



# Cloudera Upgrade

**Important Notice**

© 2010-2017 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, and any other product or service names or slogans contained in this document are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder.

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**

**US: 1-888-789-1488**

**Intl: 1-650-362-0488**

**www.cloudera.com**

**Release Information**

Version: Cloudera Enterprise 5.13.x

Date: October 12, 2017

# Table of Contents

<b>Cloudera Upgrade Overview.....</b>	<b>7</b>
Assessing the Impact of an Upgrade.....	7
Overview of Upgrading Cloudera Manager.....	8
Overview of Upgrading CDH.....	9
Overview of Upgrading Cloudera Navigator Components.....	9
Upgrading the JDK.....	10
<b>Upgrading Cloudera Manager.....</b>	<b>11</b>
Upgrading Cloudera Manager 5 Using Packages.....	11
Step 1: Collect Upgrade Information.....	12
Step 2: Complete Pre-Upgrade Steps.....	13
Step 3: Back Up Cloudera Manager Databases.....	13
Step 4: Upgrade the JDK .....	14
Step 5: Establish Access to the Software.....	14
Step 6: Prepare the Cloudera Navigator Data Management Component for Upgrade.....	14
Step 7: Upgrade the Cloudera Manager Server.....	15
Step 8: Verify and Test the Upgrade.....	23
Step 9: Upgrade any Required Navigator Components.....	24
Step 10: (Optional) Upgrade CDH.....	24
Upgrading Cloudera Manager 5 Using Tarballs.....	24
Step 1: Collect Upgrade Information.....	25
Step 2: Complete Pre-Upgrade Steps.....	25
Step 3: Upgrade the JDK.....	26
Step 4: Creating and Using a Package Repository for Cloudera Manager.....	26
Step 5: Upgrading the Cloudera Navigator Data Management Component.....	26
Step 6: Upgrading the Cloudera Manager Server and Agents.....	26
Step 7: Verify and Test the Upgrade.....	30
Step 8: Upgrading Cloudera Navigator Components.....	31
Step 9: (Optional) Upgrade CDH.....	31
Package Dependencies.....	31
<b>Upgrading CDH and Managed Services Using Cloudera Manager.....</b>	<b>34</b>
Upgrading CDH 5.x.....	34
Minor Version Upgrades.....	34
Maintenance Version Upgrades.....	34
Upgrading to CDH 5.x Using a Rolling Upgrade.....	35

Step 1: Collect Upgrade Information.....	35
Step 2: Complete Pre-Upgrade Steps.....	36
Step 3: Ensure High Availability Is Enabled.....	37
Step 4: Back Up HDFS Metadata.....	37
Step 5: Back Up Databases.....	37
Step 6: Run the Upgrade Wizard.....	38
Step 7: Recover from Failed Steps or Perform a Manual Upgrade.....	40
Step 8: Remove the Previous CDH Version Packages and Refresh Symlinks.....	40
Step 9: Finalize HDFS Rolling Upgrade.....	41
Step 10: Exit Maintenance Mode.....	41
Step 11: Clear Browser Cache (Hue only).....	41
Upgrading to CDH 5.x Using Parcels.....	42
Step 1: Collect Upgrade Information.....	42
Step 2: Complete Pre-Upgrade Steps.....	42
Step 3: Stop Cluster Services.....	44
Step 4: Back up the HDFS Metadata on the NameNode.....	44
Step 5: Back Up Databases.....	45
Step 6: Run the Upgrade Wizard.....	46
Step 7: Recover from Failed Steps or Perform a Manual Upgrade.....	48
Step 8: Remove the Previous CDH Version Packages and Refresh Symlinks.....	48
Step 9: Finalize the HDFS Metadata Upgrade.....	49
Step 10: Exit Maintenance Mode.....	49
Step 11: Clear Browser Cache (Hue only).....	49
Upgrading to CDH 5.x Using Packages.....	49
Step 1: Collect Upgrade Information.....	50
Step 2: Complete Pre-Upgrade Steps.....	50
Step 3: Upgrade Unmanaged Components.....	52
Step 4: Stop Cluster Services.....	52
Step 5: Back up the HDFS Metadata on the NameNode.....	52
Step 6: Back Up Databases.....	52
Step 7: Upgrade Managed Components.....	53
Step 8: Update Symlinks for the Newly Installed Components.....	57
Step 9: Run the Upgrade Wizard.....	57
Step 10: Recover from Failed Steps or Perform a Manual Upgrade.....	57
Step 11: Finalize the HDFS Metadata Upgrade.....	58
Step 12: Exit Maintenance Mode.....	58
Step 13: Clear Browser Cache (Hue only).....	58
Upgrade Managed Components Using a Specific Set of Packages.....	58
Performing Upgrade Wizard Actions Manually.....	61
Upgrading to CDH 5.8.0 or CDH 5.8.1 When Using the Flume Kafka Client.....	63

## **Upgrading to Oracle JDK 1.8.....65**

Upgrading to Oracle JDK 1.8 in a Cloudera Manager Deployment.....	65
---	----

Upgrading to Oracle JDK 1.8 in an Unmanaged Deployment.....	66
Using AES-256 Encryption.....	66

## **Upgrading Cloudera Navigator Components.....68**

Upgrading the Cloudera Navigator Data Management Component.....	68
Upgrading Cloudera Navigator Key Trustee Server.....	69
<i>Upgrading Cloudera Navigator Key Trustee Server 3.x to 5.4.x.....</i>	<i>69</i>
<i>Upgrading Cloudera Navigator Key Trustee Server 3.8 to 5.5 Using the ktupgrade Script.....</i>	<i>74</i>
<i>Upgrading Cloudera Navigator Key Trustee Server 5.4.x or Higher.....</i>	<i>78</i>
Upgrading Cloudera Navigator Key HSM.....	84
<i>Setting Up an Internal Repository.....</i>	<i>84</i>
<i>Upgrading Key HSM.....</i>	<i>84</i>
Upgrading Key Trustee KMS.....	85
<i>Setting Up an Internal Repository.....</i>	<i>85</i>
<i>Upgrading Key Trustee KMS Using Parcels.....</i>	<i>85</i>
<i>Upgrading Key Trustee KMS Using Packages.....</i>	<i>85</i>
Upgrading Cloudera Navigator Encrypt.....	86
<i>Setting Up an Internal Repository.....</i>	<i>86</i>
<i>Upgrading Navigator Encrypt (RHEL-Compatible).....</i>	<i>86</i>
<i>Upgrading Navigator Encrypt (SLES).....</i>	<i>87</i>
<i>Upgrading Navigator Encrypt (Debian or Ubuntu).....</i>	<i>87</i>
<i>Best Practices for Upgrading Navigator Encrypt Hosts.....</i>	<i>88</i>

## **Creating and Using a Package Repository for Cloudera Manager.....89**

Creating a Permanent Remote Repository.....	89
<i>Installing a Web Server.....</i>	<i>89</i>
Creating a Temporary Remote Repository.....	90
Modifying Clients to Find the Repository.....	90

## **Database Considerations for Cloudera Manager Upgrades.....92**

Backing up Databases.....	92
Creating New Databases.....	92
Modifying Databases to Support UTF-8.....	92
Modifying Databases to Support Appropriate Maximum Connections.....	93
Next Steps.....	94

## **Re-Running the Cloudera Manager Upgrade Wizard.....95**

## **Reverting a Failed Cloudera Manager Upgrade.....96**

Reinstall the Cloudera Manager Server Packages.....	96
Start the Server.....	98

**Upgrading Unmanaged CDH Using the Command Line.....99**

Upgrading from an Earlier CDH 5 Release to the Latest Release.....	99
<i>Important Tasks.....</i>	<i>99</i>
<i>Before Upgrading to the Latest Release of CDH.....</i>	<i>100</i>
<i>Upgrading from CDH 5.4.0 or Higher to the Latest Release.....</i>	<i>101</i>
<i>Upgrading from a Release Lower than CDH 5.4.0 to the Latest Release.....</i>	<i>114</i>

**Upgrading Host Operating Systems in a CDH Cluster.....130**

# Cloudera Upgrade Overview

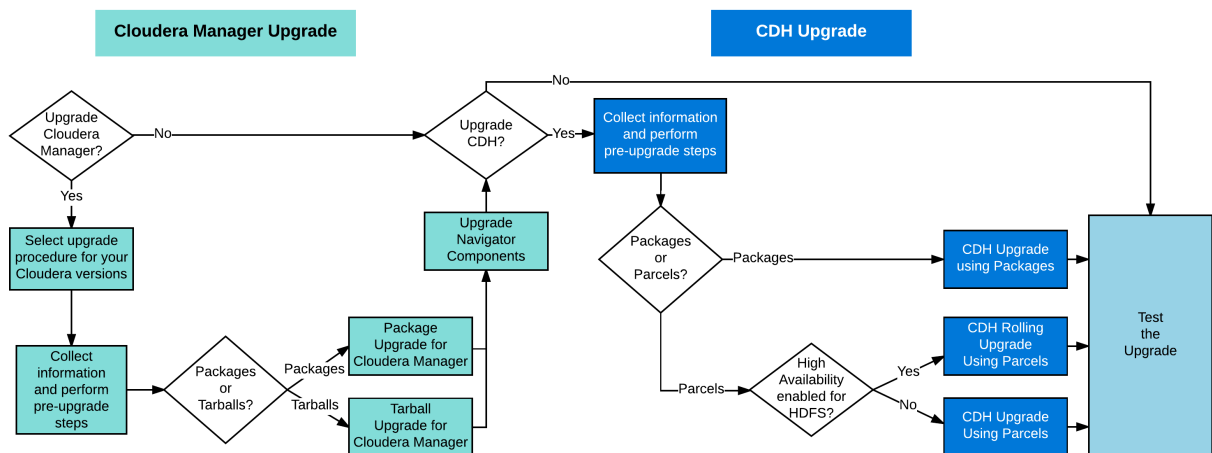
This topic provides an overview of upgrade procedures for Cloudera Manager and CDH.

The procedures described here are for clusters managed by Cloudera Manager. For information about upgrading unmanaged CDH clusters (clusters that are not managed by Cloudera Manager), see [Upgrading Unmanaged CDH Using the Command Line](#) on page 99.

When upgrading Cloudera Manager, you can use tarballs or operating system packages. When upgrading CDH, you can use packages or parcels. You might also need to install a new version of the JDK. Cloudera Navigator is also upgraded when you upgrade Cloudera Manager.

You are not required to upgrade Cloudera Manager and CDH at the same time, but the versions of Cloudera Manager and CDH must be compatible. Cloudera Manager can manage clusters for the current and previous major versions of CDH and any equal or lower minor version of CDH. For example, Cloudera Manager 5.7.1 can manage clusters with CDH 5.7.2, CDH 5.6.1, and CDH 4.8.6 but cannot manage a cluster with CDH 5.8.1. Cloudera Manager 5.x cannot manage clusters using CDH 3.x.

**You can choose from several options as you upgrade:**



## Assessing the Impact of an Upgrade

Plan for a sufficient maintenance window to perform an upgrade. Depending on which components you are upgrading, the number of hosts in your cluster, and the type of hardware, you might need up to a full day to upgrade your cluster. Before you begin the upgrade, you need to gather some information; these steps are also detailed in the Cloudera Manager and CDH upgrade procedures.

Before upgrading, consult the [release notes](#) for Cloudera Manager and CDH to learn about API changes, deprecated features, new features, and incompatible changes. Also check the [CDH 5 and Cloudera Manager 5 Requirements and Supported Versions](#) page to make sure that you are using a supported operating system, JDK, database, and other components.

There are three types of upgrades: major, minor, and maintenance:

### Major Upgrades

A major upgrade typically has the following characteristics:

- Large changes to functionality and update of Hadoop to a more recent version
- Incompatible changes in data formats
- Significant changes and additions to the user interface in Cloudera Manager

- Database schema changes for Cloudera Manager that are automatically handled by the upgrade process
- Significant down time for the cluster is required.
- Client configurations are redeployed.

You can only upgrade from one major version to the next major version, for example from version 4.8.1 to version 5.8.0. To be ready for the next major version upgrade, Cloudera recommends that you upgrade to a 5.x version.

### Minor Upgrades

Minor upgrades upgrade your software to a higher minor version of a major release—for example from version 5.4.x to version 5.8.x—and typically include the following:

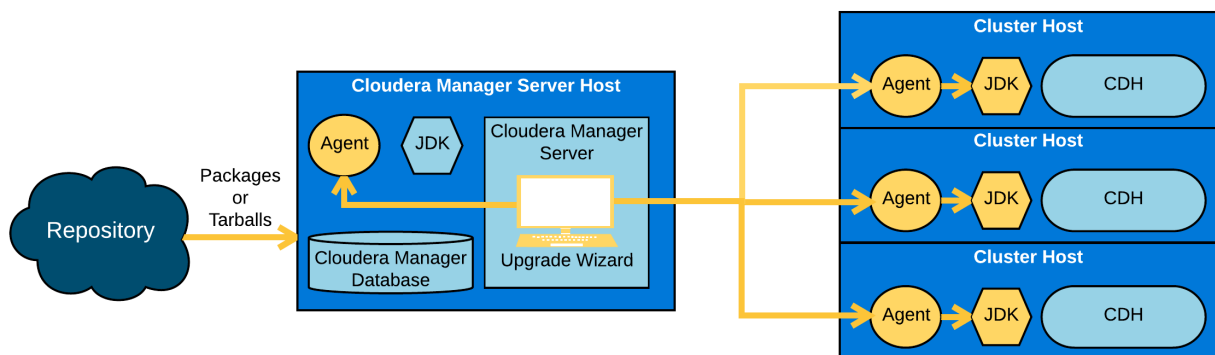
- New functionality
- Bug fixes
- Potential database schema changes for Cloudera Manager that are handled automatically

Incompatible changes or changes to data formats are generally not introduced in minor upgrades. [Client configurations](#) are redeployed.

### Maintenance Upgrades

Maintenance upgrades are used only to fix critical bugs or address security issues. No new functionality or incompatible changes are introduced.

## Overview of Upgrading Cloudera Manager



**Figure 1: Cloudera Manager 5 Upgrade**

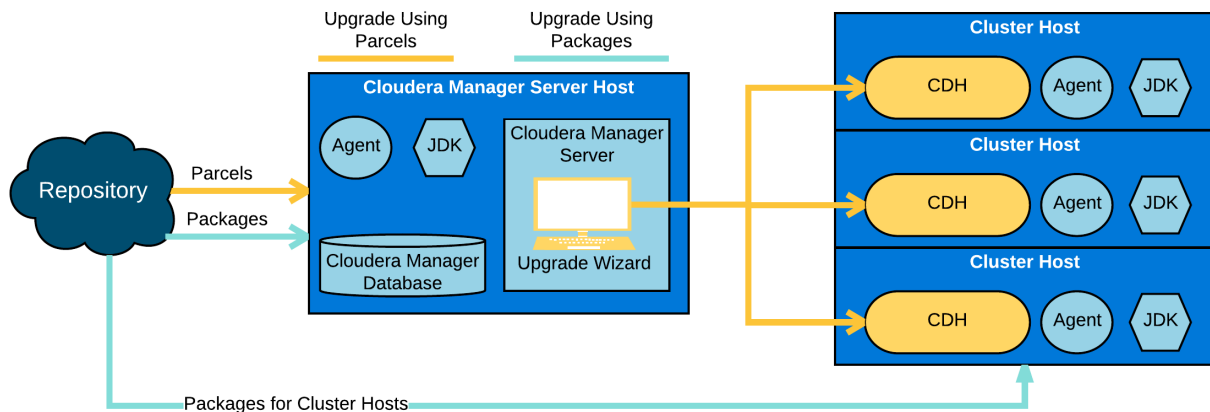
To upgrade Cloudera Manager, you perform the following tasks:

1. Upgrade the Cloudera Manager server software on the Cloudera Manager host using operating system package commands from the command line (for example, `yum` on RHEL systems). You can also manually upgrade Cloudera Manager using tarballs. Tarball upgrades are more suitable for environments in which existing tools are used to manage software distributed over many hosts. Cloudera Manager automates much of this process and is recommended for upgrading and managing your CDH clusters.
2. Upgrade the Cloudera Manager agent software on all cluster hosts. The Cloudera Manager upgrade wizard can upgrade the agent software (and, optionally, the JDK), or you can install the agent and JDK software manually from tarballs. The CDH software is not upgraded during this process.

