloudera` as part of their name.

### Backing Up ZooKeeper

On all ZooKeeper hosts, back up the ZooKeeper data directory specified with the `dataDir` property in the ZooKeeper configuration. The default location is `/var/lib/zookeeper`.

Record the permissions of the files and directories; you will need these to roll back ZooKeeper.

### Backing Up HDFS (With High Availability)

Follow this procedure to back up an HDFS deployment that has been configured for high availability.

1. On both NameNode hosts, back up one of the NameNode data directories specified with the `dfs.namenode.name.dir` property.
2. On each JournalNode, back up the JournalNode edits directory specified by the `dfs.journalnode.edits.dir` property. Note which JournalNode host the backup comes from.
3. Back up the `VERSION` files for each DataNode, noting which DataNode you are backing up. There may be multiple data directories in each node, but you need to back up only one of them on each DataNode. The location of the data directories is specified with the `dfs.datanode.data.dir` property. The `VERSION` file is located in the `current` subdirectory. You will use the `version` files to get the `storageID` when you perform the rollback steps; for example (using the default path): `/data/dfs/dn/current/VERSION`. You only need this `storageID` when rolling back the DataNodes; copying the `VERSION` file is suggested as a convenience.

### Backing Up HDFS (Without High Availability)

Use this procedure to back up an HDFS deployment that has not been configured for high availability.

1. On the NameNode host, back up one of the NameNode data directories specified with the `dfs.namenode.name.dir` property.
2. Back up the `VERSION` files for each DataNode, noting which DataNode you are backing up. There may be multiple data directories in each node, but you need to back up only one of them on each DataNode. The location of the data directories is specified with the `dfs.datanode.data.dir` property. The `VERSION` file is located in the `current` subdirectory. You will use the `version` files to get the `storageID` when you perform the rollback steps; for example (using the default path): `/data/dfs/dn/current/VERSION`. You only need this `storageID` when rolling back the DataNodes; copying the `VERSION` file is suggested as a convenience.

### Backing Up HBase

Because the rollback procedure also rolls back HDFS, the data in HBase is also rolled back. In addition, HBase metadata stored in ZooKeeper is recovered as part of the ZooKeeper rollback procedure.

If your cluster is configured to use HBase replication, Cloudera recommends that you document all replication peers. If necessary (for example, because the HBase znode has been deleted), you can roll back HBase as part of the HDFS rollback without the ZooKeeper metadata. This metadata can be reconstructed in a fresh ZooKeeper installation, with the exception of the replication peers, which you must add back. For information on enabling HBase replication, listing peers, and adding a peer, see [HBase Replication](#) in the CDH 4 documentation.

### Backing Up Hive

Back up the database that backs the Hive metastore. See [Backing up Databases](#) on page 754.

## Backing Up Oozie

Back up the Oozie database. See [Backing up Databases](#) on page 754.

## Backing Up Search

On each Solr node, back up the contents of the Solr Data directory and record the permissions for the directory. This location is specified with the **Solr Data Directory** property. The default location is:

```
/var/lib/solr
```

Search data on ZooKeeper is restored as part of the ZooKeeper rollback.

## Backing Up Sqoop 2

If you are not using the default embedded Derby database for Sqoop 2, [back up the database](#) you have configured for Sqoop 2. Otherwise, back up the repository subdirectory of the Sqoop 2 metastore directory. This location is specified with the **Sqoop 2 Server Metastore Directory** property. The default location is: /var/lib/sqoop2. For this default location, Derby database files are located in /var/lib/sqoop2/repository.

## Backing Up Hue

1. Back up the Hue database. See [Backing up Databases](#) on page 754.
2. Back up the app registry file, <HUE\_HOME>/app.reg, where HUE\_HOME is the location of your Hue installation. For package installs, this is usually /usr/lib/hue; for parcel installs, this is usually, /opt/cloudera/parcels/<parcel version>/lib/hue/.

## Backing Up Cloudera Manager

1. Stop Cloudera Management Services using Cloudera Manager:
  - a. Select **Clusters > Cloudera Management Service**.
  - b. Select **Actions > Stop**.
2. Stop Cloudera Manager Server by running the following command on the Cloudera Manager Server host:

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

3. On the host where Cloudera Manager Server is running, back up the /etc/cloudera-scm-server/db.properties file.
4. On the host where the Event Server role is configured to run, back up the contents of the directory specified with the **Event Server Index Directory** property (the default value is /var/lib/cloudera-scm-eventserver).
5. Back up the /etc/cloudera-scm-agent/config.ini file on each host in the cluster.
6. Back up the following Cloudera Manager-related databases; see [Backing up Databases](#) on page 754:
  - Cloudera Manager Server
  - Activity Monitor (depending on your deployment, this role may not be installed)
  - Reports Manager
  - Service Monitor
  - Host Monitor
  - Navigator Audit Server
  - Navigator Metadata Server

## Backing Up Other CDH Components

No backups are required for the following components:

- MapReduce
- YARN
- Spark
- Pig

## Rolling Back a CDH 4-to-CDH 5 Upgrade

- Sqoop
- Impala

### Backing up Databases

Several steps in the backup procedures require you to back up various databases used in a CDH cluster. The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.

See the following vendor resources for more information:

- **MariaDB 5.5:** <http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/>
- **MySQL 5.5:** <http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html>
- **MySQL 5.6:** <http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html>
- **PostgreSQL 8.4:** <https://www.postgresql.org/docs/8.4/static/backup.html>
- **PostgreSQL 9.2:** <https://www.postgresql.org/docs/9.2/static/backup.html>
- **PostgreSQL 9.3:** <https://www.postgresql.org/docs/9.3/static/backup.html>
- **Oracle 11gR2:** [http://docs.oracle.com/cd/E11882\\_01/backup.112/e10642/toc.htm](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm)

## Procedure for Rolling Back a CDH 4-to-CDH 5 Upgrade

You can roll back to CDH 4 after upgrading to CDH 5 only if the [HDFS upgrade has not been finalized](#). The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations. Data created after the upgrade is lost.



**Important:** When performing these rollback steps, use backups taken before you started the upgrade. For steps where you need to restore the contents of a directory, clear the contents of the directory before copying the backed-up files to the directory. If you fail to do this, artifacts from the original upgrade can cause problems if you attempt the upgrade again after the rollback.

### Rollback Steps

1. [Determining the Last Active NameNode](#) on page 754.
2. [Downgrading the Software](#) on page 755.
3. [Rolling Back ZooKeeper](#) on page 756.
4. [Rolling Back HDFS](#) on page 756.
5. [Re-enabling HDFS with High Availability](#) on page 758 (optional).
6. [Rolling Back HBase](#) on page 758.
7. [Rolling Back Hive](#) on page 759.
8. [Rolling Back Oozie](#) on page 759.
9. [Rolling Back Search](#) on page 759.
10. [Rolling Back Sqoop 2](#) on page 759.
11. [Rolling Back Hue](#) on page 759.
12. [Rolling Back Cloudera Manager](#) on page 760.

#### Determining the Last Active NameNode

If your cluster has high availability for HDFS enabled, determine which NameNode host was last active:

1. In Cloudera Manager, select **Clusters > HDFS > Instances**.

A list of role types and hosts displays.

2. Look for the Role Type **NameNode (Active)**.

The NameNode that is not active is called the *standby* NameNode and displays in the list of Role Types as **NameNode (Standby)**.

## Downgrading the Software

### 1. Stop the cluster:

- On the **Home > Status** tab, click



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

- 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.

### 2. Downgrade the JDK, if necessary.

When you upgraded your cluster to CDH 5, you were required to also upgrade the JDK. If you are rolling back your cluster to Cloudera Manager version 4.6.x or lower and CDH version 4.3.x or lower, Cloudera recommends that you downgrade your JDK on all hosts to the version deployed before the upgrade, or to JDK 1.6. You can also choose to run your cluster using the version of the JDK you installed during the upgrade. To verify that the JDK you choose is supported by Cloudera Manager and CDH, see [JDK Compatibility](#).