To upgrade Cloudera Manager, see [Upgrading Cloudera Manager](#) on page 11.



## Overview of Upgrading CDH



**Figure 2: CDH Upgrades**

CDH upgrades contain updated versions of the Hadoop software and other components. You can use Cloudera Manager to upgrade CDH using either [parcels or packages](#).

- Upgrade Using Parcels

(Also applies to rolling upgrades.)

Upgrading CDH using parcels is the preferred method because parcels are managed by Cloudera Manager, which automatically downloads, distributes, and activates the correct versions of the software. There are two types of parcel upgrades:

- **Parcels** – Requires you to restart the cluster to complete the upgrade.
- **Rolling Upgrade**– If you have enabled high availability for HDFS, you can perform a rolling upgrade to upgrade CDH without cluster down time. For an easier upgrade experience, consider [switching from packages to parcels](#) so that Cloudera Manager can automate more of the process. You can also switch from packages to parcels when upgrading CDH 5.

- Upgrade Using Packages

Upgrading CDH using packages requires you to download updated packages and manually run package upgrade commands on the Cloudera Manager server and *all cluster hosts*.

See [Upgrading to CDH 5.x Using Packages](#) on page 49.

Cloudera Manager 5.3 introduced 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.

See [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 34

## Overview of Upgrading Cloudera Navigator Components

Cloudera Navigator Metadata and Audit servers are automatically upgraded when you upgrade Cloudera Manager. You can also optionally upgrade other Cloudera Navigator components such as Cloudera Navigator Key Trustee Server, Cloudera Navigator Key HSM, and Cloudera Navigator Encrypt. You do not have to upgrade these components along with Cloudera Manager or CDH upgrades.

See [Upgrading Cloudera Navigator Components](#) on page 68.

### Upgrading the JDK

Before upgrading Cloudera Manager or CDH, ensure that all cluster hosts are using a [supported version of the Oracle Java Development Kit \(JDK\)](#). All cluster hosts must use the same version of the JDK. See:

- [Upgrading to Oracle JDK 1.8](#) on page 65

# Upgrading Cloudera Manager

The procedures for upgrading Cloudera Manager differ depending on the version of Cloudera Manager from which you are upgrading and whether you installed Cloudera Manager using packages or tarballs. See the following links for detailed upgrade instructions:

- Upgrading Cloudera Manager 5
  - [Upgrading Cloudera Manager 5 Using Packages](#) on page 11  
Upgrade Cloudera Manager from any version of Cloudera Manager 5 to a higher version of Cloudera Manager 5 using operating system package-management commands.
  - [Upgrading Cloudera Manager 5 Using Tarballs](#) on page 24  
Upgrade Cloudera Manager from any version of Cloudera Manager 5 to a higher version of Cloudera Manager 5 manually using tarballs. Recommended for advanced users only.



**Note:** Upgrading Cloudera Manager does not upgrade CDH. See [Cloudera Upgrade Overview](#) on page 7 for more complete information about the upgrade process.

## Upgrading Cloudera Manager 5 Using Packages

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

This topic describes how to upgrade Cloudera Manager 5.x using *packages*. These steps apply to both [minor and maintenance](#) upgrades. When you upgrade Cloudera Manager using packages, you run operating system package commands from the command line, and then complete the upgrade using Cloudera Manager.

You can also manually upgrade Cloudera Manager using [tarballs](#).

In most cases, you can upgrade Cloudera Manager without shutting down most CDH services. Depending on which services are deployed in your cluster, you might need to stop some dependent services. CDH daemons can run unaffected while Cloudera Manager is upgraded.

The upgrade process does not affect your CDH installation. After upgrading Cloudera Manager, you might want to upgrade CDH 4 clusters to CDH 5, or upgrade to a more recent minor version of CDH.

The Cloudera Manager upgrade process does the following:

- Upgrades the database schema to reflect the current version.
- Upgrades the Cloudera Manager Server and all supporting services.
- Upgrades the Cloudera Manager agent.
- Redeploys client configurations to ensure that client services have the most current configuration.

**Steps to upgrade Cloudera Manager 5 using packages:**

Step	Description	Link
1	Collect the information you need to upgrade Cloudera Manager. This includes user accounts, passwords, database URLs, and other items. You must gather this information before beginning the upgrade because some information is available only from the Cloudera Manager Admin Console, which is not available during the upgrade process.	<a href="#">Step 1: Collect Upgrade Information</a> on page 12
2	Complete the pre-upgrade steps and review special warnings about upgrades.	<a href="#">Step 2: Complete Pre-Upgrade Steps</a> on page 13

Step	Description	Link
3	Back up the Cloudera Manager databases.	<a href="#">Step 3: Back Up Cloudera Manager Databases</a> on page 13
4	If your Cloudera Manager hosts use an unsupported version of the JDK, you must upgrade the hosts to a <a href="#">supported version of the JDK</a> before upgrading Cloudera Manager. If you plan to upgrade CDH, you must also upgrade the JDK on all cluster hosts.	Step 4: <a href="#">Java Development Kit Installation</a> <a href="#">Upgrading to Oracle JDK 1.8</a> on page 65
5	If the Cloudera Manager host does not have access to the internet, or you install a version lower than the latest version of Cloudera Manager, configure access to the Cloudera Manager software from either the Cloudera public repository or a local package repository that you create.	Step 5: <a href="#">Creating and Using a Package Repository for Cloudera Manager</a> on page 89
6	If you are upgrading from Cloudera Navigator 2.6 or lower, upgrade the Cloudera Navigator data management component.	<a href="#">Step 6: Prepare the Cloudera Navigator Data Management Component for Upgrade</a> on page 14
7	Upgrade the Cloudera Manager server and agent software.	<a href="#">Step 7: Upgrade the Cloudera Manager Server</a> on page 15
8	Verify and test your upgrade.	<a href="#">Step 8: Verify and Test the Upgrade</a> on page 23
9	Upgrade any required Cloudera Navigator components: <ul style="list-style-type: none"> <li>• Cloudera Manager Key Trustee Server</li> <li>• Cloudera Navigator Key HSM</li> <li>• Cloudera Navigator Key Trustee KMS</li> <li>• Cloudera Navigator Encrypt.</li> </ul> <p>The Cloudera Navigator Data Management Component is upgraded automatically when you upgrade Cloudera Manager.</p>	Step 9: <a href="#">Upgrading Cloudera Navigator Components</a> on page 68
10	(Optional) Upgrade CDH. You are not required to upgrade CDH after upgrading Cloudera Manager. You can upgrade CDH at a later time.	<a href="#">Step 10: (Optional) Upgrade CDH</a> on page 24

If you encounter problems during an upgrade, see the following:

- [Troubleshooting Installation and Upgrade Problems](#)
- [Re-Running the Cloudera Manager Upgrade Wizard](#) on page 95
- [Reverting a Failed Cloudera Manager Upgrade](#) on page 96

### Step 1: Collect Upgrade Information

Before starting an upgrade, collect the following information:

1. Host credentials. You must have SSH access and be able to log in using a root account or an account that has password-less sudo permission.
2. The version of Cloudera Manager used in your cluster. Go to **Support > About**.
3. The version of the JDK deployed in the cluster. Go to **Support > About**.
4. The version of CDH. The CDH version number displays next to the cluster name on the **Home** page.
5. Whether the cluster was installed using parcels or packages. This information displays next to the CDH version on the **Home** page of Cloudera Manager.
6. The services enabled in your cluster. Go to **Clusters > Cluster name**.
7. Operating system type and version. Go to **Hosts** and click on a hostname in the list. The operating system type and version displays in the **Distribution** row in the **Details** section.

## Step 2: Complete Pre-Upgrade Steps

Before beginning a Cloudera Manager upgrade, do the following:

1. Review the [CDH 5 and Cloudera Manager 5 Requirements and Supported Versions](#) for the new versions you are upgrading to.
2. Read the [Cloudera Manager 5 Release Notes](#).
3. Read the [Cloudera Security Bulletins](#).
4. Note the following:
  - **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 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. You will be instructed to stop some services in a later step.
5. If you have previously installed Kafka 1.2, and are upgrading from Cloudera Manager 5.4 or lower, remove the Kafka CSD:
  - a. 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.
  - b. Delete the Kafka CSD from the directory.
6. Review package (RPM) dependencies. A Cloudera Manager upgrade might introduce new package dependencies. If your organization has restrictions or requires prior approval for installation of packages, see the list of [Package Dependencies](#) on page 31 before upgrading Cloudera Manager.

## Step 3: Back Up Cloudera Manager Databases

1. Stop the Cloudera Management Service:
  - a. Select **Clusters > Cloudera Management Service**.
  - b. Select **Actions > Stop**.
2. Back up the following Cloudera Manager databases:
  - Cloudera Manager Server
  - Cloudera Navigator Audit Server
  - Cloudera Navigator Metadata Server
  - Activity Monitor
  - Reports Manager

To locate information about these databases (database type, hostname, and credentials):

- Cloudera Manager Server – Log in to the Cloudera Manager host and examine the `/etc/cloudera-scm-server/db.properties` file. For example:

```
more /etc/cloudera-scm-server/db.properties
# Auto-generated by scm_prepare_database.sh on Fri Dec 9 08:51:29 PST 2016
#
# For information describing how to configure the Cloudera Manager Server
# to connect to databases, see the "Cloudera Manager Installation Guide."
#
```

```
com.cloudera.cmf.db.type=mysql
com.cloudera.cmf.db.host=localhost
com.cloudera.cmf.db.name=cm
com.cloudera.cmf.db.user=cm
com.cloudera.cmf.db.password=cm
```

- For the other databases, go to **Clusters > Cloudera Management Service > Configuration** and select the **Database** category. You might need to contact your database administrator to obtain passwords.

See [Backing Up Databases](#) for detailed instructions for each supported type of database.

### 3. Start the Cloudera Management Service:

- a. Select **Clusters > Cloudera Management Service**.
- b. Select **Actions > Start**.

## Step 4: Upgrade the JDK

If your Cloudera Manager hosts use an unsupported version of the JDK, you must upgrade the hosts to a [supported version of the JDK](#) before upgrading Cloudera Manager. If you plan to upgrade CDH, you must also upgrade the JDK on all cluster hosts.

See:

- [Java Development Kit Installation](#)
- [Upgrading to Oracle JDK 1.8](#) on page 65

If you have enabled TLS/SSL, you must reinstall CA certificates to your truststores after upgrading the JDK. See [Recommended Keystore and Truststore Configuration](#).

## Step 5: Establish Access to the Software

If the Cloudera Manager host does not have access to the internet, or you install a version lower than the latest version of Cloudera Manager, configure access to the Cloudera Manager software from either the Cloudera public repository or a local package repository that you create.

See [Creating and Using a Package Repository for Cloudera Manager](#) on page 89.


## Step 6: Prepare the Cloudera Navigator Data Management Component for Upgrade

Cloudera Manager upgrades Cloudera Navigator as part of the Cloudera Manager upgrade process. If you are upgrading from Cloudera Navigator 2.6 or lower, follow the steps in this section to prepare the Cloudera Navigator data management component for upgrade and then continue with [Step 7: Upgrade the Cloudera Manager Server](#) on page 15. If you are upgrading from Cloudera Navigator 2.7 or higher, skip this section and continue with [Step 7: Upgrade the Cloudera Manager Server](#) on page 15.

To find the version of Cloudera Navigator:

1. Go to **Clusters > Cloudera Navigator**.

The Cloudera Navigator user interface displays.

2. Log in to Cloudera Navigator.
3. Click the  icon and select **About**.

A dialog box displays the version number and other information about Cloudera Navigator.

To upgrade the Cloudera Navigator data management component:

1. Stop the Navigator Metadata Server role:
  - a. Go to **Clusters > Cloudera Management Service > Instances**.
  - b. Select the **Navigator Metadata Server**.
  - c. Click **Actions for Selected > Stop**.

2. Back up the Navigator Metadata Server storage directory. To find the location of this directory:

- a. Go to **Clusters > Cloudera Management Service > Instances**.
- b. Click the **Configuration** tab.
- c. Select **Scope > Navigator Metadata Server**.

The **Navigator Metadata Server Storage Dir** property stores the location of the directory.

3. Start the Navigator Metadata Server role.

- a. Go to **Clusters > Cloudera Management Service > Instances**.
- b. Select the **Navigator Metadata Server**.
- c. Click **Actions for Selected > Start**.

4. Purge the Navigator Metadata Server of stale and deleted entities. See [Managing Metadata Storage with Purge](#)

5. Stop the Navigator Metadata Server role:

- a. Go to **Clusters > Cloudera Management Service > Instances**.
- b. Select the **Navigator Metadata Server**.
- c. Click **Actions for Selected > Stop**.

6. Make sure that the [Navigator Metadata Server has sufficient memory](#) to complete the upgrade.

7. 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.

## Step 7: Upgrade the Cloudera Manager Server

1. If your cluster is running the embedded PostgreSQL database, stop all services that are using the embedded database. These can include:

- Hive service and all services such as Impala and Hue that use the Hive metastore
- Oozie
- Sentry

2. If your cluster is running the Cloudera Navigator data management component and the following services are enabled for auditing, [stop](#) the following roles. (You also can elect to skip this step and leave these services running, but some audits by Cloudera Navigator may not occur during the Cloudera Manager upgrade process.)

- **HDFS** - NameNode
- **HBase** - Master and RegionServers
- **Hive** - HiveServer2
- **Hue** - Beeswax Server



**Note:** Stopping these roles renders any service depending on these roles unavailable. For the HDFS - NameNode role, most of the services in the cluster are unavailable until the upgrade is finished.

To determine which services are enabled for auditing:

- a. Go to the **Home** page in Cloudera Manager.
- b. Click **Configuration > Navigator Settings**.
- c. Type "Enable Audit" in the search box. The **Enable Audit Collection** property displays the services for which Cloudera Navigator auditing is enabled.

3. Stop the Cloudera Management Service:

- a. Select **Clusters > Cloudera Management Service**.
- b. Select **Actions > Stop**.

#### 4. Stop Cloudera Manager Server, Database, and Agent:

- a. Use the Cloudera Manager Admin Console to stop any running commands. These include user commands and commands Cloudera Manager automatically triggers in response to a state change or a schedule. You can either wait for commands to complete, or stop any running commands. For more information on viewing and stopping running commands, see [Viewing Running and Recent Commands](#).



**Important:** If you do not stop all commands, the Cloudera Manager Server *fails to start after upgrade*.

- b. On the host running the Cloudera Manager Server, stop the Cloudera Manager Server:

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

- c. If you are using the embedded PostgreSQL database with Cloudera Manager, stop the database on the host where the database runs, usually the Cloudera Manager Server host:

```
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
```

- d. If the Cloudera Manager host is also running the Cloudera Manager Agent, stop the Cloudera Manager Agent:

```
sudo service cloudera-scm-agent stop
```

#### 5. Back up the following directories on the Cloudera Manager server host:

- /etc/cloudera-scm-server
- /etc/cloudera-scm-agent

6. Establish access to the Cloudera Manager Server packages. You can either upgrade from the Cloudera repository at <https://archive.cloudera.com/cm5/>, or you can create your own package repository, as described in [Creating and Using a Package Repository for Cloudera Manager](#) on page 89. You must create your own repository if Cloudera Manager does not have Internet access or you want to upgrade to a version of Cloudera Manager lower than the latest version.