You can have two versions of a JDK on a host machine, but you must make sure the correct version is used by the software. See: [Java Development Kit Installation](#) on page 78.

### 3. Depending on whether your cluster was installed using *parcels* or *packages*, do one of the following:

- **Parcels**

1. Log in to the Cloudera Manager Admin Console.
2. Select **Hosts > Parcels**.

A list of parcels displays.

3. Locate the CDH 4 parcel and click **Activate**. (This automatically deactivates the CDH 5 parcel.) See [Activating a Parcel](#) on page 59 for more information. If the parcel is not available, use the **Download** button to download the parcel.
4. If you include any additional components in your cluster, such as Search or Impala, click **Activate** for those parcels.



#### Important:

Do not start any services.

If you accidentally restart services, stop your cluster before proceeding.

- **Packages**

1. Log in as a privileged user to all hosts in your cluster.
2. Run the following command to uninstall CDH 5:

Operating System	Command
RHEL	\$ sudo yum remove bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc
SLES	\$ sudo zypper remove bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc
Ubuntu or Debian	\$ sudo apt-get purge bigtop-jsvc bigtop-utils bigtop-tomcat hue-common sqoop2-client hbase-solr-doc solr-doc

## Rolling Back a CDH 4-to-CDH 5 Upgrade

3. Remove the CDH 5 repository files from the system repository directory. For example, on a RHEL or similar system, remove all files in `/etc/yum.repos.d` that have `cloudera` as part of the name. (Make sure that you have backed up these files, as instructed in [Backing Up CDH 4 and Cloudera Manager Repository Files](#) on page 751.)
4. Restore the CDH 4 repository files that you previously backed up to the repository directory.
5. Re-install the CDH 4 packages using the same installation path you used for the initial installation; see [Installing Cloudera Manager and CDH](#). Repeat only the installation steps and not the configuration steps. (The configurations are already stored in Cloudera Manager.) Make sure that you include any additional components used in your cluster, such as MapReduce 1, YARN, Spark, Sqoop, or Impala.

### Rolling Back ZooKeeper

1. Restore the contents of the `dataDir` on each ZooKeeper server. These files are located in a directory specified with the `dataDir` property in the ZooKeeper configuration. The default location is `/var/lib/zookeeper`.
2. Make sure that the permissions of all the directories and files are as they were before the upgrade.
3. Start ZooKeeper using Cloudera Manager.

### Rolling Back HDFS

You cannot roll back HDFS while high availability is enabled. The rollback procedure in this topic creates a temporary configuration without high availability. Regardless of whether high availability is enabled, follow the steps in this section to create the temporary configuration, and to re-enable high availability, follow the steps in [Re-enabling HDFS with High Availability](#) on page 758.

1. Create a rollback configuration directory on each host that has the HDFS role; for example `/etc/hadoop/conf.rollback`.
2. Create a `core-site.xml` file in this directory on all hosts with the HDFS role. The `<value>` element in this file references the NameNode host. For HDFS with high availability enabled, choose the last active NameNode host (see [Determining the Last Active NameNode](#) on page 754). The file needs to contain only the following:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://NameNode_host:port</value>
    <!-- For example:
        hdfs://a1234.cloudera.com:8020
    -->
  </property>
</configuration>
```

3. Create an `hdfs-site.xml` file in the rollback configuration directory on all hosts with the HDFS role. The file specifies values for the following properties:

- `dfs.namenode.name.dir`
- `dfs.datanode.data.dir`
- `dfs.namenode.checkpoint.dir`

You can omit the `dfs.namenode.checkpoint.dir` property if high availability was enabled for HDFS in your cluster.

To find values you need to create the `hdfs-site.xml` file:

1. Open Cloudera Manager.
2. Go to the HDFS service.
3. Click the **Configuration** tab.
4. Enter the property name in the search field.
5. Copy each path defined for the property into the `<value>` element for the property. Precede each path with `file://` and separate each directory path with a comma.

For example:

```
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///data/1/dfs/nn,file:///data/2/dfs/nn</value>
  </property>
  <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,file:///data/5/dfs/dn
    </value>
  </property>
  <property>
    <name>dfs.namenode.checkpoint.dir</name>
    <value>file:///data/1/dfs/snn</value>
  </property>
</configuration>
```

4. Restore the `storageID` values from the `VERSION` files. On each DataNode, edit the `VERSION` file in each data directory and replace the value of the `storageID` field with the value from the `VERSION` file you backed up from the same DataNode. The `storageID` is the same for each data directory on a DataNode host.



**Important:**

Restore the `storageID` on *all data directories* on *all DataNodes*.

Do not restore the entire `VERSION` file.

5. Copy the `/etc/hadoop/conf/log4j.properties` file to the `/etc/hadoop/conf.rollback` directory (the rollback configuration directory you created previously).

6. Verify that the cluster is now running CDH 4 by running the following command on all Hadoop nodes in your cluster:

```
hadoop version
```

The version number appears right after the string `cdh`. For example:

```
$ hadoop version
Hadoop 2.0.0-cdh4.7.1
...
```

7. Run the following command on the NameNode host. (If high availability is enabled for HDFS, run the command on the active NameNode.)

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback namenode -rollback
```

This command also starts the NameNode.

8. While the NameNode is running, run the following command on all DataNode hosts:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback datanode -rollback
```

This command also starts the DataNodes.

9. If your cluster uses a Secondary NameNode, do the following on the Secondary NameNode host:

- a. Delete the directory defined with the `dfs.namenode.checkpoint.dir` parameter.
- b. Run the following command while the active NameNode and all DataNodes are running:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback secondarynamenode -format
```

## Rolling Back a CDH 4-to-CDH 5 Upgrade

- 10 When rollback is completed, stop all daemons by typing `CTRL-C` in all open terminal sessions (the NameNode hosts and DataNode hosts on which you ran the `hdfs` commands). You can monitor the progress of the rollback by opening the NameNode web interface in a web browser at `NameNode_Host:50070`. On that page, when all the blocks are reported and the NameNode no longer reports that it is in **Safemode**, the rollback is complete.
- 11 If your cluster does not have high availability enabled for HDFS, use the Cloudera Manager Admin Console to start the HDFS service and then continue your cluster rollback with [Rolling Back HBase](#) on page 758:
  - a. Go to the HDFS service.
  - b. Select **Actions > Start**.

If your cluster has high availability enabled for HDFS, continue with the next section to remove the temporary non-HA configuration and re-enable high availability.

### Re-enabling HDFS with High Availability

The rollback steps performed so far have restored all of the DataNodes and one of the NameNodes. In a high-availability configuration, you also need to roll back the other (standby) NameNode and the JournalNodes.

1. On each JournalNode, restore the backed-up edits directory to the same location.
2. On the standby NameNode (the NameNode host that has not been rolled back), delete the directories specified in the `dfs.namenode.name.dir` property. Failing to delete these directories causes the rollback to fail.
3. Using Cloudera Manager, start the last active NameNode:
  - a. Go to the HDFS service.
  - b. Click the **Instances** tab.
  - c. Select **NameNode(Active)** from the list of Role Types.
  - d. Select **Actions for Selected > Start**.
4. Using Cloudera Manager, bootstrap the standby NameNode:
  - a. Go to the HDFS service.
  - b. Click the **Instances** tab.
  - c. Select **NameNode(Standby)**.
  - d. Select **Actions > Bootstrap Standby NameNode**.

The `fsImage` is restored from the last active NameNode to the standby NameNode.

5. Restart the HDFS service:
  - a. Go to the HDFS service.
  - b. Select **Actions > Restart**.

### Rolling Back HBase

No additional steps are required to roll back HBase. When you rolled back HDFS, all of the HBase data was also rolled back.