To upgrade using the Cloudera repository:

- a. Back up the current Cloudera Manager repo file, located in one of the following directories:



**RHEL**

/etc/yum.repos.d/

**SLES**

/etc/zypp/repos.d/

**Ubuntu or Debian**

/etc/apt/sources.list.d/

- b.** Download 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 Red Hat or CentOS 6, go to the appropriate release directory, for example, [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
```

- 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
```

For example, you can use the following command to download the .repo file for Cloudera Manager version 5 and RHEL version 6:

```
$ wget https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo
```

- c.** Do one of the following on the Cloudera Manager Server host:

- Upgrade to the most recent version of Cloudera Manager:**

Copy the cloudera-manager.repo file to the configuration location for the package management software for your system:

**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/

- Upgrade to an specific version of Cloudera Manager:**

**RHEL-compatible or SLES**

- Edit the cloudera-manager.repo file to change the 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/.
```

2. Save the edited file:

- For RHEL or CentOS, save it in `/etc/yum.repos.d/`.
- For SLES, save it in `/etc/zypp/repos.d/`.

### Ubuntu or Debian

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 `cloudera.list` 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 `/etc/apt/sources.list.d/` directory.

- d. Run the following command to clean the cache directories and upgrade the software:

#### RHEL

```
sudo yum clean all  
sudo yum upgrade cloudera-manager-server cloudera-manager-daemons 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 run. `yum` indicates which components must be upgraded.
- If the Cloudera Manager instance you are upgrading uses the embedded PostgreSQL database, add `cloudera-manager-server-db-2` to the list of packages in the `yum upgrade` command. *The embedded PostgreSQL database should not be used in production environments.*

#### SLES

```
sudo zypper clean --all  
sudo zypper up -r https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/
```

To upgrade from your own repository:

```
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
```

**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-agent
```

If the Cloudera Manager instance you are upgrading uses the embedded PostgreSQL database, add `cloudera-manager-server-db-2` to the list of packages in the `apt-get install` command. *The embedded PostgreSQL database should not be used in production environments.*

During this process, you might 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.
```

You will receive a similar prompt for `/etc/cloudera-scm-server/db.properties`. Answer **N** to both prompts.

- If you customized the `/etc/cloudera-scm-agent/config.ini` file, your customized file is renamed with the extension `.rpmsave` or `.dpkg-old`. Merge any customizations into the `/etc/cloudera-scm-agent/config.ini` file that is installed by the package manager.
- On the Cloudera Manager Server host, verify that you now have the following packages, corresponding to the version of Cloudera Manager you installed, by running the following command:

**RPM-based distributions**

```
$ rpm -qa 'cloudera-manager-*'
cloudera-manager-server-5.13.0-0.cm5130.p0.38.el6.x86_64
cloudera-manager-agent-5.13.0-0.cm5130.p0.38.el6.x86_64
cloudera-manager-daemons-5.13.0-0.cm5130.p0.38.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 5.13.0-0.cm5130.p0.38~sq The Cloudera Manager Agent
ii cloudera-manager-daemo 5.13.0-0.cm5130.p0.38~sq Provides daemons for monitoring
Hadoop and related tools.
ii cloudera-manager-serve 5.13.0-0.cm5130.p0.38~sq The Cloudera Manager Server
```



**Note:** You might also see an entry for the `cloudera-manager-server-db-2` if you are using the embedded PostgreSQL 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.

- Start Cloudera Manager Server. On the Cloudera Manager Server host (the host on which you installed the `cloudera-manager-server` package), do the following:

- a. If you are using the embedded PostgreSQL database for Cloudera Manager, start the database. If your installation uses other databases, Cloudera Manager reconnects with them after start up.

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

- b. Start the Cloudera Manager Server:

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

You should see the following:

```
Starting cloudera-scm-server: [ OK ]
```

- 10 Log in to the Cloudera Manager Admin Console. It can take several minutes for Cloudera Manager Server to start, and the console is unavailable until the server startup is complete.

The **Upgrade Wizard** displays.

- 11 Upgrade the Cloudera Manager Agent [using Cloudera Manager](#) or by [manually upgrading the packages](#):

### Cloudera Manager upgrades Agent software

When Cloudera Manager upgrades the Cloudera Manager agent, Cloudera Manager handles the upgrade and cleanup, and optionally upgrades the JDK.

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 location must contain the matched Agent version. See [Creating and Using a Package Repository for Cloudera Manager](#) on page 89. Custom repositories are required when your cluster cannot access the Internet, or when you need to upgrade to a version other than the most current version of Cloudera Manager.
3. Click **Continue**. The JDK Installation Options page displays.
  - If you want Cloudera Manager to install JDK 1.7 on all cluster hosts, select **Install Oracle Java SE Development Kit (JDK)**.
  - If 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.
4. Click **Continue**.
5. 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.

6. Click **Continue**.

The Cloudera Manager Agent packages and, if selected, the JDK are installed.

7. Click **Continue**.

The Host Inspector runs to inspect your managed hosts for correct versions and configurations. If problems occur, you can make changes and then rerun the inspector.

When you are satisfied with the inspection results, click **Continue**.

## Manually upgrade Agent software

To manually upgrade the Cloudera Manager Agent software, you use package commands to clean up old versions, download the new version, and upgrade the software.

### To manually upgrade the Cloudera Manager agent:

1. On all cluster hosts except the Cloudera Manager Server host, stop the Agent:

```
sudo service cloudera-scm-agent stop
```

2. Select **No, I would like to skip the agent upgrade now** and click **Continue**.
3. Copy the repo file as described in step 6 on page 16.
4. Run the following commands on all hosts except the Cloudera Manager Server host:

#### RHEL

```
sudo yum clean all
sudo yum upgrade cloudera-manager-daemons 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 run. `yum` indicates which components must be upgraded.
- If the Cloudera Manager instance you are upgrading uses the embedded PostgreSQL database, add `cloudera-manager-server-db-2` to the list of packages in the `yum upgrade` command. *Do not use the embedded PostgreSQL database in production environments.*

#### SLES

```
sudo zypper clean --all
sudo zypper up -r https://archive.cloudera.com/cm5/sles/11/x86_64/cm/5/
```

To upgrade from your own repository:

```
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
```

#### Ubuntu or Debian

Use the following commands to 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-agent cloudera-manager-daemons
```

If the Cloudera Manager instance you are upgrading uses the embedded PostgreSQL database, add `cloudera-manager-server-db-2` to the list of packages in the `apt-get install` command. *Do not use the embedded PostgreSQL database in production environments.*

During this process, you might 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.
```

You will receive a similar prompt for `/etc/cloudera-scm-server/db.properties`. Answer **N** to both prompts.

5. If you customized the `/etc/cloudera-scm-agent/config.ini` file, your customized file is renamed with the extension `.rpmsave` or `.dpkg-old`. Merge any customizations into the `/etc/cloudera-scm-agent/config.ini` file that is 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 inspects your managed hosts for correct versions and configurations. If problems occur, you can make changes and then rerun the inspector. When you are satisfied with the inspection results, click **Continue**.

- 12 Click **Finish**.

- 13 If you are upgrading from Cloudera Manager 5.0 and are using an external database for Cloudera Navigator, the Database Setup page displays. Configure these database settings:

- a. Enter the database host, database type, database name, username, and password for the database.
- b. Click **Test Connection** to confirm that Cloudera Manager can communicate with the Cloudera Navigator database using the information you 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 in a later step.)

- 14 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.

- 15 Click **Continue**.

If you keep the default selection, the Upgrade wizard restarts the Cloudera Management Service.

- 16 Click **Finish**.

The **Home** page displays.

All services (except for any services you stopped) should now be running.

- 17 If, as part of this upgrade, you stopped some selected services and roles, [restart](#) the following roles:

- **HDFS** - NameNode
- **HBase** - Master and RegionServers
- **Hive** - HiveServer2
- **Hue** - Beeswax Server

- 18 If you upgraded the JDK, do the following:

- a. 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](#).

- b. 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**.

- 19 If Cloudera Manager reports [stale configurations](#) after the upgrade, restart the cluster services and redeploy the client configurations. If you are also upgrading CDH, this step is not required. Stale configurations can occur after a Cloudera Manager upgrade when a configuration value has changed, or new configuration options are added. Configuration changes that result in Cloudera Manager reporting stale configurations are described the [Cloudera Manager release notes](#).

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



next to the cluster name and select **Restart**.

- b. In the confirmation dialog box, click **Restart**.

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



next to the cluster name and select **Deploy Client Configuration**.

- d. In the confirmation dialog box, click **Deploy Client Configuration**.

- 20 If upgrading from Navigator 2.6 or lower:

- a. [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 *three to four hours* or longer.

- 21 When the upgrade is complete, click **Continue**. The Cloudera Navigator landing page is displayed.

- 22 If you are upgrading from CDH 5.5.0 or lower to CDH 5.5.0 or higher, hard restart the agent on all hosts to update and restart the `supervisord` process:

**RHEL 7 and higher:**

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

**Other Linux Distributions:**

```
sudo service cloudera-scm-agent hard_restart
```

## Step 8: Verify and Test the Upgrade

If the commands to update and start the Cloudera Manager Server complete without errors, the upgrade has completed successfully. To verify, check that the server versions have been updated.

1. Verify that the agents are sending heartbeats to Cloudera Manager:
  - a. Go to **Hosts > All Hosts**.
  - b. Click the column header labeled **Last Heartbeat** to sort it.
  - c. Verify that the last heartbeat for each host has occurred within one minute.
2. In the Cloudera Manager Admin Console, click the **Hosts** tab.
3. Click **Inspect All Hosts**. On large clusters, the host inspector can take some time to finish running. You must wait for the process to complete before proceeding to the next step.
4. 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.
5. Verify that the monitoring features are working as expected; follow the instructions in [Testing the Installation](#).

### Step 9: Upgrade any Required Navigator Components

Upgrade any Cloudera Navigator components deployed in your cluster:

- Cloudera Manager Key Trustee Server
- Cloudera Navigator Key HSM
- Cloudera Navigator Key Trustee KMS
- Cloudera Navigator Encrypt.

You can upgrade other Cloudera Navigator components at any time. You do not have to perform these upgrades when upgrading Cloudera Manager or CDH.

See [Upgrading Cloudera Navigator Components](#) on page 68.

### Step 10: (Optional) Upgrade CDH

Cloudera Manager 5 can manage both CDH 4 and CDH 5, so upgrading existing CDH 4 and CDH 5 installations is not required. To upgrade CDH, see [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 34.

## Upgrading Cloudera Manager 5 Using Tarballs

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

This topic describes how to upgrade Cloudera Manager 5.x using *tarballs*. Tarballs contain both the Cloudera Manager Server and Cloudera Manager Agent in a single file.

In most cases it is possible to upgrade Cloudera Manager 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, or upgrade to a more recent minor version of CDH.

**Steps to upgrade Cloudera Manager 5 using tarballs:**

Step	Description	Link
1	Collect the information you need to upgrade Cloudera Manager. This includes user accounts, passwords, database URLs, and other items. You must gather this information before beginning the upgrade because some information is available only from the Cloudera Manager Admin Console, which is not available during the upgrade process.	<a href="#">Step 1: Collect Upgrade Information</a> on page 25
2	Complete the pre-upgrade steps and review special warnings about upgrades.	<a href="#">Step 2: Complete Pre-Upgrade Steps</a> on page 25
3	If your Cloudera Manager hosts use an unsupported version of the JDK, you must upgrade the hosts to a <a href="#">supported version of the JDK</a> before upgrading Cloudera Manager. If you plan to upgrade CDH, you must also upgrade the JDK on all cluster hosts.	Step 3: Upgrade the JDK <a href="#">Java Development Kit Installation</a> <a href="#">Upgrading to Oracle JDK 1.8</a> on page 65
4	If the Cloudera Manager host does not have access to the internet, or you install a version lower than the latest version of Cloudera Manager, configure access to the Cloudera Manager software from either the Cloudera public repository or a local package repository that you create.	Step 4: <a href="#">Creating and Using a Package Repository for Cloudera Manager</a> on page 89
5	If you are upgrading from Cloudera Navigator 2.6 or lower, upgrade the Cloudera Navigator data management component.	Step 5: <a href="#">Upgrading the Cloudera Navigator Data Management Component</a> on page 68



Step	Description	Link
6	Upgrade the Cloudera Manager server and agent software.	<a href="#">Step 6: Upgrading the Cloudera Manager Server and Agents</a> on page 26
7	Verify and test your upgrade.	<a href="#">Step 7: Verify and Test the Upgrade</a> on page 30
8	Upgrade any required Cloudera Navigator components: <ul style="list-style-type: none"> <li>• Cloudera Manager Key Trustee Server</li> <li>• Cloudera Navigator Key HSM</li> <li>• Cloudera Navigator Key Trustee KMS</li> <li>• Cloudera Navigator Encrypt.</li> </ul> The Cloudera Navigator Data Management Component is upgraded automatically when you upgrade Cloudera Manager.	Step 8: <a href="#">Upgrading Cloudera Navigator Components</a> on page 68
9	(Optional) Upgrade CDH. You are not required to upgrade CDH after upgrading Cloudera Manager. You can upgrade CDH at a later time.	<a href="#">Step 9: (Optional) Upgrade CDH</a> on page 31

## Step 1: Collect Upgrade Information

Before starting an upgrade, collect the following information:

1. Host credentials. You must have SSH access and be able to log in using a root account or an account that has password-less sudo permission.
2. The version of Cloudera Manager used in your cluster. Go to **Support > About**.
3. The version of the JDK deployed in the cluster. Go to **Support > About**.
4. The version of CDH. The CDH version number displays next to the cluster name on the **Home** page.
5. Whether the cluster was installed using parcels or packages. This information displays next to the CDH version on the **Home** page of Cloudera Manager.
6. The services enabled in your cluster. Go to **Clusters > Cluster name**.
7. Operating system type and version. Go to **Hosts** and click on a hostname in the list. The operating system type and version displays in the **Distribution** row in the **Details** section.

## Step 2: Complete Pre-Upgrade Steps

Before beginning a Cloudera Manager upgrade, do the following:

1. Review the [CDH 5 and Cloudera Manager 5 Requirements and Supported Versions](#) for the new versions you are upgrading to.
2. Read the [Cloudera Manager 5 Release Notes](#).
3. Read the [Cloudera Security Bulletins](#).
4. Note the following:
  - **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 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. You will be instructed to stop some services in a later step.

5. If you have previously installed Kafka 1.2, and are upgrading from Cloudera Manager 5.4 or lower, remove the Kafka CSD:
  - a. 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.
  - b. Delete the Kafka CSD from the directory.
6. Review package (RPM) dependencies. A Cloudera Manager upgrade might introduce new package dependencies. If your organization has restrictions or requires prior approval for installation of packages, see the list of [Package Dependencies](#) on page 31 before upgrading Cloudera Manager.

### Step 3: Upgrade the JDK

See:

- [Java Development Kit Installation](#)
- [Upgrading to Oracle JDK 1.8](#) on page 65

### Step 4: Creating and Using a Package Repository for Cloudera Manager

If the Cloudera Manager host does not have access to the internet, or you install a version lower than the latest version of Cloudera Manager, configure access to the Cloudera Manager software from either the Cloudera public repository or a local package repository that you create.

See: [Creating and Using a Package Repository for Cloudera Manager](#) on page 89

### Step 5: Upgrading the Cloudera Navigator Data Management Component

If you are upgrading from Cloudera Navigator 2.6 or lower, upgrade the Cloudera Navigator data management component.

See [Upgrading the Cloudera Navigator Data Management Component](#) on page 68

### Step 6: Upgrading the Cloudera Manager Server and Agents

1. If your cluster is running the embedded PostgreSQL database, stop all services that are using the embedded database. These can include:
  - Hive service and all services such as Impala and Hue that use the Hive metastore
  - Oozie
  - Sentry
2. Stop Cloudera Manager Server, Database, and Agent:
  - a. Use the Cloudera Manager Admin Console to stop any running commands. These include user commands and commands Cloudera Manager automatically triggers in response to a state change or a schedule. You can either wait for commands to complete, or stop any running commands. For more information on viewing and stopping running commands, see [Viewing Running and Recent Commands](#).



**Important:** If you do not stop all commands, the Cloudera Manager Server *fails to start after upgrade*.

- b. On the host running the Cloudera Manager Server, stop the Cloudera Manager Server:

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

- c. If you are using the embedded PostgreSQL database with Cloudera Manager, stop the database on the host where the database runs, usually the Cloudera Manager Server host:

```
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
```

- d. If the Cloudera Manager host is also running the Cloudera Manager Agent, stop the Cloudera Manager Agent:

```
sudo service cloudera-scm-agent stop
```

3. Download tarballs from the locations listed in [Cloudera Manager Version and Download Information](#).
4. 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
```

5. 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.