Using Cloudera Manager, start HBase:

1. Go to the HBase service.
2. Select **Actions > Start**.

If you encounter errors when starting HBase, delete the znode in ZooKeeper and then start HBase again:

1. In Cloudera Manager, look up the value of the `zookeeper.znode.parent` property. The default value is `/hbase`.
2. Connect to the ZooKeeper ensemble by running the following command from any HBase gateway host:

```
zookeeper-client -server zookeeper_ensemble
```

To find the value to use for `zookeeper_ensemble`, open the `/etc/hbase/conf/hbase-site.xml` file on any HBase gateway host. Use the value of the `hbase.zookeeper.quorum` property.

**Note:**

If you have deployed a secure cluster, you must connect to ZooKeeper using a client `jaas.conf` file. You can find such a file in an HBase process directory (`/var/run/cloudera-scm-agent/process/`). Specify the `jaas.conf` using the JVM flags by running the following commands in the ZooKeeper client:

```
CLIENT_JVMFLAGS=
"-Djava.security.auth.login.config=/var/run/cloudera-scm-agent/process/Base process directory/jaas.conf"
zookeeper-client -server <zookeeper_ensemble>
```

The ZooKeeper command-line interface opens.

**3.** Enter the following command:

```
rmr /hbase
```

### **Rolling Back Hive**

Restore the Hive metastore database from your backup. [See the documentation for your database](#) for details.

### **Rolling Back Oozie**

Restore the Oozie database from your backup. [See the documentation for your database](#) for details.

### **Rolling Back Search**

Restore the contents of the `/var/lib/solr` directory from your backup. Ensure that the file permissions are the same as they were before rolling back. (The permissions are typically `solr:solr`.)

### **Rolling Back Sqoop 2**

Restore the Sqoop 2 database from your backup. [See the documentation for your database](#) for details.

If you are not using the default embedded Derby database for Sqoop 2, [restore the database](#) you have configured for Sqoop 2. Otherwise, restore the `repository` subdirectory of the Sqoop 2 metastore directory from your backup. This location is specified with the **Sqoop 2 Server Metastore Directory** property. The default location is `/var/lib/sqoop2`. For this default location, Derby database files are located in `/var/lib/sqoop2/repository`.

### **Rolling Back Hue**

- 1.** Restore the Hue database from your backup.
- 2.** Restore the file, `app.reg`, from your backup. Place it in the location of your Hue installation (referred to as `HUE_HOME`). For package installs, this is usually `/usr/lib/hue`; for parcel installs, this is usually, `/opt/cloudera/parcels/<parcel version>/lib/hue/`.
- 3.** Using Cloudera Manager, install the Beeswax role:
  - a.** Go to the Hue service.
  - b.** Select the **Instances** tab.
  - c.** Click **Add**.
  - d.** Locate the row for the host with the Hue Server role and select **Beeswax Server**.
  - e.** Click **Continue**.

## Rolling Back a CDH 4-to-CDH 5 Upgrade

### Rolling Back Cloudera Manager

After you complete the rollback steps, your cluster is using Cloudera Manager 5 to manage your CDH 4 cluster. You can continue to use Cloudera Manager 5 to manage your CDH 4 cluster, or you can downgrade to Cloudera Manager 4 by following these steps:

1. Stop your CDH cluster using Cloudera Manager:
  - a. Go to the **Home** page.
  - b. In the drop-down list next to your cluster, select **Stop**.
2. Stop Cloudera Management Services:
  - a. Select **Clusters > Cloudera Management Service**.
  - b. Select **Actions > Stop**
3. Stop Cloudera Manager Server by running the following command on the Cloudera Manager Server host:

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

4. Stop the Cloudera Manager Agents by running the following command on all hosts in your cluster:

```
sudo service cloudera-scm-agent stop
```

5. Downgrade the JDK, if necessary.

If you are rolling back to Cloudera Manager version 4.6.x or lower, downgrade the version of your JDK to the version deployed before the upgrade, or the latest Oracle upgrade for JDK 1.6. You can also choose to run Cloudera Manager using the version of the JDK you installed during the upgrade. To verify that the JDK you choose is supported by Cloudera Manager and CDH, see [JDK Compatibility](#).