6. 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.

7. 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 configure 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.

### 8. Start Cloudera Manager Server.

The way in which you start the Cloudera Manager Server varies depending on which 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`.

9. To stop the Cloudera Manager Agent, run this command on each Agent host:

```
$ sudo tarball_root/etc/init.d/cloudera-scm-agent hard_stop_confirmed
```

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 hard_stop_confirmed
```

- 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 hard_stop_confirmed
```

10 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`.

#### 11 Log in to the Cloudera Manager Admin Console.

#### 12 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**.

#### 13 Upgrade any required Cloudera Navigator components:

- [Upgrading the Cloudera Navigator Data Management Component](#) on page 68
- [Upgrading Cloudera Navigator Key Trustee Server](#) on page 69
- [Upgrading Cloudera Navigator Key HSM](#) on page 84
- [Upgrading Key Trustee KMS](#) on page 85
- [Upgrading Cloudera Navigator Encrypt](#) on page 86

## Step 7: Verify and Test the Upgrade

If the commands to update and start the Cloudera Manager Server complete without errors, the upgrade has completed successfully. To verify, check that the server versions have been updated.

#### 1. Verify that the agents are sending heartbeats to Cloudera Manager:

- ##### a. Go to **Hosts** > **All Hosts**.
- ##### b. Click the column header labeled **Last Heartbeat** to sort it.
- ##### c. Verify that the last heartbeat for each host has occurred within one minute.

#### 2. In the Cloudera Manager Admin Console, click the **Hosts** tab.

#### 3. Click **Inspect All Hosts**. On large clusters, the host inspector can take some time to finish running. You must wait for the process to complete before proceeding to the next step.

#### 4. 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.

#### 5. Verify that the monitoring features are working as expected; follow the instructions in [Testing the Installation](#).

## Step 8: Upgrading Cloudera Navigator Components

Upgrade any required Cloudera Navigator components:

- Cloudera Manager Key Trustee Server
- Cloudera Navigator Key HSM
- Cloudera Navigator Key Trustee KMS
- Cloudera Navigator Encrypt.

See [Upgrading Cloudera Navigator Components](#) on page 68.

The Cloudera Navigator Data Management Component is upgraded automatically when you upgrade Cloudera Manager.

## Step 9: (Optional) Upgrade CDH

Cloudera Manager 5 can manage both CDH 4 and CDH 5, so upgrading existing CDH 4 and CDH 5 installations is not required. To upgrade CDH, see [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 34.

## Package Dependencies

When upgrading Cloudera Manager note that the following packages may be installed or upgraded as part of the steps taken by the upgrade wizard.

### RHEL/CentOS

- bind-utils
- chkconfig
- cyrus-sasl-gssapi
- cyrus-sasl-plain
- fuse
- fuse-libs
- gcc
- httpd
- init-functions
- libxslt
- mod\_ssl
- MySQL-python
- openssl
- openssl-devel
- openssl-devel
- perl
- portmap
- postgresql-server >= 8.4
- psmisc
- python >= 2.4.3-43
- python-devel >= 2.4.3-43
- python-psycopg2
- python-setuptools
- sed
- service
- sqlite
- swig
- useradd
- zlib

### SLES

- apache2
- bind-utils
- chkconfig
- cyrus-sasl-gssapi
- cyrus-sasl-plain
- fuse
- gcc
- libfuse2
- libxslt
- openssl
- openssl-devel
- perl
- portmap
- postgresql-server >= 8.4
- psmisc
- python >= 2.4.3-43
- python-devel >= 2.4.3-43
- python-mysql
- python-setuptools
- python-xml
- sed
- service
- sqlite
- swig
- useradd
- zlib

### Debian/Ubuntu

- ant
- apache2
- bash
- chkconfig
- debhelper (>= 7)
- fuse-utils | fuse
- gcc
- libfuse2
- libsasl2-modules
- libsasl2-modules-gssapi-mit
- libsqlite3-0
- libssl-dev
- libxslt1.1
- lsb-base
- make
- openssl
- perl
- postgresql-client@@PG\_PKG\_VERSION@@
- postgresql@@PG\_PKG\_VERSION@@
- psmisc
- python-dev (>=2.4)



- python-mysqldb
- python-psycopg2
- python-setuptools
- rpcbind
- sed
- swig
- useradd
- zlib1g

## Upgrading CDH and Managed Services Using Cloudera Manager

You can use Cloudera Manager to upgrade CDH for [major, minor, and maintenance upgrades](#). The procedures vary depending on the version of Cloudera Manager you are using and from which versions of CDH you are upgrading. Procedures to upgrade Cloudera Manager installations are different when using parcels compared to packages.

After completing preparatory steps, you use the Cloudera Manager upgrade wizard to complete the upgrade. If you use parcels (recommended), have enabled [HDFS High Availability](#), and have a Cloudera Enterprise license, you can perform a *rolling upgrade* that does not require you to take the cluster offline during the upgrade.

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. To upgrade Cloudera Manager, see [Overview of Upgrading Cloudera Manager](#) on page 8.

Choose one of the following procedures to upgrade CDH using Cloudera Manager:

### Upgrading CDH 5.x

The procedures for upgrading to CDH 5.x differ for minor release upgrades and maintenance release upgrades.

#### Minor Version Upgrades

To upgrade CDH 5.x to a higher version of CDH, for example from CDH 5.4 to CDH 5.8, select one of the following options:

- [Upgrading to CDH 5.x Using a Rolling Upgrade](#) on page 35

This option uses Cloudera Manager [parcels](#) and allows you to upgrade your cluster without having to stop the cluster. You must have HDFS high availability enabled and have a Cloudera Enterprise license.

- [Upgrading to CDH 5.x Using Parcels](#) on page 42

This option upgrades your cluster using parcels, but requires you to restart the cluster before completing the upgrade.

- [Upgrading to CDH 5.x Using Packages](#) on page 49

This option is the most time consuming and requires you to log in using `ssh` and execute a series of package commands on *all hosts* in your cluster. Cloudera recommends that you instead upgrade your cluster using [parcels](#), which allows Cloudera Manager to distribute the upgraded software to all hosts in the cluster without having to log in to each host. If you installed the cluster using packages, you can upgrade using parcels and the cluster will use parcels for subsequent upgrades.

#### Maintenance Version Upgrades

Maintenance upgrades fix critical bugs or address security issues. No new functionality or incompatible changes are introduced. The version numbers for maintenance releases differ only in the third digit, for example, when upgrading from CDH 5.8.0 to CDH 5.8.2.

To upgrade to a maintenance release, you only need to perform a subset of the Minor version upgrade steps. Follow the same procedures as for minor version upgrades but skip the steps that are labeled as follows:

**[Not required for CDH maintenance release upgrades.]**

## Upgrading to CDH 5.x Using a Rolling Upgrade

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

This topic describes how to perform a rolling upgrade from any version of CDH 5.x to a higher version of CDH 5.x, using Cloudera Manager and parcels. The minor version of Cloudera Manager you use to perform the upgrade must be equal to or greater than the CDH minor version. To upgrade Cloudera Manager, see [Overview of Upgrading Cloudera Manager](#) on page 8.

A rolling upgrade allows you to upgrade your cluster software and restart the upgraded services without taking the cluster offline. Performing a rolling upgrade requires the following:

- CDH was installed using Cloudera Manager and parcels. You can [migrate your cluster from using packages to using parcels](#).
- The cluster uses a Cloudera Enterprise license.
- High availability is enabled for HDFS.



**Note:** If you are upgrading to a *maintenance* version of CDH, skip any steps that are labeled **[Not required for CDH maintenance release upgrades.]**.

The version numbers for maintenance releases differ only in the third digit, for example when upgrading from CDH 5.8.0 to CDH 5.8.2. See [Maintenance Version Upgrades](#) on page 34.

To upgrade CDH using a rolling upgrade:

### Step 1: Collect Upgrade Information

Before starting an upgrade, collect the following information:

1. Host credentials. You must have SSH access and be able to log in using a root account or an account that has password-less sudo permission.
2. The version of Cloudera Manager used in your cluster. Go to **Support > About**.
3. The version of the JDK deployed in the cluster. Go to **Support > About**.
4. The version of CDH. The CDH version number displays next to the cluster name on the **Home** page.
5. Whether the cluster was installed using parcels or packages. This information displays next to the CDH version on the **Home** page of Cloudera Manager.
6. The services enabled in your cluster. Go to **Clusters > Cluster name**.
7. Operating system type and version. Go to **Hosts** and click on a hostname in the list. The operating system type and version displays in the **Distribution** row in the **Details** section.
8. Database information for the databases used by Sqoop, Oozie, Hue, Hive Metastore, and Sentry Server (information is only required if these services are enabled in the cluster).

Gather the following information:

- Type of database (PostgreSQL, Embedded PostgreSQL, MySQL, MariaDB, or Oracle)
- Hostnames of the databases
- Credentials for the databases

To locate database information:

- **Sqoop, Oozie, and Hue** – Go to **Cluster Name > Configuration > Database Settings**.
- **Hive Metastore** – Go to the Hive service, select **Configuration**, and select the **Hive Metastore Database** category.
- **Sentry** – Go to the Sentry service, select **Configuration**, and select the **Sentry Server Database** category.

### Step 2: Complete Pre-Upgrade Steps

1. Review the [CDH 5 and Cloudera Manager 5 Requirements and Supported Versions](#) for the new versions you are upgrading to.
2. Read the [CDH 5 Release Notes](#).
3. Read the [Cloudera Security Bulletins](#).
4. Ensure that Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Java Development Kit Installation](#) or [Upgrading to Oracle JDK 1.8](#) on page 65, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
5. Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:
  - **Targeted CDH Version:** 5.0.5  
**Minimum Cloudera Manager Version:** 5.0.x
  - **Targeted CDH Version:** 5.8.2  
**Minimum Cloudera Manager Version:** 5.8.x
6. If you are upgrading from CDH 5.1 or lower, and use Hive Date partition columns, you may need to update the date format. See [Date partition columns](#).
7. If the cluster uses Impala, 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](#).
8. Run the [Host Inspector](#) and fix every issue. Go to **Cluster > Inspect Hosts**.
9. Run the [Security Inspector](#) and fix and reported errors. (Go to **Administration > Security** and click **Security Inspector**.)
- 10 Log in to any cluster node as the `hdfs` user, run the following commands, and correct any reported errors:

```
hdfs fsck /
```



**Note:** the `fsck` command may take 10 minutes or more to complete, depending on the number of files in your cluster.

```
hdfs dfsadmin -report
```

See [HDFS Commands Guide](#) in the Apache Hadoop documentation.

- 11 Log in to any DataNode as the `hbase` user, run the following command, and correct any reported errors:

```
hbase hbck
```

See [Checking and Repairing HBase Tables](#).

- 12 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.
- 13 To avoid unnecessary alerts during the upgrade process, enable [maintenance mode](#) on your cluster before you start the upgrade. This stops email alerts and SNMP traps from being sent, but does not stop checks and configuration validations. Be sure to exit maintenance mode when you have finished the upgrade to reen able Cloudera Manager alerts.
- 14 If you have configured Hue to use TLS/SSL and you are upgrading from CDH 5.2 or lower to CDH 5.3 or higher, Hue validates CA certificates and requires a truststore. To create a truststore, follow the instructions in [Hue as a TLS/SSL Client](#).
- 15 If your cluster uses the Flume Kafka client, and you are upgrading to CDH 5.8.0 or CDH 5.8.1, perform the extra steps described in [Upgrading to CDH 5.8.0 or CDH 5.8.1 When Using the Flume Kafka Client](#) on page 63 and then continue with the procedures in this topic.

- 16 If your cluster uses Impala and Llama, this role has been deprecated as of CDH 5.9 and you must remove the role from the Impala service before starting the upgrade. If you do not remove this role, the upgrade wizard will halt the upgrade.

To determine if Impala uses Llama:

1. Go to the Impala service.
2. Select the **Instances** tab.
3. Examine the list of roles in the **Role Type** column. If Llama appears, the Impala service is using Llama.

To remove the Llama role:

1. Go to the Impala service and select **Actions > Disable YARN and Impala Integrated Resource Management**.

The **Disable YARN and Impala Integrated Resource Management** wizard displays.

2. Click **Continue**.

The **Disable YARN and Impala Integrated Resource Management Command** page displays the progress of the commands to disable the role.

3. When the commands have completed, click **Finish**.

- 17 If you have deployed the Sentry service in your cluster, and are upgrading from CDH 5.12 or lower, you may need to increase the Java heap memory for Sentry. See [Performance Guidelines](#).

### Step 3: Ensure High Availability Is Enabled

See [HDFS High Availability](#) for instructions. Enabling automatic failover is optional. 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 configuring JobTracker high availability is not a requirement for performing a rolling upgrade.

### Step 4: Back Up HDFS Metadata

**[Not required for CDH maintenance release upgrades.]**

The steps in this section are only required for the following upgrades:

- 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
```

### Step 5: Back Up Databases



**Note:** Backing up databases requires that you stop some services, which may make them unavailable during backup.



Back up the databases for any of the following services that are deployed in your cluster:

**Table 1: Service Databases to Back Up**

Service	Where to find database information
Sqoop	Go to <b>Clusters &gt; Cluster Name &gt; Sqoop service &gt; Configuration</b> and select the <b>Database</b> category.
Hue	Go to <b>Clusters &gt; Cluster Name &gt; Hue service &gt; Configuration</b> and select the <b>Database</b> category.

Service	Where to find database information
Oozie	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Oozie service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Cloudera Navigator Audit Server	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Cloudera Navigator Metadata Server	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Activity Monitor	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Reports Manager	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Sentry Server	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Sentry service</b> > <b>Configuration</b> and select the <b>Sentry Server Database</b> category.
Hive Metastore	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Hive service</b> > <b>Configuration</b> and select the <b>Hive Metastore Database</b> category.

## To back up the databases:

- If not already stopped, stop the service:
  - On the **Home** > **Status** tab, click  to the right of the service name and select **Stop**.
  - Click **Stop** in the next screen to confirm. When you see a **Finished** status, the service has stopped.
- Back up the database. See [Backing Up Databases](#) for detailed instructions for each supported type of database.
- Restart the service:
  - On the **Home** > **Status** tab, click  to the right of the service name and select **Start**.
  - Click **Start** that appears in the next screen to confirm. When you see a **Finished** status, the service has started.

## Step 6: Run the Upgrade Wizard



**Note:** If Cloudera Manager detects a failure while upgrading CDH, Cloudera Manager displays a dialog box where you can create a diagnostic bundle to send to Cloudera Support so they can help you recover from the failure. The cluster name and time duration fields are pre-populated to capture the correct data.

- If your cluster has Kudu 1.4.0 (or lower) installed, deactivate the existing Kudu parcel. Starting with Kudu 1.5.0 / CDH 5.13, Kudu is part of the CDH parcel and does not need to be installed separately.
- If your cluster has Spark 2.1 or Spark 2.2 installed and you want to upgrade to CDH 5.13 or higher, your cluster must be running either Spark 2.1, release 2 or Spark 2.2 release 1 before upgrading. To install these versions of Spark do the following before running the Upgrade Wizard:
  - Install the Custom Service Descriptor (CSD) file. See
    - [Installing Spark 2.1](#)
    - [Installing Spark 2.2](#)




**Note:** Spark 2.2 requires that JDK 1.8 be deployed throughout the cluster. JDK 1.7 is not supported for Spark 2.2.

See [Java Development Kit Installation](#).

b. Download, distribute, and activate the Parcel for the version of Spark that you are installing:

- **Spark 2.1 release 2:** The parcel name includes "cloudera2" in its name.
- **Spark 2.2 release 1:** The parcel name includes "cloudera1" in its name.

See [Managing Parcels](#).

3. From the **Home > Status** tab, click  next to the cluster name and select **Upgrade Cluster**.

The **Getting Started** page of the upgrade wizard displays.

4. If the option to pick between packages and parcels displays, select **Use Parcels**.
5. In the **Choose CDH Version (Parcels)** field, select the CDH version. If no qualifying parcels are listed, or you want to upgrade to a different version, click the **Modify the Remote Parcel Repository URLs** link to go to the configuration page for **Remote Parcel Repository URLs** and add the appropriate URL to the configuration. See [Parcel Configuration Settings](#) for information about entering the correct URL for parcel repositories. Click **Continue**.
6. If you previously installed the GPLEXTRAS parcel, download and distribute the version of the GPLEXTRAS parcel that matches the version of CDH that you are upgrading to.
7. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**. If you downloaded a new version of the GPLEXTRAS parcel, the Upgrade Wizard displays a message that the GPLEXTRAS parcel conflicts with the version of the CDH parcel, similar to the following:

**GPLEXTRAS 5.10.0-1.cdh5.10.0.p0.41 conflicts with CDH 5.11.1-1.**



**Let Cloudera Manager resolve this automatically: Activate GPLEXTRAS**

Select the option to resolve the conflicts automatically and click **Continue**.

Cloudera Manager deactivates the old version of the GPLEXTRAS parcel, activates the new version and verifies that all hosts have the correct software installed.

8. Click **Continue**.

The selected parcels are downloaded and distributed.

9. Click **Continue**.

The Host Inspector runs and displays the CDH version on the hosts.

10. Click **Continue**.

The **Choose Upgrade Procedure** screen displays.

11. Select **Rolling Restart**. Cloudera Manager upgrades services and performs a rolling restart. This option is only available if you have [enabled high availability for HDFS](#). Services that do not support rolling restart undergo a normal restart, and are not available during the restart process.

12. (Optional) Configure the following parameters for the rolling restart:

### Batch Size

Number of roles to include in a batch. Cloudera Manager restarts the worker roles rack-by-rack, in alphabetical order, and within each rack, hosts are restarted in alphabetical order. If you use the default replication factor of 3, Hadoop tries to keep the replicas on at least 2 different racks. So if you have multiple racks, you can use a higher batch size than the default 1. However, using a batch size that is too high means that fewer worker roles are active at any time during the upgrade, which can cause temporary performance degradation. If you are using a single rack, restart *one worker node at a time* to ensure data availability during upgrade.

### Advanced Options > Sleep between batches

Amount of time Cloudera Manager waits before starting the next batch.

### Advanced Options > Failed threshold

The number of *batch* failures that cause the entire rolling restart to fail. For example if you have a very large cluster, you can use this option to allow some failures when you know that the cluster is functional when some worker roles are down.

### 13 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, deploys client configuration files, restarts services, and performs a rolling restart of the services that support it.

If your cluster was previously installed or upgraded using *packages*, the wizard may indicate that some services cannot start because their parcels are not available. To download the required parcels:

1. In another browser tab, open the Cloudera Manager Admin Console.
2. Select **Hosts > Parcels**.
3. Locate the row containing the missing parcel and click the button to **Download, Distribute**, and then **Activate** the parcel.
4. Return to the upgrade wizard and click the **Retry** button.

The Upgrade Wizard continues upgrading the cluster.

### 14 Click **Finish** to return to the Home page.

## Step 7: Recover from Failed Steps or Perform a Manual Upgrade

If one or more hosts fail to restart, you can resume the rolling restart after fixing the problems that caused the upgrade to fail. Cloudera Manager will skip restarting roles that have already successfully restarted.

The actions performed by the upgrade wizard are listed in [Performing Upgrade Wizard Actions Manually](#) on page 61. If any of the steps in the **Upgrade Cluster Command** screen fail, complete the steps as described in that section before proceeding.

## Step 8: Remove the Previous CDH Version Packages and Refresh Symlinks

**[Not required for CDH maintenance release upgrades.]**

*Skip this step if your previous installation or upgrade used parcels.*

If your previous installation of CDH was done using packages, remove those packages on all hosts where you installed the parcels and refresh the symlinks so that clients will run the new software versions.

1. If your Hue service uses the embedded SQLite database, back up `/var/lib/hue/desktop.db` to a location that is not `/var/lib/hue` because this directory is removed when the packages are removed.
2. Uninstall the CDH packages on each host:

- **Not including Impala and Search**

**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**

**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 database, restore the database 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.

**Step 9: Finalize HDFS Rolling Upgrade**

**[Not required for CDH maintenance release upgrades.]**

The steps in this section are only required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

To determine if you can finalize, 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. Verifying that you are ready to finalize the upgrade can take a long time.

1. Go to the HDFS service.
2. Select **Actions > Finalize Rolling Upgrade** and click **Finalize Rolling Upgrade** to confirm.

**Step 10: Exit Maintenance Mode**

If you entered maintenance mode during this upgrade, [exit maintenance mode](#).

**Step 11: Clear Browser Cache (Hue only)**

If you have enabled the Hue service in your upgraded cluster, users may need to clear the cache in their Web browsers before accessing Hue.

### Upgrading to CDH 5.x Using Parcels

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

This topic describes how to upgrade CDH from any version of CDH 5.x to a higher version of CDH 5.x using Cloudera Manager and [parcels](#). If the CDH 5 cluster you are upgrading was installed using packages, you can upgrade it using parcels, and the upgraded version of CDH will then use parcels for future upgrades or changes. You can also [migrate your cluster from using packages to using parcels](#) before starting the upgrade. The minor version of Cloudera Manager you use to perform the upgrade must be equal to or greater than the CDH minor version. To upgrade Cloudera Manager, see [Overview of Upgrading Cloudera Manager](#) on page 8.

The upgrade procedure described in this topic requires cluster downtime. If the cluster was installed using parcels, has a Cloudera Enterprise license, and has HDFS high availability enabled, you can perform a [rolling upgrade](#) that does not require cluster downtime.



**Note:** If you are upgrading to a *maintenance* version of CDH, skip any steps that are labeled **[Not required for CDH maintenance release upgrades.]**.

The version numbers for maintenance releases differ only in the third digit, for example when upgrading from CDH 5.8.0 to CDH 5.8.2. See [Maintenance Version Upgrades](#) on page 34.

To upgrade CDH using parcels:

#### Step 1: Collect Upgrade Information

Before starting an upgrade, collect the following information:

1. Host credentials. You must have SSH access and be able to log in using a root account or an account that has password-less sudo permission.
2. The version of Cloudera Manager used in your cluster. Go to **Support > About**.
3. The version of the JDK deployed in the cluster. Go to **Support > About**.
4. The version of CDH. The CDH version number displays next to the cluster name on the **Home** page.
5. Whether the cluster was installed using parcels or packages. This information displays next to the CDH version on the **Home** page of Cloudera Manager.
6. The services enabled in your cluster. Go to **Clusters > Cluster name**.
7. Operating system type and version. Go to **Hosts** and click on a hostname in the list. The operating system type and version displays in the **Distribution** row in the **Details** section.
8. Database information for the databases used by Sqoop, Oozie, Hue, Hive Metastore, and Sentry Server (information is only required if these services are enabled in the cluster).

Gather the following information:

- Type of database (PostgreSQL, Embedded PostgreSQL, MySQL, MariaDB, or Oracle)
- Hostnames of the databases
- Credentials for the databases

To locate database information:

- **Sqoop, Oozie, and Hue** – Go to **Cluster Name > Configuration > Database Settings**.
- **Hive Metastore** – Go to the Hive service, select **Configuration**, and select the **Hive Metastore Database** category.
- **Sentry** – Go to the Sentry service, select **Configuration**, and select the **Sentry Server Database** category.

#### Step 2: Complete Pre-Upgrade Steps

1. Review the [CDH 5 and Cloudera Manager 5 Requirements and Supported Versions](#) for the new versions you are upgrading to.

2. Read the [CDH 5 Release Notes](#).
3. Read the [Cloudera Security Bulletins](#).
4. Ensure that Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Java Development Kit Installation](#) or [Upgrading to Oracle JDK 1.8](#) on page 65, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
5. Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:

- **Targeted CDH Version:** 5.0.5

**Minimum Cloudera Manager Version:** 5.0.x

- **Targeted CDH Version:** 5.8.2

**Minimum Cloudera Manager Version:** 5.8.x

6. If you are upgrading from CDH 5.1 or lower, and use Hive Date partition columns, you may need to update the date format. See [Date partition columns](#).
7. If the cluster uses Impala, 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](#).
8. Run the [Host Inspector](#) and fix every issue. Go to **Cluster > Inspect Hosts**.
9. Run the [Security Inspector](#) and fix and reported errors. (Go to **Administration > Security** and click **Security Inspector**.)
- 10 Log in to any cluster node as the `hdfs` user, run the following commands, and correct any reported errors:

```
hdfs fsck /
```



**Note:** the `fsck` command may take 10 minutes or more to complete, depending on the number of files in your cluster.

```
hdfs dfsadmin -report
```

See [HDFS Commands Guide](#) in the Apache Hadoop documentation.

- 11 Log in to any DataNode as the `hbase` user, run the following command, and correct any reported errors:

```
hbase hbck
```

See [Checking and Repairing HBase Tables](#).

- 12 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.
- 13 To avoid unnecessary alerts during the upgrade process, enable [maintenance mode](#) on your cluster before you start the upgrade. This stops email alerts and SNMP traps from being sent, but does not stop checks and configuration validations. Be sure to exit maintenance mode when you have finished the upgrade to reen able Cloudera Manager alerts.
- 14 If you have configured Hue to use TLS/SSL and you are upgrading from CDH 5.2 or lower to CDH 5.3 or higher, Hue validates CA certificates and requires a truststore. To create a truststore, follow the instructions in [Hue as a TLS/SSL Client](#).
- 15 If your cluster uses the Flume Kafka client, and you are upgrading to CDH 5.8.0 or CDH 5.8.1, perform the extra steps described in [Upgrading to CDH 5.8.0 or CDH 5.8.1 When Using the Flume Kafka Client](#) on page 63 and then continue with the procedures in this topic.
- 16 If your cluster uses Impala and Llama, this role has been deprecated as of CDH 5.9 and you must remove the role from the Impala service before starting the upgrade. If you do not remove this role, the upgrade wizard will halt the upgrade.

To determine if Impala uses Llama:

1. Go to the Impala service.
2. Select the **Instances** tab.
3. Examine the list of roles in the **Role Type** column. If Llama appears, the Impala service is using Llama.

To remove the Llama role:

1. Go to the Impala service and select **Actions > Disable YARN and Impala Integrated Resource Management**.

The **Disable YARN and Impala Integrated Resource Management** wizard displays.

2. Click **Continue**.

The **Disable YARN and Impala Integrated Resource Management Command** page displays the progress of the commands to disable the role.

3. When the commands have completed, click **Finish**.

17. If you have deployed the Sentry service in your cluster, and are upgrading from CDH 5.12 or lower, you may need to increase the Java heap memory for Sentry. See [Performance Guidelines](#).

### Step 3: Stop Cluster Services

[Not required for CDH maintenance release upgrades.]

The steps in this section are only required for the following upgrades:

- 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.

### Step 4: Back up the HDFS Metadata on the NameNode

[Not required for CDH maintenance release upgrades.]

The steps in this section are only required for the following upgrades:

- 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, because 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 a file with the extension *lock* exists in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, beginning with shutting down the NameNode role.

## Step 5: Back Up Databases



**Note:** Backing up databases requires that you stop some services, which may make them unavailable during backup.

Back up the databases for any of the following services that are deployed in your cluster:

**Table 2: Service Databases to Back Up**

Service	Where to find database information
Sqoop	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Sqoop service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Hue	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Hue service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Oozie	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Oozie service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Cloudera Navigator Audit Server	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Cloudera Navigator Metadata Server	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Activity Monitor	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Reports Manager	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Sentry Server	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Sentry service</b> > <b>Configuration</b> and select the <b>Sentry Server Database</b> category.
Hive Metastore	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Hive service</b> > <b>Configuration</b> and select the <b>Hive Metastore Database</b> category.

### To back up the databases:

1. If not already stopped, stop the service:

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



to the right of the service name and select **Stop**.

- b. Click **Stop** in the next screen to confirm. When you see a **Finished** status, the service has stopped.

2. Back up the database. See [Backing Up Databases](#) for detailed instructions for each supported type of database.
3. Restart the service:

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



to the right of the service name and select **Start**.

- b. Click **Start** that appears in the next screen to confirm. When you see a **Finished** status, the service has started.

### Step 6: Run the Upgrade Wizard



**Note:** If Cloudera Manager detects a failure while upgrading CDH, Cloudera Manager displays a dialog box where you can create a diagnostic bundle to send to Cloudera Support so they can help you recover from the failure. The cluster name and time duration fields are pre-populated to capture the correct data.

1. If your cluster has Kudu 1.4.0 (or lower) installed, deactivate the existing Kudu parcel. Starting with Kudu 1.5.0 / CDH 5.13, Kudu is part of the CDH parcel and does not need to be installed separately.
2. If your cluster has Spark 2.1 or Spark 2.2 installed and you want to upgrade to CDH 5.13 or higher, your cluster must be running either Spark 2.1, release 2 or Spark 2.2 release 1 before upgrading. To install these versions of Spark do the following before running the Upgrade Wizard:

- a. Install the Custom Service Descriptor (CSD) file. See

- [Installing Spark 2.1](#)
- [Installing Spark 2.2](#)




**Note:** Spark 2.2 requires that JDK 1.8 be deployed throughout the cluster. JDK 1.7 is not supported for Spark 2.2.

See [Java Development Kit Installation](#).

- b. Download, distribute, and activate the Parcel for the version of Spark that you are installing:

- **Spark 2.1 release 2:** The parcel name includes "cloudera2" in its name.
- **Spark 2.2 release 1:** The parcel name includes "cloudera1" in its name.

See [Managing Parcels](#).

3. From the **Home > Status** tab, click  next to the cluster name and select **Upgrade Cluster**.

The **Getting Started** page of the upgrade wizard displays.

4. If the option to pick between packages and parcels displays, select **Use Parcels**.
5. In the **Choose CDH Version (Parcels)** field, select the CDH version. If no qualifying parcels are listed, or you want to upgrade to a different version, click the **Modify the Remote Parcel Repository URLs** link to go to the configuration page for **Remote Parcel Repository URLs** and add the appropriate URL to the configuration. See [Parcel Configuration Settings](#) for information about entering the correct URL for parcel repositories. Click **Continue**.
6. If you previously installed the GPLEXTRAS parcel, download and distribute the version of the GPLEXTRAS parcel that matches the version of CDH that you are upgrading to.
7. Read the notices for steps you must complete before upgrading, click the **Yes, I ...** checkboxes after completing the steps, and click **Continue**. If you downloaded a new version of the GPLEXTRAS parcel, the Upgrade Wizard displays a message that the GPLEXTRAS parcel conflicts with the version of the CDH parcel, similar to the following:

**GPLEXTRAS 5.10.0-1.cdh5.10.0.p0.41 conflicts with CDH 5.11.1-1.**



**Let Cloudera Manager resolve this automatically: Activate GPLEXTRAS**

Select the option to resolve the conflicts automatically and click **Continue**.

Cloudera Manager deactivates the old version of the GPLEXTRAS parcel, activates the new version and verifies that all hosts have the correct software installed.

**8. Click Continue.**

The Host Inspector runs and displays the CDH version on the hosts.

**9. Click Continue.**

The **Choose Upgrade Procedure** screen displays the available types of upgrades:

- **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster.
- **Manual upgrade** Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually upgrading is difficult and for advanced users only. To perform a manual upgrade:
  1. Select the **Let me upgrade the cluster** checkbox.
  2. Click **Continue**.
  3. See [Performing Upgrade Wizard Actions Manually](#) on page 61 for the required steps.

**10 Select Full Cluster Restart.**

**11 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, deploys client configuration files, and restarts services. If any of the steps fail, correct any reported errors and click the **Retry** button. If you click the **Abort** button, the **Retry** button at the top right is enabled.

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](#).

**12 Click Continue.**

The wizard reports the result of the upgrade.

If your cluster was previously installed or upgraded using *packages*, the wizard may indicate that some services cannot start because their parcels are not available. To download the required parcels:

1. In another browser tab, open the Cloudera Manager Admin Console.
2. Select **Hosts > Parcels**.
3. Locate the row containing the missing parcel and click the button to **Download, Distribute**, and then **Activate** the parcel.
4. Return to the upgrade wizard and click the **Retry** button.

The Upgrade Wizard continues upgrading the cluster.

**13 Click Finish** to return to the Home page.

### Step 7: Recover from Failed Steps or Perform a Manual Upgrade

The actions performed by the upgrade wizard are listed in [Performing Upgrade Wizard Actions Manually](#) on page 61. If any of the steps in the **Upgrade Cluster Command** screen fail, complete the steps as described in that section before proceeding.

### Step 8: Remove the Previous CDH Version Packages and Refresh Symlinks

**[Not required for CDH maintenance release upgrades.]**

*Skip this step if your previous installation or upgrade used parcels.*

If your previous installation of CDH was done using packages, remove those packages on all hosts where you installed the parcels and refresh the symlinks so that clients will run the new software versions.

1. If your Hue service uses the embedded SQLite database, back up `/var/lib/hue/desktop.db` to a location that is not `/var/lib/hue` because this directory is removed when the packages are removed.
2. Uninstall the CDH packages on each host:

- **Not including Impala and Search**

**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**

**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 database, restore the database 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.



## Step 9: Finalize the HDFS Metadata Upgrade

**[Not required for CDH maintenance release upgrades.]**

The steps in this section are only required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

To determine if you can finalize, 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. 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 that the following behavior continues until the upgrade is finalized:

- 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. Select the **NameNode** instance. If you have enabled high availability for HDFS, select **NameNode (Active)**.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

## Step 10: Exit Maintenance Mode

If you entered maintenance mode during this upgrade, [exit maintenance mode](#).

## Step 11: Clear Browser Cache (Hue only)

If you have enabled the Hue service in your upgraded cluster, users may need to clear the cache in their Web browsers before accessing Hue.

## Upgrading to CDH 5.x Using Packages

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

This topic describes how to upgrade CDH from any version of CDH 5.x to a higher version of CDH 5.x using Cloudera Manager and *packages*. The minor version of Cloudera Manager you use to perform the upgrade must be equal to or greater than the CDH minor version. To upgrade Cloudera Manager, see [Overview of Upgrading Cloudera Manager](#) on page 8.

The upgrade procedure described in this topic requires cluster downtime. If the cluster was installed using parcels, has a Cloudera Enterprise license, and has HDFS high availability enabled, you can perform a [rolling upgrade](#) that does not require cluster downtime.

If the CDH 5 cluster you are upgrading was installed using packages, you can [upgrade it using parcels](#), and the upgraded version of CDH will then use parcels for future upgrades or changes. You can also [migrate your cluster from using packages to using parcels](#) before starting the upgrade.

Upgrading using packages is for advanced users. Some parts of this procedure require you to run commands from the command line on *all cluster hosts*. This may require significant time. Additionally, if other software packages are installed on cluster hosts, the upgrade steps described in this topic could update dependencies that affect those packages.



**Note:** If you are upgrading to a *maintenance* version of CDH, skip any steps that are labeled **[Not required for CDH maintenance release upgrades.]**.

The version numbers for maintenance releases differ only in the third digit, for example when upgrading from CDH 5.8.0 to CDH 5.8.2. See [Maintenance Version Upgrades](#) on page 34.

To upgrade CDH using packages:

### Step 1: Collect Upgrade Information

Before starting an upgrade, collect the following information:

1. Host credentials. You must have SSH access and be able to log in using a root account or an account that has password-less sudo permission.
2. The version of Cloudera Manager used in your cluster. Go to **Support > About**.
3. The version of the JDK deployed in the cluster. Go to **Support > About**.
4. The version of CDH. The CDH version number displays next to the cluster name on the **Home** page.
5. Whether the cluster was installed using parcels or packages. This information displays next to the CDH version on the **Home** page of Cloudera Manager.
6. The services enabled in your cluster. Go to **Clusters > Cluster name**.
7. Operating system type and version. Go to **Hosts** and click on a hostname in the list. The operating system type and version displays in the **Distribution** row in the **Details** section.
8. Database information for the databases used by Sqoop, Oozie, Hue, Hive Metastore, and Sentry Server (information is only required if these services are enabled in the cluster).

Gather the following information:

- Type of database (PostgreSQL, Embedded PostgreSQL, MySQL, MariaDB, or Oracle)
- Hostnames of the databases
- Credentials for the databases

To locate database information:

- **Sqoop, Oozie, and Hue** – Go to **Cluster Name > Configuration > Database Settings**.
- **Hive Metastore** – Go to the Hive service, select **Configuration**, and select the **Hive Metastore Database** category.
- **Sentry** – Go to the Sentry service, select **Configuration**, and select the **Sentry Server Database** category.

### Step 2: Complete Pre-Upgrade Steps

1. Review the [CDH 5 and Cloudera Manager 5 Requirements and Supported Versions](#) for the new versions you are upgrading to.
2. Read the [CDH 5 Release Notes](#).
3. Read the [Cloudera Security Bulletins](#).
4. Ensure that Java 1.7 or 1.8 is installed across the cluster. For installation instructions and recommendations, see [Java Development Kit Installation](#) or [Upgrading to Oracle JDK 1.8](#) on page 65, and make sure you have read [Known Issues and Workarounds in Cloudera Manager 5](#) before you proceed with the upgrade.
5. Ensure that the Cloudera Manager minor version is *equal to or greater than* the CDH minor version. For example:
  - **Targeted CDH Version:** 5.0.5  
**Minimum Cloudera Manager Version:** 5.0.x
  - **Targeted CDH Version:** 5.8.2  
**Minimum Cloudera Manager Version:** 5.8.x

6. If you are upgrading from CDH 5.1 or lower, and use Hive Date partition columns, you may need to update the date format. See [Date partition columns](#).
7. If the cluster uses Impala, 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](#).
8. Run the [Host Inspector](#) and fix every issue. Go to **Cluster > Inspect Hosts**.
9. Run the [Security Inspector](#) and fix and reported errors. (Go to **Administration > Security** and click **Security Inspector**.)
- 10 Log in to any cluster node as the `hdfs` user, run the following commands, and correct any reported errors:

```
hdfs fsck /
```



**Note:** the `fsck` command may take 10 minutes or more to complete, depending on the number of files in your cluster.

```
hdfs dfsadmin -report
```

See [HDFS Commands Guide](#) in the Apache Hadoop documentation.

- 11 Log in to any DataNode as the `hbase` user, run the following command, and correct any reported errors:

```
hbase hbck
```

See [Checking and Repairing HBase Tables](#).

- 12 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.
- 13 To avoid unnecessary alerts during the upgrade process, enable [maintenance mode](#) on your cluster before you start the upgrade. This stops email alerts and SNMP traps from being sent, but does not stop checks and configuration validations. Be sure to exit maintenance mode when you have finished the upgrade to reenale Cloudera Manager alerts.
- 14 If you have configured Hue to use TLS/SSL and you are upgrading from CDH 5.2 or lower to CDH 5.3 or higher, Hue validates CA certificates and requires a truststore. To create a truststore, follow the instructions in [Hue as a TLS/SSL Client](#).
- 15 If your cluster uses the Flume Kafka client, and you are upgrading to CDH 5.8.0 or CDH 5.8.1, perform the extra steps described in [Upgrading to CDH 5.8.0 or CDH 5.8.1 When Using the Flume Kafka Client](#) on page 63 and then continue with the procedures in this topic.
- 16 If your cluster uses Impala and Llama, this role has been deprecated as of CDH 5.9 and you must remove the role from the Impala service before starting the upgrade. If you do not remove this role, the upgrade wizard will halt the upgrade.

To determine if Impala uses Llama:

1. Go to the Impala service.
2. Select the **Instances** tab.
3. Examine the list of roles in the **Role Type** column. If Llama appears, the Impala service is using Llama.

To remove the Llama role:

1. Go to the Impala service and select **Actions > Disable YARN and Impala Integrated Resource Management**.

The **Disable YARN and Impala Integrated Resource Management** wizard displays.

2. Click **Continue**.

The **Disable YARN and Impala Integrated Resource Management Command** page displays the progress of the commands to disable the role.

3. When the commands have completed, click **Finish**.

17. If you have deployed the Sentry service in your cluster, and are upgrading from CDH 5.12 or lower, you may need to increase the Java heap memory for Sentry. See [Performance Guidelines](#).

### Step 3: Upgrade Unmanaged Components

Upgrade any unmanaged components before upgrading components that are managed by Cloudera Manager.

### Step 4: 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.

### Step 5: Back up the HDFS Metadata on the NameNode

**[Not required for CDH maintenance release upgrades.]**

The steps in this section are only required for the following upgrades:

- 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, because 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 a file with the extension *lock* exists in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, beginning with shutting down the NameNode role.

### Step 6: Back Up Databases





**Note:** Backing up databases requires that you stop some services, which may make them unavailable during backup.

Back up the databases for any of the following services that are deployed in your cluster:

**Table 3: Service Databases to Back Up**

Service	Where to find database information
Sqoop	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Sqoop service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Hue	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Hue service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Oozie	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Oozie service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Cloudera Navigator Audit Server	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Cloudera Navigator Metadata Server	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Activity Monitor	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Reports Manager	Go to <b>Clusters</b> > <b>Cloudera Management Service</b> > <b>Configuration</b> and select the <b>Database</b> category.
Sentry Server	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Sentry service</b> > <b>Configuration</b> and select the <b>Sentry Server Database</b> category.
Hive Metastore	Go to <b>Clusters</b> > <b>Cluster Name</b> > <b>Hive service</b> > <b>Configuration</b> and select the <b>Hive Metastore Database</b> category.

**To back up the databases:**

1. If not already stopped, stop the service:
  - a. On the **Home** > **Status** tab, click  to the right of the service name and select **Stop**.
  - b. Click **Stop** in the next screen to confirm. When you see a **Finished** status, the service has stopped.
2. Back up the database. See [Backing Up Databases](#) for detailed instructions for each supported type of database.
3. Restart the service:
  - a. On the **Home** > **Status** tab, click  to the right of the service name and select **Start**.
  - b. Click **Start** that appears in the next screen to confirm. When you see a **Finished** status, the service has started.

**Step 7: Upgrade Managed Components**

Use *one* of the following strategies to upgrade CDH 5:

**Use the Cloudera 1-Click Package**

Using the Cloudera "1-click Install" package is the simplest way to upgrade only the Cloudera packages.

1. Check whether you have the CDH 5 "1-click" repository installed by running the following command on each cluster host:

## RHEL/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 Step 3 on page 56, **Install the CDH packages**. If instead you see:

```
package CDH 5-repository is not installed
```

proceed with Step 2 on page 54, **Install the 1-click package**.

## Ubuntu and Debian

```
dpkg -l | grep CDH 5-repository
```

If the repository is installed, skip to Step 3 on page 56, **Install the CDH packages**; otherwise proceed with Step 2 on page 54, **Install the 1-click package**.

2. **Install the CDH5 "1-click" package.** 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.

## RHEL compatible

1. Download and install the "1-click Install" package on each cluster host.

- 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. Using a repository key helps to verify that you are using a signed version of the package. If you add a repository key, you can omit the `--nogpgcheck` option when running `yum` commands. Run the following command to add the 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 on each cluster host:

- 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 the following:

```
$ sudo zypper refresh
```

2. Optionally add a repository key. Using a repository key helps to verify that you are using a signed version of the package.

```
$ 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 on each cluster host:

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

OS Version	Package Link
Jessie	<a href="#">Jessie package</a>
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 (Using a repository key helps to verify that you are using a signed version of the package.):

- **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 by running the following command on all cluster hosts:



**Note:**

- Installing these packages also installs all the other CDH packages required for a full CDH 5 installation.
- Kudu is only supported on a subset of the operating systems supported by CDH. These are: RHEL/CentOS 6, RHEL/CentOS 7, SLES 12 SP1, Ubuntu 14.04, 16.04, Debian 8.2, 8.4. Remove the `kudu` package from the command if you are installing on any other operating systems.

### RHEL 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-webhcat hue-beeswax hue-hbase hue-impala hue-pig
hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell
kudu 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-webhcat hue-beeswax hue-hbase hue-impala hue-pig
hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell
kudu 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-webhcat hue-beeswax hue-hbase hue-impala
hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala
impala-shell kudu kite llama mahout oozie parquet pig pig-udf-datafu search sentry solr
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```

## Use Operating System Package Management Tools

- Use your operating system's package management tools to update all packages to the latest version using standard repositories. This approach minimizes the amount of configuration required and uses the simplest commands. This can take a considerable amount of time if you have not upgraded the system recently. To update all packages on your system, run the following command on each cluster host:

### RHEL

```
$ sudo yum update
```

### SLES

```
$ sudo zypper up
```

### Ubuntu or Debian

```
$ sudo apt-get upgrade
```



### Use a Specific Set of Packages


To upgrade managed components to a specific version of CDH, specify the packages you want to use for the upgrade. Follow the procedure at [Upgrade Managed Components Using a Specific Set of Packages](#) on page 58 and then continue with the procedures in this topic.

## Step 8: Update Symlinks for the Newly Installed Components

Restart the Cloudera Manager Agents to force an update of the symlinks to point to the newly installed components on each host. Run the following command on all cluster hosts:

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

## Step 9: Run the Upgrade Wizard

1. Log in to 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 cluster hosts have the correct software installed. If the packages have not been installed, a warning displays to that effect. Install the missing packages and click **Retry**. When there are no errors, click **Continue**.
7. The Host Inspector runs. Correct any errors displayed and click **Continue**.

The **Choose Upgrade Procedure** screen displays the available types of upgrades:

- **Full Cluster Restart** - Cloudera Manager performs all service upgrades and restarts the cluster.
- **Manual upgrade** Cloudera Manager configures the cluster to the specified CDH version but performs no upgrades or service restarts. Manually upgrading is difficult and for advanced users only. To perform a manual upgrade:

1. Select the **Let me upgrade the cluster** checkbox.
2. Click **Continue**.
3. See [Performing Upgrade Wizard Actions Manually](#) on page 61 for the required steps.

8. Select **Full Cluster Restart**.
9. Click **Continue**. The **Upgrade Cluster Command** screen displays the result of the commands run by the wizard as it shuts down all services, upgrades services, deploys client configuration files, and restarts services. If any of the steps fails or if you click the **Abort** button, the **Retry** button at the top right is enabled.

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. See [Performing Upgrade Wizard Actions Manually](#) on page 61.

- 10 Click **Continue**. The wizard reports the result of the upgrade.
- 11 Click **Finish** to return to the Home page.

## Step 10: Recover from Failed Steps or Perform a Manual Upgrade

The actions performed by the upgrade wizard are listed in [Performing Upgrade Wizard Actions Manually](#) on page 61. If any of the steps in the **Upgrade Cluster Command** screen fail, complete the steps as described in that section before proceeding.

### Step 11: Finalize the HDFS Metadata Upgrade

[Not required for CDH maintenance release upgrades.]

The steps in this section are only required for the following upgrades:

- CDH 5.0 or 5.1 to 5.2 or higher
- CDH 5.2 or 5.3 to 5.4 or higher

To determine if you can finalize, 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. 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 that the following behavior continues until the upgrade is finalized:

- 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. Select the **NameNode** instance. If you have enabled high availability for HDFS, select **NameNode (Active)**.
4. Select **Actions > Finalize Metadata Upgrade** and click **Finalize Metadata Upgrade** to confirm.

### Step 12: Exit Maintenance Mode

If you entered maintenance mode during this upgrade, [exit maintenance mode](#).

### Step 13: Clear Browser Cache (Hue only)

If you have enabled the Hue service in your upgraded cluster, users may need to clear the cache in their Web browsers before accessing Hue.

### Upgrade Managed Components Using a Specific Set of Packages



**Important:** The procedures in this topic are part of the procedure [Upgrading to CDH 5.x Using Packages](#) on page 49. They describe how to configure Cloudera Manager to upgrade managed CDH components using a specific set of packages instead of using the latest packages available. After completing the steps in this topic, return to [Step 7: Upgrade Managed Components](#) on page 53 to complete the upgrade.

1. Download and save the repo file.

- **RHEL-compatible systems**

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.

OS Version	Link
RHEL/CentOS/Oracle 5	<a href="#">RHEL/CentOS/Oracle 5 link</a>
RHEL/CentOS 6 (64-bit)	<a href="#">RHEL/CentOS 6 link</a>

- **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
```

- **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.

- **RHEL-compatible systems**

Open the repo file you 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 [RHEL 6](#), it looks like this when you open it:

```
[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 looks 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
```

- **SLES systems**

Open the repo file that you just added 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:

```
[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 looks 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
```

- **Ubuntu and Debian systems**

Replace `-cdh5` near the end of each line (before `contrib`) with the CDH release you need to install. The following examples use CDH 5.1.0:

- **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
```

- **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:

- **RHEL-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**

#### – 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:

#### • RHEL-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-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 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-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 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-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 parquet pig pig-udf-datafu search sentry solr
solr-mapreduce spark-python sqoop sqoop2 whirr zookeeper
```



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

### 5. Return to [Step 7: Upgrade Managed Components](#) on page 53 to complete the upgrade.

## Performing Upgrade Wizard Actions Manually



**Important:** Perform the steps in this section only if the upgrade wizard reports a failure.

### 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.

## Upgrading CDH and Managed Services Using Cloudera Manager

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
  - From any version of CDH to CDH 5.12 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.

### 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 5.8.0 or CDH 5.8.1 When Using the Flume Kafka Client

Due to the change of offset storage from ZooKeeper to Kafka in the CDH 5.8 Flume Kafka client, data might not be consumed by the Flume agents, or might be duplicated (if `kafka.auto.offset.reset=smallest`) during an upgrade to CDH 5.8.0 or CDH 5.8.1. To prevent this, perform the steps described below before you upgrade your system.



**Important:** This issue has been fixed for CDH 5.8.2 and higher. If you are upgrading to CDH 5.8.2 or higher, you do not need to perform this procedure. For more information, see [Cloudera Distribution of Apache Kafka Known Issues](#).

The upgrade process is based on this example configuration:

```
tier1.sources = source1
tier1.channels = channel1
tier1.sinks = sink1

tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource
tier1.sources.source1.zookeeperConnect = zkhost:2181
tier1.sources.source1.topic = flumetopic
tier1.sources.source1.groupId = flume
tier1.sources.source1.channels = channel1
tier1.sources.source1.interceptors = i1 i2
tier1.sources.source1.interceptors.i1.type = timestamp
tier1.sources.source1.interceptors.i2.type = host
tier1.sources.source1.kafka.consumer.timeout.ms = 100

tier1.channels.channel1.type = org.apache.flume.channel.kafka.KafkaChannel
tier1.channels.channel1.brokerList=broker1:9092,broker2:9092
tier1.channels.channel1.zookeeperConnect=zkhost:2181
tier1.channels.channel1.topic=flumechannel1
tier1.channels.channel1.groupId = flumechannel
tier1.channels.channel1.capacity = 10000
tier1.channels.channel1.transactionCapacity = 1000

tier1.sinks.sink1.type = hdfs
tier1.sinks.sink1.hdfs.path = /tmp/kafka/{topic}/
tier1.sinks.sink1.hdfs.filePrefix = %{host}-
tier1.sinks.sink1.hdfs.rollInterval = 60
tier1.sinks.sink1.hdfs.rollSize = 0
tier1.sinks.sink1.hdfs.rollCount = 0
tier1.sinks.sink1.hdfs.fileType = DataStream
tier1.sinks.sink1.channel = channel1
tier1.sinks.sink1.hdfs.kerberosKeytab = $KERBEROS_KEYTAB
tier1.sinks.sink1.hdfs.kerberosPrincipal = $KERBEROS_PRINCIPAL
```

Perform the following steps to upgrade CDH.

1. If you are using a version lower than CDH 5.7, first [upgrade to CDH 5.7](#). If for some reason you cannot upgrade to CDH 5.7, contact your Cloudera Sales Engineer for assistance, or file a support case with specific versions from which and to which you are upgrading.
2. Add the following sections to the source and channel for the Flume configuration:

```
# Added for source upgrade compatability
tier1.sources.source1.kafka.bootstrap.servers = broker1:9092,broker2:9092
tier1.sources.source1.kafka.offsets.storage = kafka
tier1.sources.source1.kafka.dual.commit.enabled = true
tier1.sources.source1.kafka.consumer.group.id = flume
tier1.sources.source1.kafka.topics = flumetopic
```

```
# Added for channel upgrade compatability
tier1.channels.channell1.kafka.topic = flumechannell1
tier1.channels.channell1.kafka.bootstrap.servers = broker1:9092,broker2:9092
tier1.channels.channell1.kafka.consumer.group.id = flumechannel
tier1.channels.channell1.kafka.offsets.storage = kafka
tier1.channels.channell1.kafka.dual.commit.enabled = true
```

3. Restart (or rolling restart) the Flume agents. This switches `offsets.storage` to Kafka, but keeps both the Kafka and ZooKeeper offsets updated because the `dual.commit.enabled` property is set to `true`. Confirm that Kafka messages are flowing through the Flume servers. Updating the offsets only occurs when new messages are consumed, so there must be at least one Kafka message consumed by the Flume agent, or one event passed through the Flume channel. Use the following commands to verify that Flume is properly updating the offsets in Kafka (the `egrep` command is used to match the correct topic names: in this example, `flumetopic` and `flumechannell1`):

```
echo "exclude.internal.topics=false" > /tmp/consumer.config
kafka-console-consumer --consumer.config /tmp/consumer.config
--formatter "kafka.coordinator.GroupMetadataManager\$OffsetsMessageFormatter"
--zookeeper zkhost:2181 --topic __consumer_offsets |egrep -e "flumetopic|flumechannell1"
```

Output should be similar to the following and show that the Flume source and/or channel topics offsets are being incremented:

```
[flume,flumetopic,0]::[OffsetMetadata[70,cf9e5630-214e-4689-9869-5e077c936ffb],CommitTime
1469827951129,ExpirationTime 1469914351129]
[flumechannell,flumechannell1,0]::[OffsetMetadata[61,875e7a82-1c22-43be-acaa-eb4d63e7f71e],CommitTime
1469827951128,ExpirationTime 1469914351128]
[flumechannell,flumechannell1,0]::[OffsetMetadata[62,66bda888-0a70-4a02-a286-7e2e7d14050d],CommitTime
1469827951131,ExpirationTime 1469914351131]
```

4. Perform the upgrade to CDH 5.8.0 or CDH 5.8.1. The Flume agents are restarted during the process. Flume continues to consume the source topic where it left off, and the sinks continue draining from the Kafka channels where they left off. Post upgrade, remove the following deprecated properties from `flume.conf` because they are no longer used in CDH 5.8.0 or higher:

```
tier1.sources.source1.zookeeperConnect = zkhost:2181
tier1.sources.source1.topic = flumetopic
tier1.sources.source1.groupId = flume

tier1.channels.channell1.zookeeperConnect=zkhost:2181
tier1.channels.channell1.topic=flumechannell1
tier1.channels.channell1.groupId = flumechannel
```



## 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 [CDH and Cloudera Manager Supported JDK Versions](#).



### 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, you automatically install the policy files; for unmanaged deployments, install them manually.) See [Using AES-256 Encryption](#) on page 66.

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

The process for upgrading to Oracle JDK 1.8 varies depending on whether you have a [Cloudera Manager Deployment](#) or an [Unmanaged Deployment](#).

## 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. 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](#) for instructions.
8. On the Cloudera Manager Server host, configure the location of the JDK in `/etc/default/cloudera-scm-server`.
9. Start the Cloudera Manager Server.
- 10 Start all Cloudera Manager Agents.
- 11 Configure the location of the JDK on cluster hosts as described in [Configuring a Custom Java Home Location](#).
- 12 If you have configured TLS for Cloudera Manager, as described in [Level 0: Basic TLS/SSL Configuration](#), copy the `jssecacerts` file from the previous JDK installation to the new JDK installation. For example:

```
cp previous_java_home/jre/lib/security/jssecacerts new_java_home/jre/lib/security
```

(Substitute *previous\_java\_home* and *new\_java\_home* with the paths to the JDK installations.)

- 13 Start all clusters.
- 14 Start the Cloudera Management Service.
- 15 Delete your previous Java version files.

## Upgrading to Oracle JDK 1.8 in an Unmanaged Deployment



### Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.13.x. See [Cloudera Documentation](#) for information specific to other releases.

1. [Upgrade to CDH 5.3 or higher](#) if you have not already done so.
2. [Shut down the cluster.](#)
3. On each cluster host:
  - a. Install the same [supported version](#) of JDK 1.8. See [Java Development Kit Installation](#) 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.
  - c. If you have configured TLS for Cloudera Manager, as described in [Level 0: Basic TLS/SSL Configuration](#), copy the `jssecacerts` file from the previous JDK installation to the new JDK installation. For example:

```
cp previous_java_home/jre/lib/security/jssecacerts new_java_home/jre/lib/security
```

(Substitute *previous\_java\_home* and *new\_java\_home* with the paths to the JDK installations.)

4. [Start the cluster.](#)
5. Delete your previous Java version files.

## Using AES-256 Encryption

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_encetypes` 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
```

## Upgrading Cloudera Navigator Components

[Cloudera Navigator Data Management](#) is a comprehensive auditing, data governance, compliance, data stewardship, and data lineage discovery component that is fully integrated with Hadoop. Cloudera Navigator comprises two different services, the Navigator Metadata Server and the Navigator Audit Server. Both are upgraded automatically whenever Cloudera Manager Server is upgraded, but can also be upgraded individually. Before upgrading, always review the [release notes](#) and any [upgrade-specific release notes](#).

[Cloudera Navigator Data Encryption](#) is a data-at-rest encryption and key management suite that includes [Cloudera Navigator Encrypt](#), [Cloudera Navigator Key Trustee Server](#), among other components. These can optionally be upgraded during a Cloudera Manager or CDH upgrade, and can also be upgraded individually.

### Upgrading the Cloudera Navigator Data Management Component

**Cloudera Manager Required Role:** [Full Administrator](#)

The Cloudera Navigator Data Management component is upgraded automatically during the Cloudera Manager upgrade process. The component is not supported on Cloudera Express. It requires a Cloudera Enterprise license. See [Managing Licenses](#) for details about upgrading from trial licenses or Cloudera Express to Cloudera Enterprise.

#### Before Upgrading the Cluster

- Review the [Cloudera Navigator product compatibility matrix](#) and confirm the compatibility of the Cloudera Navigator version and the Cloudera Manager version for the upgrade.
- Review the [Cloudera Navigator release notes](#), especially [upgrade issues and limitations](#) before upgrading.
- Use the Purge capability provided in Cloudera Navigator 2.9 (and higher) to avoid out-of-memory errors during the upgrade process. See [Avoiding Out-of-Memory Errors During an Upgrade](#) on page 68 for details.

#### Avoiding Out-of-Memory Errors During an Upgrade

A very large Navigator Metadata Server storage directory size can cause out-of-memory (OOM) errors, or can cause the upgrade to take an extremely long time (10 hours or more). To reduce the chances of OOM errors and the time it takes to upgrade:

- Run the purge command before upgrading, or
- Temporarily increase available JVM memory to 31 GB (but not more). Increasing JVM memory decreases available OS RAM, which is also needed for the upgrade, so give the JVM 31 GB max for the upgrade.

See [Navigator Metadata Server Tuning](#) for more information.

#### Upgrading Cloudera Navigator

The upgrade process can take three to four hours, depending on the amount of data in the Navigator Metadata Server storage directory.

1. Before upgrading Cloudera Navigator 2.6 (or lower) to a higher release, you must:

- Stop the Navigator Metadata Server role.
- Back up the [Navigator Metadata Server storage directory](#).
- Ensure the Navigator Metadata Server has sufficient memory to complete the upgrade.
- For systems using an Oracle database, grant additional privileges to the `nav` (or appropriate user name for the Navigator Audit Server). Log in to the database using SQL\*Plus and run the following:

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

2. Upgrade Cloudera Manager, following the steps in [Cloudera Upgrade Overview](#) on page 7.
3. To upgrade from Cloudera Navigator 2.6 (and lower), [log in to the Cloudera Navigator console](#). The Upgrading Navigator page displays. Depending on the amount of data in the Navigator Metadata Server storage directory, the upgrade process can take 3–4 hours or longer.
4. When the upgrade is complete, click **Continue**. The Cloudera Navigator console displays.

## 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 74.



**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` file for RHEL 6 or RHEL 7 located in the [How can I use these extra packages?](#) section of the EPEL wiki 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-psycpg2*
```

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.

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.pgsql
$ chown keytrustee:keytrustee /var/lib/keytrustee/ktdbexport.pgsql
```

3. Start the Key Trustee Server database and import ktdbexport.pgsql:

```
$ 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.pgsql
```



**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/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
```

#### (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 Higher](#) on page 78.

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/
```

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](#):

```
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](#). 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 70 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:

- [Upgrading Cloudera Navigator Key Trustee Server 3.x to 5.4.x](#) on page 69
- [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or Higher](#) on page 78

### 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 Higher](#) on page 78 to upgrade to the version you want.

- Before upgrading Key Trustee Server, upgrade Cloudera Manager and CDH. See [Cloudera Upgrade Overview](#) on page 7 and [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 34. 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 85 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 4: 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/RPMS/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/RPMS/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/RPMS/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/RPMS/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 --downloadaddir=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](#):
  - a. Select **Packages** from the **SELECT DOWNLOAD TYPE** drop-down menu.
  - b. Select your operating system from the **SELECT AN OS** drop-down menu.
  - c. Click **DOWNLOAD NOW**.
  - d. 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 74 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 75.

### 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**)

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 Parcel Repository for Cloudera Manager](#).

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](#). 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 Higher](#) on page 78.

## Upgrading Cloudera Navigator Key Trustee Server 5.4.x or Higher

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 74.



**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 Parcel Repository for Cloudera Manager](#)

- [Creating and Using a Package Repository for Cloudera Manager](#) on page 89

## Upgrading Cloudera Navigator Key Trustee Server 5.4.x or Higher 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 Higher Using the Command Line \(CherryPy Web Server\)](#) on page 79 or [Upgrading Cloudera Navigator Key Trustee Server 5.4.x or Higher Using the Command Line \(Apache Web Server\)](#) on page 80 section.

1. Add your internal parcel repository to Cloudera Manager following the instructions in [Configuring Cloudera Manager Server Parcel Settings](#).
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](#).



**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 Higher 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 Higher Using the Command Line \(Apache Web Server\)](#) on page 80.

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` file for RHEL 6 or RHEL 7 located in the [How can I use these extra packages?](#) section of the EPEL wiki 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 90 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](#) 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
```

### Migrate Key Trustee Server to Cloudera Manager

Skip to [Migrating Unmanaged Key Trustee Server to Cloudera Manager](#) on page 83 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 Higher 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 `keytrusteed` 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/ktdbexport.pgsql
$ chown keytrustee:keytrustee /var/lib/keytrustee/ktdbexport.pgsql
```

3. Start the Key Trustee Server database and import `ktdbexport.pgsql`:

```
$ 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.pgsql
```



**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"
```

## Upgrading Cloudera Navigator Components

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` file for RHEL 6 or RHEL 7 located in the [How can I use these extra packages?](#) section of the EPEL wiki 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 90 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 83 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 (if you have not already done so) 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](#). 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 81 for more information.

For Apache web servers:

```
$ sudo service httpd stop
```

For CherryPy web servers:

```
$ sudo service keytrusteed stop
```

**4. 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
```

## Upgrading Cloudera Navigator Components

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 /usr/pgsql-9.3/bin/pg_ctl -D /var/lib/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 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 89.

### 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 90 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](#) 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 69 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 Parcel Repository for Cloudera Manager](#) if you are using parcels, or [Creating and Using a Package Repository for Cloudera Manager](#) on page 89 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 for Hosted Repositories](#) 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](#) 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

1. After [Setting Up an Internal Repository](#) on page 85, configure the Key Trustee KMS host to use the repository. See [Modifying Clients to Find the Repository](#) on page 90 for more information.
2. 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](#).
3. 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 Parcel Repository for Cloudera Manager](#)
- [Creating and Using a Package Repository for Cloudera Manager](#) on page 89

### Upgrading Navigator Encrypt (RHEL-Compatible)



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

- Only RHEL 7.3, 7.2 and 7.1 are supported. RHEL 7.0 is not supported.
- RHEL 7.1 is only supported with CDH 5.5 and higher.
- RHEL 7.2 is only supported with CDH 5.7 and higher.
- 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 90 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
```

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 90 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 90 for more information.

## Upgrading Cloudera Navigator Components

- **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 the product compatibility matrix for [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.



# 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 89
- [Creating a Temporary Remote Repository](#) on page 90

Once you have created a repository, go to [Modifying Clients to Find the Repository](#) on page 90.

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 89.

#### Installing Apache HTTPD

You may need to respond to some prompts to confirm you want to complete the installation.

OS	Command
RHEL	<code>[root@localhost yum.repos.d]\$ yum install httpd</code>
SLES	<code>[root@localhost zypp]\$ zypper install httpd</code>
Ubuntu or Debian	<code>[root@localhost apt]\$ apt-get install httpd</code>

#### Starting Apache HTTPD

OS	Command
RHEL	<code>[root@localhost tmp]\$ service httpd start</code>
SLES	<code>[root@localhost tmp]\$ service apache2 start</code>
Ubuntu or Debian	<code>[root@localhost tmp]\$ service apache2 start</code>

### 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](#)

## Creating and Using a Package Repository for Cloudera Manager

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](#)

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	Create files on client systems with the following information and format, where <i>hostname</i> is the name of the web server:  [myrepo] name=myrepo baseurl=http://hostname/cm/5 enabled=1 gpgcheck=0

OS	Command
	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.
<b>SLES</b>	Use the <code>zypper</code> utility to update client system repo information by issuing the following command:  <pre>\$ zypper addrepo http://hostname/cm alias</pre>
<b>Ubuntu or Debian</b>	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:  <pre>\$ cat /etc/apt/sources.list.d/my-private-cloudera-repo.list deb http://hostname/cm codename components</pre> <p>You can find the <i>codename</i> and <i>component</i> 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>

## 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 lower than 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 cluster functionality as it existed before upgrade. Cloudera recommends switching to a different supported database because of the limitations of a Derby-backed Hive metastore.

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](#).

### 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](#).

### 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.

#### 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 username 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.

## Database Considerations for Cloudera Manager Upgrades

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 5 Using Packages](#) on page 11.

## 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](#).

- 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):

OS	Command
<b>RHEL</b>	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/
<b>Ubuntu or Debian</b>	Replacedeb 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
<b>RHEL</b>	Copy <code>cloudera-manager.repo</code> to <code>/etc/yum.repos.d/</code> .
<b>SLES</b>	Copy <code>cloudera-manager.repo</code> to <code>/etc/zypp/repos.d/</code> .
<b>Ubuntu or Debian</b>	Copy <code>cloudera.list</code> to <code>/etc/apt/sources.list.d/</code> .

- c. Run the following commands:

Operating System	Commands
<b>RHEL</b>	<code>\$ sudo yum downgrade 'cloudera-*</code>
<b>SLES</b>	<code>\$ sudo zypper clean --all</code> <code>\$ sudo zypper dup -r</code> <code>https://archive.cloudera.com/cm4/sles/11/x86_64/cm/4/</code>  To download from your own repository: <code>\$ sudo zypper clean --all</code> <code>\$ sudo zypper dup -r http://myhost.example.com/path_to_cm_repo</code>
<b>Ubuntu or Debian</b>	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 <code>apt-get install</code> 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
```

## Reverting a Failed Cloudera Manager Upgrade

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-daemo 5.0.5-1.cm505.p0.163~sq Provides daemons for monitoring Hadoop  
and related tools.  
ii cloudera-manager-serve 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 Unmanaged CDH Using the Command Line

This section provides instructions for upgrading CDH to the latest release, using the command line instead of Cloudera Manager. (A cluster you are not managing with Cloudera Manager is referred to as an **unmanaged** cluster.)



### Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.13.x. See [Cloudera Documentation](#) for information specific to other releases.

To proceed with the upgrade, choose one of the following sections:

### 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 in [Upgrading CDH and Managed Services Using Cloudera Manager](#) on page 34 to upgrade to the latest version of CDH 5 in a Cloudera Manager deployment.
- 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, use Cloudera Manager [to manage these services](#).



### Note: Running Services

Use the `service` command to start, stop, and restart CDH components, instead of 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 information and instructions:

## Upgrading Unmanaged CDH Using the Command Line

- [Before Upgrading to the Latest Release of CDH](#) on page 100
- [Upgrading from CDH 5.4.0 or Higher to the Latest Release](#) on page 101
- [Upgrading from a Release Lower than CDH 5.4.0 to the Latest Release](#) on page 114

### 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:** Make sure to read [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 from 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 101
  - [Upgrading from a Release Lower than CDH 5.4.0 to the Latest Release](#) on page 114

### Troubleshooting: Upgrading hadoop-kms from 5.2.x and 5.3.x Releases on SLES

This section describes issues that affect 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 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 (either option works):

```
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
```

You should now 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](#).

### Step 1: Prepare the cluster for the upgrade

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. 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
```

## Upgrading Unmanaged CDH Using the Command Line

3. Check each host to make sure that there are no processes running as the `hdfs`, `yarn`, `mapred` or `httpfs` users from root:

```
# ps -aef | 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
```

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).

## Upgrading Unmanaged CDH Using the Command Line

### 2. Install the RPM:

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

### 3. Update your system package index by running the following:

```
$ 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
Jessie	<a href="#">Jessie package</a>
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](#).



#### Important: Clean cached packages and headers to ensure that 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, 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 hosts at the same time.
- If you are using [HA for the NameNode](#), do not install `hadoop-hdfs-secondarynamenode`.

**Before installing 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. 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>	<code>\$ sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-jobtracker</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-jobtracker</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-jobtracker</code>
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode</code>
<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>
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode</code>
All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts, running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode</code>
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-client</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-client</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-client</code>

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>	<code>\$ sudo yum clean all; sudo yum install hadoop-yarn-resourcemanager</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-resourcemanager</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-yarn-resourcemanager</code>
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-namenode</code>
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode</code>
All cluster hosts except the Resource Manager (analogous to MRv1 TaskTrackers) running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>

Where to install	Install commands
<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 update; sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
One host in the cluster running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-client</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-client</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get 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.

## 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 RHEL-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.

#### 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 hosts 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](#).

## Upgrading Unmanaged CDH Using the Command Line

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.

## Step 6b: Start MapReduce with YARN

**Important:**

Make sure you are not trying to run MRv1 and YARN on the same set of hosts 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
```



**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](#)
- [Flume Installation](#)
- [HBase Installation](#)
- [HCatalog Installation](#)
- [Hive Installation](#)
- [HttpFS Installation](#)
- [Hue Installation](#)
- [Impala Installation](#)
- [KMS Installation and Upgrade](#)
- [Mahout Installation](#)
- [Oozie Installation](#)
- [Pig Installation](#)
- [Search Installation](#)
- [Sentry Installation](#)

## Upgrading Unmanaged CDH Using the Command Line

- [Snappy Installation](#)
- [Spark Installation](#)
- [Sqoop 1 Installation](#)
- [Sqoop 2 Installation](#)
- [Whirr Installation](#)
- [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 Lower 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 [Upgrading from CDH 5.4.0 or Higher to the Latest Release](#) on page 101.

### 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 99
- [Before Upgrading to the Latest Release of CDH](#) on page 100

#### 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. Run a `saveNamespace` operation:

```
$ sudo -u hdfs hdfs dfsadmin -saveNamespace
```

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 -aef | 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 5: Checking for the 1-click Repository**

Operating System	Command to Run	Results and Actions
RHEL-compatible	<code>rpm -q cdh5-repository</code>	<p>If the command returns, <code>cdh5-repository-1-0</code>, the 1-click repository is installed. Skip to <a href="#">Step 3</a>.</p> <p>If the command returns package <code>cdh5-repository</code> is not installed, go to the 1-click instructions for your OS:</p> <ul style="list-style-type: none"> <li>• <a href="#">RHEL</a></li> </ul>
Ubuntu and Debian	<code>dpkg -l   grep cdh5-repository</code>	

## Upgrading Unmanaged CDH Using the Command Line

Operating System	Command to Run	Results and Actions
		<ul style="list-style-type: none"><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 the following:

```
$ 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](#).



**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
Jessie	<a href="#">Jessie package</a>
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](#).



**Important:** Clean cached packages and headers to ensure that 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
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

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>
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-client</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-client</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-client</code>

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>	<code>\$ sudo yum clean all; sudo yum install hadoop-yarn-resourcemanager</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-resourcemanager</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-yarn-resourcemanager</code>
NameNode host running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-hdfs-namenode</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-namenode</code>
Secondary NameNode host (if used) running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode</code>

Where to install	Install commands
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-hdfs-secondarynamenode</code>
All cluster hosts except the Resource Manager (analogous to MRv1 TaskTrackers) running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce</code>
One host in the cluster running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
<i>Ubuntu or Debian</i>	<code>\$ sudo apt-get update; sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver</code>
All client hosts, running:	
<i>Red Hat/CentOS compatible</i>	<code>\$ sudo yum clean all; sudo yum install hadoop-client</code>
<i>SLES</i>	<code>\$ sudo zypper clean --all; sudo zypper install hadoop-client</code>
<i>Ubuntu or Debian</i>	<code>sudo apt-get 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.

## 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 RHEL-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 hosts 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
```

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:

- 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](#)).

### 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](#)
- [Flume Installation](#)
- [HBase Installation](#)
- [HCatalog Installation](#)
- [Hive Installation](#)
- [HttpFS Installation](#)
- [Hue Installation](#)
- [Impala Installation](#)
- [KMS Installation and Upgrade](#)
- [Mahout Installation](#)
- [Oozie Installation](#)
- [Pig Installation](#)
- [Search Installation](#)
- [Sentry Installation](#)

- [Snappy Installation](#)
- [Spark Installation](#)
- [Sqoop 1 Installation](#)
- [Sqoop 2 Installation](#)
- [Whirr Installation](#)
- [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.

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 have 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 that the following behavior continues until the upgrade is finalized:
  - 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/hdfs.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 121, 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.hdfs.server.namenode.NameNode:
Exception in namenode join
    java.io.IOException: File system image contains an old layout version -55.An
upgrade to version -59 is required.
    Please restart NameNode with the "-rollingUpgrade started" option if a rolling
upgrade is already
        started; or restart NameNode with the "-upgrade" option to start a new upgrade.

        at
org.apache.hadoop.hdfs.server.namenode.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.hdfs.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 114 through [Step 4: In an HA Deployment, Upgrade and Start the JournalNodes](#) on page 120; or [Step 1: Prepare the Cluster for the Upgrade](#) on page 114 through [Step 3: Upgrade the Packages on the Appropriate Hosts](#) on page 117 if this is not an HA deployment).
2. Starting with [Step 5: Upgrade the HDFS Metadata](#) on page 121, complete all the remaining steps through [Step 10: Finalize the HDFS Metadata Upgrade](#) on page 127.

# Upgrading Host Operating Systems in a CDH Cluster

This topic describes the steps to upgrade the operating system (OS) on hosts in an active CDH cluster.

## Prerequisites

1. If you are upgrading CDH or Cloudera Manager as well as the OS, upgrade the OS first.
2. Read the release notes of the specific CDH version to ensure that the new OS version is supported. In most cases, upgrading to a minor version of the OS is supported. Read the release note of the new OS release to check for new default settings or changes in behavior. (For example, Transparent Huge Pages is on by default in some versions of RHEL 6).
3. If any files have a replication factor lower than the global default, verify that bringing down any node does not make the file unavailable.

## Upgrading Hosts

For each host in the cluster, do the following:

1. If the host runs a DataNode service, determine whether it needs to be decommissioned. See [Deciding Whether to Decommission DataNodes](#) on page 130.
2. Stop all Hadoop-related services on the host.
3. Take the host offline (for example, switch to single-user mode or restart the host to boot off the network).
4. Upgrade the OS partition, leaving the data partitions (for example, `dfs.data.dir`) unchanged.
5. Bring the host back online.
6. If the host is decommissioned, recommission it.
7. Verify in the NameNode UI (or Cloudera Manager) that the host is healthy and all services are running.

## Upgrading Hosts With High Availability Enabled

If you have enabled high availability for the NameNode or JobTracker, follow this procedure:

1. Stop the backup NameNode or JobTracker.
2. Upgrade the OS partition.
3. Start the services and ensure that they are running properly.
4. Fail over to the backup NameNode or JobTracker.
5. Upgrade the primary NameNode or JobTracker.
6. Bring the primary NameNode or JobTracker back online.
7. Reverse the failover.

## Upgrading Hosts Without High Availability Enabled

If you have not enabled high availability, upgrading a primary host causes an outage for that service. The procedure to upgrade is the same as upgrading a secondary host, except that you must decommission and recommission the host. When upgrading hosts that are part of a ZooKeeper quorum, ensure that the majority of the quorum is available. Cloudera recommends that you upgrade only one host at a time.

## Deciding Whether to Decommission DataNodes

When a DataNode is decommissioned, the NameNode ensures that every block from the DataNode is still available across the cluster as specified by the replication factor. This procedure involves copying blocks off the DataNode in small batches. In cases where a DataNode has several thousand blocks, decommissioning takes several hours.

When a DataNode is turned off without being decommissioned:

- The NameNode marks the DataNode as dead after a default of 10m 30s (controlled by `dfs.heartbeat.interval` and `dfs.heartbeat.recheck.interval`).

- The NameNode schedules the missing replicas to be placed on other DataNodes.
- When the DataNode comes back online and reports to the NameNode, the NameNode schedules blocks to be copied to it while other nodes are decommissioned or when new files are written to HDFS.

If the OS upgrade procedure is quick (for example, under 30 mins per node), do not decommission the DataNode.

Speed up the decommissioning of a DataNode by increasing values for these properties:

- `dfs.max-repl-streams`: The number of simultaneous streams to copy data.
- `dfs.balance.bandwidthPerSec`: The maximum amount of bandwidth that each DataNode can utilize for balancing, in bytes per second.
- `dfs.namenode.replication.work.multiplier.per.iteration`: NameNode configuration requiring a restart, defaults to 2 but can be raised to 10 or higher.

This determines the total amount of block transfers to begin in parallel at a DataNode for replication, when such a command list is being sent over a DataNode heartbeat by the NameNode. The actual number is obtained by multiplying this value by the total number of live nodes in the cluster. The result number is the number of blocks to transfer immediately, per DataNode heartbeat.

For more information, see [Decommissioning and Recommissioning Hosts](#).