You can keep two versions of a JDK on a host machine, but you must make sure the correct version is used by the software. See [Java Development Kit Installation](#) on page 78 and [Configuring a Custom Java Home Location](#) on page 176 for information about specifying a JDK.

6. Downgrade the software:

- a. Run the following command on the Cloudera Manager Server host:

Operating System	Command
RHEL	\$ sudo yum remove cloudera-manager-server
SLES	\$ sudo zypper remove cloudera-manager-server
Ubuntu or Debian	\$ sudo apt-get purge cloudera-manager-server

- b. Run the following command on all hosts:

Operating System	Command
RHEL	\$ sudo yum remove cloudera-manager-daemons cloudera-manager-agent
SLES	\$ sudo zypper remove cloudera-manager-daemons cloudera-manager-agent
Ubuntu or Debian	\$ sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent

- c. Restore the Cloudera Manager repository files that you previously backed up to the Cloudera repository directory.
  - d. Run the following commands on all hosts:

Operating System	Command
RHEL	\$ sudo yum clean all \$ sudo yum install cloudera-manager-agent
SLES	\$ sudo zypper refresh -s \$ sudo zypper install cloudera-manager-agent
Ubuntu or Debian	\$ sudo apt-get update \$ sudo apt-get install cloudera-manager-agent

- e. Run the following commands on the Cloudera Manager server host:

Operating System	Command
RHEL	\$ sudo yum install cloudera-manager-server
SLES	\$ sudo zypper install cloudera-manager-server
Ubuntu or Debian	\$ sudo apt-get install cloudera-manager-server

7. Restore the following Cloudera Manager databases:

- Cloudera Manager Server
- Activity Monitor (depending on your deployment, this role may not be installed)
- Reports Manager
- Service Monitor
- Host Monitor
- Navigator Audit Server
- Navigator Metadata Server

[See the documentation for your database](#) for details.



**Important:** Restore the databases to their pre-upgrade state. If the Cloudera Manager databases are restored in a way that leaves tables that were created during the upgrade, there could be problems if you attempt to upgrade Cloudera Manager again after the rollback.

8. On the Cloudera Manager Server, restore the following from your backup to the same location:

- a. The /etc/cloudera-scm-server/db.properties file
- b. The contents of the /var/lib/cloudera-scm-eventserver directory

9. On each host in the cluster, restore the /etc/cloudera-scm-agent/config.ini file from your backup.

- 10 Start Cloudera Manager by running the following command on the Cloudera Manager Server host:

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

- 11 Start the Cloudera Manager Agents by running the following command on all hosts in your clusters:

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

- 12 Log in to the Cloudera Manager Admin console and start the Cloudera Management Service:

- a. Select **Clusters > Cloudera Management Service**.

## Rolling Back a CDH 4-to-CDH 5 Upgrade

- b. Select **Actions > Start**.

13 Start your cluster:

- a. Select **Clusters > Cluster Name**.
- b. Select **Actions > Start**.

### Restoring Databases

Several steps in the rollback procedures require you to restore previously backed-up databases. The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.



**Important:** Restore the databases to their exact state as of when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- **MariaDB 5.5:** <http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/>
- **MySQL 5.5:** <http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html>
- **MySQL 5.6:** <http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html>
- **PostgreSQL 8.4:** <https://www.postgresql.org/docs/8.4/static/backup.html>
- **PostgreSQL 9.2:** <https://www.postgresql.org/docs/9.2/static/backup.html>
- **PostgreSQL 9.3:** <https://www.postgresql.org/docs/9.3/static/backup.html>
- **Oracle 11gR2:** [http://docs.oracle.com/cd/E11882\\_01/backup.112/e10642/toc.htm](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm)