cloudera®

Cloudera Installation

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

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

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

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

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

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

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

Release Information

Version: Cloudera Enterprise 5.12.x
Date: February 3, 2021

Table of Contents

Cloudera Installation Guide....11

Configuration Requirements for Cloudera Manager, Cloudera Navigator, and CDH 5....12

Permission Requirements for Package-based Installations and Upgrades of CDH....12
Cluster Hosts and Role Assignments....14
CDH Cluster Hosts and Role Assignments....14
Allocating Hosts for Key Trustee Server and Key Trustee KMS....16
Required Tomcat Directories....17
Ports....17
Ports Used by Cloudera Manager and Cloudera Navigator....17
Ports Used by Cloudera Navigator Encryption....21
Ports Used by Components of CDH 5....21
Ports Used by Impala....28
Ports Used by Cloudera Search....29
Ports Used by DistCp....29
Ports Used by Third-Party Components....30
Ports Used by Apache Flume and Apache Solr....31

Managing Software Installation Using Cloudera Manager....32

Parcels....32
Advantages of Parcels....33
Parcel Life Cycle....34
Parcel Locations....34
Managing Parcels....34
Viewing Parcel Usage....37
Parcel Configuration Settings....40
Creating Virtual Images of Cluster Hosts....42
Creating a Pre-Deployed Cloudera Manager Host....42
Instantiating a Cloudera Manager Image....43
Creating a Pre-Deployed Worker Host....43
Instantiating a Worker Host....45
Migrating from Packages to Parcels....45
Migrating from Parcels to Packages....47
Install CDH and Managed Service Packages....47
Deactivate Parcels....53

Restart the Cluster.....53
Remove and Delete Parcels.....53

Installing Cloudera Manager and CDH.....54
Cloudera Manager Deployment.....54
Cloudera Manager Installation Phases.....55
Cloudera Manager Installation Software.....56
Unmanaged Deployment.....57
Java Development Kit Installation.....58
Installing the Oracle JDK.....58
Configuring Single User Mode.....59
Limitations.....59
Using a Non-default Single User.....59
Configuration Steps Before Starting Cloudera Manager Agents in Installation Paths B and C.....60
Configuration Steps Before Running the Installation Wizard.....60
Configuration Steps Before Starting the Installation Wizard in Installation Paths B and C.....61
Configuration Steps While Running the Installation Wizard.....61
Configuration for Secure Clusters.....62
Controlling Access to sudo Commands.....62
Cloudera Manager and Managed Service Datastores.....68
Required Databases.....69
Setting up the Cloudera Manager Server Database.....70
External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server.....73
External Databases for Hue.....73
Embedded PostgreSQL Database.....73
External PostgreSQL Database.....76
MariaDB Database.....82
MySQL Database.....89
Oracle Database.....97
Configuring an External Database for Oozie.....107
Configuring an External Database for Sqoop.....110
Backing Up Databases.....111
Data Storage for Monitoring Data.....113
Storage Space Planning for Cloudera Manager.....116
Installation Path A - Automated Installation by Cloudera Manager (Non-Production Mode).....127
Before You Begin.....128
Download and Run the Cloudera Manager Server Installer.....129
Start and Log into the Cloudera Manager Admin Console.....130
Use the Cloudera Manager Wizard for Software Installation and Configuration.....130
Configure Database Settings.....135
Review Configuration Changes and Start Services.....135
Change the Default Administrator Password.....136

Configure Oozie Data Purge Settings....136
Test the Installation....136
Installation Path B - Installation Using Cloudera Manager Parcels or Packages....136
Before You Begin....137
Establish Your Cloudera Manager Repository Strategy....137
Install Cloudera Manager Server Software....138
(Optional) Manually Install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Service Packages....139
Start the Cloudera Manager Server....141
Start and Log into the Cloudera Manager Admin Console....141
Choose Cloudera Manager Edition....141
Choose Cloudera Manager Hosts....142
Choose the Software Installation Type and Install Software....143
Add Services....144
Configure Database Settings....146
Review Configuration Changes and Start Services....146
Change the Default Administrator Password....146
Configure Oozie Data Purge Settings....146
Test the Installation....146
(Optional) Manually Install CDH and Managed Service Packages....146
Installation Path C - Manual Installation Using Cloudera Manager Tarballs....149
Before You Begin....150
Install the Cloudera Manager Server and Agents....150
Create Parcel Directories....152
Start the Cloudera Manager Server....153
Start the Cloudera Manager Agents....154
Install Package Dependencies....155
Start and Log into the Cloudera Manager Admin Console....157
Choose Cloudera Manager Edition....157
Choose Cloudera Manager Hosts....158
Install CDH and Managed Service Software....158
Add Services....159
Configure Database Settings....160
Review Configuration Changes and Start Services....160
(Optional) Change the Cloudera Manager User....161
Change the Default Administrator Password....161
Configure Oozie Data Purge Settings....161
Test the Installation....161
Installing Impala....161
Installing Kudu....162
Install Kudu Using Parcels....162
Install Kudu Using Packages....163
Verify the Installation....164
Installing Cloudera Search....164
Deploying Cloudera Search....164

Installing Spark....164
Installing the GPL Extras Parcel....165
Understanding Custom Installation Solutions....166
Understanding Parcels....167
Understanding Package Management....167
Creating and Using a Parcel Repository for Cloudera Manager....168
Creating and Using a Package Repository for Cloudera Manager....171
Configuring a Custom Java Home Location....173
Installing Lower Versions of Cloudera Manager 5....174
Creating a CDH Cluster Using a Cloudera Manager Template....189
Deploying Clients....195
Testing the Installation....195
Checking Host Heartbeats....196
Running a MapReduce Job....196
Testing with Hue....196
Uninstalling Cloudera Manager and Managed Software....196
Uninstalling Cloudera Manager and Managed Software....196
Uninstalling a CDH Component From a Single Host....200
Installing the Cloudera Navigator Data Management Component....201
Installing Cloudera Navigator Key Trustee Server....203
Prerequisites....203
Setting Up an Internal Repository....203
Installing Key Trustee Server....204
Securing Key Trustee Server Host....206
Leveraging Native Processor Instruction Sets....207
Initializing Key Trustee Server....208
Installing Cloudera Navigator Key HSM....208
Prerequisites....208
Setting Up an Internal Repository....208
Installing Navigator Key HSM....208
Installing Key Trustee KMS....209
Setting Up an Internal Repository....209
Installing Key Trustee KMS Using Parcels....209
Installing Key Trustee KMS Using Packages....210
Post-Installation Configuration....210
Installing Navigator HSM KMS Backed by Thales HSM 210
Client Prerequisites....210
Setting Up an Internal Repository....211
Installing Navigator HSM KMS Backed by Thales HSM Using Parcels....211
Installing Navigator HSM KMS Backed by Thales HSM Using Packages....211
Installing Navigator HSM KMS Backed by Luna HSM 212
Client Prerequisites....212
Setting Up an Internal Repository....212

Installing Navigator HSM KMS Backed by Luna HSM Using Parcels 212
Installing Navigator HSM KMS Backed by Luna HSM Using Packages 212
Post-Installation Configuration 213
Installing Cloudera Navigator Encrypt 213
Prerequisites 213
Setting Up an Internal Repository 213
Installing Navigator Encrypt (RHEL-Compatible) 213
Installing Navigator Encrypt (SLES) 215
Installing Navigator Encrypt (Debian or Ubuntu) 216
Post Installation 217
AES-NI and RDRAND 217
Setting Up TLS for Navigator Encrypt Clients 217
Entropy Requirements 218
Uninstalling and Reinstalling Navigator Encrypt 219

Installing and Deploying CDH Using the Command Line 221
Before You Install CDH 5 on a Cluster 221
Scheduler Defaults 221
High Availability 221
Creating a Local Yum Repository 222
Installing the Latest CDH 5 Release 223
CDH 5 Installation Options 223
Before You Begin Installing CDH 5 Manually 223
Steps to Install CDH 5 Manually 224
Installing an Earlier CDH 5 Release 232
Downloading and Installing an Earlier Release 232
CDH 5 and MapReduce 235
YARN (MRv2) 235
Migrating from MapReduce (MRv1) to MapReduce (MRv2) 236
Introduction 236
Terminology and Architecture 236
For MapReduce Programmers: Writing and Running Jobs 236
For Administrators: Configuring and Running MRv2 Clusters 238
Web UI 243
Summary of Configuration Changes 243
Deploying CDH 5 on a Cluster 248
Configuring Dependencies Before Deploying CDH on a Cluster 248
Deploying HDFS on a Cluster 252
Deploying MapReduce v2 (YARN) on a Cluster 263
Deploying MapReduce v1 (MRv1) on a Cluster 269
Configuring Hadoop Daemons to Run at Startup 272
Installing CDH 5 Components 273
CDH 5 Components 273

Crunch Installation.....273
Flume Installation.....275
HBase Installation.....282
HCatalog Installation.....312
Impala Installation.....317
Hive Installation.....331
HttpFS Installation.....338
Hue Installation.....341
KMS Installation and Upgrade.....362
Kudu Installation.....364
Mahout Installation.....368
Oozie Installation.....370
Pig Installation.....386
Search Installation.....389
Sentry Installation.....395
Snappy Installation.....396
Spark Installation.....397
Sqoop 1 Installation.....398
Sqoop 2 Installation.....403
Whirr Installation.....411
ZooKeeper Installation.....415
Building RPMs from CDH Source RPMs.....419
Prerequisites.....419
Setting Up an Environment for Building RPMs.....420
Building an RPM.....420
Apache and Third-Party Licenses.....420
Apache License.....420
Third-Party Licenses.....420
Uninstalling CDH Components.....420
Uninstalling from Red Hat, CentOS, and Similar Systems.....421
Uninstalling from Debian and Ubuntu.....422
Uninstalling from SLES.....422
Additional clean-up.....423
Viewing the Apache Hadoop Documentation.....424

Troubleshooting Installation and Upgrade Problems.....425

The Cloudera Manager Server fails to start after upgrade.....425
Possible Reasons.....425
Possible Solutions.....425
Navigator HSM KMS Backed by Thales HSM installation fails.....425
Possible Reasons.....425
Possible Solutions.....425
Failed to start server reported by cloudera-manager-installer.bin.....425

Possible Reasons....425
Possible Solutions....425
Installation interrupted and installer does not restart....426
Possible Reasons....426
Possible Solutions....426
Cloudera Manager Server fails to start with MySQL....426
Possible Reasons....426
Possible Solutions....426
Agents fail to connect to Server....426
Possible Reasons....426
Possible Solutions....426
Cluster hosts do not appear....426
Possible Reasons....426
Possible Solutions....426
"Access denied" in install or update wizard....427
Possible Reasons....427
Possible Solutions....427
Databases fail to start....427
Possible Reasons....427
Possible Solutions....427
Cannot start services after upgrade....427
Possible Reasons....427
Possible Solutions....427
Cloudera services fail to start....428
Possible Reasons....428
Possible Solutions....428
Activity Monitor displays a status of **BAD**....428
Possible Reasons....428
Possible Solutions....428
Activity Monitor fails to start....428
Possible Reasons....428
Possible Solutions....428
Attempts to reinstall lower version of Cloudera Manager fail....428
Possible Reasons....428
Possible Solutions....429
Create Hive Metastore Database Tables command fails....429
Possible Reasons....429
Possible Solutions....429
HDFS DataNodes fail to start....429
Possible Reasons....429
Possible Solutions....429
Create Hive Metastore Database Tables command fails....431
Possible Reasons....432

Possible Solutions.....*432*
Oracle invalid identifier.....432
Possible Reasons.....*432*
Possible Solutions.....*432*

Appendix: Apache License, Version 2.0.....433

Cloudera Installation Guide

This guide provides instructions for installing Cloudera software, including Cloudera Manager, CDH, and other managed services, in a production environment.

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

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

For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

This guide includes the following sections:

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

This section describes configuration requirements for Cloudera Manager, Cloudera Navigator, and CDH 5. See also Version and Download Information and CDH 5 and Cloudera Manager 5 Requirements and Supported Versions.

Permission Requirements for Package-based Installations and Upgrades of CDH

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

Permission Requirements for Package-Based CDH Installation with Cloudera Manager

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

Table 1: Permission Requirements with Cloudera Manager

Task	Permissions Required
Install Cloudera Manager (using `cloudera-manager-installer.bin`)	root or sudo access on a single host
Manually start/stop/restart the Cloudera Manager Server (that is, log onto the host running Cloudera Manager and execute: `service cloudera-scm-server` *`action`*)	root or sudo
Run Cloudera Manager Server.	`cloudera-scm`
Install CDH components through Cloudera Manager.	One of the following, configured during initial installation of Cloudera Manager: • Direct access to root user using the root password. • Direct access to root user using a SSH key file. • Passwordless sudo access for a specific user. This is the same requirement as the installation of CDH components on individual hosts, which is a requirement of the UNIX system in general. You *cannot* use another system (such as PowerBroker) that provides root/sudo privileges.
Install the Cloudera Manager Agent through Cloudera Manager.	One of the following, configured during initial installation of Cloudera Manager: • Direct access to root user using the root password. • Direct access to root user using a SSH key file. • Passwordless sudo access for a specific user. This is the same requirement as the installation of CDH components on individual hosts, which is a requirement of the UNIX system in general. You *cannot* use another system (such as PowerBroker) that provides root/sudo privileges.

Task	Permissions Required
Run the Cloudera Manager Agent.	If single user mode is not enabled, access to the root account during runtime, through one of the following scenarios: • During Cloudera Manager and CDH installation, the Agent is automatically started if installation is successful. It is then started using one of the following, as configured during the initial installation of Cloudera Manager: – Direct access to root user using the root password – Direct access to root user using a SSH key file – Passwordless sudo access for a specific user Using another system (such as PowerBroker) that provides root/sudo privileges is *not* acceptable. • Through automatic startup during system boot, using init.
Manually start/stop/restart the Cloudera Manager Agent process.	If single user mode is not enabled, root or sudo access. This permission requirement ensures that services managed by the Cloudera Manager Agent assume the appropriate user (that is, the HDFS service assumes the `hdfs` user) for correct privileges. Any action request for a CDH service managed within Cloudera Manager *does not* require root or sudo access, because the action is handled by the Cloudera Manager Agent, which is already running under the root user.

Permission Requirements for Package-Based CDH Installation without Cloudera Manager

Table 2: Permission Requirements without Cloudera Manager

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

`sudo` Commands Run by Cloudera Manager

The `sudo` commands are:

- `yum` (RHEL/CentOS/Oracle)
- `zypper` (SLES)
- `apt-get` (Debian/Ubuntu)
- `apt-key` (Debian/Ubuntu)
- `sed`
- `service`

- `/sbin/chkconfig` (RHEL/CentOS/Oracle)
- `/usr/sbin/update-rc.d` (Debian/Ubuntu)
- `id`
- `rm`
- `mv`
- `chown`
- `install`

Cluster Hosts and Role Assignments

This topic describes suggested role assignments for a CDH cluster managed by Cloudera Manager. The actual assignments you choose for your deployment can vary depending on the types and volume of work loads, the services deployed in your cluster, hardware resources, configuration, and other factors.

When you install CDH using the Cloudera Manager installation wizard, Cloudera Manager attempts to spread the roles among cluster hosts (except for roles assigned to Edge hosts) based on the resources available in the hosts. You can change these assignments on the **Customize Role Assignments** page that appears in the wizard. You can also change and add roles at a later time using Cloudera Manager. See Role Instances.

If your cluster uses data-at-rest encryption, see Allocating Hosts for Key Trustee Server and Key Trustee KMS on page 16.

> **Note:** For information about where to locate various databases that are required for Cloudera Manager and other services, see Cloudera Manager and Managed Service Datastores on page 68.

CDH Cluster Hosts and Role Assignments

The Table 3: Cluster Hosts and Role Assignments on page 14 table describes allocations for the following types of hosts:

- **Master hosts** run Hadoop master processes such as the HDFS NameNode and YARN Resource Manager.
- **Utility hosts** run other cluster processes that are not master processes such as Cloudera Manager and the Hive Metastore.
- **Edge hosts** are client access points for launching jobs in the cluster. The number of Edge hosts required varies depending on the type and size of the workloads.
- **Worker hosts** primarily run DataNodes and other distributed processes such as Impalad.

> **Important:** Cloudera recommends that you always enable high availability when CDH is used in a production environment.

Table 3: Cluster Hosts and Role Assignments

Cluster Size	Master Hosts	Utility Hosts	Edge Hosts	Worker Hosts
Very Small, without High Availability • Up to 10 worker hosts • High availability	**Master Host 1:** • NameNode • YARN ResourceManager • JobHistory Server • ZooKeeper • Impala StateStore	**One host for all Utility and Edge roles:** • Secondary NameNode • Cloudera Manager • Cloudera Manager Management Service • Hive Metastore • HiveServer2 • Impala Catalog • Hue		**3 - 10 Worker Hosts:** • DataNode • NodeManager • Impalad

Cluster Size	Master Hosts	Utility Hosts	Edge Hosts	Worker Hosts
not enabled		• Oozie • Flume • Gateway configuration		
Small, with High Availability • Up to 20 worker hosts • High availability enabled	**Master Host 1:** • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • JobHistory Server **Master Host 2:** • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • Impala StateStore	**One host for all Utility and Edge roles:** • Cloudera Manager • Cloudera Manager Management Service • Hive Metastore • HiveServer2 • Impala Catalog • Hue • Oozie • Flume • Gateway configuration • ZooKeeper (requires dedicated disk) • JournalNode (requires dedicated disk)		**3 - 20 Worker Hosts:** • DataNode • NodeManager • Impalad
Medium, with High Availability • Up to 200 worker hosts • High availability enabled	**Master Host 1:** • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper **Master Host 2:** • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper **Master Host 3:** • ZooKeeper • JournalNode • JobHistory Server • Impala StateStore	**Utility Host 1:** • Cloudera Manager **Utility Host 2:** • Cloudera Manager Management Service • Hive Metastore • Catalog Server • Oozie	**One or more Edge Hosts:** • Hue • HiveServer2 • Flume • Gateway configuration	**50 - 200 Worker nodes:** • DataNode • NodeManager • Impalad
Large, with High Availability • Up to 500 worker hosts • High availability enabled	**Master Host 1:** • NameNode • JournalNode • FailoverController • ZooKeeper **Master Host 2:** • NameNode • JournalNode • FailoverController	**Utility Host 1:** • Cloudera Manager **Utility Host 2:** • Cloudera Manager Management Service • Hive Metastore	**One or more Edge Hosts:** • Hue • HiveServer2 • Flume • Gateway configuration	**200 - 500 Worker Hosts:** • DataNode • NodeManager • Impalad

Cluster Size	Master Hosts	Utility Hosts	Edge Hosts	Worker Hosts
	• ZooKeeper **Master Host 3:** • YARN ResourceManager • ZooKeeper • JournalNode **Master Host 4:** • YARN ResourceManager • ZooKeeper • JournalNode **Master Host 5:** • JobHistory Server • Impala StateStore • ZooKeeper • JournalNode	• Catalog Server • Oozie		
Extra Large, with High Availability • Up to 1000 worker hosts • High availability enabled	**Master Host 1:** • NameNode • JournalNode • FailoverController • ZooKeeper **Master Host 2:** • NameNode • JournalNode • FailoverController • ZooKeeper **Master Host 3:** • YARN ResourceManager • ZooKeeper • JournalNode **Master Host 4:** • YARN ResourceManager • ZooKeeper • JournalNode **Master Host 5:** • JobHistory Server • Impala StateStore • ZooKeeper • JournalNode	**Utility Host 1:** • Cloudera Manager Server **Utility Host 2:** • Service Monitor **Utility Host 3:** • Reports Manager **Utility Host 4:** • Cloudera Management Service Roles: (Host Monitor, Navigator, Alert Publisher) **Utility Host 5:** • Hive Metastore • Catalog Server • Oozie	**One or more Edge Hosts:** • Hue • HiveServer2 • Flume • Gateway configuration	**500 - 1000 Worker Hosts:** • DataNode • NodeManager • Impalad

Allocating Hosts for Key Trustee Server and Key Trustee KMS

If you are enabling data-at-rest encryption for a CDH cluster, Cloudera recommends that you isolate the Key Trustee Server from other enterprise data hub (EDH) services by deploying the Key Trustee Server on dedicated hosts in a separate cluster managed by Cloudera Manager. Cloudera also recommends deploying Key Trustee KMS on dedicated

hosts in the same cluster as the EDH services that require access to Key Trustee Server. This architecture helps users avoid having to restart the Key Trustee Server when restarting a cluster.

See Encrypting Data at Rest.

For production environments in general, or if you have enabled high availability for HDFS and are using data-at-rest encryption, Cloudera recommends that you enable high availability for Key Trustee Server and Key Trustee KMS.

See:

- Cloudera Navigator Key Trustee Server High Availability
- Enabling Key Trustee KMS High Availability

Required Tomcat Directories

Most directories used by Cloudera services are configurable. Services that use Tomcat – for example, HttpFS, Oozie, Solr, Sqoop, KMS – require the directory `/var/lib/<service-specific-directory>`. For example, the Sqoop service requires the directory `/var/lib/sqoop`.

Ports

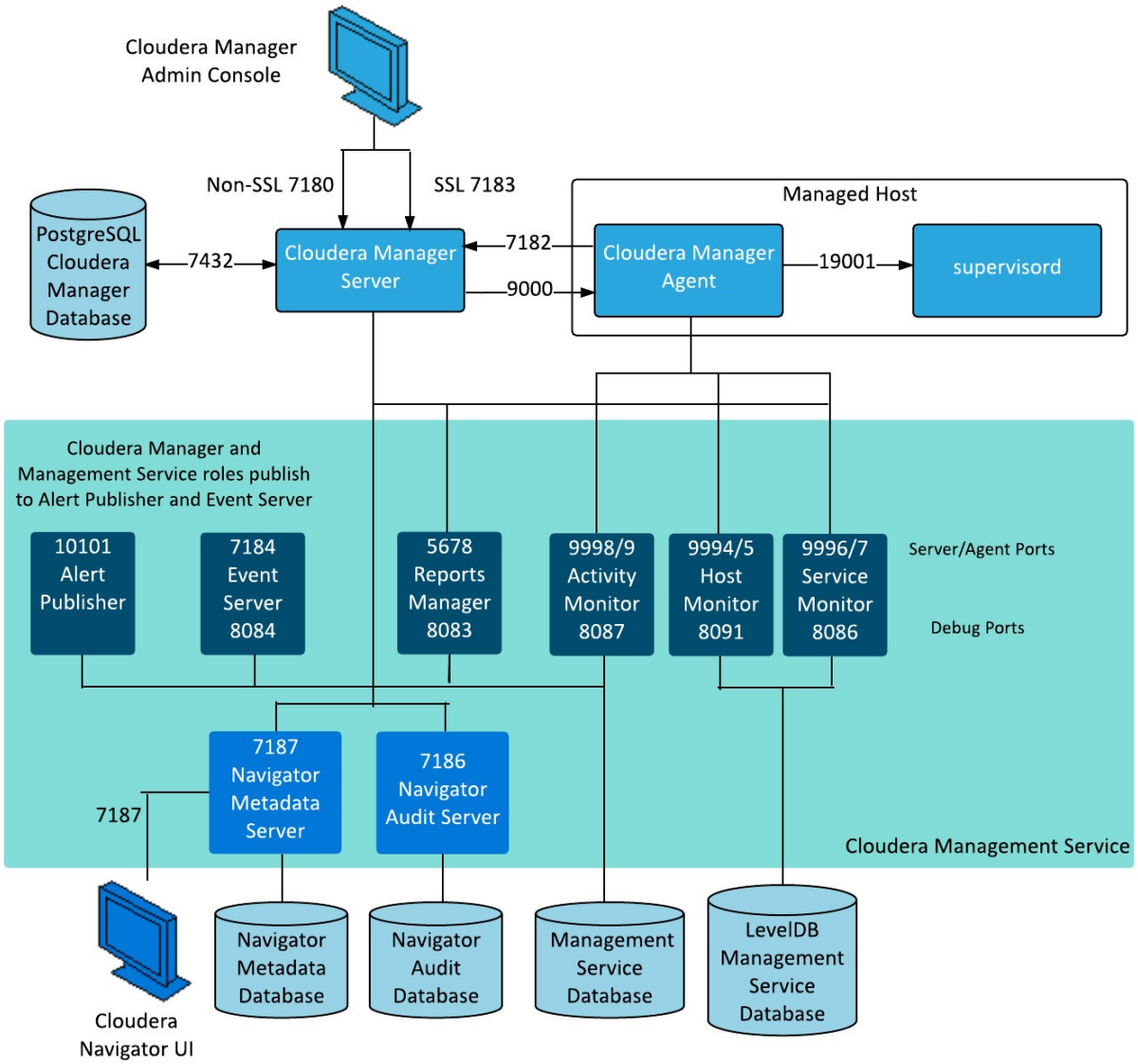
Cloudera Manager, CDH components, managed services, and third-party components use the ports listed in the tables that follow. Before you deploy Cloudera Manager, CDH, and managed services, and third-party components make sure these ports are open on each system. If you are using a firewall, such as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

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

Unless otherwise specified, the ports access requirement is unidirectional, meaning that inbound connections to the specified ports must be allowed. In most modern stateful firewalls, it is not necessary to create a separate rule for return traffic on a permitted session.

Ports Used by Cloudera Manager and Cloudera Navigator

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



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

Component	Service	Port	Access Requirement	Configuration	Comment
Cloudera Manager Server	HTTP (Web UI)	7180	External	Administration > Settings > Ports and Addresses	
	HTTPS (Web UI)	7183	External		Used for HTTPS on master, if enabled. HTTP is the default; port 7180 is open, but is redirected to port 7183.
	Avro (RPC)	7182	Internal		Used for Agent to Server heartbeats
	PostgreSQL database managed by	7432	Internal		The optional embedded PostgreSQL database used for storing configuration

Component	Service	Port	Access Requirement	Configuration	Comment
	cloudera-scm-server-db service				information for Cloudera Manager Server.
	Peer-to-peer parcel distribution	4443, 7190, 7191	Internal	Hosts > All Hosts > Configuration > P2P Parcel Distribution Port	Used to distribute parcels to cluster hosts during installation and upgrade operations.
Cloudera Manager Agent	HTTP (Debug)	9000	Internal	/etc/cloudera-scm-agent/config.ini	
	Internal supervisord	localhost: 19001	localhost		supervisord status and control port; used for communication between the Agent and supervisord; only open internally (on localhost)
Event Server	Listens for the publication of events.	7184	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	Listens for queries for events.	7185	Internal		
	HTTP (Debug)	8084	Internal		Allows access to debugging and diagnostic information
Alert Publisher	Internal API	10101	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
Service Monitor	HTTP (Debug)	8086	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	Listening for Agent messages (private protocol)	9997	Internal		
	Internal query API (Avro)	9996	Internal		
Activity Monitor	HTTP (Debug)	8087	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	Listening for Agent messages (private protocol)	9999	Internal		
	Internal query API (Avro)	9998	Internal		

Component	Service	Port	Access Requirement	Configuration	Comment
Host Monitor	HTTP (Debug)	8091	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	Listening for Agent messages (private protocol)	9995	Internal		
	Internal query API (Avro)	9994	Internal		
Reports Manager	Queries (Thrift)	5678	Internal	Cloudera Management Service > Configuration > Ports and Addresses	
	HTTP (Debug)	8083	Internal		
Cloudera Navigator				Cloudera Management Service > Configuration > Ports and Addresses	
Audit Server	HTTP	7186	Internal		
	HTTP (Debug)	8089	Internal		The port where Navigator Audit Server runs a debug web server. Set to -1 to disable debug server.
Metadata Server	HTTP (Web UI)	7187	External		
Task Tracker Plug-in (used for activity monitoring)	HTTP (Debug)	localhost: 4867	localhost		Used only on localhost interface by monitoring agent
Backup and Disaster Recovery	HTTP (Web UI)	7180	External	Administration > Settings > Ports and Addresses	Used for communication to peer (source) Cloudera Manager.
	HDFS NameNode	8020	External	HDFS > Configuration > Ports and Addresses > NameNode Port	HDFS and Hive/Impala replication: communication from destination HDFS and MapReduce hosts to source HDFS NameNode(s). Hive/Impala Replication: communication from source Hive hosts to destination HDFS NameNode(s).
	HDFS DataNode	50010	External	HDFS > Configuration > Ports and Addresses > DataNode Transceiver Port	HDFS and Hive/Impala replication: communication from destination HDFS and MapReduce hosts to source HDFS DataNode(s). Hive/Impala Replication: communication from source

Component	Service	Port	Access Requirement	Configuration	Comment
					Hive hosts to destination HDFS DataNode(s).

Ports Used by Cloudera Navigator Encryption

All ports listed are TCP.

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

Ports Used by Components of CDH 5

All ports listed are TCP.

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

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
					or `dfs.namenode.http-address`	is deprecated (but still works)

	NameNode	Secure	50470	External	`dfs.https.address`
					or
					`dfs.namenode.https-address`
						`dfs.https.address`
						is deprecated (but still works)

	Secondary NameNode		50090	Internal	`dfs.secondary.http.address`
					or
					`dfs.namenode.secondary.http-address`
						`dfs.secondary.http.address`
						is deprecated (but still works)

	Secondary NameNode	Secure	50495	Internal	`dfs.secondary.https.address`	
	JournalNode		8485	Internal	`dfs.namenode.shared.edits.dir`	
	JournalNode		8480	Internal	`dfs.journalnode.http-address`	
	JournalNode		8481	Internal	`dfs.journalnode.https-address`	
	Failover Controller		8019	Internal		Used for NameNode HA

	NFS gateway		2049	External		`nfs`
						port (
						`nfs3.server.port`
						)

	NFS gateway		4242	External		`mountd`
						port (
						`nfs3.mountd.port`

	NFS gateway		111	External		`portmapper`
						or
						`rpcbind`
						port

	NFS gateway		50079	External	`nfs.http.port`	CDH 5.4.0 and higher. The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
	NFS gateway	Secure	50579	External	`nfs.https.port`	CDH 5.4.0 and higher. The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	HttpFS		14000	External		
	HttpFS		14001	External		
Hadoop MapReduce (MRv1)	JobTracker		8021	External	`mapred.job.tracker`	
	JobTracker		8023	External	`mapred.ha.job.tracker`	High availability service protocol port for the JobTracker. The JobTracker listens on a separate port for HA operations.
	JobTracker		50030	External	`mapred.job.tracker.http.address`	
	JobTracker	Thrift Plugin	9290	Internal	`jobtracker.thrift.address`	Required by Hue and Cloudera Manager Activity Monitor
	TaskTracker		50060	External	`mapred.task.tracker.http.address`	
	TaskTracker		0	Localhost	`mapred.task.tracker.report.address`	Communicating with child (umbilical)
	Failover Controller		8018	Internal	`mapred.ha.zkfc.port`	Used for JobTracker HA
Hadoop YARN (MRv2)	ResourceManager		8032	External	`yarn.resourcemanager.address`	
	ResourceManager		8030	Internal	`yarn.resourcemanager.scheduler.address`	
	ResourceManager		8031	Internal	`yarn.resourcemanager.resource-tracker.address`	
	ResourceManager		8033	External	`yarn.resourcemanager.admin.address`	
	ResourceManager		8088	External	`yarn.resourcemanager.webapp.address`	
	ResourceManager		8090	External	`yarn.resourcemanager.webapp.https.address`	
	NodeManager		8040	Internal	`yarn.nodemanager.localizer.address`	

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	NodeManager		8041	Internal	`yarn.nodemanager.address`	
	NodeManager		8042	External	`yarn.nodemanager.webapp.address`	
	NodeManager		8044	External	`yarn.nodemanager.webapp.https.address`	
	JobHistory Server		10020	Internal	`mapreduce.jobhistory.address`	
	JobHistory Server		10033	Internal	`mapreduce.jobhistory.admin.address`	
	Shuffle HTTP		13562	Internal	`mapreduce.shuffle.port`	
	JobHistory Server		19888	External	`mapreduce.jobhistory.webapp.address`	
	JobHistory Server		19890	External	`mapreduce.jobhistory.webapp.https.address`	
	ApplicationMaster			External		The ApplicationMaster serves an HTTP service using an ephemeral port that cannot be restricted. This port is never accessed directly from outside the cluster by clients. All requests to the ApplicationMaster web server is routed using the YARN ResourceManager (proxy service). Locking down access to ephemeral port ranges within the cluster's network might restrict your access to the ApplicationMaster UI and its logs, along with the ability to look at running applications.
Flume	Flume Agent		41414	External		
Hadoop KMS	Key Management Server		16000	External	`kms_http_port`	CDH 5.2.1 and higher. Applies to both Java KeyStore KMS and Key Trustee KMS.
	Key Management Server		16001	Localhost	`kms_admin_port`	CDH 5.2.1 and higher. Applies to both Java

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
						KeyStore KMS and Key Trustee KMS.
HBase	Master		60000	External	`hbase.master.port`	IPC
	Master		60010	External	`hbase.master.info.port`	HTTP
	RegionServer		60020	External	`hbase.regionserver.port`	IPC
	RegionServer		60030	External	`hbase.regionserver.info.port`	HTTP
	HQuorumPeer		2181	Internal	`hbase.zookeeper.property.clientPort`	HBase-managed ZooKeeper mode
	HQuorumPeer		2888	Internal	`hbase.zookeeper.peerport`	HBase-managed ZooKeeper mode
	HQuorumPeer		3888	Internal	`hbase.zookeeper.leaderport`	HBase-managed ZooKeeper mode
	REST	Non-Cloudera Manager-managed	8080	External	`hbase.rest.port`	The default REST port in HBase is 8080. Because this is a commonly used port, Cloudera Manager sets the default to 20550 instead.
	REST	Cloudera Manager-managed	20550	External	`hbase.rest.port`	The default REST port in HBase is 8080. Because this is a commonly used port, Cloudera Manager sets the default to 20550 instead.
	REST UI		8085	External		
	Thrift Server	Thrift Server	9090	External	Pass `-p <port>` on CLI	
	Thrift Server		9095	External		
		Avro server	9090	External	Pass `--port <port>` on CLI	
	hbase-solr-indexer	Lily Indexer	11060	External		
Hive	Metastore		9083	External		
	HiveServer2		10000	External	`hive.server2.thrift.port`	The Beeline command interpreter requires that you specify this port on the command line.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
						If you use Oracle database, you must manually reserve this port. For more information, see Reserving Ports for HiveServer 2 on page 97.
	HiveServer2 Web User Interface (UI)		10002	External	`hive.server2.webui.port` in `hive-site.xml`	
	WebHCat Server		50111	External	`templeton.port`	
Hue	Server		8888	External		
	Load Balancer		8889	External		
Oozie	Oozie Server		11000	External	`OOZIE_HTTP_PORT` in `oozie-env.sh`	HTTP
	Oozie Server	SSL	11443	External		HTTPS
	Oozie Server		11001	localhost	`OOZIE_ADMIN_PORT` in `oozie-env.sh`	Shutdown port
Sentry	Sentry Server		8038	External	`sentry.service.server.rpc-port`	
	Sentry Server		51000	External	`sentry.service.web.port`	
Solr	Solr Server		8983	External		HTTP port for all Solr-specific actions, update/query.
	Solr Server		8984	Internal		Solr administrative use.
	Solr Server		8985	External		HTTPS port for all Solr-specific actions, update/query.
Spark	Default Master RPC port		7077	External		
	Default Worker RPC port		7078	External		
	Default Master web UI port		18080	External		
	Default Worker web UI port		18081	External		
	History Server		18088	External	`history.port`	

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
	Shuffle service		7337	Internal		
Sqoop	Metastore		16000	External	`sqoop.metastore.server.port`	
Sqoop 2	Sqoop 2 server		8005	Localhost	`SQOOP_ADMIN_PORT` environment variable	
	Sqoop 2 server		12000	External		
	Sqoop 2		12001	External		Admin port
ZooKeeper	Server (with CDH 5 or Cloudera Manager 5)		2181	External	clientPort	Client port
	Server (with CDH 5 only)		2888	Internal	`X in server.N =host:X:Y`	Peer
	Server (with CDH 5 only)		3888	Internal	`X in server.N =host:X:Y`	Peer
	Server (with CDH 5 and Cloudera Manager 5)		3181	Internal	`X in server.N =host:X:Y`	Peer
	Server (with CDH 5 and Cloudera Manager 5)		4181	Internal	`X in server.N =host:X:Y`	Peer
	ZooKeeper JMX port		9010	Internal		ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must do *one* of the following: • Open up all ports when the connection originates from the Cloudera Manager Server • Do the following: 1. Open a non-ephemeral port (such as 9011) in the firewall. 2. Install Oracle Java 7u4 JDK or higher.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
						3. Add the port configuration to the advanced configuration snippet, for example: `-Dcom.sun.management.jmxremote.rmi.port=9011` 4. Restart ZooKeeper.

Ports Used by Impala

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

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

Component	Service	Port	Access Requirement	Comment
Impala Catalog Daemon	Catalog HTTP Server Port	25020	External	Catalog service web interface for administrators to monitor and troubleshoot. New in Impala 1.2 and higher.
Impala StateStore Daemon	StateStore Service Port	24000	Internal	Internal use only. The statestore daemon listens on this port for registration/unregistration requests.
Impala Catalog Daemon	Catalog Service Port	26000	Internal	Internal use only. The catalog service uses this port to communicate with the Impala daemons. New in Impala 1.2 and higher.
Impala Daemon	Llama Callback Port	28000	Internal	Internal use only. Impala daemons use to communicate with Llama. New in and higher.
Impala Llama ApplicationMaster	Llama Thrift Admin Port	15002	Internal	Internal use only. New in and higher.
Impala Llama ApplicationMaster	Llama Thrift Port	15000	Internal	Internal use only. New in and higher.
Impala Llama ApplicationMaster	Llama HTTP Port	15001	External	Llama service web interface for administrators to monitor and troubleshoot. New in and higher.

Ports Used by Cloudera Search

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

Ports Used by DistCp

All ports listed are TCP.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hadoop HDFS	NameNode		8020	External	`fs.default.name`	`fs.default.name`

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
					or `fs.defaultFS`	is deprecated (but still works)
	DataNode	Secure	1004	External	`dfs.datanode.address`	
	DataNode		50010	External	`dfs.datanode.address`	
WebHDFS	NameNode		50070	External	`dfs.http.address` or `dfs.namenode.http-address`	`dfs.http.address` is deprecated (but still works)
	DataNode	Secure	1006	External	`dfs.datanode.http.address`	
HttpFS	web		14000			

Ports Used by Third-Party Components

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
Ganglia	ganglia-gmond		8649	UDP/TCP	Internal		
	ganglia-web		80	TCP	External	Via Apache `httpd`	
Kerberos	KRB5 KDC Server	Secure	88	UDP/TCP	External	`kdc_ports` and `kdc_tcp_ports` in either the `[kdcdefaults]` or `[realms]` sections of `kdc.conf`	By default only UDP
	KRB5 Admin Server	Secure	749	TCP	External	`kadmind_port` in the `[realms]` section of `kdc.conf`	
	kpasswd		464	UDP/TCP	External		
SSH	ssh		22	TCP	External		
PostgreSQL			5432	TCP	Internal		
MariaDB			3306	TCP	Internal		

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

Ports Used by Apache Flume and Apache Solr

Ports used by Apache Flume to communicate with Apache Solr might vary, depending on your configuration and whether you are using security (for example, SSL). A typical cluster set up with Flume writing to HDFS and Solr sinks uses the ports listed below:

Port	Description
41414	HTTP Port - port on which the Flume web server listens for requests. Flume uses this port continuously.
7184	Cloudera Manager Event Publish Port - port on which the Event Server listens for the publication of events. Flume uses this port continuously.
8020	NameNode Port, used by the HDFS sink.
8983	Solr HTTP Port, used by MorphlineSolrSink.
2181	ZooKeeper client port, used by MorphlineSolrSink.

Managing Software Installation Using Cloudera Manager

A major function of Cloudera Manager is to install CDH and managed service software. Cloudera Manager installs software for new deployments and to upgrade existing deployments. Cloudera Manager supports two software distribution formats: packages and parcels.

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

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

- Parcels are self-contained and installed in a versioned directory, which means that multiple versions of a given parcel can be installed side-by-side. You can then designate one of these installed versions as the active one. With packages, only one package can be installed at a time so there is no distinction between what is installed and what is active.
- You can install parcels at any location in the filesystem. They are installed by default in `/opt/cloudera/parcels`. In contrast, packages are installed in `/usr/lib`.
- When you install from the Parcels page, Cloudera Manager automatically downloads, distributes, and activates the correct parcel for the operating system running on each host in the cluster. All CDH hosts that make up a logical cluster need to run on the same major OS release to be covered by Cloudera Support. Cloudera Manager needs to run on the same OS release as one of the CDH clusters it manages, to be covered by Cloudera Support. The risk of issues caused by running different minor OS releases is considered lower than the risk of running different major OS releases. Cloudera recommends running the same minor release cross-cluster, because it simplifies issue tracking and supportability. You can, however, use RHEL/Centos 7.2 as the operating system for gateway hosts. See Operating System Support for Gateway Hosts (CDH 5.11 and higher only).

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

Parcels

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

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

- Parcels are self-contained and installed in a versioned directory, which means that multiple versions of a given parcel can be installed side-by-side. You can then designate one of these installed versions as the active one. With packages, only one package can be installed at a time so there is no distinction between what is installed and what is active.
- You can install parcels at any location in the filesystem. They are installed by default in `/opt/cloudera/parcels`. In contrast, packages are installed in `/usr/lib`.
- When you install from the Parcels page, Cloudera Manager automatically downloads, distributes, and activates the correct parcel for the operating system running on each host in the cluster. All CDH hosts that make up a logical cluster need to run on the same major OS release to be covered by Cloudera Support. Cloudera Manager needs to run on the same OS release as one of the CDH clusters it manages, to be covered by Cloudera Support. The risk of issues caused by running different minor OS releases is considered lower than the risk of running different major OS releases. Cloudera recommends running the same minor release cross-cluster, because it simplifies issue tracking and supportability. You can, however, use RHEL/Centos 7.2 as the operating system for gateway hosts. See Operating System Support for Gateway Hosts (CDH 5.11 and higher only).

Important: Cloudera Manager manages the parcels without the need for users to manipulate parcels in the filesystem. You can cause failures in your cluster if you perform any of the following unsupported actions:

- Installing parcels within custom RPM packages and saving them to the Cloudera Manager parcel directory.
- Downloading parcels and manually placing them in the Cloudera Manager parcel directory.

For detailed installation instructions using parcels, and other methods, see Installing Cloudera Manager and CDH on page 54.

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

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

Advantages of Parcels

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

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

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

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

- **Compatibility with other distribution tools** - Cloudera Manager works with other tools you use for download and distribution. For example, you can use Puppet. Or, you can download the parcel to Cloudera Manager Server manually if your cluster has no Internet connectivity and then have Cloudera Manager distribute the parcel to the cluster.

Parcel Life Cycle

To enable upgrades and additions with minimal disruption, parcels have following phases:

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

  If you are upgrading CDH or managed services when you activate a parcel, follow the instructions in Upgrading CDH and Managed Services Using Cloudera Manager to complete the upgrade.

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

Cloudera Manager detects when new parcels are available. You can configure Cloudera Manager to download and distribute parcels automatically. See Configuring Cloudera Manager Server Parcel Settings on page 40.

Parcel Locations

The default location for the local parcel directory on the Cloudera Manager Server is `/opt/cloudera/parcel-repo`. To change this location, follow the instructions in Configuring Cloudera Manager Server Parcel Settings on page 40.

The default location for the distributed parcels on managed hosts is `/opt/cloudera/parcels`. To change this location, set the `parcel_dir` property in `/etc/cloudera-scm-agent/config.ini` file of the Cloudera Manager Agent and restart the Cloudera Manager Agent or by following the instructions in Configuring the Host Parcel Directory on page 41.

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

Managing Parcels

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

Accessing the Parcels Page

Access the Parcels page by doing one of the following:

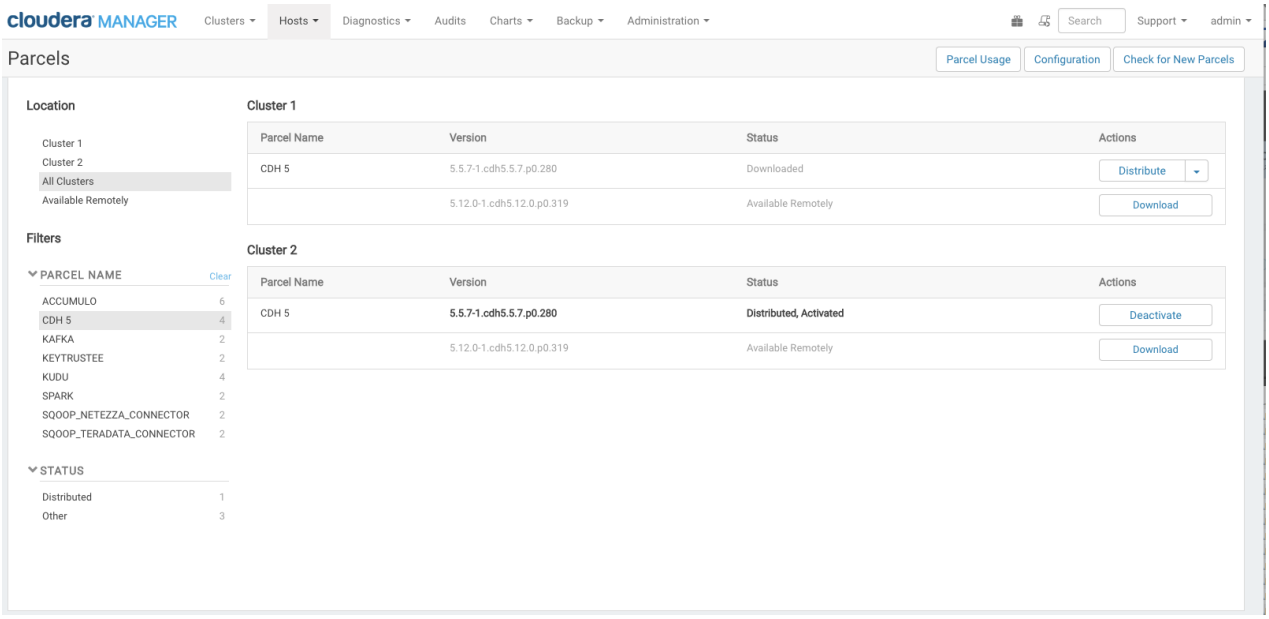
- Click the parcel indicator in the top navigation bar.
- Click the **Hosts** in the top navigation bar, then the **Parcels** tab.

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

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

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

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



Downloading a Parcel

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

   If the parcel you want is not shown here—for example, you want to upgrade to a version of CDH that is not the most current version—you can make additional remote parcel repositories available. You can also configure the location of the local parcel repository and other settings. See Parcel Configuration Settings on page 40.

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

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

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

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

Distributing a Parcel

Downloaded parcels can be distributed to the hosts in your cluster and made available for activation. Parcels are downloaded to the Cloudera Manager Server, so with multiple clusters, the downloaded parcels are shown as available to *all* clusters managed by the Cloudera Manager Server. However, you select distribution to a specific cluster's hosts on a cluster-by-cluster basis.

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

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

If you have a large number of hosts to which parcels must be distributed, you can control how many concurrent uploads Cloudera Manager performs. See Parcel Configuration Settings on page 40.

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

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

Activating a Parcel

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

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

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

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

the HDFS, YARN, Impala, and Spark services, the parcel page shows that parcel as **Still in use** until you restart these remaining services.

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

Deactivating a Parcel

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

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

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

Removing a Parcel

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

Deleting a Parcel

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

Changing the Parcel Directory

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

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

Troubleshooting

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

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

Viewing Parcel Usage

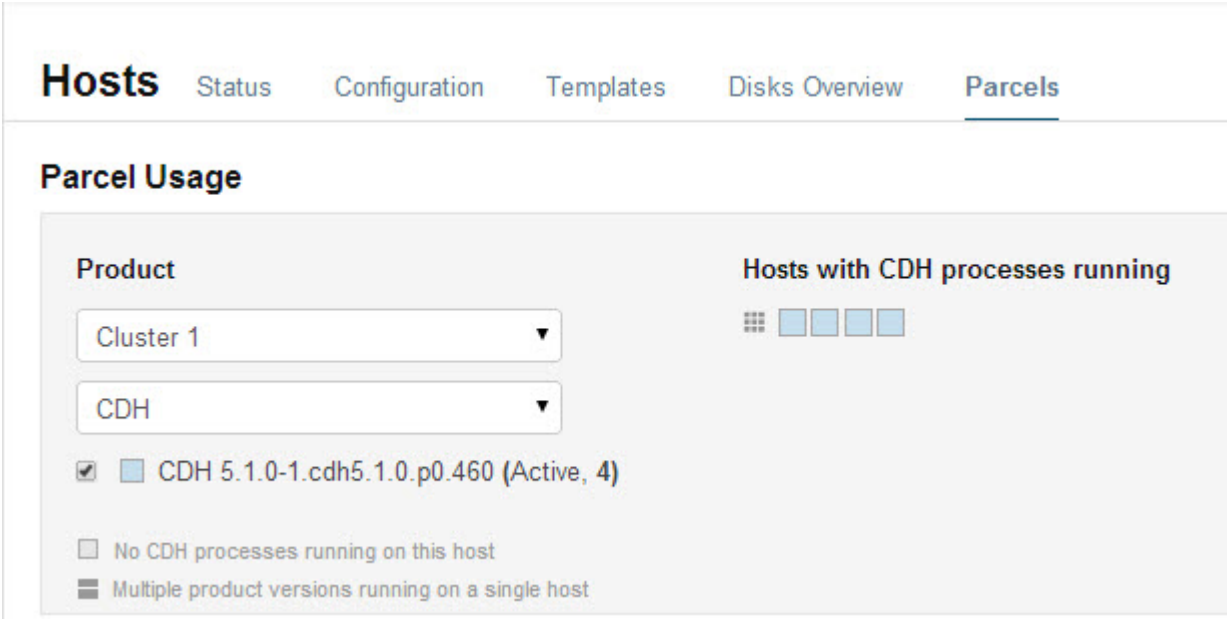
The **Parcel Usage** page shows parcels in current use in your clusters. In a large deployment, this makes it easier to keep track of different versions installed across the cluster, especially if some hosts were not available when you performed an installation or upgrade, or were added later. To display the Parcel Usage page:

1. Do one of the following:

- Click in the top navigation bar
- Click **Hosts** in the top navigation bar and click the **Parcels** tab.

2. Click the **Parcel Usage** button.

This page only shows the usage of parcels, not components that were installed as packages. If you select a cluster running packages, the cluster is not displayed, and instead you see a message indicating the cluster is not running parcels.



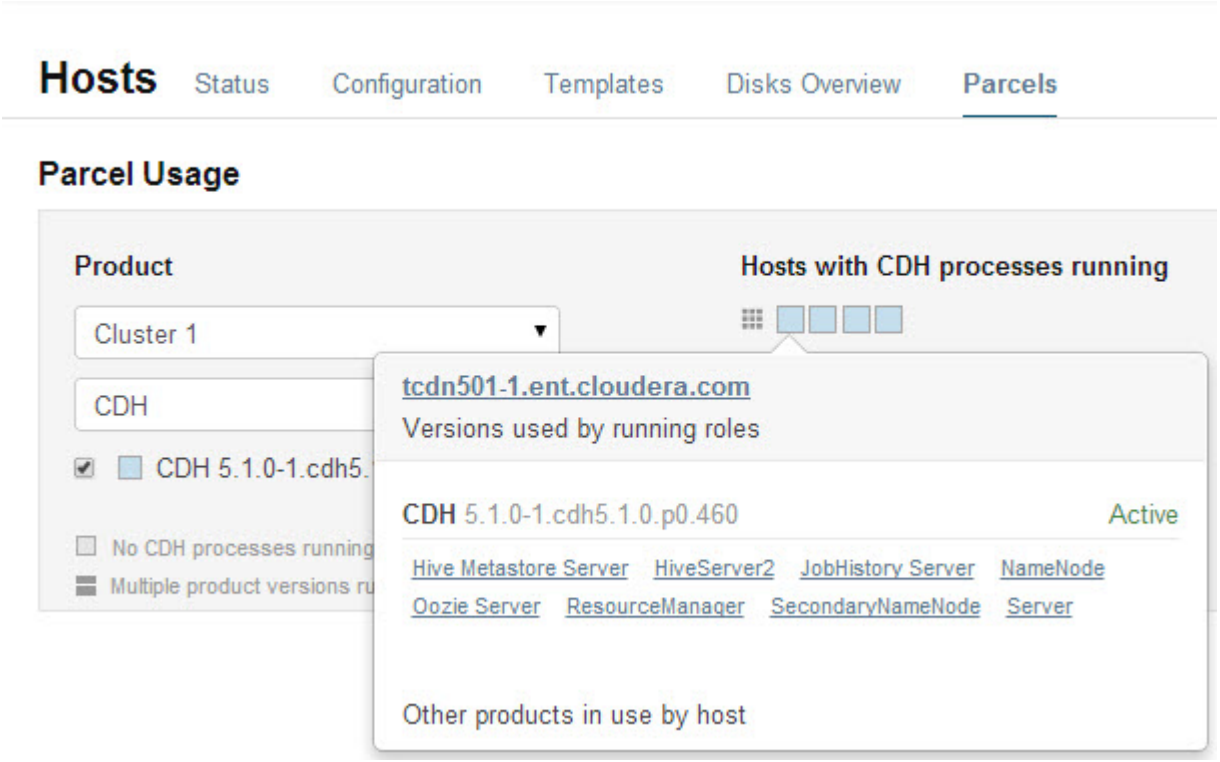
You can view parcel usage by cluster or by product.

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

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

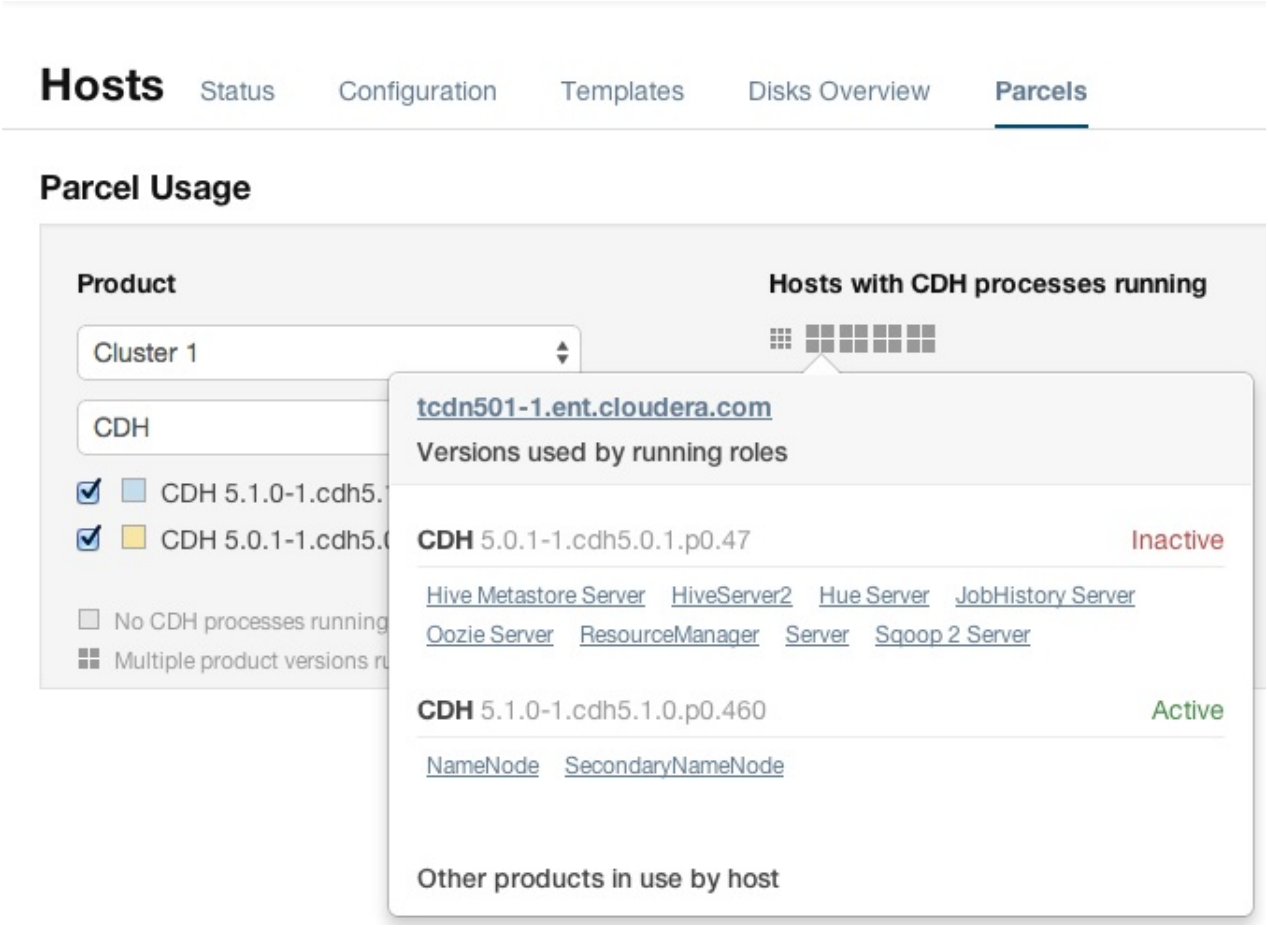
Move the cursor over the icon to see the rack to which the hosts are assigned. Hosts on different racks are displayed in separate rows.

To view the exact versions of the software running on a given host, click the square representing the host. This displays the parcel versions installed on that host.



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

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



Parcel Configuration Settings

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

Configuring Cloudera Manager Server Parcel Settings

1. Use one of the following methods to open the parcel settings page:
   - **Navigation bar**
     1. Click in the top navigation bar or click **Hosts** and click the **Parcels** tab.
     2. Click the **Configuration** button.
   - **Menu**
     1. Select **Administration** > **Settings**.
     2. Select **Category** > **Parcels** .
2. Specify a property:
   - **Local Parcel Repository Path** defines the path on the Cloudera Manager Server host where downloaded parcels are stored.
   - **Remote Parcel Repository URLs** is a list of repositories that Cloudera Manager checks for parcels. Initially this points to the latest released CDH 4, CDH 5, Impala, and Solr repositories, but you can add your own

repository locations to the list. Use this mechanism to add Cloudera repositories that are not listed by default, such as older versions of CDH, or the Sentry parcel for CDH 4.3. You can also use this to add your own custom repositories. The locations of the Cloudera parcel repositories are `https://archive.cloudera.com/product/parcels/version`, where *product* is a product name and *version* is a specific product version, `latest`, or the substitution variable `{latest_supported}`. The substitution variable appears after the parcel for the CDH version with the same major number as the Cloudera Manager version to enable substitution of the latest supported maintenance version of CDH.

To add a parcel repository:

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

3. Click **Save Changes**.

You can also:

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

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

- Theoretically, the concurrent upload count (**Maximum Parcel Uploads**) is unimportant if all hosts have the same speed Ethernet. Fifty concurrent uploads is acceptable in most cases. However, if the server has more bandwidth (for example, 10 GbE, and the normal hosts are using 1 GbE), then the count is important to maximize bandwidth. It should be at least the difference in speeds (10x in this case).
- The bandwidth limit (**Parcel Distribution Rate Limit**) should be your Ethernet speed (in MiB/seconds) divided by approximately 16. You can use a higher limit if you have QoS configured to prevent starving other services, or if you can accept the risk associated with higher bandwidth load.

Configuring a Proxy Server

To configure a proxy server through which parcels are downloaded, follow the instructions in Configuring Network Settings.

Configuring the Host Parcel Directory

Important: If you modify the parcel directory location, make sure that all hosts use the same location. Using different locations on different hosts can cause unexpected problems.

To configure the location of distributed parcels:

1. Click **Hosts** in the top navigation bar.
2. Click the **Configuration** tab.
3. Select **Category** > **Parcels**.
4. Configure the value of the **Parcel Directory** property. The setting of the `parcel_dir` property in the Cloudera Manager Agent configuration file overrides this setting.
5. Click **Save Changes** to commit the changes.
6. Restart the Cloudera Manager Agent on all hosts.

Configuring Peer-to-Peer Distribution of Parcels

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

To modify peer-to-peer distribution of parcels:

1. Open Cloudera Manager and select **Hosts** > **All Hosts** > **Configuration**.
2. Change the value of the **P2P Parcel Distribution Port** property to the new port number.

   Set the value to `0` to disable peer-to-peer distribution of parcels.
3. Click **Save Changes** to commit the changes.

Creating Virtual Images of Cluster Hosts

You can create virtual machine images, such as an PXE-boot images, Amazon AMIs, and Azure VM images of cluster hosts with pre-deployed Cloudera software that you can use to quickly spin up virtual machines. These images use Cloudera Manager parcels to install CDH software. This topic describes the procedures to create images of the Cloudera Manager host and worker host and how to instantiate hosts from those images.

Creating a Pre-Deployed Cloudera Manager Host

To create a Cloudera Manager virtual machine image:

1. Instantiate a virtual machine image (an AMI, if you are using Amazon Web Services) based on a supported operating system and start the virtual machine. See the documentation for your virtualization environment for details.
2. Install Cloudera Manager and configure a database. You can configure either a local or remote database.
3. Wait for the Cloudera Manager Admin console to become active.
4. Log in to the Cloudera Manager Admin console.
5. Download any parcels for CDH or other services managed by Cloudera Manager. Do not distribute or activate the parcels.
6. Log in to the Cloudera Manager server host:
   a. Run the following command to stop the Cloudera Manager service:

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

   b. Run the following command to disable autostarting of the `cloudera-scm-server` service:

   - RHEL6.x, CentOS 6.x and SUSE:

```
chkconfig cloudera-scm-server off
```

   - RHEL 7.x /CentOS 7.x.x:

```
systemctl disable cloudera-scm-server.service
```

   - Ubuntu:

```
update-rc.d -f cloudera-scm-server remove
```

7. Create an image of the Cloudera Manager host.See the documentation for your virtualization environment for details.
8. If you installed the Cloudera Manager database on a remote host, also create an image of the database host.

Note: Ensure that there are no clients using the remote database while creating the image.

Instantiating a Cloudera Manager Image

To create a new Cloudera Manager instance from a virtual machine image:

1. Instantiate the Cloudera Manager image.
2. If the Cloudera Manager database will be hosted on a remote host, also instantiate the database host image.
3. Ensure that the `cloudera-scm-server` service is not running by running the following command on the Cloudera Manager host:

```
service cloudera-scm-server status
```

   If it is running, stop it using the following command:

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

4. On the Cloudera Manager host, create a file named `uuid` in the `/etc/cloudera-scm-server` directory. Add a globally unique identifier to this file using the following command:

```
cat /proc/sys/kernel/random/uuid > /etc/cloudera-scm-server/uuid
```

   The existence of this file informs Cloudera Manager to reinitialize its own unique identifier when it starts.
5. Run the following command to start the Cloudera Manager service:

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

6. Run the following command to enable automatic restart for the `cloudera-scm-server`:
   - RHEL6.x, CentOS 6.x and SUSE:

```
chkconfig cloudera-scm-server on
```

   - RHEL 7.x /CentOS 7.x.x:

```
systemctl enable cloudera-scm-server.service
```

   - Ubuntu:

```
update-rc.d -f cloudera-scm-server defaults
```

Creating a Pre-Deployed Worker Host

1. Instantiate a virtual machine image (an AMI, if you are using Amazon Web Services) based on a supported operating system and start the virtual machine. See the documentation for your virtualization environment for details.
2. Download the parcels required for the worker host from the public parcel repository, or from a repository that you have created and save them to a temporary directory. See Cloudera Manager Version and Download Information.
3. From the same location where you downloaded the parcels, download the *`parcel_name`*`.parcel.sha1` file for each parcel.

4. Calculate and compare the sha1 of the downloaded parcel to ensure that the parcel was downloaded correctly. For example:

```
sha1sum KAFKA-2.0.2-1.2.0.2.p0.5-el6.parcel | awk '{print $1}' >
KAFKA-2.0.2-1.2.0.2.p0.5-el6.parcel.sha

diff KAFKA-2.0.2-1.2.0.2.p0.5-el6.parcel.sha1 KAFKA-2.0.2-1.2.0.2.p0.5-el6.parcel.sha
```

5. Unpack the parcel:

   a. Create the following directories:

      - `/opt/cloudera/parcels`
      - `/opt/cloudera/parcel-cache`

   b. Set the ownership for the two directories you just created so that they are owned by the username that the Cloudera Manager agent runs as.

   c. Set the permissions for each directory using the following command:

```
chmod 755 directory
```

      Note that the contents of these directories will be publicly available and can be safely marked as world-readable.

   d. Running as the same user that runs the Cloudera Manager agent, extract the contents of the parcel from the temporary directory using the following command:

```
tar -zxvf parcelfile -C /opt/cloudera/parcels/
```

   e. Add a symbolic link from the product name of each parcel to the `/opt/cloudera/parcels` directory.

      For example, to link `/opt/cloudera/parcels/CDH-5.9.4-1.cdh5.9.4.p0.79` to `/opt/cloudera/parcels/`**`CDH`**, use the following command:

```
ln -s /opt/cloudera/parcels/CDH-5.9.4-1.cdh5.9.4.p0.79 /opt/cloudera/parcels/CDH
```

   f. Mark the parcels to not be deleted by the Cloudera Manager agent on start up by adding a `.dont_delete` marker file (this file has no contents) to each subdirectory in the `/opt/cloudera/parcels` directory. For example:

```
touch /opt/cloudera/parcels/CDH/.dont_delete
```

6. Verify the file exists:

```
ls -l /opt/cloudera/parcels/parcelname
```

   You should see output similar to the following:

```
ls -al /opt/cloudera/parcels/CDH
total 100
drwxr-xr-x  9 root root  4096 Sep 14 14:53 .
drwxr-xr-x  9 root root  4096 Sep 14 06:34 ..
drwxr-xr-x  2 root root  4096 Sep 12 06:39 bin
-rw-r--r--  1 root root     0 Sep 14 14:53 .dont_delete
drwxr-xr-x 26 root root  4096 Sep 12 05:10 etc
drwxr-xr-x  4 root root  4096 Sep 12 05:04 include
drwxr-xr-x  2 root root 69632 Sep 12 06:44 jars
drwxr-xr-x 37 root root  4096 Sep 12 06:39 lib
drwxr-xr-x  2 root root  4096 Sep 12 06:39 meta
drwxr-xr-x  5 root root  4096 Sep 12 06:39 share
```

7. Install the Cloudera Manager agent. See Manually Install Cloudera Manager Agent Packages on page 140. If you have not already done so, Establish Your Cloudera Manager Repository Strategy on page 137.

8. Create an image of the worker host. See the documentation for your virtualization environment for details.

Instantiating a Worker Host

1. Instantiate the Cloudera worker host image.
2. Edit the following file and set the `server_host` and `server_port` properties to reference the Cloudera Manager server host.
3. If necessary perform additional steps to configure TLS/SSL. See Configuring TLS Encryption for Cloudera Manager.
4. Start the agent service:

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

Migrating from Packages to Parcels

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

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

Download, Distribute, and Activate Parcels

1. In the Cloudera Manager Admin Console, click the Parcels indicator in the top navigation bar.
2. Click **Download** for the version that matches the CDH or service version of the currently installed packages. If the parcel you want is not shown here—for example, if you want to use a version of CDH that is not the most current version—you can add parcel repositories through the Parcel Configuration Settings on page 40 page:
   - **CDH 5** - Impala, Kudu, Spark, and Search are included in the CDH parcel.
     - CDH - `https://username:password@archive.cloudera.com/p/cdh5/parcels/`
     - Accumulo--`https://username:password@archive.cloudera.com/p/accumulo-c5/parcels/`
     - GPL Extras - `https:archive.cloudera.com/gplextras5/parcels/`
   - **Cloudera Distribution of Apache Spark 2**
     - The exact parcel name is dependent on the OS. You can find all the parcels at `https://username:password@archive.cloudera.com/p/spark2/parcels/`.
   - **Key Trustee Server**
     - Go to the Key Trustee Server download page. Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee Server parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz` *`filename`*`.tar.gz` command.
   - **Key Trustee KMS**
     - Go to the Key Trustee KMS download page. Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee KMS parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz` *`filename`*`.tar.gz` command.
   - **Navigator HSM KMS**
     - Go to the Navigator HSM KMS download page. Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Navigator HSM KMS parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz` *`filename`*`.tar.gz` command. Note that the parcel name (KEYTRUSTEE) for the KMS services (both Key Trustee KMS and Navigator HSM KMS) is the same.

- **Other services**
  - Sqoop connectors - `https://username:password@archive.cloudera.com/p/sqoop-connectors/parcels/`

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel file(s) and put them into a repository. See Creating and Using a Parcel Repository for Cloudera Manager on page 168 for more details.

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

Restart the Cluster and Deploy Client Configuration

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

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

   You can optionally perform a rolling restart.
2. Redeploy client configurations:
   a. On the **Home** > **Status** tab, click
   to the right of the cluster name and select **Deploy Client Configuration**.
   b. Click **Deploy Client Configuration**.

Uninstall Packages

1. If your Hue service uses the embedded SQLite 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:

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

   - **Not including Impala and Search**

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

   - **Including Impala and Search**

Operating System	Command
RHEL	`$ sudo yum remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc 'kudu*'`
SLES	`$ sudo zypper remove 'bigtop-*' hue-common impala-shell solr-server sqoop2-client hbase-solr-doc avro-libs crunch-doc avro-doc solr-doc 'kudu*'`
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 'kudu*'`

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.

Restart Cloudera Manager Agents

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

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

Update Applications to Reference Parcel Paths

With parcels, the path to the CDH libraries is `/opt/cloudera/parcels/CDH/lib` instead of the usual `/usr/lib`. Do not link `/usr/lib/` elements to parcel-deployed paths, because the links may cause scripts that distinguish between the two paths to not work. Instead you should update your applications to reference the new library locations.

Migrating from Parcels to Packages

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

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

Install CDH and Managed Service Packages

Choose a Repository Strategy

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

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems.
- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see Understanding Custom Installation Solutions on page 166. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

Do one of the following:

- Install CDH 5 and Managed Service Packages on page 48
- Install CDH 4, Impala, and Solr Managed Service Packages on page 50

Install CDH 5 and Managed Service Packages

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

- **Red Hat**

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

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

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

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

     b. Install the RPM for all RHEL versions:

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

  2. (Optionally) add a repository key:

     - **Red Hat/CentOS/Oracle 5**

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

     - **Red Hat/CentOS/Oracle 6**

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

  3. Install the CDH packages:

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

- **SLES**

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

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

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

b. Install the RPM:

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

c. Update your system package index by running the following:

```
$ sudo zypper refresh
```

2. (Optionally) add a repository key:

- **SLES 11:**

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

- **SLES 12:**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh5/sles/12/x86_64/cdh/RPM-GPG-KEY-cloudera
```

3. Install the CDH packages:

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

- **Ubuntu and Debian**

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

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

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

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

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

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

2. Optionally add a repository key:

- **Debian Wheezy**

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

- **Ubuntu Precise**

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

3. Install the CDH packages:

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

Install CDH 4, Impala, and Solr Managed Service Packages

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

- **RHEL-compatible**

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

- **RHEL/CentOS/Oracle 5**

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

- **RHEL/CentOS 6**

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

4. Install packages on every host in your cluster:

   a. Install CDH 4 packages:

```
$ sudo yum -y install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins
 hbase hive oozie oozie-client pig zookeeper
```

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

```
$ sudo yum install hue
```

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

   a. In the table at Cloudera Impala Version and Download Information, click the entry that matches your RHEL or CentOS system.

b. Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
c. Install Impala and the Impala Shell on Impala machines:

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

6. (**Requires CDH 4.3 and higher**) Install Search
   a. In the table at Cloudera Search Version and Download Information, click the entry that matches your RHEL or CentOS system.
   b. Go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.
   c. Install the Solr Server on machines where you want Cloudera Search.

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

- **SLES**
  1. Run the following command:

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

  2. Update your system package index by running:

```
$ sudo zypper refresh
```

  3. Optionally add a repository key:

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

  4. Install packages on every host in your cluster:
     a. Install CDH 4 packages:

```
$ sudo zypper install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins
 hbase hive oozie oozie-client pig zookeeper
```

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

```
$ sudo zypper install hue
```

     c. (**Requires CDH 4.2 and higher**) Install Impala
        a. Run the following command:

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

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

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

     d. (**Requires CDH 4.3 and higher**) Install Search
        a. Run the following command:

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

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

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

- **Ubuntu or Debian**

1. In the table at CDH Version and Packaging Information, click the entry that matches your Ubuntu or Debian system.
2. Go to the list file (`cloudera.list`) for your system and save it in the `/etc/apt/sources.list.d/` directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your `cloudera.list` file should look like:

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

3. Optionally add a repository key:

   - **Ubuntu Lucid**

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

   - **Ubuntu Precise**

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

   - **Debian Squeeze**

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

4. Install packages on every host in your cluster:

   a. Install CDH 4 packages:

```
$ sudo apt-get install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins
 hbase hive oozie oozie-client pig zookeeper
```

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

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

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

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

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

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

      a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Ubuntu or Debian system.

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

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

Deactivate Parcels

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

1. Go to the Parcels page by doing one of the following:
   - Clicking the parcel indicator in the Admin Console navigation bar ( )
   - Clicking the **Hosts** in the top navigation bar, then the **Parcels** tab.
2. Click **Actions** on the activated CDH and managed service parcels and select **Deactivate**.

Restart the Cluster

1. On the **Home** > **Status** tab, click to the right of the cluster name and select **Restart**.
2. Click **Restart** that appears in the next screen to confirm. The **Command Details** window shows the progress of stopping services.

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

You can optionally perform a rolling restart.

Remove and Delete Parcels

Removing a Parcel

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

Deleting a Parcel

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

Installing Cloudera Manager and CDH

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

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

Cloudera Manager Deployment

A Cloudera Manager deployment consists of the following software components:

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

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

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

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

- **Demonstration and proof of concept deployments** - There are three installation options:
  - Installation Path A - Automated Installation by Cloudera Manager (Non-Production Mode) on page 127 - Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, Cloudera Manager Agent, CDH, and managed service software on cluster hosts. Cloudera Manager also configures databases for the Cloudera Manager Server and Hive Metastore and optionally for Cloudera Management Service roles. This path is recommended for demonstration and proof-of-concept deployments, but is *not recommended* for production deployments because its not intended to scale and may require database migration as your cluster grows. To use this method, server and cluster hosts must satisfy the following requirements:
    - Provide the ability to log in to the Cloudera Manager Server host using a root account or an account that has password-less sudo permission.
    - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See CDH and Cloudera Manager Networking and Security Requirements for further information.
    - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.
  - Installation Path B - Installation Using Cloudera Manager Parcels or Packages on page 136 - you install the Oracle JDK, Cloudera Manager Server, and embedded PostgreSQL database packages on the Cloudera Manager Server host. You have two options for installing Oracle JDK, Cloudera Manager Agent, CDH, and managed service software on cluster hosts: manually install it yourself or use Cloudera Manager to automate installation.

    In order for Cloudera Manager to automate installation of Cloudera Manager Agent packages or CDH and managed service software, cluster hosts must satisfy the following requirements:

    - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See CDH and Cloudera Manager Networking and Security Requirements for further information.

    - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.

- **Production deployments** - require you to first manually install and configure a production database for the Cloudera Manager Server and Hive Metastore. There are two installation options:
  - Installation Path B - Installation Using Cloudera Manager Parcels or Packages on page 136 - you install the Oracle JDK and Cloudera Manager Server packages on the Cloudera Manager Server host. You have two options for installing Oracle JDK, Cloudera Manager Agent, CDH, and managed service software on cluster hosts: manually install it yourself or use Cloudera Manager to automate installation.

    In order for Cloudera Manager to automate installation of Cloudera Manager Agent packages or CDH and managed service software, cluster hosts must satisfy the following requirements:

    - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See CDH and Cloudera Manager Networking and Security Requirements for further information.
    - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.
  - Installation Path C - Manual Installation Using Cloudera Manager Tarballs on page 149 - you install the Oracle JDK, Cloudera Manager Server, and Cloudera Manager Agent software using tarballs and use Cloudera Manager to automate installation of CDH and managed service software as parcels.

Note: Cloudera does not support CDH cluster deployments using hosts in Docker containers.

Cloudera Manager Installation Phases

The following table describes the phases of installing Cloudera Manager and a Cloudera Manager deployment of CDH and managed services. Every phase is required, but you can accomplish each phase in multiple ways, depending on your organization's policies and requirements. The six phases are grouped into three installation paths based on how the Cloudera Manager Server and database software are installed on the Cloudera Manager Server and cluster hosts. The criteria for choosing an installation path are discussed in Cloudera Manager Deployment on page 54.

Table 4: Cloudera Installation Phases

Phase			
Phase 1: Install JDK Install the JDK required by Cloudera Manager Server, Management Service, and CDH.	There are two options: • Use the Cloudera Manager Installer to install a supported version of the Oracle JDK in `/usr/java` and on all hosts in the cluster. • Use the command line to manually install supported versions of the Oracle JDK and set the `JAVA_HOME` environment variable to the install directory on all hosts.		
Phase 2: Set up Databases Install, configure, and start the databases that are required by the Cloudera Manager Server, Cloudera Management Service, and that are optional for some CDH services.	There are two options: • Use the Cloudera Manager Installer to install, configure, and start an embedded PostgresSQL database. • Use command-line package installation tools like `yum` to install, configure, and install the database		
	Path A	**Path B**	**Path C**

Phase			
Phase 3: Install Cloudera Manager Server Install and start Cloudera Manager Server on one host.	Use the Cloudera Manager Installer to install its packages and the server. Requires Internet access and sudo privileges on the host.	Use Linux package install commands (like yum) to install Cloudera Manager Server. Update database properties. Use service commands to start Cloudera Manager Server.	Use Linux commands to unpack tarballs and service commands to start the server.
Phase 4: Install Cloudera Manager Agents Install and start the Cloudera Manager Agent on all hosts.	Use the Cloudera Manager Installation wizard to install the Agents on all hosts.	There are two options: • Use Linux package install commands (like yum) to install Cloudera Manager Agents on all hosts. • Use the Cloudera Manager Installation wizard to install the Agents on all hosts.	Use Linux commands to unpack tarballs and service commands to start the agents on all hosts.
Phase 5: Install CDH and Managed Service software Install, configure, and start CDH and managed services on all hosts.	Use the Cloudera Manager Installation wizard to install CDH and other managed services.	There are two options: • Use the Cloudera Manager Installation wizard to install CDH and other managed services. • Use Linux package install commands (like yum) to install CDH and other managed services on all hosts.	Use Linux commands to unpack tarballs and service commands to start CDH and managed services on all hosts.
Phase 6: Create, Configure and Start CDH and Managed Services Configure and start CDH and managed services.	Use the Cloudera Manager Installation wizard to install CDH and other managed services, assign roles to hosts, and configure the cluster. Many configurations are automated.	Use the Cloudera Manager Installation wizard to install CDH and other managed services, assign roles to hosts, and configure the cluster. Many configurations are automated.	Use the Cloudera Manager Installation wizard to install CDH and other managed services, assign roles to hosts, and configure the cluster. Many configurations are automated. You can also use the Cloudera Manager API to manage a cluster, which can be useful for scripting preconfigured deployments.

Cloudera Manager Installation Software

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

- **Installation path A (non-production)** - A small self-executing Cloudera Manager installation program to install the Cloudera Manager Server and other packages. The Cloudera Manager installer, which you install on the host where you want the Cloudera Manager Server to run, performs the following:

  1. Installs the package repositories for Cloudera Manager and the Oracle Java Development Kit (JDK).
  2. Installs the Cloudera Manager packages.
  3. Installs and configures an embedded PostgreSQL database for use by the Cloudera Manager Server, some Cloudera Management Service roles, some managed services, and Cloudera Navigator roles.

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

- **Installation paths B and C** - Cloudera Manager package repositories for manually installing the Cloudera Manager Server, Agent, and embedded database packages.
- **Installation path B** - The Cloudera Manager Installation wizard for automating installation of Cloudera Manager Agent package.
- **All installation paths** - The Cloudera Manager Installation wizard for automating CDH and managed service installation and configuration on the cluster hosts. Cloudera Manager provides two methods for installing CDH and managed services: parcels and packages. Parcels simplify the installation process and allow you to download, distribute, and activate new versions of CDH and managed services from within Cloudera Manager. After you install Cloudera Manager and connect to the Cloudera Manager Admin Console for the first time, use the Cloudera Manager Installation wizard to:

  1. Discover cluster hosts.
  2. Optionally install the Oracle JDK.
  3. Optionally install CDH, managed service, and Cloudera Manager Agent software on cluster hosts.
  4. Select services.
  5. Map service roles to hosts.
  6. Edit service configurations.
  7. Start services.

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

Installation paths:

- Installation Path A - Automated Installation by Cloudera Manager (Non-Production Mode) on page 127
- Installation Path B - Installation Using Cloudera Manager Parcels or Packages on page 136
- Installation Path C - Manual Installation Using Cloudera Manager Tarballs on page 149

Unmanaged Deployment

In an deployment not managed by Cloudera Manager, you are responsible for managing all phases of the lifecycle of CDH and managed service components on each host: installation, configuration, and service lifecycle operations such as start and stop. This section describes alternatives for installing CDH 5 software in an unmanaged deployment.

- **Command-line methods:**
  - Download and install the CDH 5 "1-click Install" package
  - Add the CDH 5 repository
  - Build your own CDH 5 repository

  If you use one of these command-line methods, the first (downloading and installing the "1-click Install" package) is recommended in most cases because it is simpler than building or adding a repository.
- **Tarball** You can download a tarball from CDH downloads. Keep the following points in mind:

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

See Installing and Deploying CDH Using the Command Line on page 221 for detailed instructions for each of these options.

Java Development Kit Installation

Note: Cloudera, Inc. acquired Oracle JDK software under the Oracle Binary Code License Agreement. Pursuant to Item D(v)(a) of the SUPPLEMENTAL LICENSE TERMS of the Oracle Binary Code License Agreement, use of JDK software is governed by the terms of the Oracle Binary Code License Agreement. By installing the JDK software, you agree to be bound by these terms. If you do not wish to be bound by these terms, then do not install the Oracle JDK.

Some installation paths require that you install the Oracle Java Development Kit on hosts before deploying Cloudera Manager, CDH, and managed services. To install the Oracle JDK, follow the instructions in Installing the Oracle JDK on page 58. The completed installation, or any already existing installation, must meet the following requirements.

Requirements

- The JDK must be 64-bit. Do not use a 32-bit JDK.
- Install one of the CDH and Cloudera Manager Supported JDK Versions.
- Install the *same version* of the Oracle JDK on each host.
- Install the JDK in `/usr/java/jdk-version`.

Important:

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

  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.)

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

Installing the Oracle JDK

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

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

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

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

4. Follow the instructions in Configuring a Custom Java Home Location on page 173. This change affects all CDH processes and Cloudera Management Service roles in the cluster.

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

Configuring Single User Mode

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

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

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

By default, the single user is `cloudera-scm` and the configuration steps described in the following sections assume that user. However, other users are supported. If you choose another user, replace `cloudera-scm` in the following steps with the selected user, and perform the additional steps in Using a Non-default Single User on page 59.

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

Limitations

- Switching between conventional and single user mode is not supported.

  > **Important:** In **Administration** > **Settings** there is a **Single user mode** checkbox. Do not enable or disable the checkbox after installation.

- Single user mode is supported for clusters running CDH 5.2 and higher.
- NFS Gateway is not supported in single user mode.
- Cloudera Navigator data encryption components are not supported in single user mode.

Using a Non-default Single User

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

- Make the following directories writable by the single user:
  - `/var/log/cloudera-scm-agent/`
  - `/var/lib/cloudera-scm-agent/`
- Cloudera Manager stores parcels under `/opt/cloudera`, which by default is owned by `cloudera-scm`. Do one of the following:
  - Change `/opt/cloudera` to be writable by the single user.
  - Change the parcel directory location to be writable by the single user:

1. Go to **Administration** > **Settings** > **Parcels**.
2. Set the **Local Parcel Repository Path** property.
3. Click **Save Changes**.

- For a single user *username*, create the process limits configuration file at `/etc/security/limits.d/username.conf` with the following settings:

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

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

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

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

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

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

  - 1. Click **Hosts** in the top navigation bar.
    2. Click the **Configuration** tab.
    3. Select **Category** > **Parcels**.
    4. Configure the value of the **Parcel Directory** property. The setting of the `parcel_dir` property in the Cloudera Manager Agent configuration file overrides this setting.
    5. Click **Save Changes** to commit the changes.
    6. Restart the Cloudera Manager Agent on all hosts.

Configuration Steps Before Running the Installation Wizard

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

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

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

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

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

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

2. Add this line to the configuration file:

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

- Set up per user limits for su prior to setting up the Agent.
  1. Edit/etc/pam.d/su.
  2. Uncomment:

```
session required pam_limits.so
```

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

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

Perform the following steps for the indicated scenarios:

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

Configuration Steps While Running the Installation Wizard

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

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

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

- Paths under /var - These are log, run, and data directories for the different services.

- Per volume data directories - These are data directory configurations that list a directory per volume. Such configurations are used by HDFS, MapReduce, YARN and Impala.

Configuration for Secure Clusters

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

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

Controlling Access to sudo Commands

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

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

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

Whitelisting sudo Commands

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

Regular Operation Commands

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

Support Bundle Collection Commands

- `bash`
- `cat`
- `date`
- `df`
- `dmesg`
- `grep`
- `hostname`
- `ls`
- `netstat`

- ps
- rpm
- uname
- chkconfig
- ethtool
- ifconfig
- iptables
- lsmod
- lspci
- lvdisplay
- service
- sysctl
- curl
- dig
- host
- lsb_release
- lscpu
- nslookup
- ntpstat
- python
- sar
- top
- uptime
- vmstat
- dmidecode
- lsof
- ntpq

Overriding the sudo Command

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

To configure the location of this script:

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

```
sudo_command=path_to_script
```

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

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

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

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

```
/bin/cat /proc/cgroups
/bin/chmod -R ugo+r /etc/accumulo/*
/bin/chmod -R ugo+r /etc/hadoop/*
/bin/chmod -R ugo+r /etc/hbase/*
/bin/chmod -R ugo+r /etc/hive/*
/bin/chmod -R ugo+r /etc/solr/*
/bin/chmod -R ugo+r /etc/spark/*
/bin/chmod -R ugo+r /etc/sqoop/*
/bin/chmod 0644 /cldr/app/coolapp/var/run/cloudera-scm-agent/process/*/topology.map
/bin/chmod 0755 /cldr/app/coolapp/opt/parcels/CDH*/lib/hue/desktop
```

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

```
/bin/rm
/bin/rm -rf /etc/accumulo/*
/bin/rm -rf /etc/hadoop/*
/bin/rm -rf /etc/hbase/*
/bin/rm -rf /etc/hive/*
/bin/rm -rf /etc/solr/*
/bin/rm -rf /etc/spark/*
/bin/rm -rf /etc/sqoop/*
/bin/umount /tmp/*
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
 --display ip6tables.x86_64
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
 --display iptables.x86_64
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
 --display mta
/usr/sbin/update-alternatives --admindir /var/lib/alternatives --altdir /etc/alternatives
 --display print
/usr/sbin/update-alternatives --auto accumulo-conf
/usr/sbin/update-alternatives --auto hadoop-conf
/usr/sbin/update-alternatives --auto hbase-conf
/usr/sbin/update-alternatives --auto hive-conf
/usr/sbin/update-alternatives --auto solr-conf
/usr/sbin/update-alternatives --auto spark-conf
/usr/sbin/update-alternatives --auto sqoop-conf
/usr/sbin/update-alternatives --install /etc/accumulo/conf accumulo-conf /etc/accumulo/*
 51
/usr/sbin/update-alternatives --install /etc/flume-ng/conf flume-ng-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/flume-ng/conf.empty 10
/usr/sbin/update-alternatives --install /etc/hadoop-httpfs/conf hadoop-httpfs-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hadoop-httpfs/conf.empty 10
/usr/sbin/update-alternatives --install /etc/hadoop-kms/conf hadoop-kms-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hadoop-kms/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/* 90
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/* 91
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/* 92
/usr/sbin/update-alternatives --install /etc/hadoop/conf hadoop-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hadoop/conf.empty 10
/usr/sbin/update-alternatives --install /etc/hbase-solr/conf hbase-solr-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hbase-solr/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hbase/conf hbase-conf /etc/hbase/* 90
/usr/sbin/update-alternatives --install /etc/hbase/conf hbase-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hbase/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hive-hcatalog/conf hive-hcatalog-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hive-hcatalog/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hive-webhcat/conf hive-webhcat-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hive-webhcat/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hive/conf hive-conf /etc/hive/* 90
/usr/sbin/update-alternatives --install /etc/hive/conf hive-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hive/conf.dist 10
/usr/sbin/update-alternatives --install /etc/hue/conf hue-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/hue/conf.empty 10
/usr/sbin/update-alternatives --install /etc/impala/conf impala-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/impala/conf.dist 10
/usr/sbin/update-alternatives --install /etc/llama/conf llama-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/llama/conf.dist 10
/usr/sbin/update-alternatives --install /etc/mahout/conf mahout-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/mahout/conf.dist 10
/usr/sbin/update-alternatives --install /etc/oozie/conf oozie-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/oozie/conf.dist 10
/usr/sbin/update-alternatives --install /etc/pig/conf pig-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/pig/conf.dist 10
/usr/sbin/update-alternatives --install /etc/sentry/conf sentry-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/sentry/conf.dist 10
/usr/sbin/update-alternatives --install /etc/solr/conf solr-conf /etc/solr/* 90
/usr/sbin/update-alternatives --install /etc/solr/conf solr-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/solr/conf.dist 10
/usr/sbin/update-alternatives --install /etc/spark/conf spark-conf /etc/spark/* 51
/usr/sbin/update-alternatives --install /etc/spark/conf spark-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/spark/conf.dist 10
/usr/sbin/update-alternatives --install /etc/sqoop/conf sqoop-conf /etc/sqoop/* 50
/usr/sbin/update-alternatives --install /etc/sqoop/conf sqoop-conf
/cldr/app/coolapp/opt/parcels/CDH*/etc/sqoop/conf.dist 10
/usr/sbin/update-alternatives --install /etc/sqoop2/conf sqoop2-conf
```

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

```
/usr/sbin/update-alternatives --install /usr/bin/sqoop-create-hive-table
sqoop-create-hive-table /cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-create-hive-table
 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-eval sqoop-eval
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-eval 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-export sqoop-export
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-export 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-help sqoop-help
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-help 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-import sqoop-import
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-import 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-import-all-tables
sqoop-import-all-tables /cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-import-all-tables
 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-job sqoop-job
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-job 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-list-databases sqoop-list-databases
 /cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-list-databases 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-list-tables sqoop-list-tables
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-list-tables 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-merge sqoop-merge
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-merge 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-metastore sqoop-metastore
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-metastore 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop-version sqoop-version
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop-version 10
/usr/sbin/update-alternatives --install /usr/bin/sqoop2 sqoop2
/cldr/app/coolapp/opt/parcels/CDH*/bin/sqoop2 10
/usr/sbin/update-alternatives --install /usr/bin/statestored statestored
/cldr/app/coolapp/opt/parcels/CDH*/bin/statestored 10
/usr/sbin/update-alternatives --install /usr/bin/whirr whirr
/cldr/app/coolapp/opt/parcels/CDH*/bin/whirr 10
/usr/sbin/update-alternatives --install /usr/bin/yarn yarn
/cldr/app/coolapp/opt/parcels/CDH*/bin/yarn 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-client zookeeper-client
/cldr/app/coolapp/opt/parcels/CDH*/bin/zookeeper-client 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-server zookeeper-server
/cldr/app/coolapp/opt/parcels/CDH*/bin/zookeeper-server 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-server-cleanup
zookeeper-server-cleanup /cldr/app/coolapp/opt/parcels/CDH*/bin/zookeeper-server-cleanup
 10
/usr/sbin/update-alternatives --install /usr/bin/zookeeper-server-initialize
zookeeper-server-initialize
/cldr/app/coolapp/opt/parcels/CDH*/bin/zookeeper-server-initialize 10
/usr/sbin/update-alternatives --install /usr/bin/bigtop-detect-javahome
bigtop-detect-javahome /cldr/app/coolapp/opt/parcels/CDH*/bin/bigtop-detect-javahome 10
/usr/sbin/update-alternatives --install /usr/bin/impala-collect-minidumps
impala-collect-minidumps /cldr/app/coolapp/opt/parcels/CDH*/bin impala-collect-minidumps
 10
/usr/sbin/update-alternatives --install /usr/bin/parquet-tools parquet-tools
/cldr/app/coolapp/opt/parcels/CDH*/bin/parquet-tools 10
/usr/sbin/update-alternatives --remove accumulo-conf /etc/accumulo/*
/usr/sbin/update-alternatives --remove hadoop-conf /etc/hadoop/*
/usr/sbin/update-alternatives --remove hbase-conf /etc/hbase/*
/usr/sbin/update-alternatives --remove hive-conf /etc/hive/*
/usr/sbin/update-alternatives --remove solr-conf /etc/solr/*
/usr/sbin/update-alternatives --remove spark-conf /etc/spark/*
/usr/sbin/update-alternatives --remove sqoop-conf /etc/sqoop/*
```

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

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

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

Cloudera Manager and Managed Service Datastores

Cloudera Manager uses various databases and datastores to store information about the Cloudera Manager configuration, as well as information such as the health of the system or task progress. For quick, simple installations, Cloudera Manager can install and configure an embedded PostgreSQL database as part of the Cloudera Manager installation process. In addition, some CDH services use databases and are automatically configured to use a default database. If you plan to use the embedded and default databases provided during the Cloudera Manager installation, see Installation Path A - Automated Installation by Cloudera Manager (Non-Production Mode) on page 127 and Embedded PostgreSQL Database on page 73.

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

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

Note: Cloudera does not support CDH cluster deployments using hosts in Docker containers.

For information about planning, managing, and backing up Cloudera Manager data stores, see Storage Space Planning for Cloudera Manager on page 116.

Required Databases

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

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

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

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large.
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small.
- Activity Monitor - Contains information about past activities. In large clusters, this database can grow large. Configuring an Activity Monitor database is only necessary if a MapReduce service is deployed.
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.
- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Hue Server - Contains user account information, job submissions, and Hive queries. Relatively small.
- Sentry Server - Contains authorization metadata. Relatively small.
- Cloudera Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large.
- Cloudera Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small.

See Backing Up Databases on page 111.

The Cloudera Manager Service Host Monitor and Service Monitor roles have an internal datastore.

Cloudera Manager provides three installation paths:

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

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

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

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

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

Setting up the Cloudera Manager Server Database

The Cloudera Manager Server database stores information about service and host configurations. For demonstration and proof-of-concept deployments you can use an embedded PostgreSQL database. See Embedded PostgreSQL Database on page 73.

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

Preparing a Cloudera Manager Server External Database

Before performing these steps, install and configure a database server as described in Configuring and Starting the MariaDB Server on page 83, Configuring and Starting the MySQL Server on page 90, Configuring the Oracle Server on page 97, or Configuring and Starting the PostgreSQL Server on page 76.

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

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

   - Tarball install

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

   The script prepares the database by:

   - Creating the Cloudera Manager Server database configuration file.
   - (MariaDB, MySQL, and PostgreSQL) Creating a database for the Cloudera Manager Server to use.
   - (MariaDB, MySQL, and PostgreSQL) Setting up a user account for the Cloudera Manager Server.

2. Remove the embedded PostgreSQL properties file if it exists:
   - Installer or package install

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

- Tarball install

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

After successfully running the `scm_prepare_database.sh` script, return to Establish Your Cloudera Manager Repository Strategy.

Syntax for scm_prepare_database.sh

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

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

Table 5: Required Parameters

Parameter	Description
`database-type`	One of the supported database types: • MariaDB - `mysql` • MySQL - `mysql` • Oracle - `oracle` • PostgreSQL - `postgresql`
`database-name`	The name of the Cloudera Manager Server database to create or use.
`username`	The username for the Cloudera Manager Server database to create or use.
`password`	The password for the Cloudera Manager Server database to create or use. If you do not specify the password on the command line, the script prompts you to enter it.

Table 6: Options

Option	Description
`-? or --help`	Display help.
`--config-path`	The path to the Cloudera Manager Server configuration files. The default is `/etc/cloudera-scm-server`.
`-f`	The script does not stop if an error occurs.
`-h or --host`	The IP address or hostname of the host where the database is installed. The default is to use the local host.
`-p or --password`	The admin password for the database application. The default is no password. For -p, no space occurs between the option and the provided value.
`-P or --port`	The port number to use to connect to the database. The default port is 3306 for MariaDB, 3306 for MySQL, 5432 for PostgreSQL, and 1521 for Oracle. This option is used for a remote connection only.
`--scm-host`	The hostname where the Cloudera Manager Server is installed. Omit if the Cloudera Manager Server and the database are installed on the same host.
`--scm-password-script`	A script to execute whose `stdout` provides the password for user SCM (for the database).

Option	Description
`--schema-path`	The path to the Cloudera Manager schema files. The default is `/usr/share/cmf/schema` (the location of the script).
`-u or --user`	The admin username for the database application. For -u, no space occurs between the option and the provided value. If this option is supplied, the script creates a user and database for the Cloudera Manager Server; otherwise, it uses the user and database you created previously.

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

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

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

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

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

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

3. On myhost1, delete the temporary user:

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

Example 2: Running the script to configure Oracle

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

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

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

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

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

You can configure Cloudera Manager to use an external database for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server. If you choose this option, you must create the databases *before* you run the Cloudera Manager installation wizard. For more information, see the instructions in Configuring an External Database for Oozie on page 107, Configuring an External Database for Sqoop on page 110, MariaDB Database on page 82, MySQL Database on page 89, Oracle Database on page 97, and External PostgreSQL Database on page 76.

External Databases for Hue

See Hue Databases to configure Hue with a custom database (local or remote).

Embedded PostgreSQL Database

Installing and Starting the Embedded PostgreSQL Database

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

If you are using Installation Path B - Installation Using Cloudera Manager Parcels or Packages on page 136 and you want to use an embedded PostgreSQL database for the Cloudera Management Server, use this procedure to install and start the database:

1. Install the embedded PostgreSQL database packages:

OS	Command
RHEL-compatible, if you have a yum repo configured	`sudo yum install cloudera-manager-server-db-2`
RHEL-compatible, if you're transferring RPMs manually	`sudo yum --nogpgcheck localinstall cloudera-manager-server-db-2.noarch.rpm`
SLES	`sudo zypper install cloudera-manager-server-db-2`
Ubuntu or Debian	`sudo apt-get install cloudera-manager-server-db-2`

2. Start the PostgreSQL database:

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

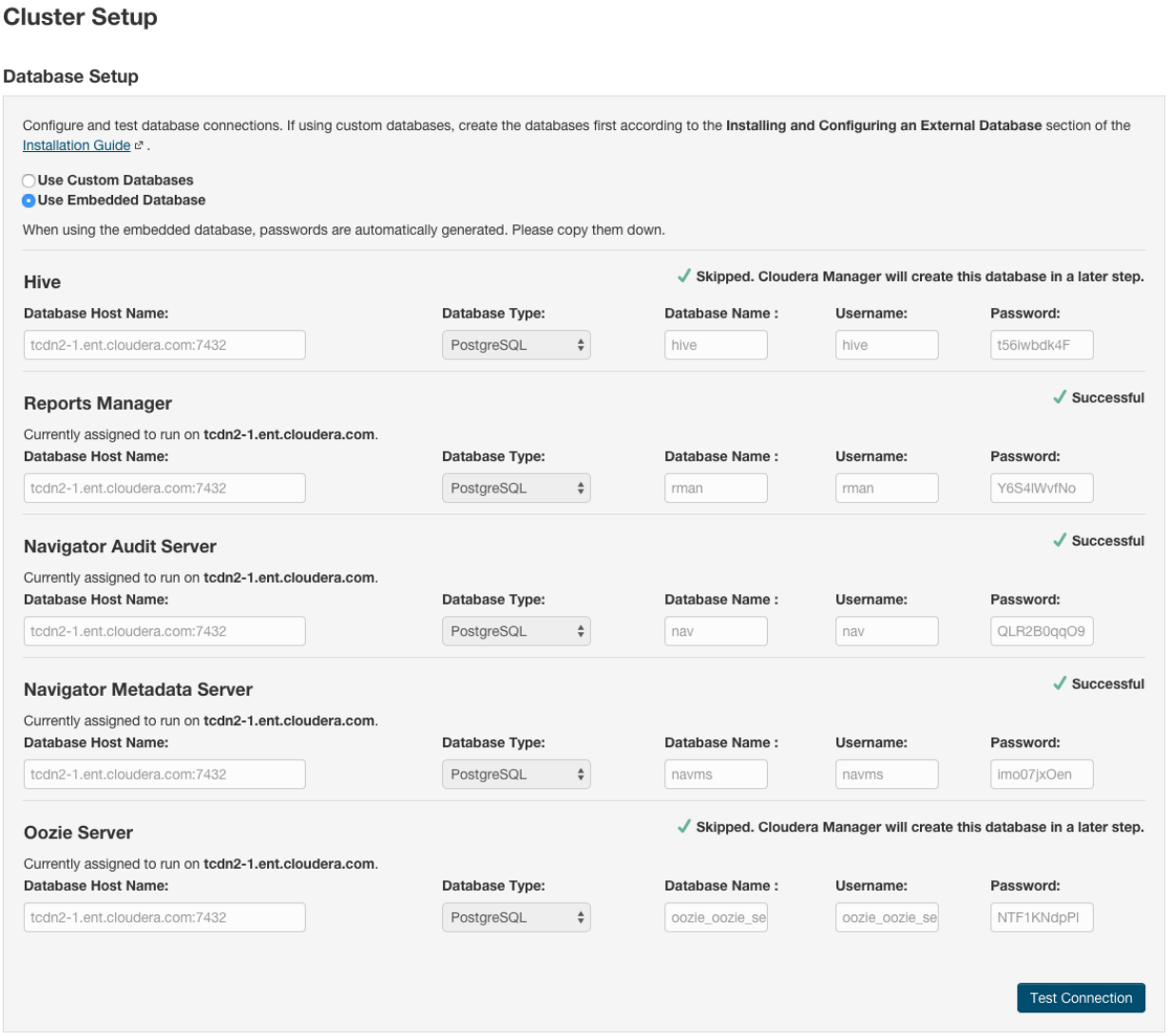
Stopping the Embedded PostgreSQL Database

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

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

Changing Embedded PostgreSQL Database Passwords

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



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

```
# cat /etc/cloudera-scm-server/db.properties

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

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

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

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

MnPwGeWaip

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

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

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

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

postgres=#
```

3. Determine the database and owner names:

```
postgres=# \l
                                      List of databases
   Name    |    Owner     | Encoding | Collation  |   Ctype    |        Access privileges
-----------+--------------+----------+------------+------------+-------------------------------

 amon      | amon         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 hive      | hive         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 nav       | nav          | UTF8     | en_US.UTF8 | en_US.UTF8 |
 navms     | navms        | UTF8     | en_US.UTF8 | en_US.UTF8 |
 postgres  | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 |
 rman      | rman         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 scm       | scm          | UTF8     | en_US.UTF8 | en_US.UTF8 |
 template0 | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 | =c/"cloudera-scm"
                                                                :
"cloudera-scm"=CTc/"cloudera-scm"
 template1 | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 | =c/"cloudera-scm"
                                                                :
"cloudera-scm"=CTc/"cloudera-scm"
(9 rows)
```

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

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

5. Configure the role with the new password:

   a. In the Cloudera Manager Admin Console, select **Clusters** > **Cloudera Management Service**.
   b. Click the **Configuration** tab.
   c. In the **Scope** section, select the role where you are configuring the database.
   d. Select **Category** > **Database** category.
   e. Set the ***Role Name*** **Database Password** property.
   f. Click **Save Changes** to commit the changes.

External PostgreSQL Database

To use an external PostgreSQL database, follow these procedures. For information on compatible versions of the PostgreSQL database, see CDH and Cloudera Manager Supported Databases.

Installing the External PostgreSQL Server

Note:

- If you already have a PostgreSQL database set up, you can skip to the section Configuring and Starting the PostgreSQL Server on page 76 to verify that your PostgreSQL configurations meet the requirements for Cloudera Manager.
- Make sure that the data directory, which by default is `/var/lib/postgresql/data/`, is on a partition that has sufficient free space.
- Cloudera Manager 5.8.x and higher supports the use of a custom schema name. For more information, see https://www.postgresql.org/docs/current/static/ddl-schemas.html.

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

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

2. Install PostgreSQL packages:

   - **RHEL**

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

   - **SLES**

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

Note: This command will install PostgreSQL 9.1. If you want to install a different version, you can use `zypper search postgresql` to search for an available supported version. See CDH and Cloudera Manager Supported Databases.

   - **Debian/Ubuntu**

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

Configuring and Starting the PostgreSQL Server

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

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

```
$ sudo service postgresql start
```

In other versions, you must explicitly initialize the database using one of the following commands:

- If the PostgreSQL database uses the SLES 12 operating system:

```
$ sudo service postgresql initdb --pgdata=/var/lib/pgsql/data --encoding=UTF-8
```

- All other operating systems:

```
$ sudo service postgresql initdb
```

See the PostgreSQL documentation for more details.

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

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

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

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

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

```
listen_addresses = '*'
```

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

- Small to mid-sized clusters - Consider the following settings as starting points. If resources are limited, consider reducing the buffer sizes and checkpoint segments further. Ongoing tuning may be required based on each host's resource utilization. For example, if the Cloudera Manager Server is running on the same host as other roles, the following values may be acceptable:
  - `shared_buffers` - 256MB
  - `wal_buffers` - 8MB
  - `checkpoint_segments` - 16

Note: The `checkpoint_segments` setting is removed in PostgreSQL 9.5 and higher, replaced by `min_wal_size` and `max_wal_size`. The PostgreSQL 9.5 release notes provides the following formula for determining the new settings:

```
max_wal_size = (3 * checkpoint_segments) * 16MB
```

  - `checkpoint_completion_target` - 0.9

- Large clusters - Can contain up to 1000 hosts. Consider the following settings as starting points.
  - `max_connection` - For large clusters, each database is typically hosted on a different host. In general, allow each database on a host 100 maximum connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL, as described at Connection Settings.
  - `shared_buffers` - 1024 MB. This requires that the operating system can allocate sufficient shared memory. See PostgreSQL information on Managing Kernel Resources for more information on setting kernel resources.

- `wal_buffers` - 16 MB. This value is derived from the `shared_buffers` value. Setting `wal_buffers` to be approximately 3% of `shared_buffers` up to a maximum of approximately 16 MB is sufficient in most cases.
- `checkpoint_segments` - 128. The PostgreSQL Tuning Guide recommends values between 32 and 256 for write-intensive systems, such as this one.

> **Note:** The `checkpoint_segments` setting is removed in PostgreSQL 9.5 and higher, replaced by `min_wal_size` and `max_wal_size`. The PostgreSQL 9.5 release notes provides the following formula for determining the new settings:
>
> ```
> max_wal_size = (3 * checkpoint_segments) * 16MB
> ```

- `checkpoint_completion_target` - 0.9. This setting is only available in PostgreSQL versions 8.3 and higher, which are highly recommended.

4. Configure the PostgreSQL server to start at boot.

   - **RHEL**

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

   - **SLES**

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

   - **Debian/Ubuntu**

```
$ sudo chkconfig postgresql on
```

5. Start or restart the PostgreSQL database:

```
$ sudo service postgresql restart
```

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

Create databases and user accounts for components that require databases:

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

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

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

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

1. Connect to PostgreSQL:

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

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

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

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

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

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

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

For PostgreSQL 8.2.23 or higher, also run:

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

Return to Establish Your Cloudera Manager Repository Strategy on page 137.

Configuring the Hue Server to Store Data in PostgreSQL

To install and configure PostgreSQL for Hue, see Connect Hue to PostgreSQL.

Connect Hue Service to PostgreSQL

Tip: To save the data in your current database (embedded or external), you must migrate (dump, synch, load) before connecting to the new database. Otherwise, skip those steps.

1. **Stop Hue Service**
   a. In Cloudera Manager, navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Stop**.

> **Note:** If necessary, refresh the page to ensure the Hue service is stopped: ◉.

2. [migration only] **Dump Current Database**

   a. Select **Actions** > **Dump Database**.
   b. Click **Dump Database**. The file is written to `/tmp/hue_database_dump.json` on the host of the Hue server.
   c. Log on to the *host of the Hue server* in a command-line terminal.
   d. Edit **`/tmp/hue_database_dump.json`** by removing all objects with `useradmin.userprofile` in the `model` field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
```

```
vi /tmp/hue_database_dump.json
```

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
  }
},
```

3. **Connect to New Database**

   a. Go to **Hue** > **Configuration**.
   b. Filter by category, **Database**.
   c. Set the following database parameters :

```
DB Hostname  = <fqdn of host with postgres server>:5432
DB Type      = <PostgreSQL>
DB Name      = hue_d
Username     = hue_u
Password     = <hue database password set when granting hue permissions>
```

   d. Click **Save Changes**.

4. [migration only] **Synchronize New Database**

   a. Select **Actions** > **Synchronize Database**
   b. Click **Synchronize Database**.

5. [migration only] **Load Data from Old Database**

   a. Log on to the *host of the PostgreSQL server* in a command-line terminal.

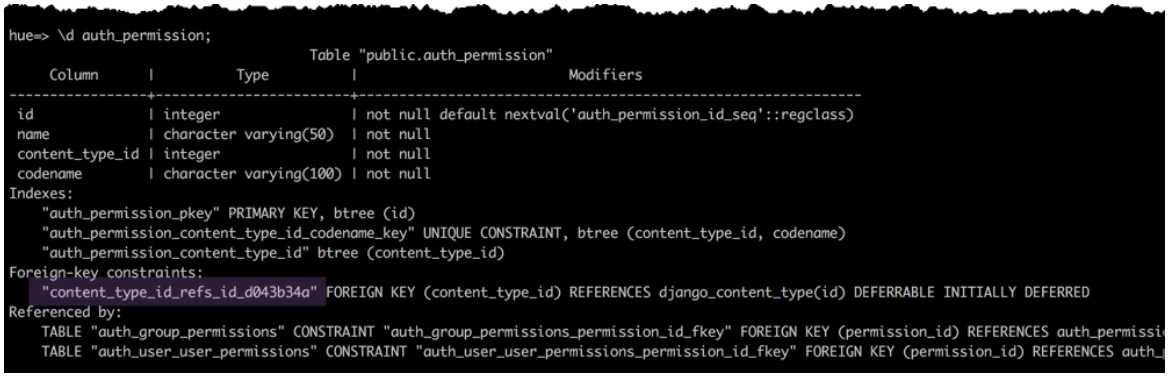
```
psql -h localhost -U hue_u -d hue_d
Password for user hue_u: <hue user password>
```

b. Drop the foreign key constraint from the `auth_permission` table in the hue database.

```
hue=# \d auth_permission;
hue=# ALTER TABLE auth_permission DROP CONSTRAINT content_type_id_refs_id_id value;
```

c. Clean the table, `django_content_type`.

```
hue=# TRUNCATE django_content_type CASCADE;
```



d. In Cloudera Manager, load the JSON file: select **Actions** > **Load Database** and click **Load Database**.
Tip: If you are blocked by a duplicate key value such as this:

```
django.db.utils.IntegrityError: Problem installing fixture '/tmp/hue_database_dump.json':
 Could not load desktop.DocumentTag(pk=1): duplicate key value violates unique constraint
 "desktop_documenttag_owner_id_1d5f76680ee9998b_uniq"
DETAIL:  Key (owner_id, tag)=(1100713, default) already exists.
```

Delete that value and try loading again, for example:

```
DELETE FROM desktop_documenttag WHERE owner_id = '1100713' and tag = 'default';
```

e. Add the foreign key back (still logged on to the Hue database):

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

6. Start Hue service

a. Navigate to **Cluster** > **Hue**, if not already there.
b. Select **Actions** > **Start**.
c. Click **Start**.
d. Click **Hue Web UI** to log on to Hue with a custom PostgreSQL database.

Configuring PostgreSQL for Oozie

Install PostgreSQL

See External PostgreSQL Database on page 76.

Create the Oozie User and Oozie Database

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

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

postgres=# CREATE ROLE oozie LOGIN ENCRYPTED PASSWORD 'oozie'
```

```
 NOSUPERUSER INHERIT CREATEDB NOCREATEROLE;
CREATE ROLE

postgres=# CREATE DATABASE "oozie" WITH OWNER = oozie
 ENCODING = 'UTF8'
 TABLESPACE = pg_default
 LC_COLLATE = 'en_US.UTF-8'
 LC_CTYPE = 'en_US.UTF-8'
 CONNECTION LIMIT = -1;
CREATE DATABASE

postgres=# \q
```

Configure PostgreSQL to Accept Network Connections for the Oozie User

1. Edit the `postgresql.conf` file and set the `listen_addresses` property to `*`, to make sure that the PostgreSQL server starts listening on all your network interfaces. Also make sure that the `standard_conforming_strings` property is set to `off`.
2. Edit the PostgreSQL `data/pg_hba.conf` file as follows:

```
host    oozie         oozie         0.0.0.0/0             md5
```

Reload the PostgreSQL Configuration

```
$ sudo -u postgres pg_ctl reload -s -D /opt/PostgreSQL/8.4/data
```

MariaDB Database

To use a MariaDB database, follow these procedures. For information on compatible versions of the MariaDB database, see CDH and Cloudera Manager Supported Databases.

Installing the MariaDB Server

Note:

- If you already have a MariaDB database set up, you can skip to the section Configuring and Starting the MariaDB Server on page 83 to verify that your MariaDB configurations meet the requirements for Cloudera Manager.
- It is important that the `datadir` directory, which, by default, is `/var/lib/mysql`, is on a partition that has sufficient free space.

1. Install the MariaDB database.

OS	Command
RHEL	`$ sudo yum install mariadb-server`
SLES	`$ sudo zypper install mariadb-server` **Note:** Some SLES systems encounter errors when using the `zypper install` command. For more information on resolving this issue, see the Novell Knowledgebase topic, error running chkconfig.
Ubuntu and Debian	`$ sudo apt-get install mariadb-server`

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

Configuring and Starting the MariaDB Server

1. Stop the MariaDB server if it is running.

```
$ sudo service mariadb stop
```

2. Move old InnoDB log files `/var/lib/mysql/ib_logfile0` and `/var/lib/mysql/ib_logfile1` out of `/var/lib/mysql/` to a backup location.
3. Determine the location of the option file, `my.cnf`.
4. Update `my.cnf` so that it conforms to the following requirements:
   - To prevent deadlocks, set the isolation level to read committed.
   - The default settings in the MariaDB installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the `innodb_flush_method` property to `O_DIRECT`.
   - Set the `max_connections` property according to the size of your cluster:
     - Small clusters (fewer than 50 hosts) - You can store more than one database (for example, both the Activity Monitor and Service Monitor) on the same host. If you do this, you should:
       - Put each database on its own storage volume.
       - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store five databases on one host (the databases for Cloudera Manager Server, Activity Monitor, Reports Manager, Cloudera Navigator, and Hive metastore), set the maximum connections to 550.
     - Large clusters (more than 50 hosts) - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts need not be reserved exclusively for databases, but each database should be on a separate host.
   - If the cluster has more than 1000 hosts, set the `max_allowed_packet` property to `16M`. Without this setting, the cluster may fail to start due to the following exception: `com.mysql.jdbc.PacketTooBigException`.
   - Although binary logging is not a requirement for Cloudera Manager installations, it provides benefits such as MariaDB replication or point-in-time incremental recovery after a database restore. The provided example configuration enables the binary log. For more information, see The Binary Log.

Here is an option file with Cloudera recommended settings:

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

key_buffer = 16M
key_buffer_size = 32M
max_allowed_packet = 32M
thread_stack = 256K
thread_cache_size = 64
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

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

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

binlog_format = mixed
```

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

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

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

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

OS	Command

RHEL	`$ sudo systemctl enable mariadb`
	`$ sudo systemctl list-unit-files | grep mariadb`
	`mariadb.service enabled`

SLES	`$ sudo chkconfig --add mariadb`
Ubuntu and Debian	`$ sudo chkconfig mariadb on` **Note:** `chkconfig` may not be available on recent Ubuntu releases. You may need to use Upstart to configure MariaDB to start automatically when the system boots. For more information, see the Ubuntu documentation or the Upstart Cookbook.

7. Start the MariaDB server:

```
$ sudo service mariadb start
```

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

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

Installing the MySQL JDBC Driver for MariaDB

The MariaDB JDBC driver is not supported. Follow the steps in this section to install and use the *MySQL* JDBC driver instead.

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

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

Note: Cloudera recommends using only version 5.1 of the JDBC driver.

OS	Command

RHEL	**Important:** Do not use the `yum install` command to install the MySQL driver package, because it installs openJDK, and then uses the Linux `alternatives` command to set the system JDK to be openJDK.
	1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html.
	2. Extract the JDBC driver JAR file from the downloaded file. For example:
	`tar zxvf mysql-connector-java-5.1.31.tar.gz`
	3. Copy the JDBC driver, renamed, to the relevant host. For example:
	`$ sudo cp mysql-connector-java-5.1.31/ mysql-connector-java-5.1.31-bin.jar /usr/share/java/ mysql-connector-java.jar`
	If the target directory does not yet exist on this host, you can create it before copying the JAR file. For example:
	`$ sudo mkdir -p /usr/share/java/`
	`$ sudo cp mysql-connector-java-5.1.31/ mysql-connector-java-5.1.31-bin.jar /usr/share/java/ mysql-connector-java.jar`

SLES	`$ sudo zypper install mysql-connector-java`
Ubuntu or Debian	`$ sudo apt-get install libmysql-java`

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

Create databases and user accounts for components that require databases:

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

  - Reports Manager

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

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

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

1. Log into MariaDB as the root user:

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

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

   Configure all databases to use the `utf8` character set.

   Include the character set for each database when you run the `CREATE DATABASE` statements described below.

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

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

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

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

Configuring the Hue Server to Store Data in MariaDB

To install and configure MariaDB for Hue, see, see Connect Hue to MySQL or MariaDB.

Connect Hue Service to MySQL

Tip: To save the data in your current database (embedded or external), you must migrate (dump, synch, load) before connecting to the new database. Otherwise, skip those steps.

1. **Stop Hue Service**
   a. In Cloudera Manager, navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Stop**.

> **Note:** Refresh the page if the Hue service does not look stopped: ⊙.

2. [migration only] **Dump Current Database**
   a. Select **Actions** > **Dump Database**.
   b. Click **Dump Database**. The file is written to `/tmp/hue_database_dump.json` on the host of the Hue server.
   c. Log on to the *host of the Hue server* in a command-line terminal.
   d. Edit **`/tmp/hue_database_dump.json`** by removing all objects with `useradmin.userprofile` in the `model` field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
```

```
vi /tmp/hue_database_dump.json
```

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
  }
},
```

3. **Connect to New Database**
   a. Go to **Hue** > **Configuration**.
   b. Filter by category, **Database**.
   c. Set the following database parameters:
      - **Hue Database Type**: MySQL
      - **Hue Database Hostname**: *FQDN of host running MySQL server*
      - **Hue Database Port**: *3306,5432, or 1521*
      - **Hue Database Username**: *username*
      - **Hue Database Password**: *password*
      - **Hue Database Name**: Hue database name or SID
   d. Click **Save Changes**.
4. [migration only] **Synchronize New Database**
   a. Select **Actions** > **Synchronize Database**
   b. Click **Synchronize Database**.
5. [migration only] **Load Data from Old Database**

a. Log on to the *host of the MySQL server* in a command-line terminal.

```
mysql -u root -p
Enter password: <root password>
```

b. Drop the foreign key constraint from the `auth_permission` table in the hue database.

```
SHOW CREATE table hue.auth_permission;
ALTER TABLE hue.auth_permission DROP FOREIGN KEY content_type_id_refs_id_id value;
```

c. Clean the table, `django_content_type`.

```
DELETE FROM hue.django_content_type;
```

d. In Cloudera Manager, load the JSON file: select **Actions** > **Load Database** and click **Load Database**.

e. Add the foreign key back:

```
ALTER TABLE hue.auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES
django_content_type (id);
```

6. **Start Hue service**

a. Navigate to **Cluster** > **Hue**, if not already there.

b. Select **Actions** > **Start**.

c. Click **Start**.

d. Click **Hue Web UI** to log on to Hue with a custom MySQL database.

Configuring MariaDB for Oozie

Install and Start MariaDB 5.5

See MariaDB Database on page 82.

Create the Oozie Database and Oozie MariaDB User

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

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

MariaDB [(none)]> create database oozie default character set utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]>  grant all privileges on oozie.* to 'oozie'@'localhost' identified by
 'oozie';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>  grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> exit
Bye
```

Add the MariaDB JDBC Driver JAR to Oozie

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

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

MySQL Database

To use a MySQL database, follow these procedures. For information on compatible versions of the MySQL database, see CDH and Cloudera Manager Supported Databases.

Installing the MySQL Server

Note:

- If you already have a MySQL database set up, you can skip to the section Configuring and Starting the MySQL Server on page 90 to verify that your MySQL configurations meet the requirements for Cloudera Manager.
- It is important that the `datadir` directory, which, by default, is `/var/lib/mysql`, is on a partition that has sufficient free space.
- Cloudera Manager installation fails if GTID-based replication is enabled in MySQL.
- For Cloudera Navigator, make sure that the MySQL server system variable `explicit_defaults_for_timestamp` is disabled (set to "0") during installation and upgrades. (MySQL 5.6.6 and later).

1. Install the MySQL database.

OS	Command

RHEL	MySQL is no longer included with RHEL. You must download the repository from the MySQL site and install it directly. You can use the following commands to install MySQL. For more information, visit the MySQL website.
	`$ wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm`
	`$ sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm`
	`$ yum update`
	`$ sudo yum install mysql-server`
	`$ sudo systemctl start mysqld`

OS	Command

SLES	`$ sudo zypper install mysql`
	`$ sudo zypper install libmysqlclient_r17`
	
	Note: Some SLES systems encounter errors when using the preceding `zypper install` command. For more information on resolving this issue, see the Novell Knowledgebase topic, error running chkconfig.

Ubuntu and Debian	`$ sudo apt-get install mysql-server`

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

Configuring and Starting the MySQL Server

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

OS	Command
RHEL	`$ sudo service mysqld stop`
SLES, Ubuntu, and Debian	`$ sudo service mysql stop`

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

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

   - The default settings in the MySQL installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the `innodb_flush_method` property to `O_DIRECT`.
   - Set the `max_connections` property according to the size of your cluster:
     - Small clusters (fewer than 50 hosts) - You can store more than one database (for example, both the Activity Monitor and Service Monitor) on the same host. If you do this, you should:
       - Put each database on its own storage volume.
       - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store five databases on one host (the databases for Cloudera Manager Server, Activity Monitor, Reports Manager, Cloudera Navigator, and Hive metastore), set the maximum connections to 550.

  - Large clusters (more than 50 hosts) - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts need not be reserved exclusively for databases, but each database should be on a separate host.

- If the cluster has more than 1000 hosts, set the `max_allowed_packet` property to `16M`. Without this setting, the cluster may fail to start due to the following exception: `com.mysql.jdbc.PacketTooBigException`.
- Binary logging is not a requirement for Cloudera Manager installations. Binary logging provides benefits such as MySQL replication or point-in-time incremental recovery after database restore. Examples of this configuration follow. For more information, see The Binary Log.

Here is an option file with Cloudera recommended settings:

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

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

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

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

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

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

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

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

sql_mode=STRICT_ALL_TABLES
```

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

OS	Command

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

SLES	`$ sudo chkconfig --add mysql`
Ubuntu and Debian	`$ sudo chkconfig mysql on` **Note:** `chkconfig` might not be available on recent Ubuntu releases. You may need to use Upstart to configure MySQL to start automatically when the system boots. For more information, see the Ubuntu documentation or the Upstart Cookbook.

8. Start the MySQL server:

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

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

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

Installing the MySQL JDBC Driver

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

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

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

Note: Cloudera recommends using only version 5.1 of the JDBC driver.

OS	Command

RHEL	**Important:** Do not use the `yum install` command to install the MySQL driver package, because it installs openJDK, and then uses the Linux `alternatives` command to set the system JDK to be openJDK.
	1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html.
	2. Extract the JDBC driver JAR file from the downloaded file. For example:
	`tar zxvf mysql-connector-java-5.1.31.tar.gz`
	3. Copy the JDBC driver, renamed, to the relevant host. For example:
	`$ sudo cp mysql-connector-java-5.1.31/ mysql-connector-java-5.1.31-bin.jar /usr/share/java/ mysql-connector-java.jar`
	If the target directory does not yet exist on this host, you can create it before copying the JAR file. For example:
	`$ sudo mkdir -p /usr/share/java/`
	`$ sudo cp mysql-connector-java-5.1.31/ mysql-connector-java-5.1.31-bin.jar /usr/share/java/ mysql-connector-java.jar`

SLES	`$ sudo zypper install mysql-connector-java`
Ubuntu or Debian	`$ sudo apt-get install libmysql-java`

Return to Establish Your Cloudera Manager Repository Strategy on page 137.

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

Create databases and user accounts for components that require databases:

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

You can create these databases on the host where the Cloudera Manager Server will run, or on any other hosts in the cluster. For performance reasons, you should install each database on the host on which the service runs, as determined

by the roles you assign during installation or upgrade. In larger deployments or in cases where database administrators are managing the databases the services use, you can separate databases from services, but use caution.

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

1. Log into MySQL as the root user:

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

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

   Configure all databases to use the `utf8` character set.

   Include the character set for each database when you run the `CREATE DATABASE` statements described below.

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

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

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

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

Configuring the Hue Server to Store Data in MySQL

To install and configure MySQL for Hue, see Connect Hue to MySQL or MariaDB.

Connect Hue Service to MySQL

Tip: To save the data in your current database (embedded or external), you must migrate (dump, synch, load) before connecting to the new database. Otherwise, skip those steps.

1. **Stop Hue Service**
   a. In Cloudera Manager, navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Stop**.

   > **Note:** Refresh the page if the Hue service does not look stopped: ◉.

2. [migration only] **Dump Current Database**
   a. Select **Actions** > **Dump Database**.

b. Click **Dump Database**. The file is written to `/tmp/hue_database_dump.json` on the host of the Hue server.
c. Log on to the *host of the Hue server* in a command-line terminal.
d. Edit **`/tmp/hue_database_dump.json`** by removing all objects with `useradmin.userprofile` in the `model` field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
```

```
vi /tmp/hue_database_dump.json
```

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
  }
},
```

3. **Connect to New Database**

a. Go to **Hue** > **Configuration**.
b. Filter by category, **Database**.
c. Set the following database parameters:

- **Hue Database Type**: MySQL
- **Hue Database Hostname**: *FQDN of host running MySQL server*
- **Hue Database Port**: *3306,5432, or 1521*
- **Hue Database Username**: *username*
- **Hue Database Password**: *password*
- **Hue Database Name**: Hue database name or SID

d. Click **Save Changes**.

4. [migration only] **Synchronize New Database**

a. Select **Actions** > **Synchronize Database**
b. Click **Synchronize Database**.

5. [migration only] **Load Data from Old Database**

a. Log on to the *host of the MySQL server* in a command-line terminal.

```
mysql -u root -p
Enter password: <root password>
```

b. Drop the foreign key constraint from the `auth_permission` table in the hue database.

```
SHOW CREATE table hue.auth_permission;
ALTER TABLE hue.auth_permission DROP FOREIGN KEY content_type_id_refs_id_id value;
```

c. Clean the table, `django_content_type`.

```
DELETE FROM hue.django_content_type;
```

```
| auth_permission | CREATE TABLE `auth_permission` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `content_type_id` int(11) NOT NULL,
  `codename` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `content_type_id` (`content_type_id`,`codename`),
  KEY `auth_permission_37ef4eb4` (`content_type_id`),
  CONSTRAINT `content_type_id_refs_id_d043b34a` FOREIGN KEY (`content_type_id`) REFERENCES `django_content_type` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=217 DEFAULT CHARSET=latin1 |
```

d. In Cloudera Manager, load the JSON file: select **Actions** > **Load Database** and click **Load Database**.

e. Add the foreign key back:

```
ALTER TABLE hue.auth_permission ADD FOREIGN KEY (content_type_id) REFERENCES
django_content_type (id);
```

6. **Start Hue service**

a. Navigate to **Cluster** > **Hue**, if not already there.

b. Select **Actions** > **Start**.

c. Click **Start**.

d. Click **Hue Web UI** to log on to Hue with a custom MySQL database.

Configuring MySQL for Oozie

Install and Start MySQL 5.x

See MySQL Database on page 89.

Create the Oozie Database and Oozie MySQL User

For example, using the MySQL `mysql` command-line tool:

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

mysql> create database oozie default character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql>  grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.00 sec)

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

mysql> exit
Bye
```

Add the MySQL JDBC Driver JAR to Oozie

Copy or symbolically link the MySQL JDBC driver JAR into one of the following directories:

- For installations that use *packages*: `/var/lib/oozie/`
- For installations that use *parcels*: `/opt/cloudera/parcels/CDH/lib/oozie/lib/`

directory.

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

Oracle Database

To use an Oracle database, follow these procedures. For information on compatible versions of the Oracle database, see CDH and Cloudera Manager Supported Databases.

Collecting Oracle Database Information

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

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

Configuring the Oracle Server

Adjusting Oracle Settings to Accommodate Larger Clusters

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

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

Reserving Ports for HiveServer 2

HiveServer2 uses port 10000 by default, but Oracle database changes the local port range. This can cause HiveServer2 to fail to start.

Manually reserve the default port for HiveServer2. For example, the following command reserves port 10000 and inserts a comment indicating the reason:

```
echo << EOF > /etc/sysctl.cnf
# HS2 uses port 10000
net.ipv4.ip_local_reserved_ports = 10000
EOF

sysctl -q -w net.ipv4.ip_local_reserved_ports=10000
```

Modifying the Maximum Number of Oracle Connections

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

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

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

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

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

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

```
transactions = 1.1 * sessions
```

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

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

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

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

Ensuring Your Oracle Database Supports UTF8

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

Installing the Oracle JDBC Connector

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

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

1. Download the Oracle JDBC Driver from the Oracle website. For example, the version 6 JAR file is named `ojdbc6.jar`.

   For more information about supported Oracle Java versions, see CDH and Cloudera Manager Supported JDK Versions.

   To download the JDBC driver, visit the Oracle JDBC and UCP Downloads page, and click on the link for your Oracle Database version. Download the `ojdbc6.jar` file.

2. Copy the Oracle JDBC JAR file to `/usr/share/java/oracle-connector-java.jar`. The Cloudera Manager databases and the Hive Mestastore database use this shared file. For example:

```
mkdir /usr/share/java
cp /tmp/ojdbc6.jar /usr/share/java/oracle-connector-java.jar
```

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

Create schema and user accounts for components that require databases:

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

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

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

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

1. Log into the Oracle client:

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

2. Create a schema and user for the Cloudera Manager Server. The minimum permissions required are:

```
create user username identified by password default tablespace tablespace;
grant CREATE SESSION to username;
grant CREATE TABLE to username;
grant CREATE SEQUENCE to username;
grant EXECUTE on sys.dbms_lob to username;
```

   where *username* and *password* are the credentials you specified in Preparing a Cloudera Manager Server External Database on page 70.

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

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

   or for unlimited space:

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

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

Role	Schema	User	Password
Activity Monitor	amon	amon	amon_password
Reports Manager	rman	rman	rman_password
Hive Metastore Server	metastore	hive	hive_password
Sentry Server	sentry	sentry	sentry_password

Role	Schema	User	Password
Cloudera Navigator Audit Server	nav	nav	nav_password
Cloudera Navigator Metadata Server	navms	navms	navms_password

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

```
create user username identified by password default tablespace tablespace;
grant CREATE SESSION to username;
grant CREATE TABLE to username;
grant CREATE SEQUENCE to username;
grant EXECUTE on sys.dbms_lob to username;
```

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

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

or for unlimited space:

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

For further information about Oracle privileges, see Authorization: Privileges, Roles, Profiles, and Resource Limitations.

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

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

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

Return to Establish Your Cloudera Manager Repository Strategy on page 137.

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

To install and configure the Oracle server and client repository for Hue, see Connect Hue to Oracle with Client Parcel

Connect Hue Service to Oracle

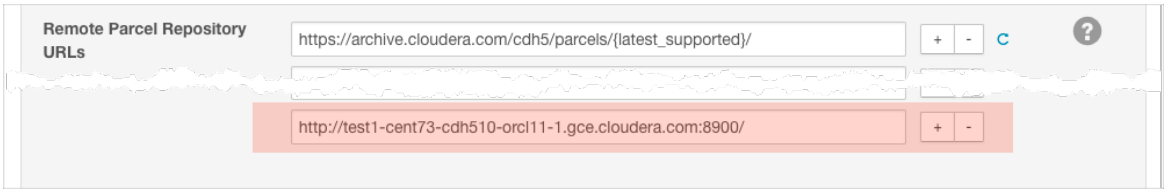
You can connect Hue to your Oracle database while installing CDH (and Hue) or with an existing installation. With existing CDH installations, you can connect and restart Hue, without saving the data in your current database, or you can migrate the old data into Oracle.

New CDH Installation

See Installing Cloudera Manager and CDH on page 54 to install Cloudera Manager (and its Installation Wizard), which you will use here to install CDH and the Oracle client.

Install CDH and Oracle Parcel

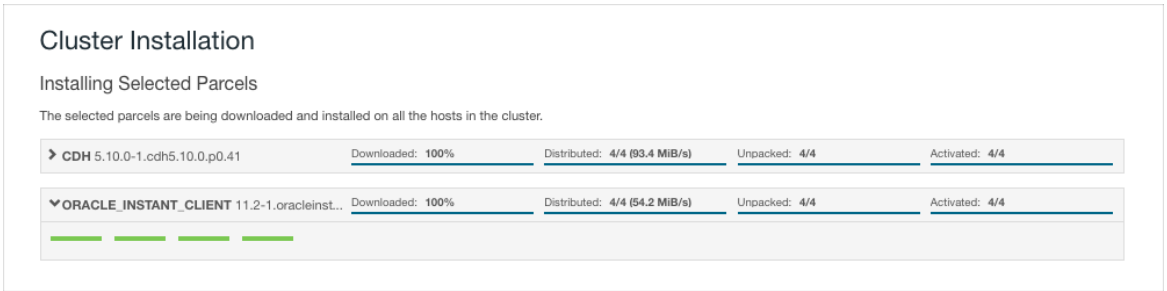
1. Open the Cloudera Manager Admin Console and run the Cloudera Manager Installation Wizard to install CDH (and Hue). The URL for Cloudera Manager is: `http://<cm server hostname>:7180`
2. Stop at **Select Repository** to add the Oracle client parcel repository (**Cluster Installation**, step 1):
   a. Choose Method **Use Parcels** and click **More Options**.
   b. +
   and add the URL for your Oracle **Remote Parcel Repository**:



c. Click **Save Changes**.

d. Select the newly added radio button by ORACLE_INSTANT_CLIENT and click **Continue**.

The Oracle parcel is downloaded, distributed, and activated at **Cluster Installation**, step 6 (**Installing Selected Parcels**).



Connect Hue to Oracle

Continuing with Cloudera Manager Installation Wizard …

1. Stop at **Database Setup** to set connection properties (**Cluster Setup**, step 3).

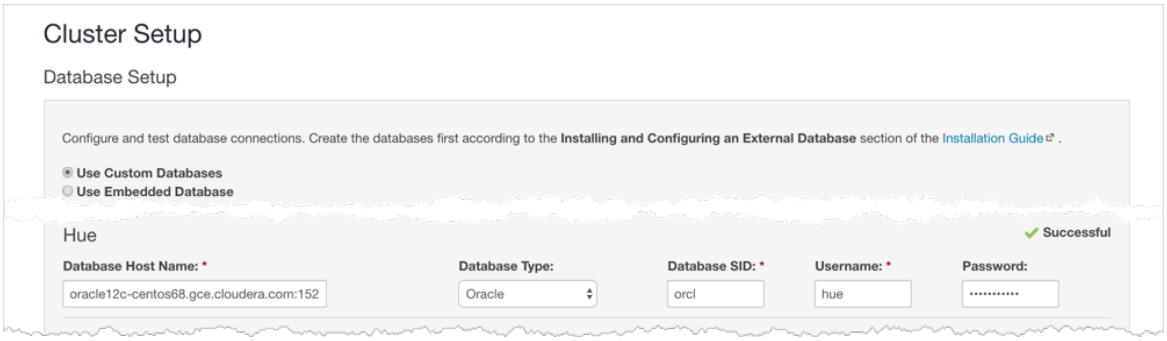
a. Select **Use Custom Database**.

b. Under **Hue**, set the connection properties to the Oracle database.

> **Note:** Copy and store the password for the Hue embedded database (just in case).

```
Database Hostname (and port): <fqdn of host with Oracle server>:1521
Database Type (or engine): Oracle
Database SID (or name): orcl
Database Username: hue
Database Password: <hue database password>
```

c. Click **Test Connection** and click **Continue** when successful.



2. Continue with the installation and click **Finish** to complete.
3. Add support for a multi-threaded environment:
   a. Go to **Clusters** > **Hue** > **Configuration**.
   b. Filter by Category, **Hue-service** and Scope, **Advanced**.
   c. Add support for a multi-threaded environment by setting **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini**:

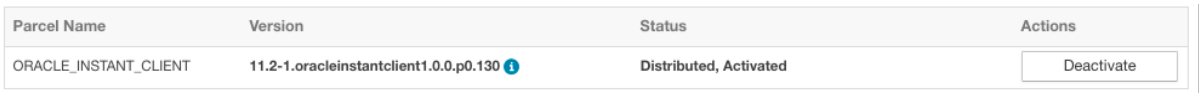
```
[desktop]
[[database]]
options={"threaded":true}
```

   d. Click **Save Changes**.
4. Restart the Hue service: select **Actions** > **Restart** and click **Restart**.
5. Log on to Hue by clicking **Hue Web UI**.

Existing CDH Installation

Activate Oracle Client Parcel

1. Log on to Cloudera Manager.
2. Go to the **Parcels** page by clicking **Hosts** > **Parcels** (or clicking the parcels icon).
3. Click the **Configuration** > **Check for New Parcels**.
4. Find ORACLE_INSTANT_CLIENT and click **Download**, **Distribute**, and **Activate**.



Connect Hue to Oracle

If you are not migrating the current (or old) database, simply connect to your new Oracle database and restart Hue (steps 3 and 6).

1. [migration only] **Stop Hue Service**
   a. In Cloudera Manager, navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Stop**.

> **Note:** If necessary, refresh the page to ensure the Hue service is stopped: .

2. [migration only] **Dump Current Database**
   a. Select **Actions** > **Dump Database**.
   b. Click **Dump Database**. The file is written to `/tmp/hue_database_dump.json` on the host of the Hue server.
   c. Log on to the *host of the Hue server* in a command-line terminal.

d. Edit **`/tmp/hue_database_dump.json`** by removing all objects with `useradmin.userprofile` in the `model` field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
```

```
vi /tmp/hue_database_dump.json
```

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
  }
},
```

3. **Connect to New Database**

   a. Configure Database connections:

   - Go to **Hue** > **Configuration** and filter by category, **Database**.
   - Set database properties and click **Save Changes**:

```
Hue Database Type (or engine): Oracle
Hue Database Hostname: <fqdn of host with Oracle server>
Hue Database Port: 1521
Hue Database Username: hue
Hue Database Password: <hue database password>
Hue Database Name (or SID): orcl
```

   b. Add support for a multi-threaded environment:

   - Filter by Category, **Hue-service** and Scope, **Advanced**.
   - Set **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini** and click **Save Changes**:

```
[desktop]
[[database]]
options={"threaded":true}
```

4. [migration only] **Synchronize New Database**

   a. Select **Actions** > **Synchronize Database**

   b. Click **Synchronize Database**.

5. [migration only] **Load Data from Old Database**

> **Important:** All user tables in the Hue database must be empty. You cleaned them at step 3 of Create Hue Database. Ensure they are still clean.

```
sqlplus hue/<your hue password> < delete_from_tables.ddl
```

6. **Re/Start Hue service**
   a. Navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Start**, and click **Start**.
   c. Click **Hue Web UI** to log on to Hue with a custom Oracle database.

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

To install and configure the Oracle server and client repository for Hue, see Connect Hue to Oracle with Client Package

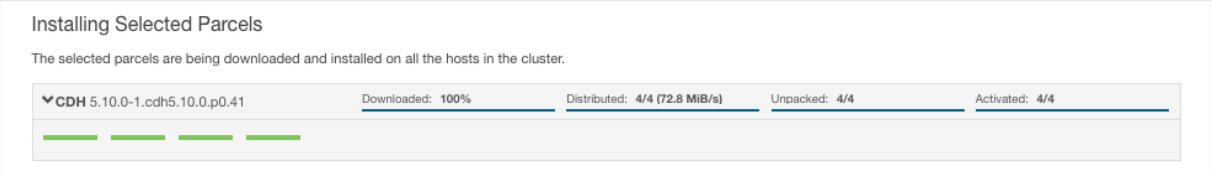
Connect Hue Service to Oracle

You can connect Hue to your Oracle database while installing CDH (and Hue) or with an existing installation. With existing CDH installations, you can connect and restart Hue, without saving the data in your current database, or you can migrate the old data into Oracle.

New CDH Installation

See Installing Cloudera Manager and CDH on page 54 to install Cloudera Manager (and its Installation Wizard), which you will use here to install CDH and the Oracle client.

1. Open the Cloudera Manager Admin Console and run the Cloudera Manager Installation Wizard to install CDH (and Hue). The URL for Cloudera Manager is: `http://<cm server hostname>:7180`
2. Stop at the end of **Cluster Installation** to copy the latest `cx_Oracle` package into Hue's Python environment.

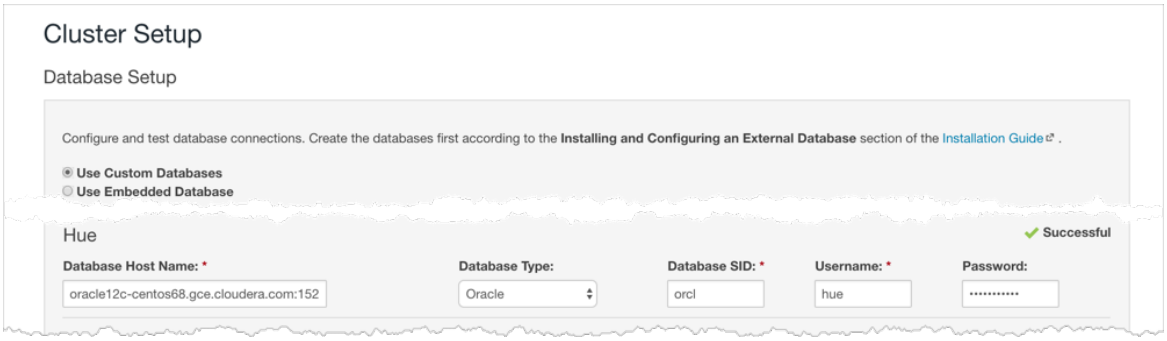


3. Stop at **Database Setup** to set connection properties (**Cluster Setup**, step 3).
   a. Select **Use Custom Database**.
   b. Under **Hue**, set the connection properties to the Oracle database.

> **Note:** Copy and store the password for the Hue embedded database (just in case).

```
Database Hostname (and port): <fqdn of host with Oracle server>:1521
Database Type (or engine): Oracle
Database SID (or name): orcl
Database Username: hue
Database Password: <hue database password>
```

   c. Click **Test Connection** and click **Continue** when successful.



4. Continue with the installation and click **Finish** to complete.
5. Add support for a multi-threaded environment:
   a. Go to **Clusters** > **Hue** > **Configuration**.
   b. Filter by Category, **Hue-service** and Scope, **Advanced**.
   c. Add support for a multi-threaded environment by setting **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini**:

```
[desktop]
[[database]]
options={"threaded":true}
```

   d. Click **Save Changes**.
6. Restart the Hue service: select **Actions** > **Restart** and click **Restart**.
7. Log on to Hue by clicking **Hue Web UI**.

Existing CDH Installation

If you are not migrating the current (or old) database, simply connect to your new Oracle database and restart Hue (steps 3 and 6).

1. [migration only] **Stop Hue Service**
   a. In Cloudera Manager, navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Stop**.

> **Note:** If necessary, refresh the page to ensure the Hue service is stopped: ◉.

2. [migration only] **Dump Current Database**
   a. Select **Actions** > **Dump Database**.
   b. Click **Dump Database**. The file is written to `/tmp/hue_database_dump.json` on the host of the Hue server.
   c. Log on to the *host of the Hue server* in a command-line terminal.
   d. Edit **`/tmp/hue_database_dump.json`** by removing all objects with `useradmin.userprofile` in the `model` field. For example:

```
# Count number of objects
grep -c useradmin.userprofile /tmp/hue_database_dump.json
```

```
vi /tmp/hue_database_dump.json
```

```
{
  "pk": 1,
  "model": "useradmin.userprofile",
  "fields": {
```

```
    "last_activity": "2016-10-03T10:06:13",
    "creation_method": "HUE",
    "first_login": false,
    "user": 1,
    "home_directory": "/user/admin"
  }
},
{
  "pk": 2,
  "model": "useradmin.userprofile",
  "fields": {
    "last_activity": "2016-10-03T10:27:10",
    "creation_method": "HUE",
    "first_login": false,
    "user": 2,
    "home_directory": "/user/alice"
  }
},
```

3. **Connect to New Database**

   a. Configure Database connections: Go to **Hue** > **Configuration**, filter by **Database**, set properties, and click **Save Changes**:

```
Hue Database Type (or engine): Oracle
Hue Database Hostname: <fqdn of host with Oracle server>
Hue Database Port: 1521
Hue Database Username: hue
Hue Database Password: <hue database password>
Hue Database Name (or SID): orcl
```

   b. Add support for a multi-threaded environment: Filter by **Hue-service**, set **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini**, and click **Save Changes**:

```
[desktop]
[[database]]
options={"threaded":true}
```

4. [migration only] **Synchronize New Database**

   a. Select **Actions** > **Synchronize Database**
   b. Click **Synchronize Database**.

5. [migration only] **Load Data from Old Database**

   Important: All user tables in the Hue database must be empty. You cleaned them at step 3 of Create Hue Database. Ensure they are still clean.

```
sqlplus hue/<your hue password> < delete_from_tables.ddl
```

6. **Re/Start Hue service**

   a. Navigate to **Cluster** > **Hue**.
   b. Select **Actions** > **Start**, and click **Start**.
   c. Click **Hue Web UI** to log on to Hue with a custom Oracle database.

Configuring Oracle for Oozie

Install and Start Oracle 11g

Use Oracle's instructions.

Create the Oozie Oracle User and Grant Privileges

The following example uses the Oracle `sqlplus` command-line tool, and shows the privileges Cloudera recommends. Oozie needs CREATE SESSION to start and manage workflows. The additional roles are needed for creating and upgrading the Oozie database.

```
$ sqlplus system@localhost

Enter password: ******

SQL> create user oozie identified by oozie default tablespace users temporary tablespace
 temp;

User created.

SQL> grant alter index to oozie;
grant alter table to oozie;
grant create index to oozie;
grant create sequence to oozie;
grant create session to oozie;
grant create table to oozie;
grant drop sequence to oozie;
grant select dictionary to oozie;
grant drop table to oozie;
alter user oozie quota unlimited on users;
alter user oozie quota unlimited on system;

SQL> exit

$
```

Important:

Do *not* make the following grant:

```
grant select any table;
```

Add the Oracle JDBC Driver JAR to Oozie

Copy or symbolically link the Oracle JDBC driver JAR into the `/var/lib/oozie/` directory.

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

Configuring an External Database for Oozie

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

- CDH 5 supported databases

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

Configuring PostgreSQL for Oozie

Install PostgreSQL

See External PostgreSQL Database on page 76.

Create the Oozie User and Oozie Database

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

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

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

postgres=# CREATE DATABASE "oozie" WITH OWNER = oozie
 ENCODING = 'UTF8'
 TABLESPACE = pg_default
 LC_COLLATE = 'en_US.UTF-8'
 LC_CTYPE = 'en_US.UTF-8'
 CONNECTION LIMIT = -1;
CREATE DATABASE

postgres=# \q
```

Configure PostgreSQL to Accept Network Connections for the Oozie User

1. Edit the `postgresql.conf` file and set the `listen_addresses` property to *, to make sure that the PostgreSQL server starts listening on all your network interfaces. Also make sure that the `standard_conforming_strings` property is set to `off`.
2. Edit the PostgreSQL `data/pg_hba.conf` file as follows:

```
host    oozie         oozie         0.0.0.0/0             md5
```

Reload the PostgreSQL Configuration

```
$ sudo -u postgres pg_ctl reload -s -D /opt/PostgreSQL/8.4/data
```

Configuring MariaDB for Oozie

Install and Start MariaDB 5.5

See MariaDB Database on page 82.

Create the Oozie Database and Oozie MariaDB User

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

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

MariaDB [(none)]> create database oozie default character set utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]>  grant all privileges on oozie.* to 'oozie'@'localhost' identified by
 'oozie';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>  grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
```

Add the MariaDB JDBC Driver JAR to Oozie

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

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

Configuring MySQL for Oozie

Install and Start MySQL 5.x

See MySQL Database on page 89.

Create the Oozie Database and Oozie MySQL User

For example, using the MySQL `mysql` command-line tool:

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

mysql> create database oozie default character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql>  grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.00 sec)

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

mysql> exit
Bye
```

Add the MySQL JDBC Driver JAR to Oozie

Copy or symbolically link the MySQL JDBC driver JAR into one of the following directories:

- For installations that use *packages*: `/var/lib/oozie/`
- For installations that use *parcels*: `/opt/cloudera/parcels/CDH/lib/oozie/lib/`

directory.

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

Configuring Oracle for Oozie

Install and Start Oracle 11g

Use Oracle's instructions.

Create the Oozie Oracle User and Grant Privileges

The following example uses the Oracle `sqlplus` command-line tool, and shows the privileges Cloudera recommends. Oozie needs CREATE SESSION to start and manage workflows. The additional roles are needed for creating and upgrading the Oozie database.

```
$ sqlplus system@localhost

Enter password: ******
```

```
SQL> create user oozie identified by oozie default tablespace users temporary tablespace
 temp;

User created.

SQL> grant alter index to oozie;
grant alter table to oozie;
grant create index to oozie;
grant create sequence to oozie;
grant create session to oozie;
grant create table to oozie;
grant drop sequence to oozie;
grant select dictionary to oozie;
grant drop table to oozie;
alter user oozie quota unlimited on users;
alter user oozie quota unlimited on system;

SQL> exit

$
```

Important:

Do *not* make the following grant:

```
grant select any table;
```

Add the Oracle JDBC Driver JAR to Oozie

Copy or symbolically link the Oracle JDBC driver JAR into the `/var/lib/oozie/` directory.

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

Configuring Oozie Data Purge Settings

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

Configuring an External Database for Sqoop

Sqoop 2 has a built-in Derby database, but Cloudera recommends that you use a PostgreSQL database instead, for the following reasons:

- Derby runs in embedded mode and it is not possible to monitor its health.
- Though it might be possible, Cloudera currently has no live backup strategy for the embedded Derby database.
- Under load, Cloudera has observed locks and rollbacks with the embedded Derby database that do not happen with server-based databases.

See CDH and Cloudera Manager Supported Databases for tested database versions.

Note:

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

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

Install PostgreSQL

See External PostgreSQL Database on page 76.

Create the Sqoop User and Sqoop Database

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

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

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

postgres=# \q
```

Configure Sqoop 2 to use PostgreSQL

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

1. Go to the Sqoop service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Sqoop 2 Server**.
4. Select **Category** > **Database**.
5. Set the following properties:
   - Sqoop Repository Database Type - postgresql
   - Sqoop Repository Database Host - the hostname on which you installed the PostgreSQL server. If the port is non-default for your database type, use host:port notation.
   - Sqoop Repository Database Name, User, Password - the properties you specified in Create the Sqoop User and Sqoop Database on page 111.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

Backing Up Databases

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

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

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

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large.
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small.
- Activity Monitor - Contains information about past activities. In large clusters, this database can grow large. Configuring an Activity Monitor database is only necessary if a MapReduce service is deployed.
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.

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

Backing Up PostgreSQL Databases

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

1. Log in to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from `/etc/cloudera-scm-server/db.properties`:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

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

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

4. Enter the password from the `com.cloudera.cmf.db.password` property in step 2.
5. To back up a database created for one of the roles described in Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server on page 78, on the local host as the *roleuser* user:

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

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

Backing Up MariaDB Databases

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

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

For example, to back up the Activity Monitor database `amon` created in Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server on page 85, on the local host as the root user, with the password `amon_password`:

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

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

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

Backing Up MySQL Databases

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

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

For example, to back up the Activity Monitor database amon created in Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server on page 93, on the local host as the root user, with the password amon_password:

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

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

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

Backing Up Oracle Databases

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

Database Vendor Resources

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

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

Data Storage for Monitoring Data

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

Monitoring Data Migration During Cloudera Manager Upgrade

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

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

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

Configuring Service Monitor Data Storage

The Service Monitor stores time series data and health data, Impala query metadata, and YARN application metadata. By default, the data is stored in `/var/lib/cloudera-service-monitor/` on the Service Monitor host. You can change this by modifying the **Service Monitor Storage Directory** configuration (`firehose.storage.base.directory`). To change this configuration on an active system, see Moving Monitoring Data on an Active Cluster on page 115.

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

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

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see Data Granularity and Time-Series Metric Data on page 114.

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

Configuring Host Monitor Data Storage

The Host Monitor stores time series data and health data. By default, the data is stored in `/var/lib/cloudera-host-monitor/` on the Host Monitor host. You can change this by modifying the **Host Monitor Storage Directory** configuration. To change this configuration on an active system, follow the procedure in Moving Monitoring Data on an Active Cluster on page 115.

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

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

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see Data Granularity and Time-Series Metric Data on page 114.

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

Viewing Host and Service Monitor Data Storage

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

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

Data Granularity and Time-Series Metric Data

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

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

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

Moving Monitoring Data on an Active Cluster

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

Basic: Changing the Configured Directory

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

Advanced: High Performance

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

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

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

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

4. Start the Service Monitor.

Host Monitor and Service Monitor Memory Configuration

You can configure Java heap size and non-Java memory size. The memory recommended for these configuration options depends on the number of hosts in the cluster, the services running on the cluster, and the number of monitored entities. Monitored entities are the objects monitored by the Service Monitor or Host Monitor. As the number of hosts and services increases, the number of monitored entities also increases.

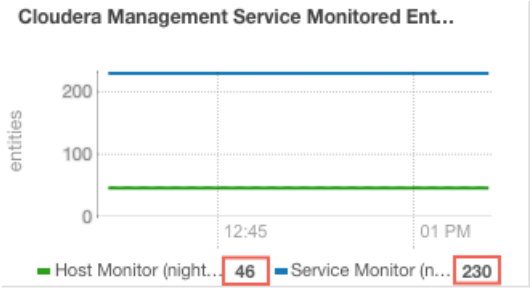
In addition to the memory configured, the Host Monitor and Service Monitor use the Linux page cache. Memory available for page caching on the Host Monitor and Service Monitor hosts improves performance.

To configure memory allocations, determine how many entities are being monitored and then consult the tables below for required and recommended memory configurations.

To determine the number of entities being monitored:

1. Go to **Clusters** > **Cloudera Management Service**.
2. Locate the chart with the title **Cloudera Management Service Monitored Entities**.

   The number of monitored entities for the Host Monitor and Service Monitor displays at the bottom of the chart. In the following example, the Host Monitor has 46 monitored entities and the Service Monitor has 230 monitored entities.



3. Use the number of monitored entities for the Host Monitor to determine its memory requirements and recommendations in the tables below.
4. Use the number of monitored entities for the Service Monitor to determine its memory requirements and recommendations in the tables below.

Clusters with HDFS, YARN, or Impala

Use the recommendations in this table for clusters where the only services having worker roles are HDFS, YARN, or Impala.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-2,000	0-100	1 GB	6 GB
2,000-4,000	100-200	1.5 GB	6 GB
4,000-8,000	200-400	1.5 GB	12 GB
8,000-16,000	400-800	2.5 GB	12 GB
16,000-20,000	800-1,000	3.5 GB	12 GB

Clusters with HBase, Solr, Kafka, or Kudu

Use the recommendations when services such as HBase, Solr, Kafka, or Kudu are deployed in the cluster. These services typically have larger quantities of monitored entities.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-30,000	0-100	2 GB	12 GB
30,000-60,000	100-200	3 GB	12 GB
60,000-120,000	200-400	3.5 GB	12 GB
120,000-240,000	400-800	8 GB	20 GB

Storage Space Planning for Cloudera Manager

Minimum Required Role: Full Administrator

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

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

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

- The cluster does not have historical operational data to meet internal requirements.
- The cluster is missing critical audit information that was not gathered nor retained for the required length of time.
- Administrators are unable to research past events or health status.
- Administrators do not have historical MR1, YARN, or Impala usage data when they need to reference or report on them later.
- There are gaps in metrics collection and charts.
- The cluster experiences data loss due to filling storage locations to 100% of capacity. The resulting damage from such an event can impact many other components.

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

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

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

Cloudera Manager Server

Table 7: Cloudera Manager Server

Entity	Cloudera Manager Server Configuration
Default Storage Location	**RDBMS:** Use any supported RDBMS to store the core configuration of your Cloudera Manager database and all cluster, service, and role configurations. See Cloudera Manager and Managed Service Datastores on page 68. **Disk:** Cloudera Manager Server Local Data Storage Directory (command_storage_path) on the host where the Cloudera Manager Server is configured to run. This local path is used by Cloudera Manager for storing data, including command result files. Critical configurations are not stored in this location. `/var/lib/cloudera-scm-server/`
Storage Configuration Defaults, Minimum, or Maximum	There are no direct storage defaults relevant to this entity.
Where to Control Data Retention or Size	The size of the Cloudera Manager Server database varies depending on the number of managed hosts and the number of discrete commands that have been run in the cluster. To configure the size of the retained command results in the Cloudera Manager Administration Console, select **Administration** > **Settings** and edit the following property: **Command Eviction Age** Length of time after which inactive commands are evicted from the database. Default is two years.

Entity	Cloudera Manager Server Configuration
Sizing, Planning & Best Practices	The Cloudera Manager Server database is the most vital configuration store in a Cloudera Manager deployment. This database holds the configuration for clusters, services, roles, and other necessary information that defines a deployment of Cloudera Manager and its managed hosts. You should perform regular, verified, remotely-stored backups of the Cloudera Manager Server database.

Cloudera Management Service

Table 8: Cloudera Management Service - Activity Monitor Configuration

Entity	Activity Monitor
Default Storage Location	Any supported RDBMS. See Cloudera Manager and Managed Service Datastores on page 68.
Storage Configuration Defaults / Minimum / Maximum	Default: 14 Days worth of MapReduce (MRv1) jobs/tasks
Where to Control Data Retention or Size	You control Activity Monitor storage usage by configuring the number of days or hours of data to retain. Older data are purged. To configure data retention in the Cloudera Manager Administration Console: 1. Go the Cloudera Management Service. 2. Click the **Configuration** tab. 3. Select **Scope** > **Activity Monitor or Cloudera Management Service (Service-Wide)**. 4. Select **Category** > **Main**. 5. Locate the ***propertyName*** property or search for it by typing its name in the Search box. **Purge Activities Data at This Age** In Activity Monitor, purge data about MapReduce jobs and aggregate activities when the data reaches this age in hours. By default, Activity Monitor keeps data about activities for 336 hours (14 days). **Purge Attempts Data at This Age** In the Activity Monitor, purge data about MapReduce attempts when the data reaches this age in hours. Because attempt data may consume large amounts of database space, you may want to purge it more frequently than activity data. By default, Activity Monitor keeps data about attempts for 336 hours (14 days). **Purge MapReduce Service Data at This Age** The number of hours of past service-level data to keep in the Activity Monitor database, such as total slots running. The default is to keep data for 336 hours (14 days). 6. Click **Save Changes** to commit the changes.
Sizing, Planning, and Best Practices	The Activity Monitor only monitors MapReduce jobs, and does not monitor not YARN applications. If you no longer use MapReduce (MRv1) in your cluster, the Activity Monitor is not required for Cloudera Manager 5 (or higher) or CDH 5 (or higher).

Entity	Activity Monitor
	The amount of storage space needed for 14 days worth of MapReduce activities can vary greatly and directly depends on the size of your cluster and the level of activity that uses MapReduce. It may be necessary to adjust and readjust the amount of storage as you determine the "stable state" and "burst state" of the MapReduce activity in your cluster. For example, consider the following test cluster and usage: • A simulated 1000-host cluster, each host with 32 slots • Synthetic MapReduce jobs with 200 attempts (tasks) per activity (job) Sizing observations for this cluster: • Each attempt takes 10 minutes to complete. • This usage results in roughly 20 thousand jobs a day with some 5 million total attempts. • For a retention period of 7 days, this Activity Monitor database required 200 GB.

Table 9: Cloudera Management Service - Service Monitor Configuration

Entity	Service Monitor Configuration
Default Storage Location	`/var/lib/cloudera-service-monitor/` on the host where the Service Monitor role is configured to run.
Storage Configuration Defaults / Minimum / Maximum	• 10 GiB Services Time Series Storage • 1 GiB Impala Query Storage • 1 GiB YARN Application Storage Total: ~12 GiB Minimum (No Maximum)
Where to Control Data Retention or Size	Service Monitor data growth is controlled by configuring the maximum amount of storage space it may use. To configure data retention in Cloudera Manager Administration Console: 1. Go the Cloudera Management Service. 2. Click the **Configuration** tab. 3. Select **Scope** > **Service Monitor or Cloudera Management Service (Service-Wide)**. 4. Select **Category** > **Main**. 5. Locate the ***propertyName*** property or search for it by typing its name in the Search box. **Time-Series Storage** The approximate amount of disk space dedicated to storing time series and health data. When the store has reached its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data once it reaches the limit. Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. The Service Monitor stores metric data not only as

Entity	Service Monitor Configuration
	raw data points but also as ten-minute, hourly, six-hourly, daily, and weekly summary data points. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time while weekly summary points are unlikely to ever be deleted. Select **Cloudera Management Service** > **Charts Library** tab in Cloudera Manager for information about how space is consumed within the Service Monitor. These pre-built charts also show information about the amount of data retained and time window covered by each data granularity. **Impala Storage** The approximate amount of disk space dedicated to storing Impala query data. When the store reaches its maximum size, it deletes older to make room for newer queries. The disk usage is approximate because the store only begins deleting data when it reaches the limit. **YARN Storage** The approximate amount of disk space dedicated to storing YARN application data. Once the store reaches its maximum size, it deletes older data to make room for newer applications. The disk usage is approximate because Cloudera Manager only begins deleting data when it reaches the limit. 6. Click **Save Changes** to commit the changes.
Sizing, Planning, and Best Practices	The Service Monitor gathers metrics about configured roles and services in your cluster and also runs active health tests. These health tests run regardless of idle and use periods, because they are always relevant. The Service Monitor gathers metrics and health test results regardless of the level of activity in the cluster. This data continues to grow, even in an idle cluster.

Table 10: Cloudera Management Service - Host Monitor

Entity	Host Monitor
Default Storage Location	`/var/lib/cloudera-host-monitor/` on the host where the Host Monitor role is configured to run.
Storage Configuration Defaults / Minimum/ Maximum	Default + Minimum: 10 GiB Host Time Series Storage
Where to Control Data Retention or Size	Host Monitor data growth is controlled by configuring the maximum amount of storage space it may use. See Data Storage for Monitoring Data on page 113. To configure these data retention in Cloudera Manager Administration Console: 1. Go the Cloudera Management Service. 2. Click the **Configuration** tab. 3. Select **Scope** > **Host Monitor or Cloudera Management Service (Service-Wide)**. 4. Select **Category** > **Main**. 5. Locate each property or search for it by typing its name in the Search box.

Entity	Host Monitor
	Time-Series Storage The approximate amount of disk space dedicated to storing time series and health data. When the store reaches its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data when it reaches the limit. Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. Host Monitor stores metric data not only as raw data points but also ten-minutely, hourly, six-hourly, daily, and weekly summary data points. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time, while weekly summary points are unlikely to ever be deleted. See the **Cloudera Management Service** > **Charts Library** tab in Cloudera Manager for information on how space is consumed within the Host Monitor. These pre-built charts also show information about the amount of data retained and the time window covered by each data granularity. 6. Click **Save Changes** to commit the changes.
Sizing, Planning and Best Practices	The Host Monitor gathers metrics about host-level items of interest (for example: disk space usage, RAM, CPU usage, swapping, etc) and also informs host health tests. The Host Monitor gathers metrics and health test results regardless of the level of activity in the cluster. This data continues to grow fairly linearly, even in an idle cluster.

Table 11: Cloudera Management Service - Event Server

Entity	Event Server
Default Storage Location	`/var/lib/cloudera-scm-eventserver/` on the host where the Event Server role is configured to run.
Storage Configuration Defaults	5,000,000 events retained
Where to Control Data Retention or Minimum /Maximum	The amount of storage space the Event Server uses is influenced by configuring how many discrete events it may retain. To configure data retention in Cloudera Manager Administration Console, 1. Go the Cloudera Management Service. 2. Click the **Configuration** tab. 3. Select **Scope** > **Event Server or Cloudera Management Service (Service-Wide)**. 4. Select **Category** > **Main**. 5. Edit the following property: **Maximum Number of Events in the Event Server Store** The maximum size of the Event Server store, in events. Once this size is exceeded, events are deleted starting with the oldest first until the size of the store is below this threshold 6. Click **Save Changes** to commit the changes.
Sizing, Planning, and Best Practices	The Event Server is a managed Lucene index that collects relevant events that happen within your cluster, such as results of health tests, log events that are

Entity	Event Server
	created when a log entry matches a set of rules for identifying messages of interest and makes them available for searching, filtering and additional action. You can view and filter events on the **Diagnostics** > **Events** tab of the Cloudera Manager Administration Console. You can also poll this data using the Cloudera Manager API. **Note:** The Cloudera Management Service role Alert Publisher sources all the content for its work by regularly polling the Event Server for entries that are marked to be sent out using SNMP or SMTP(S). The Alert Publisher is not discussed because it has no noteworthy storage requirements of its own.

Table 12: Cloudera Management Service - Reports Manager

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

Cloudera Navigator

Table 13: Cloudera Navigator - Navigator Audit Server

Entity	Navigator Audit Server
Default Storage Location	Any Supported RDBMS. See Installing and Configuring Databases.
Storage Configuration Defaults	Default: 90 Days retention
Where to Control Data Retention or Min/Max	Navigator Audit Server storage usage is controlled by configuring how many days of data it may retain. Any older data are purged. To configure data retention in the Cloudera Manager Administration Console: 1. Go the Cloudera Management Service. 2. Click the **Configuration** tab. 3. Select **Scope** > **Navigator Audit Server or Cloudera Management Service (Service-Wide)**. 4. Select **Category** > **Main**. 5. Locate the **Navigator Audit Server Data Expiration Period** property or search for it by typing its name in the Search box. **Navigator Audit Server Data Expiration Period** In Navigator Audit Server, purge audit data of various auditable services when the data reaches this age in days. By default, Navigator Audit Server keeps data about audits for 90 days. 6. Click **Save Changes** to commit the changes.
Sizing, Planning, and Best Practices	The size of the Navigator Audit Server database directly depends on the number of audit events the cluster's audited services generate. Normally the volume of HDFS audits exceed the volume of other audits (all other components like MRv1, Hive and Impala read from HDFS, which generates additional audit events). The average size of a discrete HDFS audit event is ~1 KB. For a busy cluster of 50 hosts with ~100K audit events generated per hour, the Navigator Audit Server database would consume ~2.5 GB per day. To retain 90 days of audits at that level, plan for a database size of around 250 GB. If other configured cluster services generate roughly the same amount of data as the HDFS audits, plan for the Navigator Audit Server database to require around 500 GB of storage for 90 days of data. Notes: • Individual Hive and Impala queries themselves can be very large. Since the query itself is part of an audit event, such audit events consume space in proportion to the length of the query. • The amount of space required increases as activity on the cluster increases. In some cases, Navigator Audit Server databases can exceed 1TB for 90 days of audit events. Benchmark your cluster periodically and adjust accordingly. Use this table to map Product Compatibility Matrix for Cloudera Navigator versions to Cloudera Manager versions.

Table 14: Cloudera Navigator - Navigator Metadata Server

Entity	Navigator Metadata Server
Default Storage Location	**RDBMS:** Any Supported RDBMS. See Installing and Configuring Databases. **Disk:** `/var/lib/cloudera-scm-navigator/` on the host where the Navigator Metadata Server role is configured to run.
Storage Configuration Defaults	**RDBMS:** There are no exposed defaults or configurations to directly cull or purge the size of this data set. **Disk:** There are no configuration defaults to influence the size of this location. You can change the location itself with the **Navigator Metadata Server Storage Dir** property. The size of the data in this location depends on the amount of metadata in the system (HDFS fsimage size, Hive Metastore size) and activity on the system (the number of MapReduce Jobs run, Hive queries executed, etc).
Where to Control Data Retention or Min/Max	**RDBMS:** The Navigator Metadata Server database should be carefully tuned to support large volumes of metadata. **Disk:** The Navigator Metadata Server index (an embedded Solr instance) can consume lots of disk space at the location specified for the **Navigator Metadata Server Storage Dir** property. Ongoing maintenance tasks include Purging the Navigator Metadata Server of Deleted and Stale Metadata
Sizing, Planning, and Best Practices	**Memory:** See Memory Sizing Considerations for Navigator Metadata Server for details. **RDBMS:** The database is used to store policies and authorization data. The dataset is small, but this database is also used during a Solr schema upgrade, where Solr documents are extracted and inserted again in Solr. This has same space requirements as above use case, but the space is only used temporarily during product upgrades. Use the Product Compatibility Matrix for Cloudera Navigator product compatibility matrix to map Cloudera Navigator and Cloudera Manager versions. **Disk:** This filesystem location contains all the metadata that is extracted from managed clusters. The data is stored in Solr, so this is the location where Solr stores its index and documents. Depending on the size of the cluster, this data can occupy tens of gigabytes. A guideline is to look at the size of HDFS fsimage and allocate two to three times that size as the initial size. The data here is

Entity	Navigator Metadata Server
	incremental and continues to grow as activity is performed on the cluster. The rate of growth can be on order of tens of megabytes per day.

General Performance Notes

When possible:

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

Cluster Lifecycle Management with Cloudera Manager

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

Table 15: Parcel Lifecycle Management

Parcel Lifecycle Path (default)	Notes
Local Parcel Repository Path `/opt/cloudera/parcel-repo`	This path exists only on the host where Cloudera Manager Server (`cloudera-scm-server`) runs. The Cloudera Manager Server stages all new parcels in this location as it fetches them from any external repositories. Cloudera Manager Agents are then instructed to fetch the parcels from this location when the administrator distributes the parcel using the Cloudera Manager Administration Console or the Cloudera Manager API. **Sizing and Planning** The default location is `/opt/cloudera/parcel-repo` but you can configure another local filesystem location on the host where Cloudera Manager Server runs. See Parcel Configuration Settings on page 40. Provide sufficient space to hold all the parcels you download from all configured Remote Parcel Repository URLs (See Parcel Configuration Settings on page 40). Cloudera Manager deployments that manage multiple clusters store all applicable parcels for all clusters. Parcels are provided for each operating system, so be aware that heterogeneous clusters (distinct operating systems represented in the cluster) require more space than clusters with homogeneous operating systems. For example, a cluster with both RHEL5.x and 6.x hosts must hold -el5 and -el6 parcels in the Local Parcel Repository Path, which requires twice the amount of space. **Lifecycle Management and Best Practices** Delete any parcels that are no longer in use from the Cloudera Manager Administration Console, (never delete them manually from the command line) to recover disk space in the Local Parcel Repository Path and simultaneously across all managed cluster hosts which hold the parcel. **Backup Considerations** Perform regular backups of this path, and consider it a non-optional accessory to backing up Cloudera Manager Server. If you migrate Cloudera Manager Server to a new host or restore it from a backup (for example, after a hardware failure), recover the full content of this path to the new host, in the

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

Table 16: Management Service Lifecycle - Space Reclamation Tasks

Task	Description
Activity Monitor (One-time)	The Activity Monitor only works against a MapReduce (MR1) service, not YARN. So if your deployment has fully migrated to YARN and no longer uses

Task	Description
	a MapReduce (MR1) service, your Activity Monitor database is no longer growing. If you have waited longer than the default Activity Monitor retention period (14 days) to address this point, then the Activity Monitor has already purged it all for you and your database is mostly empty. If your deployment meets these conditions, consider cleaning up by dropping the Activity Monitor database (again, only when you are satisfied that you no longer need the data or have confirmed that it is no longer in use) and the Activity Monitor role.
Service Monitor and Host Monitor (One-time)	For those who used Cloudera Manager version 4.x and have now upgraded to version 5.x: The Service Monitor and Host Monitor were migrated from their previously-configured RDBMS into a dedicated time series store used solely by each of these roles respectively. After this happens, there is still legacy database connection information in the configuration for these roles. This was used to allow for the initial migration but is no longer being used for any active work. After the above migration has taken place, the RDBMS databases previously used by the Service Monitor and Host Monitor are no longer used. Space occupied by these databases is now recoverable. If appropriate in your environment (and you are satisfied that you have long-term backups or do not need the data on disk any longer), you can drop those databases.
Ongoing Space Reclamation	Cloudera Management Services are automatically rolling up, purging or otherwise consolidating aged data for you in the background. Configure retention and purging limits per-role to control how and when this occurs. These configurations are discussed per-entity above. Adjust the default configurations to meet your space limitations or retention needs.

Log Files

All CDH cluster hosts write out separate log files for each role instance assigned to the host. Cluster administrators can monitor and manage the disk space used by these roles and configure log rotation to prevent log files from consuming too much disk space.

See Managing Disk Space for Log Files.

Conclusion

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

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

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

Before proceeding with this path for a new installation, review Cloudera Manager Deployment on page 54. If you are upgrading an existing Cloudera Manager installation, see Cloudera Upgrade Overview.

In Installation Path A, Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, Cloudera Manager Agent, CDH, and managed service software on cluster hosts. Cloudera Manager also configures databases for the Cloudera Manager Server and Hive Metastore and optionally for Cloudera Management Service roles. This path is recommended for demonstration and proof-of-concept deployments,

but is *not recommended* for production deployments because its not intended to scale and may require database migration as your cluster grows. To use this method, server and cluster hosts must satisfy the following requirements:

The automated installer binary (`cloudera-manager-installer.bin`) installs the highest version of Cloudera Manager. To use a lower version, you must install Cloudera Manager using packages. See Installing Lower Versions of Cloudera Manager 5 on page 174.

- Provide the ability to log in to the Cloudera Manager Server host using a root account or an account that has password-less sudo permission.
- Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See CDH and Cloudera Manager Networking and Security Requirements for further information.
- All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.

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

Before You Begin

Install and Configure Databases

By default, Installation Path A installs an embedded PostgreSQL database. You can also choose to configure an external database. Read Cloudera Manager and Managed Service Datastores on page 68. If you are using an external database for services or Cloudera Management Service roles, install and configure it following the instructions in External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server on page 73.

Perform Configuration Required by Single User Mode

If you are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in Configuring Single User Mode on page 59.

On CentOS 5 and RHEL 5, Install Python 2.6/2.7 and psycopg2 for Hue

Hue in CDH 5 only works with the operating system's native version of Python *when that version is 2.6 and higher*.

CentOS/RHEL 5 ships with Python 2.4 so you must install Python 2.6 (or Python 2.7) and the Python-PostgreSQL Database Adapter, psycopg2 (not psycopg).

If the Hue server is already installed, you must import the psycopg2 connector into Hue's environment **or** create a symbolic link.

```
## Navigate to Hue within your specific CDH parcel version
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print
 $9}'`/lib/hue/build/env/bin
./python2.6
>>>> import psycopg2
```

or ...

```
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print
 $9}'`/lib/hue/build/env/lib/python2.6/site-packages/
ln -s /usr/lib64/python2.6/site-packages/psycopg2 psycopg2
```

Configure an HTTP Proxy

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

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

Install the Oracle JDK

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

- CDH 5 - Java Development Kit Installation on page 58.
- CDH 4 - Java Development Kit Installation.

Download and Run the Cloudera Manager Server Installer

Download the Cloudera Manager installer to the cluster host to which you are installing the Cloudera Manager Server:

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

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

   The **Cloudera Standard License** page displays.
5. Accept the license agreement and click **Submit**.

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

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

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

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

8. Run the Cloudera Manager Server installer by doing one of the following:
   - Install Cloudera Manager packages from the Internet:

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

   - Install Cloudera Manager packages from a local repository:

```
$ sudo ./cloudera-manager-installer.bin --skip_repo_package=1
```

9. Read the Cloudera Manager README and then press **Return** or **Enter** to choose **Next**.
10. Read the Cloudera Express License and then press **Return** or **Enter** to choose **Next**. Use the arrow keys and press **Return** or **Enter** to choose **Yes** to confirm you accept the license.
11. Read the Oracle Binary Code License Agreement and then press **Return** or **Enter** to choose **Next**.
12. Use the arrow keys and press **Return** or **Enter** to choose **Yes** to confirm you accept the Oracle Binary Code License Agreement. The following occurs:

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

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

Note: If the installation is interrupted, you may need to clean up before you can re-run it. See Uninstalling Cloudera Manager and Managed Software on page 196.

Start and Log into the Cloudera Manager Admin Console

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

1. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.
2. In a web browser, enter `http://Server host:7180`, where *Server host* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.

3. Log into Cloudera Manager Admin Console. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can change the password using Cloudera Manager after you run the installation wizard. Although you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.
4. After you log in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

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

Use the Cloudera Manager Wizard for Software Installation and Configuration

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

- Select the edition of Cloudera Manager to install
- Find the cluster hosts you specify using hostname and IP address ranges
- Connect to each host with SSH to install the Cloudera Manager Agent and other components
- Optionally install the Oracle JDK on the cluster hosts.
- Install CDH and managed service packages or parcels
- Configure CDH and managed services automatically and start the services

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

- Internet access to allow the wizard to install software packages or parcels from `archive.cloudera.com`.
- A custom internal repository that the host(s) can access. For example, for a Red Hat host, you could set up a Yum repository. See Creating and Using a Package Repository for Cloudera Manager on page 171 for more information about this option.

Choose Cloudera Manager Edition

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

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

   If you choose Cloudera Express or Cloudera Enterprise Enterprise Data Hub Edition Trial, you can upgrade the license at a later time. See Managing Licenses.
2. If you elect Cloudera Enterprise, install a license:
   a. Click **Upload License**.
   b. Click the document icon to the left of the **Select a License File** text field.
   c. Go to the location of your license file, click the file, and click **Open**.
   d. Click **Upload**.
3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.
4. Click **Continue** to proceed with the installation.

Choose Cloudera Manager Hosts

Choose which hosts will run CDH and managed services:

1. To enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the cluster hostnames or IP addresses. You can also specify hostname and IP address ranges. For example:

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

   You can specify multiple addresses and address ranges by separating them with commas, semicolons, tabs, or blank spaces, or by placing them on separate lines. Use this technique to make more specific searches instead of searching overly wide ranges. The scan results will include all addresses scanned, but only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. If you do not know the IP addresses of all of the hosts, you can enter an address range that spans over unused addresses and then clear the hosts that do not exist (and are not discovered) later in this procedure. However, keep in mind that wider ranges will require more time to scan.
2. Click **Search**. Cloudera Manager identifies the hosts on your cluster to allow you to configure them for services. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking **Abort Scan**. To find additional hosts, click **New Search**, add the host names or IP addresses and click **Search** again. Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts. Common causes of loss of connectivity are firewalls and interference from SELinux.

3. Verify that the number of hosts shown matches the number of hosts where you want to install services. Clear host entries that do not exist and clear the hosts where you do not want to install services.

Choose Software Installation Method and Install Software

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

1. Select the repository type to use for the installation. In the **Choose Method** section select one of the following:
   - **Use Parcels**
     1. Choose the parcels to install. The choices you see depend on the repositories you have chosen – a repository may contain multiple parcels. Only the parcels for the latest supported service versions are configured by default.

        You can add additional parcels for previous versions by specifying custom repositories. For example, you can find the locations of the previous CDH 4 parcels at `https://archive.cloudera.com/cdh4/parcels/`. Or, if you are installing CDH 4.3 and want to use policy-file authorization, you can add the Sentry parcel using this mechanism.

        1. To specify the **Parcel Directory** or **Local Parcel Repository Path**, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:
           - **Parcel Directory** and **Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host.
           - **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the ➕ button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the Configuring Cloudera Manager Server Parcel Settings on page 40 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you will see all the unique parcels contained in all your repositories.
           - **Proxy Server** - Specify the properties of a proxy server.
        2. Click **OK**. Parcels available from the configured remote parcel repository URLs are displayed in the parcels list. If you specify a URL for a parcel version too new to be supported by the Cloudera Manager version, the parcel is not displayed in the parcel list.

   - **Use Packages**
     1. Select the major release of CDH to install.
     2. Select the specific release of CDH to install. You can choose either the latest version, a specific version, or use a custom repository. If you specify a custom repository for a CDH version too new to be supported by the Cloudera Manager version, Cloudera Manager will install the packages but you will not be able to create services using those packages and will have to manually uninstall those packages and manually reinstall packages for a supported CDH version.
     3. Select the specific releases of other services to install. You can choose either the latest version or use a custom repository. Choose **None** if you do not want to install that service.

2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories.
3. Click **Continue**.

   The **Cluster Installation JDK Installation Options** screen displays.

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

strength encryption, and you are running a secure cluster, select the **Install Java Unlimited Strength Encryption Policy Files** checkbox.

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

5. Click **Continue**.
6. (Optional) Select **Single User Mode** to configure the Cloudera Manager Agent and all service processes to run as the same user. This mode requires extra configuration steps that must be done manually on all hosts in the cluster. If you have not performed the steps, directory creation will fail in the installation wizard. In most cases, you can create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**.
7. Specify host installation properties:
   - Select **root** or enter the username for an account that has password-less sudo permission.
   - Select an authentication method:
     - If you choose password authentication, enter and confirm the password.
     - If you choose public-key authentication, provide a passphrase and path to the required key files.
   - You can specify an alternate SSH port. The default value is 22.
   - You can specify the maximum number of host installations to run at once. The default value is 10.
8. Click **Continue**. Cloudera Manager performs the following:
   - **Parcels** - Installs the Oracle JDK and the Cloudera Manager Agent packages and starts the Agent. . During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.
   - **Packages** - Configures package repositories, installs the Oracle JDK, CDH and managed service and the Cloudera Manager Agent packages, and starts the Agent. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed. If the installation has completed successfully on some hosts but failed on others, you can click **Continue** if you want to skip installation on the failed hosts and continue to the next screen to start configuring services on the successful hosts.

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

   The Host Inspector runs to validate the installation and provides a summary of the results, including all the versions of the installed components. If the validation is successful, click **Finish**.

Add Services

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

CDH 4	CDH 5
• **Core Hadoop** - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • **Core with HBase** • **Core with Impala**	• **Core Hadoop** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, and Hue • **Core with HBase** • **Core with Impala**

CDH 4	CDH 5
• **All Services** - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • **Custom Services** - Any combination of services.	• **Core with Search** • **Core with Spark** • **All Services** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • **Custom Services** - Any combination of services.

Keep the following in mind:

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

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

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

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

- The Flume service can be added only after your cluster has been set up.

- If you have chosen Enterprise Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See Cloudera Navigator Data Management Overview.

2. Click **Continue**.
3. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

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

   The following shortcuts for specifying hostname patterns are supported:

   - Range of hostnames (without the domain portion)

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

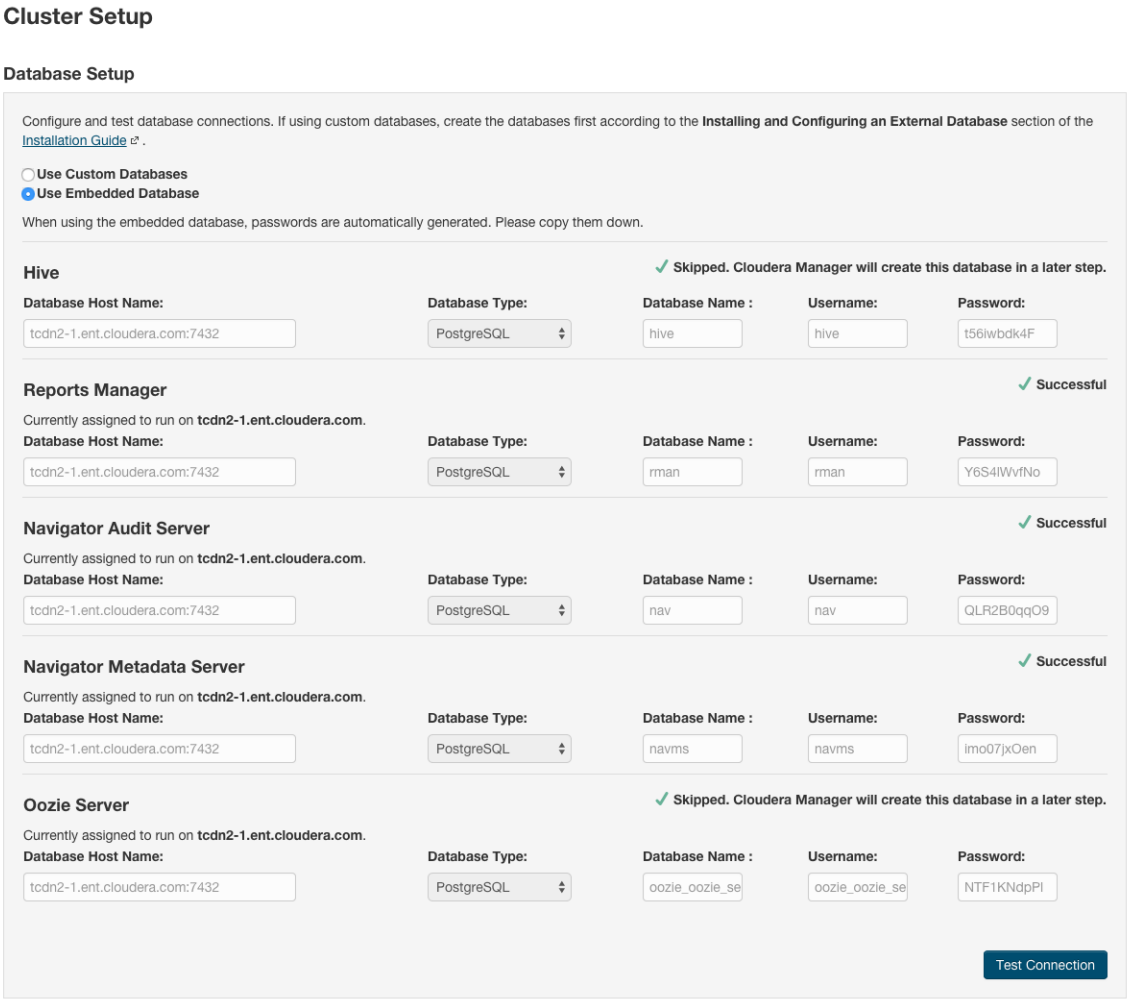
   - IP addresses
   - Rack name

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

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

Configure Database Settings

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



   - Select **Use Custom Databases** to specify the external database host and enter the database type, database name, username, and password for the custom database.
   - If you are adding the Oozie service, you can change your Oozie configuration to control when data is purged to improve performance, reduce database disk usage, improve upgrade performance, or to keep the history for a longer period of time. See Configuring Oozie Data Purge Settings Using Cloudera Manager.

2. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

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

Review Configuration Changes and Start Services

1. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use

the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

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

2. Click **Continue**.

   The wizard starts the services.

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

Change the Default Administrator Password

As soon as possible, change the default administrator password:

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

Configure Oozie Data Purge Settings

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

Test the Installation

You can test the installation following the instructions in Testing the Installation on page 195.

Installation Path B - Installation Using Cloudera Manager Parcels or Packages

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

- Install these items manually using packages. You can use utilities such as Puppet or Chef to help with the installation of these items across all the hosts in a cluster.
- Cloudera Manager can install them for you on all of the hosts in your cluster. If you choose Cloudera Manager installation, you can select installation using *packages* or Cloudera Manager *parcels*. In order for Cloudera Manager to automate installation of Cloudera Manager Agent packages or CDH and managed service software, cluster hosts must satisfy the following requirements:
  - Allow the Cloudera Manager Server host to have uniform SSH access on the same port to all hosts. See CDH and Cloudera Manager Networking and Security Requirements for further information.
  - All hosts must have access to standard package repositories and either `archive.cloudera.com` or a local repository with the required installation files.

You can also install Cloudera Manager and CDH using tarballs. See Installation Path C - Manual Installation Using Cloudera Manager Tarballs on page 149.

Before proceeding with this path for a new installation, review Cloudera Manager Deployment on page 54. If you are upgrading a Cloudera Manager existing installation, see Cloudera Upgrade Overview.

> **Note:** Cloudera does not support CDH cluster deployments using hosts in Docker containers.

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

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

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

For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

Before You Begin

Perform Configuration Required by Single User Mode

If you are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in Configuring Single User Mode on page 59.

On CentOS 5 and RHEL 5, Install Python 2.6/2.7 and psycopg2 for Hue

Hue in CDH 5 only works with the operating system's native version of Python *when that version is 2.6 and higher*.

CentOS/RHEL 5 ships with Python 2.4 so you must install Python 2.6 (or Python 2.7) and the Python-PostgreSQL Database Adapter, psycopg2 (not psycopg).

If the Hue server is already installed, you must import the psycopg2 connector into Hue's environment **or** create a symbolic link.

```
## Navigate to Hue within your specific CDH parcel version
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print
 $9}'`/lib/hue/build/env/bin
./python2.6
>>>> import psycopg2
```

or ...

```
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print
 $9}'`/lib/hue/build/env/lib/python2.6/site-packages/
ln -s /usr/lib64/python2.6/site-packages/psycopg2 psycopg2
```

Install and Configure External Databases

Read Cloudera Manager and Managed Service Datastores on page 68. Install and configure an external database for services or Cloudera Management Service roles using the instructions in External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server on page 73.

Cloudera Manager also requires a database. Prepare the Cloudera Manager Server database as described in Preparing a Cloudera Manager Server External Database on page 70.

Establish Your Cloudera Manager Repository Strategy

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

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems. For Cloudera Manager repository locations and client repository files, see .

- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see Understanding Custom Installation Solutions on page 166. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

RHEL-compatible

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

   See the **Repo File** column in the Table 1 table for the URL.

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

SLES

1. Update your system package index by running:

```
$ sudo zypper refresh
```

2. Run the appropriate command for your version of SLES.

   See the **Repo File** column in the Table 1 table for the URL.

   For example:

OS Version	Command
SLES 11:	`$ sudo zypper addrepo -f https://archive.cloudera.com/cm5/sles/11/x86_64/cm/cloudera-manager.repo`
SLES 12:	`$ sudo zypper addrepo -f https://archive.cloudera.com/cm5/sles/12/x86_64/cm/cloudera-cm.repo`

Ubuntu or Debian

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

   See the **Repo File** column in the Table 1 table for the URL.

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

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

Install Cloudera Manager Server Software

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

Install the Oracle JDK on the Cloudera Manager Server Host

> **Note:** Cloudera, Inc. acquired Oracle JDK software under the Oracle Binary Code License Agreement. Pursuant to Item D(v)(a) of the SUPPLEMENTAL LICENSE TERMS of the Oracle Binary Code License Agreement, use of JDK software is governed by the terms of the Oracle Binary Code License Agreement. By installing the JDK software, you agree to be bound by these terms. If you do not wish to be bound by these terms, then do not install the Oracle JDK.

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

- **Install the JDK from a repository**

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

OS	Command
RHEL	`$ sudo yum install oracle-j2sdk1.7`
SLES	`$ sudo zypper install oracle-j2sdk1.7`
Ubuntu or Debian	`$ sudo apt-get install oracle-j2sdk1.7`

- **Install the JDK manually**

  See Java Development Kit Installation on page 58.

Install the Cloudera Manager Server Packages

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

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

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

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

You can use Cloudera Manager to install the Oracle JDK, Cloudera Manager Agent packages, CDH, and managed service packages or you can install any of these packages manually. To use Cloudera Manager to install the packages, you must meet the requirements described in Cloudera Manager Deployment on page 54.

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

If you are going to use Cloudera Manager to install all of the software, *skip this section* and continue with Start the Cloudera Manager Server on page 141. Otherwise, to manually install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Services, continue with the procedures linked below and then return to this page to continue the installation. in this section. You can choose to manually install any of the following software and, in a later step, Cloudera Manager installs any software that you do not install manually:

Manually Install the Oracle JDK

You can use Cloudera Manager to install the Oracle JDK on all cluster hosts or you can install the JDKs manually. If you choose to have Cloudera Manager install the JDKs, *skip this section*. To use Cloudera Manager to install the JDK, you must meet the requirements described in Cloudera Manager Deployment on page 54.

Install the Oracle JDK on every cluster hosts. For more information, see Java Development Kit Installation on page 58.

Manually Install Cloudera Manager Agent Packages

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

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

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

OS	Command
RHEL, if you have a yum repo configured:	`$ sudo yum install cloudera-manager-agent cloudera-manager-daemons`
RHEL, if you're manually transferring RPMs:	`$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons`
SLES	`$ sudo zypper install cloudera-manager-agent cloudera-manager-daemons`
Ubuntu or Debian	`$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons`

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

Property	Description
`server_host`	Name of the host where Cloudera Manager Server is running.
`server_port`	Port on the host where Cloudera Manager Server is running.

   For more information on Agent configuration options, see Agent Configuration File.
3. Start the Agents by running the following command on all hosts:

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

When the Agent starts, it contacts the Cloudera Manager Server. If communication fails between a Cloudera Manager Agent and Cloudera Manager Server, see Troubleshooting Installation and Upgrade Problems on page 425. When the Agent hosts reboot, `cloudera-scm-agent` starts automatically.

Manually Install CDH and Managed Service Packages

The CDH and Managed Service Packages contain all of the CDH software. You can choose to manually install CDH and the Managed Service Packages, or you can choose to let Cloudera Manager perform this installation in a later step. To use Cloudera Manager perform the installation, continue with Start the Cloudera Manager Server on page 141. Otherwise, follow the steps in (Optional) Manually Install CDH and Managed Service Packages on page 146 and then return to this page to continue the installation.

Start the Cloudera Manager Server

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

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

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

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

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

If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.

Start and Log into the Cloudera Manager Admin Console

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

1. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.
2. In a web browser, enter `http://Server host:7180`, where *Server host* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.

3. Log into Cloudera Manager Admin Console. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can change the password using Cloudera Manager after you run the installation wizard. Although you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.
4. After you log in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

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

Choose Cloudera Manager Edition

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

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

- Basic Edition
- Flex Edition
- Enterprise Data Hub Edition

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

2. If you elect Cloudera Enterprise, install a license:
   a. Click **Upload License**.
   b. Click the document icon to the left of the **Select a License File** text field.
   c. Go to the location of your license file, click the file, and click **Open**.
   d. Click **Upload**.
3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.
4. Click **Continue** to proceed with the installation.

Choose Cloudera Manager Hosts

Choose which hosts will run CDH and managed services

1. Do one of the following depending on whether you are using Cloudera Manager to install software:
   - If you are using Cloudera Manager to install software, search for and choose hosts:
     1. To enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the cluster hostnames or IP addresses. You can also specify hostname and IP address ranges. For example:

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

        You can specify multiple addresses and address ranges by separating them with commas, semicolons, tabs, or blank spaces, or by placing them on separate lines. Use this technique to make more specific searches instead of searching overly wide ranges. The scan results will include all addresses scanned, but only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. If you do not know the IP addresses of all of the hosts, you can enter an address range that spans over unused addresses and then clear the hosts that do not exist (and are not discovered) later in this procedure. However, keep in mind that wider ranges will require more time to scan.

     2. Click **Search**. Cloudera Manager identifies the hosts on your cluster to allow you to configure them for services. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking **Abort Scan**. To find additional hosts, click **New Search**, add the host names or IP addresses and click **Search** again. Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts. Common causes of loss of connectivity are firewalls and interference from SELinux.
     3. Verify that the number of hosts shown matches the number of hosts where you want to install services. Clear host entries that do not exist and clear the hosts where you do not want to install services.
   - If you installed Cloudera Agent packages in Manually Install Cloudera Manager Agent Packages on page 140, choose from among hosts with the packages installed:
     1. Click the **Currently Managed Hosts** tab.

2. Choose the hosts to add to the cluster.

2. Click **Continue**.

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

Choose the Software Installation Type and Install Software

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

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

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

        You can add additional parcels for lower versions by specifying custom repositories. For example, you can find the locations of the lower CDH 4 parcels at `https://username:password@archive.cloudera.com/p/cdh4/parcels/`. Or, if you are installing CDH 4.3 and want to use policy-file authorization, you can add the Sentry parcel using this mechanism.

        1. To specify the parcel directory, specify the local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:
           - **Parcel Directory** and **Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host. If you change the default value for **Parcel Directory** and have already installed and started Cloudera Manager Agents, restart the Agents:

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

           - **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the + button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the Configuring Cloudera Manager Server Parcel Settings on page 40 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you see all the unique parcels contained in all your repositories.
           - **Proxy Server** - Specify the properties of a proxy server.
        2. Click **OK**.
     2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories.
   - **Use Packages** - Do one of the following:
     – If Cloudera Manager is installing the packages:
       1. Click the package version.
       2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories.

  - If you manually installed packages in Manually Install CDH and Managed Service Packages on page 140, select the CDH version (CDH 4 or CDH 5) that matches the packages you installed manually.

2. If you installed the Agent and JDK manually on all cluster hosts:
   - Click **Continue**.

     The Host Inspector runs to validate the installation and provides a summary of the results, including all the versions of the installed components. If the validation is successful, click **Finish**.
   - Skip the remaining steps in this section and continue with Add Services on page 144
3. Select **Install Oracle Java SE Development Kit (JDK)** to allow Cloudera Manager to install the JDK on each cluster host. If you have already installed the JDK, do not select this option. If your local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, select the **Install Java Unlimited Strength Encryption Policy Files** checkbox.

   Note: If you already manually installed the JDK on each cluster host, this option to install the JDK does not display.
4. (Optional) Select **Single User Mode** to configure the Cloudera Manager Agent and all service processes to run as the same user. This mode requires extra configuration steps that must be done manually on all hosts in the cluster. If you have not performed the steps, directory creation will fail in the installation wizard. In most cases, you can create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**.
5. If you chose to have Cloudera Manager install software, specify host installation properties:
   - Select **root** or enter the username for an account that has password-less sudo permission.
   - Select an authentication method:
     - If you choose password authentication, enter and confirm the password.
     - If you choose public-key authentication, provide a passphrase and path to the required key files.
   - You can specify an alternate SSH port. The default value is 22.
   - You can specify the maximum number of host installations to run at once. The default value is 10.

   The root password (or any password used at this step) is not saved in Cloudera Manager or CDH. You can change these passwords after install without any impact to Cloudera Manager or CDH.
6. Click **Continue**. If you chose to have Cloudera Manager install software, Cloudera Manager installs the Oracle JDK, Cloudera Manager Agent, packages and CDH and managed service parcels or packages. During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.
7. Click **Continue**.

   The Host Inspector runs to validate the installation and provides a summary of the results, including all the versions of the installed components. If the validation is successful, click **Finish**.

Add Services

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

CDH 4	CDH 5
• **Core Hadoop** - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue	• **Core Hadoop** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, and Hue

CDH 4	CDH 5
• **Core with HBase** • **Core with Impala** • **All Services** - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • **Custom Services** - Any combination of services.	• **Core with HBase** • **Core with Impala** • **Core with Search** • **Core with Spark** • **All Services** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • **Custom Services** - Any combination of services.

Keep the following in mind:

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

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

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

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

- The Flume service can be added only after your cluster has been set up.

- If you have chosen Enterprise Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See Cloudera Navigator Data Management Overview.

2. Click **Continue**.
3. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

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

   The following shortcuts for specifying hostname patterns are supported:

   - Range of hostnames (without the domain portion)

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

   - IP addresses
   - Rack name

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

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

Configure Database Settings

On the Database Setup page, configure settings for required databases:

1. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
2. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

   The **Review Changes** screen displays.

Review Configuration Changes and Start Services

1. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

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

2. Click **Continue**.

   The wizard starts the services.

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

Change the Default Administrator Password

As soon as possible, change the default administrator password:

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

Configure Oozie Data Purge Settings

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

Test the Installation

You can test the installation following the instructions in Testing the Installation on page 195.

(Optional) Manually Install CDH and Managed Service Packages

The procedures in this topic are an optional part of the Path B Installation instructions. Begin with the steps in Installation Path B - Installation Using Cloudera Manager Parcels or Packages on page 136 before following the steps in this topic. For an overview of the installation process, see Installing Cloudera Manager and CDH on page 54.

The CDH and Managed Service Packages contain all of the CDH software. You can choose to manually install CDH and the Managed Service Packages, or you can choose to let Cloudera Manager perform this installation in a later step. Otherwise, follow the steps in this topic to manually install CDH and Managed Service packages and then continue the installation with Start the Cloudera Manager Server on page 141.

> **Note:** If you choose to install CDH manually using these instructions, you cannot use Cloudera Manager to install additional parcels, you must use the packages option in Cloudera Manager. See Managing Software Installation Using Cloudera Manager on page 32.

Install CDH 5 and Managed Service Packages

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

- **Red Hat**

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

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

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

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

     b. Install the RPM for all RHEL versions:

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

  2. (Optionally) add a repository key:

     - **Red Hat/CentOS/Oracle 5**

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

     - **Red Hat/CentOS/Oracle 6**

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

  3. Install the CDH packages:

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

- **SLES**

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

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

   b. Install the RPM:

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

   c. Update your system package index by running the following:

```
$ sudo zypper refresh
```

2. (Optionally) add a repository key:
   - **SLES 11:**

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

   - **SLES 12:**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh5/sles/12/x86_64/cdh/RPM-GPG-KEY-cloudera
```

3. Install the CDH packages:

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

- **Ubuntu and Debian**

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

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

   b. Install the package by doing one of the following:
      - Choose **Open with** in the download window to use the package manager.

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

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

2. Optionally add a repository key:

   - **Debian Wheezy**

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

   - **Ubuntu Precise**

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

3. Install the CDH packages:

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

Continue the installation with Start the Cloudera Manager Server on page 141.

Installation Path C - Manual Installation Using Cloudera Manager Tarballs

Before proceeding with this path for a new installation, review Cloudera Manager Deployment on page 54. If you are upgrading an existing Cloudera Manager installation, see Cloudera Upgrade Overview.

In this procedure, you install the Oracle JDK, Cloudera Manager Server, and Cloudera Manager Agent software as tarballs and use Cloudera Manager to automate installation of CDH and managed service software as parcels. For a full discussion of deployment options, see Installing Cloudera Manager and CDH on page 54.

> **Important:** Tarball installation of Cloudera Manager is deprecated as of Cloudera Manager 5.9.0 and will be removed in version 6.0.0.

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

> **Important:** As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.
>
> This can affect new installations, upgrades, adding new hosts to a cluster, and adding a new cluster.
>
> For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

Before You Begin

Install the Oracle JDK

See Java Development Kit Installation on page 58.

Install and Configure External Databases

Read Cloudera Manager and Managed Service Datastores on page 68. Install and configure an external database for services or Cloudera Management Service roles using the instructions in External Databases for Oozie Server, Sqoop Server, Activity Monitor, Reports Manager, Hive Metastore Server, Hue Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server on page 73.

Cloudera Manager also requires a database. Prepare the Cloudera Manager Server database as described in Preparing a Cloudera Manager Server External Database on page 70.

On CentOS 5 and RHEL 5, Install Python 2.6/2.7 and psycopg2 for Hue

Hue in CDH 5 only works with the operating system's native version of Python *when that version is 2.6 and higher*.

CentOS/RHEL 5 ships with Python 2.4 so you must install Python 2.6 (or Python 2.7) and the Python-PostgreSQL Database Adapter, psycopg2 (not psycopg).

If the Hue server is already installed, you must import the psycopg2 connector into Hue's environment **or** create a symbolic link.

```
## Navigate to Hue within your specific CDH parcel version
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print
 $9}'`/lib/hue/build/env/bin
./python2.6
>>>> import psycopg2
```

or ...

```
cd /opt/cloudera/parcels/`ls -l /opt/cloudera/parcels | grep CDH | tail -1 | awk '{print
 $9}'`/lib/hue/build/env/lib/python2.6/site-packages/
ln -s /usr/lib64/python2.6/site-packages/psycopg2 psycopg2
```

Install the Cloudera Manager Server and Agents

To install the Cloudera Manager Server and Agents, you download and extract tarballs, create users, and configure the server and agents.

Download and Extract Tarballs

Tarballs contain the Cloudera Manager Server and Cloudera Manager Agent in a single file.

1. Download tarballs from the locations listed in .
2. 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 you choose. You can create a new directory to accommodate the files you extract from the tarball. For example, if `/opt/cloudera-manager` does not exist, create it using a command similar to:

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

3. Extract the contents of the tarball to the selected 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 required later and is referred to as *$CMF_DEFAULTS* directory.

Perform Configuration Required by Single User Mode

If you are creating a Cloudera Manager deployment that employs single user mode, perform the configuration steps described in Configuring Single User Mode on page 59.

Create Users

The Cloudera Manager Server and managed services require a user account. When installing Cloudera Manager from tarballs, you must create this user account on all hosts manually. Because Cloudera Manager Server and managed services are configured to use `cloudera-scm` by default, creating a user with this name is the simplest approach. This created user is used automatically after installation is complete.

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

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

Ensure that the `--home` argument path matches your environment. This argument varies according to where you place the tarball, and the version number varies among releases. For example, the `--home` location could be `/opt/cm-5.6.0/run/cloudera-scm-server`.

Create the Cloudera Manager Server Local Data Storage Directory

1. Create the following directory: `/var/lib/cloudera-scm-server`.
2. Change the owner of the directory so that the `cloudera-scm` user and group have ownership of the directory. For example:

```
$ sudo mkdir /var/lib/cloudera-scm-server
$ sudo chown cloudera-scm:cloudera-scm /var/lib/cloudera-scm-server
```

Configure Cloudera Manager Agents

On every Cloudera Manager Agent host, configure the Cloudera Manager Agent to point to the Cloudera Manager Server. Use the CMF_DEFAULTS environment variable in the environment of the user running Cloudera Manager Server and Agent. In *$CMF_DEFAULTS*`/cloudera-scm-agent/config.ini`, set the following environment variables:

Property	Description
`server_host`	Name of the host where Cloudera Manager Server is running.
`server_port`	Port on the host where Cloudera Manager Server is running.

By default, a tarball installation has a `var` subdirectory where state is stored. In a non-tarball installation, state is stored in `/var`. Cloudera recommends that you reconfigure the tarball installation to use an external directory as the `/var` equivalent (`/var` or any other directory outside the tarball) so that when you upgrade Cloudera Manager, the new tarball installation can access this state. Configure the installation to use an external directory for storing state by editing *$CMF_DEFAULTS*`/etc/default/cloudera-scm-agent` and setting the `CMF_VAR` variable to the location of the `/var` equivalent. If you do not reuse the state directory between different tarball installations, duplicate Cloudera Manager Agent entries can occur in the Cloudera Manager database.

Configuring for a Custom Cloudera Manager User and Custom Directories

You can change the default username and directories used by Cloudera Manager. If you do not change the default, skip to Cloudera Manager and Managed Service Datastores on page 68. By default, Cloudera Manager creates the following directories in `/var/log` and `/var/lib`:

- /var/log/cloudera-scm-headlamp
- /var/log/cloudera-scm-firehose
- /var/log/cloudera-scm-alertpublisher
- /var/log/cloudera-scm-eventserver
- /var/lib/cloudera-scm-headlamp
- /var/lib/cloudera-scm-firehose
- /var/lib/cloudera-scm-alertpublisher
- /var/lib/cloudera-scm-eventserver
- /var/lib/cloudera-scm-server

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

- **Change ownership of existing directories:**

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

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

- **Use alternate directories:**

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

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

  2. Connect to the Cloudera Manager Admin Console.
  3. Select **Clusters** > **Cloudera Management Service**
  4. Select **Scope** > ***role name***.
  5. Click the **Configuration** tab.
  6. Enter a term in the **Search** field to find the settings to be changed. For example, you can enter /var or directory.
  7. Update each value with the new locations for Cloudera Manager to use.

     Note: The configuration property for the **Cloudera Manager Server Local Data Storage Directory** (/var/lib/cloudera-scm-server by default) is located on a different page:

     1. Select **Administration** > **Settings**.
     2. Type directory in the Search box.
     3. Enter the directory path in the **Cloudera Manager Server Local Data Storage Directory** property.

  8. Click **Save Changes** to commit the changes.

Create Parcel Directories

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

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

2. Change the directory ownership to be the username you are using to run Cloudera Manager:

```
$ sudo chown username:groupname /opt/cloudera/parcel-repo
```

where *username* and *groupname* are the user and group names (respectively) you are using to run Cloudera Manager. For example, if you use the default username `cloudera-scm`, you would run the command:

```
$ sudo chown cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo
```

3. On each cluster host, create a parcels directory:

```
$ sudo mkdir -p /opt/cloudera/parcels
```

4. Change the directory ownership to be the username you are using to run Cloudera Manager:

```
$ sudo chown username:groupname /opt/cloudera/parcels
```

where *username* and *groupname* are the user and group names (respectively) you are using to run Cloudera Manager. For example, if you use the default username `cloudera-scm`, you would run the command:

```
$ sudo chown cloudera-scm:cloudera-scm /opt/cloudera/parcels
```

Start the Cloudera Manager Server

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

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

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

The way in which you start the Cloudera Manager Server varies according to which account you want the Server to run under:

- As root:

```
$ sudo $CMF_DEFAULTS/etc/init.d/cloudera-scm-server start
```

- As another user. If you run as another user, ensure that 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 $CMF_DEFAULTS/etc/init.d/cloudera-scm-server 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 *$CMF_DEFAULTS*/etc/default/cloudera-scm-server:

```
export CMF_SUDO_CMD=" "
```

2. Change the user and group in *$CMF_DEFAULTS*/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 $CMF_DEFAULTS/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 $CMF_DEFAULTS/etc/init.d/cloudera-scm-server /etc/init.d/cloudera-scm-server
$ chkconfig cloudera-scm-server on
```

- **Debian/Ubuntu**

```
$ cp $CMF_DEFAULTS/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 ${CMF_DEFAULTS:-/etc/default} to *$CMF_DEFAULTS*/etc/default.

If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.

Start the Cloudera Manager Agents

Start the Cloudera Manager Agent according to the account you want the Agent to run under:

- To start the Cloudera Manager Agent, run this command on each Agent host:

```
$ sudo $CMF_DEFAULTS/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 $CMF_DEFAULTS/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 *$CMF_DEFAULTS*/etc/default/cloudera-scm-agent:

```
export CMF_SUDO_CMD=" "
```

2. Change the user and group in *$CMF_DEFAULTS*/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 $CMF_DEFAULTS/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 $CMF_DEFAULTS/etc/init.d/cloudera-scm-agent /etc/init.d/cloudera-scm-agent
$ chkconfig cloudera-scm-agent on
```

     - **Debian/Ubuntu**

```
$ cp $CMF_DEFAULTS/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 *$CMF_DEFAULTS*/etc/init.d/cloudera-scm-agent file and change the value of CMF_DEFAULTS from ${CMF_DEFAULTS:-/etc/default} to *$CMF_DEFAULTS*/etc/default.

Install Package Dependencies

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

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

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

Start and Log into the Cloudera Manager Admin Console

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

1. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.
2. In a web browser, enter `http://Server host:7180`, where *Server host* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.

3. Log into Cloudera Manager Admin Console. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can change the password using Cloudera Manager after you run the installation wizard. Although you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.
4. After you log in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

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

Choose Cloudera Manager Edition

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

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

- Flex Edition
- Enterprise Data Hub Edition

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

2. If you elect Cloudera Enterprise, install a license:
   a. Click **Upload License**.
   b. Click the document icon to the left of the **Select a License File** text field.
   c. Go to the location of your license file, click the file, and click **Open**.
   d. Click **Upload**.
3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.
4. Click **Continue** to proceed with the installation.

Choose Cloudera Manager Hosts

1. Click the **Currently Managed Hosts** tab.
2. Choose the hosts to add to the cluster.
3. Click **Continue**.

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

Install CDH and Managed Service Software

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

         You can add additional parcels for lower versions by specifying custom repositories. For example, you can find the locations of the lower CDH 4 parcels at `https://username:password@archive.cloudera.com/p/cdh4/parcels/`. Or, if you are installing CDH 4.3 and want to use policy-file authorization, you can add the Sentry parcel using this mechanism.

         1. To specify the parcel directory, specify the local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:
            - **Parcel Directory** and **Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host. If you change the default value for **Parcel Directory** and have already installed and started Cloudera Manager Agents, restart the Agents:

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

            - **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the ➕ button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the Configuring Cloudera Manager Server Parcel Settings on page 40 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you see all the unique parcels contained in all your repositories.
            - **Proxy Server** - Specify the properties of a proxy server.
         2. Click **OK**.
      b. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. You can choose either the version that matches the Cloudera Manager Server you are currently using

or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories.

b. Click **Continue**. Cloudera Manager installs the CDH and managed service parcels. During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed. Click **Continue**.

2. Click **Continue**.

The Host Inspector runs to validate the installation and provides a summary of the results, including all the versions of the installed components. If the validation is successful, click **Finish**.

Add Services

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

- Select the combination of services to install:

CDH 4	CDH 5
• **Core Hadoop** - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • **Core with HBase** • **Core with Impala** • **All Services** - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • **Custom Services** - Any combination of services.	• **Core Hadoop** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, and Hue • **Core with HBase** • **Core with Impala** • **Core with Search** • **Core with Spark** • **All Services** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • **Custom Services** - Any combination of services.

Keep the following in mind:

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

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

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

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

  - The Flume service can be added only after your cluster has been set up.

- If you have chosen Enterprise Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See Cloudera Navigator Data Management Overview.

2. Click **Continue**.
3. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

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

   The following shortcuts for specifying hostname patterns are supported:

   - Range of hostnames (without the domain portion)

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

   - IP addresses
   - Rack name

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

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

Configure Database Settings

On the Database Setup page, configure settings for required databases:

1. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
2. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

   The **Review Changes** screen displays.

Review Configuration Changes and Start Services

1. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

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

2. Click **Continue**.

   The wizard starts the services.

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

(Optional) Change the Cloudera Manager User

After configuring your services, the installation wizard automatically starts the Cloudera Management Service, assuming that it runs using `cloudera-scm`. If you configured this service to run using a user other than `cloudera-scm`, the Cloudera Management Service roles do not start automatically. To change the service configuration to use the user account that you selected:

1. Connect to the Cloudera Manager Admin Console.
2. Do one of the following:
   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
3. Click the **Configuration** tab.
4. Use the search box to find the property to change. For example, you might enter "system" to find the **System User** and **System Group** properties.
5. Make any changes required to the System User and System Group to ensure Cloudera Manager uses the proper user accounts.
6. Click **Save Changes**.
7. Start the Cloudera Management Service roles.

Change the Default Administrator Password

As soon as possible, change the default administrator password:

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

Configure Oozie Data Purge Settings

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

Test the Installation

You can test the installation following the instructions in Testing the Installation on page 195.

Installing Impala

Impala is included with CDH 5. In a parcel-based configuration, it is part of the CDH parcel rather than a separate parcel. Starting with CDH 5.4 (corresponding to Impala 2.2 in the Impala versioning scheme) new releases of Impala are only available on CDH 5, not CDH 4.

Although these installation instructions primarily focus on CDH 5, you can also manage CDH 4 clusters using Cloudera Manager 5. In CDH 4, Impala has packages and parcels that you download and install separately from CDH. To use Impala with CDH 4, you must install both CDH and Impala on the hosts that will run Impala.

Note:

- See CDH 5 and Cloudera Manager 5 Requirements and Supported Versions for supported versions.
- Before proceeding, review the installation options described in Cloudera Manager Deployment on page 54.

Installing Impala after Upgrading Cloudera Manager

If you have just upgraded Cloudera Manager from a version that did not support Impala, the Impala software is not installed automatically. (Upgrading Cloudera Manager does not automatically upgrade CDH or other managed services). You can add Impala using parcels; go to the **Hosts** tab, and select the **Parcels** tab. If you have installed CDH 4, you should see at least one Impala parcel available for download. See Parcels on page 32 for detailed instructions on using parcels to install or upgrade Impala. If you do not see any Impala parcels available, click the **Edit Settings** button on the **Parcels** page to go to the Parcel configuration settings and verify that the Impala parcel repo URL (https://archive.cloudera.com/impala/parcels/latest/) has been configured in the **Parcels** configuration page. See Parcel Configuration Settings on page 40 for more details.

Post Installation Configuration

See The Impala Service for instructions on configuring the Impala service.

Installing Kudu

Important: If you have previously installed the beta `IMPALA_KUDU` service, make sure you remove it from your cluster before you proceed.

You can install Kudu on a Cloudera Manager deployment using either parcels or packages. Depending on your deployment, follow the instructions in either Install Kudu Using Parcels on page 162 or Install Kudu Using Packages on page 163.

Install Kudu Using Parcels

Use the following steps to install Kudu using parcels.

1. In Cloudera Manager, go to **Hosts** > **Parcels**. Find KUDU in the list, and click **Download**.
2. When the download is complete, select your cluster from the **Locations** selector, and click **Distribute**. If you only have one cluster, it is selected automatically.
3. When distribution is complete, click **Activate** to activate the parcel. Restart the cluster when prompted. This might take several minutes.
4. Install the Kudu service on your cluster. Go to the cluster where you want to install Kudu. Click **Actions** > **Add a Service**. Select **Kudu** from the list, and click **Continue**.
5. Select a host for the master role and one or more hosts for the tablet server roles. A host can act as both a master and a tablet server, but this might cause performance problems on a large cluster. The Kudu master process is not resource-intensive and can be collocated with other similar processes such as the HDFS NameNode or YARN ResourceManager. After selecting hosts, click **Continue**.
6. Configure the storage locations for Kudu data and write-ahead log (WAL) files on masters and tablet servers. Cloudera Manager will create the directories.
   - You can use the same directory to store data and WALs.
   - You cannot store WALs in a subdirectory of the data directory.
   - If any host is both a master and tablet server, configure different directories for master and tablet server. For instance, `/data/kudu/master` and `/data/kudu/tserver`.
   - If you have chosen a filesystem that does not support hole punching, the Kudu service will fail to start. In this case *only*, exit the wizard by clicking the Cloudera logo at the top left, and enable the file block manager. This is not appropriate for production. See Enabling the File Block Manager on page 163.
7. If your filesystem supports hole punching, do not exit the wizard. Click **Continue**. Kudu masters and tablet servers are started. Otherwise, go to the **Kudu** service, and click **Actions** > **Start**.
8. Verify the Installation on page 164.
9. To manage roles, go to the **Kudu** service and use the **Actions** menu to stop, start, restart, or otherwise manage the service.

Enabling the File Block Manager

If your filesystem supports hole punching, do not use the file blocker manager. The file blocker manager does not perform well at scale and must only be used for small-scale development and testing.

If your filesystem does not support hole punching, but you want to experiment with Kudu, you must enable the file block manager. If you do not enable the file block manager, Kudu will not start.

1. If you are still in the Cloudera configuration wizard, exit the configuration wizard by clicking the Cloudera logo at the top of the Cloudera Manager interface.
2. Go to the **Kudu** service.
3. Go to **Configuration** and search for the **Kudu Service Advanced Configuration Snippet (Safety Valve) for gflagfile** configuration option.
4. Add the following line to it, and save your changes:

```
--block_manager=file
```

Install Kudu Using Packages

Table 17: Kudu Repository and Package Links

Operating System	Repository Package	Individual Packages
RHEL	RHEL 6 or RHEL 7	RHEL 6
Ubuntu	Trusty, Xenial	Trusty, Xenial
SLES	SLES 12	SLES 12
Debian	Jessie	Jessie

1. Cloudera recommends installing the Kudu repositories for your operating system. Use the links in Table 17: Kudu Repository and Package Links on page 163 to download the appropriate repository installer. Save the repository installer to `/etc/yum.repos.d/` for RHEL, `/etc/apt/sources.list.d/` for Ubuntu/Debian, or `/etc/zypp/repos.d` for SLES.
   - If you use Cloudera Manager, you only need to install the kudu package:

Operating System	Install Commands
RHEL/CentOS	`sudo yum install kudu`
Ubuntu/Debian	`sudo apt-get install kudu`
SLES	`sudo zypper install kudu`

   - If you need the C++ client development libraries or the Kudu SDK, install `kudu-client` and `kudu-client-devel` packages for RHEL, or `libkuduclient0` and `libkuduclient-dev` packages for Ubuntu.
   - **Do not** install the `kudu-master` and `kudu-tserver` packages. They provide operating system startup scripts for using Kudu without Cloudera Manager.
2. Install the Kudu service on your cluster. Go to the cluster where you want to install Kudu. Click **Actions** > **Add a Service**. Select **Kudu** from the list, and click **Continue**.
3. Select a host for the master role and one or more hosts for the tablet server roles. A host can act as both a master and a tablet server, but this might cause performance problems on a large cluster. The Kudu master process is

not resource-intensive and can be collocated with other similar processes such as the HDFS NameNode or YARN ResourceManager. After selecting hosts, click **Continue**.

4. Configure the storage locations for Kudu data and write-ahead log (WAL) files on masters and tablet servers. Cloudera Manager will create the directories.
   - You can use the same directory to store data and WALs.
   - You cannot store WALs in a subdirectory of the data directory.
   - If any host is both a master and tablet server, configure different directories for master and tablet server. For instance, `/data/kudu/master` and `/data/kudu/tserver`.
   - If you have chosen a filesystem that does not support hole punching, the Kudu service will fail to start. In this case *only*, exit the wizard by clicking the Cloudera logo at the top left, and enable the file block manager. This is not appropriate for production. See Enabling the File Block Manager on page 163.
5. If your filesystem supports hole punching, do not exit the wizard. Click **Continue**. Kudu masters and tablet servers are started. Otherwise, go to the **Kudu** service, and click **Actions** > **Start**.
6. Verify the Installation on page 164.
7. To manage roles, go to the **Kudu** service and use the **Actions** menu to stop, start, restart, or otherwise manage the service.

Verify the Installation

1. Verify that the Kudu master and tablet servers are running using one of the following methods:
   - Examine the output of the `ps` command on servers to verify that the `kudu-master` and `kudu-tserver` processes are running.
   - Access the master or tablet server web UI by going to `http://<_host_name_>:8051/` for masters, or `http://<_host_name_>:8050/` for tablet servers.
2. If Kudu isn't running, look at the log files in `/var/log/kudu`, and if there's a file ending with `.FATAL`, that means Kudu did not start.
   - If the error is related to a failed hole punch test or the file block manager, it might be a problem with your operating system.
   - If the error is related to clock synchronization, it is most likely a problem with the Network Time Protocol.

Installing Cloudera Search

Cloudera Search (powered by Apache Solr) is included in CDH 5. If you have installed CDH 5.0 or higher, you do not need to perform any additional actions to install Search. For more information on installing Cloudera Manager and CDH, see Installing Cloudera Manager and CDH on page 54.

Deploying Cloudera Search

For information on deploying the Cloudera Search service in your cluster, see Deploying Cloudera Search.

Installing Spark

Apache Spark is included with CDH 5. To use Apache Spark with CDH 4, you must install both CDH and Spark on the hosts that will run Spark.

Note:

- See CDH 5 and Cloudera Manager 5 Requirements and Supported Versions for supported versions.
- Before proceeding, review the installation options described in Cloudera Manager Deployment on page 54.

Installing Spark after Upgrading Cloudera Manager

If you have just upgraded Cloudera Manager from a version that did not support Spark, the Spark software is not installed automatically. (Upgrading Cloudera Manager does not automatically upgrade CDH or other managed services).

You can add Spark using parcels; go to the **Hosts** tab, and select the **Parcels** tab. You should see at least one Spark parcel available for download. See Parcels on page 32 for detailed instructions on using parcels to install or upgrade Spark. If you do not see any Spark parcels available, click the **Edit Settings** button on the **Parcels** page to go to the Parcel configuration settings and verify that the Spark parcel repo URL (https://archive.cloudera.com/spark/parcels/latest/) has been configured in the **Parcels** configuration page. See Parcel Configuration Settings on page 40 for more details.

Post Installation Configuration

See Managing Spark Using Cloudera Manager for instructions on adding the Spark service.

Installing the GPL Extras Parcel

GPL Extras contains LZO functionality for compressing data.

To install the GPL Extras parcel:

1. Add the appropriate repository to the Cloudera Manager list of parcel repositories. The public repositories can be found at:
   - **CDH 5.4 and higher** - `https://archive.cloudera.com/gplextras5/parcels/{latest_supported}`. The substitution variable `{latest_supported}` appears after the parcel to enable substitution of the latest supported maintenance version of the parcel.
   - **CDH 5.0-5.3** - `https://archive.cloudera.com/gplextras5/parcels/latest`
   - **CDH 4** - `https://archive.cloudera.com/gplextras/parcels/latest`

   If you are using LZO with Impala, you must choose a specific version of the GPL Extras parcel for the Impala version according to the following tables:

Table 18: CDH 5

Impala Version	Parcels Version Subdirectory	GPL Extras Parcel Version
CDH 5.*x.y*	`5.x.y/`	GPLEXTRAS-5.*x.y*

Table 19: CDH 4

Impala Version	Parcels Version Subdirectory	GPL Extras Parcel Version
2.1.0	`0.4.15.101/`	HADOOP_LZO-0.4.15-1.gplextras.p0.101
2.0.0	`0.4.15.101/`	HADOOP_LZO-0.4.15-1.gplextras.p0.101
1.4.0	`0.4.15.85/`	HADOOP_LZO-0.4.15-1.gplextras.p0.85

Impala Version	Parcels Version Subdirectory	GPL Extras Parcel Version
1.3.1	`0.4.15.64/`	HADOOP_LZO-0.4.15-1.gplextras.p0.64
1.2.4	`0.4.15.58/`	HADOOP_LZO-0.4.15-1.gplextras.p0.58
1.2.3	`0.4.15.39/`	HADOOP_LZO-0.4.15-1.gplextras.p0.39
1.2.2	`0.4.15.37/`	HADOOP_LZO-0.4.15-1.gplextras.p0.37
1.2.1	`0.4.15.33/`	HADOOP_LZO-0.4.15-1.gplextras.p0.33

To create the repository URL, append the version directory to the URL (CDH 4) `https://archive.cloudera.com/gplextras/parcels/` or (CDH 5) `https://archive.cloudera.com/gplextras5/parcels/` respectively. For example: `https://archive.cloudera.com/gplextras5/parcels/5.0.2`.

2. Download, distribute, and activate the parcel.
3. If not already installed, on all cluster hosts, install the `lzo` package on RHEL or the `liblzo2-2` package on SLES, Debian, or Ubuntu:

   RHEL:

```
sudo yum install lzo
```

   Debian or Ubuntu:

```
sudo apt-get install liblzo2-2
```

   SLES:

```
sudo zypper install liblzo2-2
```

Understanding Custom Installation Solutions

Cloudera hosts two types of software repositories that you can use to install products such as Cloudera Manager or CDH—parcel repositories and RHEL and SLES RPM and Debian/Ubuntu package repositories.

These repositories are effective solutions in most cases, but custom installation solutions are sometimes required. Using the software repositories requires client access over the Internet and results in the installation of the latest version of products. An alternate solution is required if:

- You need to install older product versions. For example, in a CDH cluster, all hosts must run the same CDH version. After completing an initial installation, you may want to add hosts. This could be to increase the size of your cluster to handle larger tasks or to replace older hardware.
- The hosts on which you want to install Cloudera products are not connected to the Internet, so they are unable to reach the Cloudera repository. (For a parcel installation, only the Cloudera Manager Server needs Internet access, but for a package installation, all cluster members need access to the Cloudera repository). Some organizations choose to partition parts of their network from outside access. Isolating segments of a network can provide greater assurance that valuable data is not compromised by individuals out of maliciousness or for personal gain. In such a case, the isolated computers are unable to access Cloudera repositories for new installations or upgrades.

In both of these cases, using a custom repository solution allows you to meet the needs of your organization, whether that means installing older versions of Cloudera software or installing any version of Cloudera software on hosts that are disconnected from the Internet.

Understanding Parcels

Parcels are a packaging format that facilitate upgrading software from within Cloudera Manager. You can download, distribute, and activate a new software version all from within Cloudera Manager. Cloudera Manager downloads a parcel to a local directory. Once the parcel is downloaded to the Cloudera Manager Server host, an Internet connection is no longer needed to deploy the parcel. Parcels are available for CDH 4.1.3 and onwards. For detailed information about parcels, see Parcels on page 32.

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel files and put them into a parcel repository. See Creating and Using a Parcel Repository for Cloudera Manager on page 168.

Understanding Package Management

Before getting into the details of how to configure a custom package management solution in your environment, it can be useful to have more information about:

- Package management tools
- Package repositories

See Creating and Using a Package Repository for Cloudera Manager on page 171.

Package Management Tools

Packages (`rpm` or `deb` files) help ensure that installations complete successfully by encoding each package's dependencies. That means that if you request the installation of a solution, all required elements can be installed at the same time. For example, `hadoop-0.20-hive` depends on `hadoop-0.20`. Package management tools, such as `yum` (RHEL), `zypper` (SLES), and `apt-get` (Debian/Ubuntu) are tools that can find and install any required packages. For example, for RHEL, you might enter `yum install hadoop-0.20-hive`. `yum` would inform you that the hive package requires `hadoop-0.20` and offers to complete that installation for you. `zypper` and `apt-get` provide similar functionality.

Package Repositories

Package management tools operate on package repositories.

Repository Configuration Files

Information about package repositories is stored in configuration files, the location of which varies according to the package management tool.

- RHEL/CentOS `yum` - `/etc/yum.repos.d`
- SLES `zypper` - `/etc/zypp/zypper.conf`
- Debian/Ubuntu `apt-get` - `/etc/apt/apt.conf` (Additional repositories are specified using `*.list` files in the `/etc/apt/sources.list.d/` directory.)

For example, on a typical CentOS system, you might find:

```
[user@localhost ~]$ ls -l /etc/yum.repos.d/
total 24
-rw-r--r-- 1 root root 2245 Apr 25  2010 CentOS-Base.repo
-rw-r--r-- 1 root root  626 Apr 25  2010 CentOS-Media.repo
```

The `.repo` files contain pointers to one or many repositories. There are similar pointers inside configuration files for `zypper` and `apt-get`. In the following snippet from `CentOS-Base.repo`, there are two repositories defined: one

named Base and one named Updates. The mirrorlist parameter points to a website that has a list of places where this repository can be downloaded.

```
# ...
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
# ...
```

Listing Repositories

You can list the repositories you have enabled. The command varies according to operating system:

- RHEL/CentOS - `yum repolist`
- SLES - `zypper repos`
- Debian/Ubuntu - `apt-get` does not include a command to display sources, but you can determine sources by reviewing the contents of `/etc/apt/sources.list` and any files contained in `/etc/apt/sources.list.d/`.

The following shows an example of what you might find on a CentOS system in `repolist`:

```
[root@localhost yum.repos.d]$ yum repolist
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * addons: mirror.san.fastserv.com
 * base: centos.eecs.wsu.edu
 * extras: mirrors.ecvps.com
 * updates: mirror.5ninesolutions.com
repo id                         repo name                                  status
addons                          CentOS-5 - Addons                          enabled:
0
base                            CentOS-5 - Base                           enabled: 3,434
extras                          CentOS-5 - Extras                         enabled:   296
updates                         CentOS-5 - Updates                        enabled: 1,137
repolist: 4,867
```

Creating and Using a Parcel Repository for Cloudera Manager

You must create a repository and direct hosts in your Cloudera Manager deployment to use that repository.

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

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

For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

After you have created a hosted repository, go to Configuring the Cloudera Manager Server to Use the Parcel URL for Hosted Repositories on page 170.

Alternatively, you can create a local file repository. See Using a Local Parcel Repository on page 170.

Note: Using a local parcel repository does not support parcel relations (Depends, Conflicts, and Replaces), which are defined in manifest files. Hosted parcel repositories are required.

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.

Hosting a Parcel Repository

Installing a Web Server

The repository is typically hosted using HTTP on a host inside your network. If you already have a web server in your organization, you can move the repository directory, which will include both the RPMs and the `repodata/` subdirectory, to a location hosted by the web server. An easy web server to install is the Apache HTTPD. If you are able to use an existing web server, then note the URL and skip to Downloading the Parcel and Publishing Files on page 169.

Installing Apache HTTPD

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

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

Starting Apache HTTPD

OS	Command
RHEL	`[root@localhost tmp]$  service httpd start`
SLES	`[root@localhost tmp]$ service apache2 start`
Ubuntu or Debian	`[root@localhost tmp]$ service apache2 start`

Downloading the Parcel and Publishing Files

1. Download the parcel and `manifest.json` files for your OS distribution from
   - **CDH 5** - Impala, Kudu, Spark, and Search are included in the CDH parcel.
     - CDH - `https://username:password@archive.cloudera.com/p/cdh5/parcels/`
     - Accumulo--`https://username:password@archive.cloudera.com/p/accumulo-c5/parcels/`
     - GPL Extras - `https:archive.cloudera.com/gplextras5/parcels/`
   - **Cloudera Distribution of Apache Spark 2**
     - The exact parcel name is dependent on the OS. You can find all the parcels at `https://username:password@archive.cloudera.com/p/spark2/parcels/`.
   - **Key Trustee Server**
     - Go to the Key Trustee Server download page. Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee Server parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz` *`filename`*`.tar.gz` command.
   - **Key Trustee KMS**
     - Go to the Key Trustee KMS download page. Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee KMS parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz` *`filename`*`.tar.gz` command.

- **Navigator HSM KMS**
  - Go to the Navigator HSM KMS download page. Select **Parcels** from the **Package or Parcel** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Navigator HSM KMS parcels and `manifest.json` files in a `.tar.gz` file. Extract the files with the `tar xvfz filename.tar.gz` command. Note that the parcel name (KEYTRUSTEE) for the KMS services (both Key Trustee KMS and Navigator HSM KMS) is the same.
- **Other services**
  - Sqoop connectors - `https://username:password@archive.cloudera.com/p/sqoop-connectors/parcels/`

2. Move the `.parcel` and `manifest.json` files to the web server directory, and modify file permissions. For example, you might use the following commands:

```
[root@localhost tmp]$ mkdir /var/www/html/cdh4.6
[root@localhost tmp]$ mv CDH-4.6.0-1.cdh4.6.0.p0.26-lucid.parcel /var/www/html/cdh4.6
[root@localhost tmp]$ mv manifest.json /var/www/html/cdh4.6
[root@localhost tmp]$ chmod -R ugo+rX /var/www/html/cdh4.6
```

After moving the files and changing permissions, visit `http://hostname:80/cdh4.6/` to verify that you can access the parcel. Apache may have been configured to not show indexes, which is also acceptable.

Configuring the Cloudera Manager Server to Use the Parcel URL for Hosted Repositories

1. Use one of the following methods to open the parcel settings page:
   - **Navigation bar**
     1. Click in the top navigation bar or click **Hosts** and click the **Parcels** tab.
     2. Click the **Configuration** button.
   - **Menu**
     1. Select **Administration** > **Settings**.
     2. Select **Category** > **Parcels** .
2. In the **Remote Parcel Repository URLs** list, click the addition symbol to open an additional row.
3. Enter the path to the parcel. For example, `http://hostname:port/cdh4.6/`.
4. Click **Save Changes** to commit the changes.

Using a Local Parcel Repository

To use a local parcel repository, complete the following steps:

1. Open the Cloudera Manager web UI and navigate to the **Parcels** page.
2. Select **Configuration** and verify that you have a **Local Parcel Repository** path set. By default, the directory is `/opt/cloudera/parcel-repo`.
3. Remove any **Remote Parcel Repository URLs** you are not using, including ones that point to Cloudera archives.
4. Add the parcel you want to use to the local parcel repository directory that you specified.
5. In the command line, navigate to the local parcel repository directory.
6. Create a SHA1 hash for the parcel you added and save it to a file named *parcel_name*`.parcel.sha`.

   For example, the following command generates a SHA1 hash for the parcel `CDH-5.10.1-1.cdh5.10.1.p0.10-el5`:

```
sha1sum CDH-5.10.1-1.cdh5.10.1.p0.10-el5.parcel | awk '{ print $1 }' >
CDH-5.10.1-1.cdh5.10.1.p0.10-el5.parcel.sha
```

7. Change the ownership of the parcel and hash files to `cloudera-scm`:

```
sudo chown cloudera-scm:cloudera-scm CDH*
```

8. In the Cloudera Manager web UI, navigate to the **Parcels** page.
9. Click **Check for New Parcels**.

   The new parcel appears.

10. Download, distribute, and activate the parcel.

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 171
- Creating a Temporary Remote Repository on page 172

Once you have created a repository, go to Modifying Clients to Find the Repository on page 173.

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.

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

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

For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

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 172.

Installing Apache HTTPD

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

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

Starting Apache HTTPD

OS	Command
RHEL	`[root@localhost tmp]$  service httpd start`

OS	Command
SLES	`[root@localhost tmp]$ service apache2 start`
Ubuntu or Debian	`[root@localhost tmp]$ service apache2 start`

Downloading the Tarball and Publishing Repository Files

1. Download the tarball for your OS distribution from the repo as tarball archive.

   For Cloudera Navigator data encryption components, go to the download page for each component, select your OS version, and click **Download**:

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

2. Unpack the tarball, move the files to the web server directory, and modify file permissions. For example, you might use the following commands:

```
[root@localhost tmp]$ tar xvfz cm5.0.0-centos6.tar.gz
[root@localhost tmp]$ mv cm /var/www/html
[root@localhost tmp]$ chmod -R ugo+rX /var/www/html/cm
```

   After moving files and changing permissions, visit `http://hostname:port/cm` to verify that you see an index of files. Apache may have been configured to not show indexes, which is also acceptable.

Creating a Temporary Remote Repository

You can quickly create a temporary remote repository to deploy a package once. It is convenient to perform this on the same host that runs Cloudera Manager, or a gateway role. In this example, python SimpleHTTPServer is used from a directory of your choosing.

1. Download the tarball for your OS distribution from the repo as tarball archive.

   For Cloudera Navigator data encryption components, go to the download page for each component, select your OS version, and click **Download**:

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

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 *hostname* is the name of the web server:
	`[myrepo]`
	`name=myrepo`
	`baseurl=http://hostname/cm/5`
	`enabled=1`
	`gpgcheck=0`
	See `man yum.conf` for more details. Put that file into `/etc/yum.repos.d/myrepo.repo` on all of your hosts to enable them to find the packages that you are hosting.

SLES	Use the `zypper` utility to update client system repo information by issuing the following command: `$ zypper addrepo http://hostname/cm alias`

Ubuntu or Debian	Add a new `.list` file to `/etc/apt/sources.list.d/` on client systems. For example, you might create the file `/etc/apt/sources.list.d/my-private-cloudera-repo.list`. In that file, create an entry to your newly created repository. For example:
	`$ cat /etc/apt/sources.list.d/my-private-cloudera-repo.list`
	`deb http://hostname/cm codename components`
	You can find the *codename* and *component* variables in the `./conf/distributions` file in the repository.
	After adding your `.list` file, ensure `apt-get` uses the latest information by issuing the following command:
	`$ sudo apt-get update`

Configuring a Custom Java Home Location

Note: This procedure changes the JDK for Cloudera Management Services and CDH cluster processes only. It does not affect the JDK used by other non-Cloudera processes, or gateway roles.

Java, which Cloudera services require, may be installed at a custom location. Follow the installation instructions in Java Development Kit Installation on page 58.

If you choose to use a custom Java location, modify the host configuration to ensure the JDK can be found. If you do not update the configuration, Cloudera services will be unable to find this resource and will not start:

1. Open the Cloudera Manager Admin Console.
2. In the main navigation bar, click the **Hosts** tab and optionally click a specific host link.
3. Click the **Configuration** tab.
4. Select **Category** > **Advanced**.
5. Set the **Java Home Directory** property to the custom location.
6. Click **Save Changes**.
7. Restart all services.

Installing Lower Versions of Cloudera Manager 5

When you install Cloudera Manager—for example, by using the installer downloadable from the Cloudera Downloads website—the most recent version is installed by default. This ensures that you install the latest features and bug fixes. In some cases, however, you may want to install a lower version.

For example, you might install a lower version if you want to expand an existing cluster. In this case, follow the instructions in Adding a Host to the Cluster.

You can also add a cluster to be managed by the same instance of Cloudera Manager by using the **Add Cluster** feature from the Services page in the Cloudera Manager Admin Console. Follow the instructions in Adding a Cluster.

You may also want to install a lower version of the Cloudera Manager Server on a new cluster if, for example, you have validated a specific version and want to deploy that version on additional clusters. Installing an older version of Cloudera Manager requires several manual steps to install and configure the database and the correct version of the Cloudera Manager Server. After completing these steps, run the Installation wizard to complete the installation of Cloudera Manager and CDH.

Before You Begin

Install and Configure Databases

Cloudera Manager Server, Cloudera Management Service, and the Hive metastore data are stored in a database. Install and configure required databases following the instructions in Cloudera Manager and Managed Service Datastores on page 68.

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

CDH 5 Hue works only with the default system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6, you need Python 2.6 to start Hue.

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

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

Establish Your Cloudera Manager Repository Strategy

- **Download and Edit the Repo File for RHEL-compatible OSs or SLES**
  1. Download the Cloudera Manager repo file (`cloudera-manager.repo`) for your OS version using the links provided on the Cloudera Manager Version and Download Information page. For example, for Red Hat/CentOS 6, the file is located at `https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/cloudera-manager.repo`.
  2. Edit the file to change `baseurl` to point to the version of Cloudera Manager you want to download. For example, to install Cloudera Manager version 5.0.1, change: `baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/` to `baseurl=https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5.0.1/`.
  3. Save the edited file:
     - For RHEL or CentOS, save it in `/etc/yum.repos.d/`.
     - For SLES, save it in `/etc/zypp/repos.d`.
- **Download and Edit the cloudera.list file for Debian or Apt**
  1. Download the Cloudera Manager list file (`cloudera.list`) using the links provided at Cloudera Manager Version and Download Information. For example, for Ubuntu 10.04 (lucid), this file is located at `https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm/cloudera.list`.

2. Edit the file to change the second-to-last element to specify the version of Cloudera Manager you want to install. For example, with Ubuntu lucid, if you want to install Cloudera Manager version 5.0.1, change: `deb https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm lucid-cm5 contrib` to `deb https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm lucid-cm5.0.1 contrib`.
3. Save the edited file in the directory `/etc/apt/sources.list.d/`.

Install the Oracle JDK on the Cloudera Manager Server Host

Note: Cloudera, Inc. acquired Oracle JDK software under the Oracle Binary Code License Agreement. Pursuant to Item D(v)(a) of the SUPPLEMENTAL LICENSE TERMS of the Oracle Binary Code License Agreement, use of JDK software is governed by the terms of the Oracle Binary Code License Agreement. By installing the JDK software, you agree to be bound by these terms. If you do not wish to be bound by these terms, then do not install the Oracle JDK.

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

- **Install the JDK from a repository**

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

OS	Command
RHEL	`$ sudo yum install oracle-j2sdk1.7`
SLES	`$ sudo zypper install oracle-j2sdk1.7`
Ubuntu or Debian	`$ sudo apt-get install oracle-j2sdk1.7`

- **Install the JDK manually**

  See Java Development Kit Installation on page 58.

Install the Cloudera Manager Server Packages

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

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

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

Set up a Database for the Cloudera Manager Server

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

- External database - Prepare the Cloudera Manager Server database as described in Preparing a Cloudera Manager Server External Database on page 70.
- Embedded database - Install an embedded PostgreSQL database as described in Embedded PostgreSQL Database on page 73.

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

You can use Cloudera Manager to install the Oracle JDK, Cloudera Manager Agent packages, CDH, and managed service packages or you can install any of these packages manually. To use Cloudera Manager to install the packages, you must meet the requirements described in Cloudera Manager Deployment on page 54.

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

If you are going to use Cloudera Manager to install all of the software, *skip this section* and continue with Start the Cloudera Manager Server on page 141. Otherwise, to manually install the Oracle JDK, Cloudera Manager Agent, and CDH and Managed Services, continue with the procedures linked below and then return to this page to continue the installation. in this section. You can choose to manually install any of the following software and, in a later step, Cloudera Manager installs any software that you do not install manually:

Manually Install the Oracle JDK

You can use Cloudera Manager to install the Oracle JDK on all cluster hosts or you can install the JDKs manually. If you choose to have Cloudera Manager install the JDKs, *skip this section*. To use Cloudera Manager to install the JDK, you must meet the requirements described in Cloudera Manager Deployment on page 54.

Install the Oracle JDK on every cluster hosts. For more information, see Java Development Kit Installation on page 58.

Manually Install Cloudera Manager Agent Packages

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

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

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

OS	Command
RHEL, if you have a yum repo configured:	`$ sudo yum install cloudera-manager-agent cloudera-manager-daemons`
RHEL, if you're manually transferring RPMs:	`$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons`
SLES	`$ sudo zypper install cloudera-manager-agent cloudera-manager-daemons`
Ubuntu or Debian	`$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons`

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

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

For more information on Agent configuration options, see Agent Configuration File.

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

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

When the Agent starts, it contacts the Cloudera Manager Server. If communication fails between a Cloudera Manager Agent and Cloudera Manager Server, see Troubleshooting Installation and Upgrade Problems on page 425. When the Agent hosts reboot, `cloudera-scm-agent` starts automatically.

Manually Install CDH and Managed Service Packages

The CDH and Managed Service Packages contain all of the CDH software. You can choose to manually install CDH and the Managed Service Packages, or you can choose to let Cloudera Manager perform this installation in a later step. To use Cloudera Manager perform the installation, continue with Start the Cloudera Manager Server on page 141. Otherwise, follow the steps in (Optional) Manually Install CDH and Managed Service Packages on page 146 and then return to this page to continue the installation.

Install CDH and Managed Service Packages

Choose a Repository Strategy

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

- Standard Cloudera repositories. For this method, ensure you have added the required repository information to your systems.
- Internally hosted repositories. You might use internal repositories for environments where hosts do not have access to the Internet. For information about preparing your environment, see Understanding Custom Installation Solutions on page 166. When using an internal repository, you must copy the repo or list file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

Do one of the following:

- Install CDH 5 and Managed Service Packages on page 177
- Install CDH 4, Impala, and Solr Managed Service Packages on page 180

Install CDH 5 and Managed Service Packages

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

- **Red Hat**

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

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

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

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

b. Install the RPM for all RHEL versions:

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

2. (Optionally) add a repository key:

- **Red Hat/CentOS/Oracle 5**

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

- **Red Hat/CentOS/Oracle 6**

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

3. Install the CDH packages:

```
$ sudo yum clean all
$ sudo yum install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

- **SLES**

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

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

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

b. Install the RPM:

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

c. Update your system package index by running the following:

```
$ sudo zypper refresh
```

2. (Optionally) add a repository key:

- **SLES 11:**

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

- **SLES 12:**

```
$ sudo rpm --import
https://archive.cloudera.com/cdh5/sles/12/x86_64/cdh/RPM-GPG-KEY-cloudera
```

3. Install the CDH packages:

```
$ sudo zypper clean --all
$ sudo zypper install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

- **Ubuntu and Debian**

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

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

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

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

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

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

2. Optionally add a repository key:

   - **Debian Wheezy**

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

   - **Ubuntu Precise**

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

3. Install the CDH packages:

```
$ sudo apt-get update
$ sudo apt-get install avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3
hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase
hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper
impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry
solr-mapreduce spark-core spark-master spark-worker spark-python sqoop sqoop2 whirr
```

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

Install CDH 4, Impala, and Solr Managed Service Packages

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

- **RHEL-compatible**

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

     - **RHEL/CentOS/Oracle 5**

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

     - **RHEL/CentOS 6**

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

  4. Install packages on every host in your cluster:

     a. Install CDH 4 packages:

```
$ sudo yum -y install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins
 hbase hive oozie oozie-client pig zookeeper
```

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

```
$ sudo yum install hue
```

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

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

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

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

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

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

- **SLES**

  1. Run the following command:

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

2. Update your system package index by running:

```
$ sudo zypper refresh
```

3. Optionally add a repository key:

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

4. Install packages on every host in your cluster:

   a. Install CDH 4 packages:

```
$ sudo zypper install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins
 hbase hive oozie oozie-client pig zookeeper
```

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

```
$ sudo zypper install hue
```

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

      a. Run the following command:

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

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

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

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

      a. Run the following command:

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

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

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

- **Ubuntu or Debian**

1. In the table at CDH Version and Packaging Information, click the entry that matches your Ubuntu or Debian system.
2. Go to the list file (`cloudera.list`) for your system and save it in the `/etc/apt/sources.list.d/` directory. For example, to install CDH 4 for 64-bit Ubuntu Lucid, your `cloudera.list` file should look like:

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

3. Optionally add a repository key:

- **Ubuntu Lucid**

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

- **Ubuntu Precise**

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

- **Debian Squeeze**

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

4. Install packages on every host in your cluster:

   a. Install CDH 4 packages:

```
$ sudo apt-get install bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs
hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins
 hbase hive oozie oozie-client pig zookeeper
```

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

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

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

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

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

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

      a. In the table at Cloudera Search Version and Download Information, click the entry that matches your Ubuntu or Debian system.
      b. Install Solr Server on machines where you want Cloudera Search:

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

Start the Cloudera Manager Server

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

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

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

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

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

If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.

Start the Cloudera Manager Agents

If you using Cloudera Manager to install the Cloudera Manager Agent packages, *skip this section*. Otherwise, run the following command on each Agent host:

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

When the Agent starts, it contacts the Cloudera Manager Server. If communication fails between a Cloudera Manager Agent and Cloudera Manager Server, see Troubleshooting Installation and Upgrade Problems on page 425.

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

Start and Log into the Cloudera Manager Admin Console

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

1. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run `tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log` on the Cloudera Manager Server host. If the Cloudera Manager Server does not start, see Troubleshooting Installation and Upgrade Problems on page 425.
2. In a web browser, enter `http://Server host:7180`, where *Server host* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.
3. Log into Cloudera Manager Admin Console. The default credentials are: **Username:** `admin` **Password:** `admin`. Cloudera Manager does not support changing the `admin` username for the installed account. You can change the password using Cloudera Manager after you run the installation wizard. Although you cannot change the `admin` username, you can add a new user, assign administrative privileges to the new user, and then delete the default `admin` account.
4. After you log in, the **Cloudera Manager End User License Terms and Conditions** page displays. Read the terms and conditions and then select **Yes** to accept them.
5. Click **Continue**.

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

Choose Cloudera Manager Edition

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

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

   If you choose Cloudera Express or Cloudera Enterprise Enterprise Data Hub Edition Trial, you can upgrade the license at a later time. See Managing Licenses.
2. If you elect Cloudera Enterprise, install a license:
   a. Click **Upload License**.
   b. Click the document icon to the left of the **Select a License File** text field.
   c. Go to the location of your license file, click the file, and click **Open**.
   d. Click **Upload**.
3. Information is displayed indicating what the CDH installation includes. At this point, you can click the **Support** drop-down menu to access online Help or the Support Portal.
4. Click **Continue** to proceed with the installation.

Choose Cloudera Manager Hosts

Choose which hosts will run CDH and managed services

1. Do one of the following depending on whether you are using Cloudera Manager to install software:
   - If you are using Cloudera Manager to install software, search for and choose hosts:
     1. To enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the cluster hostnames or IP addresses. You can also specify hostname and IP address ranges. For example:

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

        You can specify multiple addresses and address ranges by separating them with commas, semicolons, tabs, or blank spaces, or by placing them on separate lines. Use this technique to make more specific searches instead of searching overly wide ranges. The scan results will include all addresses scanned, but only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. If you do not know the IP addresses of all of the hosts, you can enter an address range that spans over unused addresses and then clear the hosts that do not exist (and are not discovered) later in this procedure. However, keep in mind that wider ranges will require more time to scan.

     2. Click **Search**. Cloudera Manager identifies the hosts on your cluster to allow you to configure them for services. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking **Abort Scan**. To find additional hosts, click **New Search**, add the host names or IP addresses and click

Search again. Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts. Common causes of loss of connectivity are firewalls and interference from SELinux.

3. Verify that the number of hosts shown matches the number of hosts where you want to install services. Clear host entries that do not exist and clear the hosts where you do not want to install services.

- If you installed Cloudera Agent packages in Manually Install Cloudera Manager Agent Packages on page 140, choose from among hosts with the packages installed:

  1. Click the **Currently Managed Hosts** tab.
  2. Choose the hosts to add to the cluster.

2. Click **Continue**.

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

Choose the Software Installation Type and Install Software

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

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

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

        You can add additional parcels for lower versions by specifying custom repositories. For example, you can find the locations of the lower CDH 4 parcels at `https://username:password@archive.cloudera.com/p/cdh4/parcels/`. Or, if you are installing CDH 4.3 and want to use policy-file authorization, you can add the Sentry parcel using this mechanism.

        1. To specify the parcel directory, specify the local parcel repository, add a parcel repository, or specify the properties of a proxy server through which parcels are downloaded, click the **More Options** button and do one or more of the following:
           - **Parcel Directory** and **Local Parcel Repository Path** - Specify the location of parcels on cluster hosts and the Cloudera Manager Server host. If you change the default value for **Parcel Directory** and have already installed and started Cloudera Manager Agents, restart the Agents:

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

           - **Parcel Repository** - In the **Remote Parcel Repository URLs** field, click the ➕ button and enter the URL of the repository. The URL you specify is added to the list of repositories listed in the Configuring Cloudera Manager Server Parcel Settings on page 40 page and a parcel is added to the list of parcels on the Select Repository page. If you have multiple repositories configured, you see all the unique parcels contained in all your repositories.
           - **Proxy Server** - Specify the properties of a proxy server.
        2. Click **OK**.
     2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. You can choose either the version that matches the Cloudera Manager Server you are currently using

or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories.

- **Use Packages** - Do one of the following:
  - If Cloudera Manager is installing the packages:
    1. Click the package version.
    2. If you are using Cloudera Manager to install software, select the release of Cloudera Manager Agent. You can choose either the version that matches the Cloudera Manager Server you are currently using or specify a version in a custom repository. If you opted to use custom repositories for installation files, you can provide a GPG key URL that applies for all repositories.
  - If you manually installed packages in Manually Install CDH and Managed Service Packages on page 140 , select the CDH version (CDH 4 or CDH 5) that matches the packages you installed manually.

2. If you installed the Agent and JDK manually on all cluster hosts:
   - Click **Continue**.

     The Host Inspector runs to validate the installation and provides a summary of the results, including all the versions of the installed components. If the validation is successful, click **Finish**.
   - Skip the remaining steps in this section and continue with Add Services on page 144
3. Select **Install Oracle Java SE Development Kit (JDK)** to allow Cloudera Manager to install the JDK on each cluster host. If you have already installed the JDK, do not select this option. If your local laws permit you to deploy unlimited strength encryption, and you are running a secure cluster, select the **Install Java Unlimited Strength Encryption Policy Files** checkbox.

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

4. (Optional) Select **Single User Mode** to configure the Cloudera Manager Agent and all service processes to run as the same user. This mode requires extra configuration steps that must be done manually on all hosts in the cluster. If you have not performed the steps, directory creation will fail in the installation wizard. In most cases, you can create the directories but the steps performed by the installation wizard may have to be continued manually. Click **Continue**.
5. If you chose to have Cloudera Manager install software, specify host installation properties:
   - Select **root** or enter the username for an account that has password-less sudo permission.
   - Select an authentication method:
     - If you choose password authentication, enter and confirm the password.
     - If you choose public-key authentication, provide a passphrase and path to the required key files.
   - You can specify an alternate SSH port. The default value is 22.
   - You can specify the maximum number of host installations to run at once. The default value is 10.

   The root password (or any password used at this step) is not saved in Cloudera Manager or CDH. You can change these passwords after install without any impact to Cloudera Manager or CDH.
6. Click **Continue**. If you chose to have Cloudera Manager install software, Cloudera Manager installs the Oracle JDK, Cloudera Manager Agent, packages and CDH and managed service parcels or packages. During parcel installation, progress is indicated for the phases of the parcel installation process in separate progress bars. If you are installing multiple parcels, you see progress bars for each parcel. When the **Continue** button at the bottom of the screen turns blue, the installation process is completed.
7. Click **Continue**.

   The Host Inspector runs to validate the installation and provides a summary of the results, including all the versions of the installed components. If the validation is successful, click **Finish**.

Add Services

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

CDH 4	CDH 5
• **Core Hadoop** - HDFS, MapReduce, ZooKeeper, Oozie, Hive, and Hue • **Core with HBase** • **Core with Impala** • **All Services** - HDFS, MapReduce, ZooKeeper, HBase, Impala, Oozie, Hive, Hue, and Sqoop • **Custom Services** - Any combination of services.	• **Core Hadoop** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, and Hue • **Core with HBase** • **Core with Impala** • **Core with Search** • **Core with Spark** • **All Services** - HDFS, YARN (includes MapReduce 2), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer • **Custom Services** - Any combination of services.

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

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

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

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

     - The Flume service can be added only after your cluster has been set up.
   - If you have chosen Enterprise Data Hub Edition Trial or Cloudera Enterprise, optionally select the **Include Cloudera Navigator** checkbox to enable Cloudera Navigator. See Cloudera Navigator Data Management Overview.
2. Click **Continue**.
3. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

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

   The following shortcuts for specifying hostname patterns are supported:
   - Range of hostnames (without the domain portion)

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

- IP addresses
- Rack name

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

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

Configure Database Settings

On the Database Setup page, configure settings for required databases:

1. Enter the database host, database type, database name, username, and password for the database that you created when you set up the database.
2. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

   The **Review Changes** screen displays.

Review Configuration Changes and Start Services

1. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

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

2. Click **Continue**.

   The wizard starts the services.

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

Change the Default Administrator Password

As soon as possible, change the default administrator password:

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

Test the Installation

You can test the installation following the instructions in Testing the Installation on page 195.

Creating a CDH Cluster Using a Cloudera Manager Template

You can create a new CDH cluster by exporting a *cluster template* from an existing CDH cluster managed by Cloudera Manager. You can then modify the template and use it to create new clusters with the same configuration on a new set of hosts. Use cluster templates to:

- Duplicate clusters for use in developer, test, and production environments.
- Quickly create a cluster for a specific workload.
- Reproduce a production cluster for testing and debugging.

Follow these general steps to create a template and a new cluster:

1. Export the cluster configuration from the source cluster. The exported configuration is a JSON file that details all of the configurations of the cluster. The JSON file includes an `instantiator` section that contains some values you must provide before creating the new cluster.

   See Exporting the Cluster Configuration on page 189.

2. Set up the hosts for the new cluster by installing Cloudera Manager agents and the JDK on all hosts. For secure clusters, also configure a Kerberos key distribution center (KDC) in Cloudera Manager.

   See Preparing a New Cluster on page 190

3. Create any local repositories required for the cluster.

   See Establish Your Cloudera Manager Repository Strategy on page 137.

4. Complete the `instantiator` section of the cluster configuration JSON document to create a template.

   See Creating the Template on page 190.

5. Import the cluster template to the new cluster.

   See Importing the Template to a New Cluster on page 194.

Exporting the Cluster Configuration

To create a cluster template, you begin by exporting the configuration from the source cluster. The cluster must be running and managed by Cloudera Manager 5.7 or higher.

To export the configuration:

1. Any Host Templates you have created are used to export the configuration. If you do not want to use those templates in the new cluster, delete them. In Cloudera Manager, go to **Hosts** > **Host Templates** and click **Delete** next to the Host Template you want to delete.
2. Delete any Host Templates created by the Cloudera Manager Installation Wizard. They typically have a name like `Template - 1`).
3. Run the following command to download the JSON configuration file to a convenient location for editing:

```
curl -u admin_username:admin_user_password
"http://Cloudera Manager URL/api/v12/clusters/Cluster name/export" >
path_to_file/file_name.json
```

   For example:

```
curl -u adminuser:adminpass
"http://myCluster-1.myDomain.com:7180/api/v12/clusters/Cluster1/export" >
myCluster1-template.json
```

Note: Add the `?exportAutoConfig=true` parameter to the command above to include configurations made by Autoconfiguration. These configurations are included for reference only and are not used when you import the template into a new cluster. For example:

```
curl -u admin_username:admin_user_password
"http://Cloudera Manager URL/api/v12/clusters/Cluster
name/export?exportAutoConfig=true" >
path_to_file/file_name.json
```

Preparing a New Cluster

The new cluster into which you import the cluster template must meet the following requirements:

- Database for Cloudera Manager is installed and configured.
- Cloudera Manager 5.7 or higher is installed and running.
- All required databases for CDH services are installed. See Cloudera Manager and Managed Service Datastores on page 68.
- The JDK is installed on all cluster hosts.
- The Cloudera Manager Agent is installed and configured on all cluster hosts.
- If the source cluster uses Kerberos, the new cluster must have KDC properties and privileges configured in Cloudera Manager.
- If the source cluster used *packages* to install CDH and managed services, install those packages manually before importing the template. See Managing Software Installation Using Cloudera Manager on page 32.

Creating the Template

To create a template, modify the `instantiator` section of the JSON file you downloaded. Lines that contain the string <changeme> require a value that you must supply. Here is a sample `instantiator` section:

```
"instantiator" : {
  "clusterName" : "<changeme>",
  "hosts" : [ {
    "hostName" : "<changeme>",
    "hostTemplateRefName" : "<changeme>",
    "roleRefNames" : [ "HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e" ]
  }, {
    "hostName" : "<changeme>",
    "hostTemplateRefName" : "<changeme>"
  }, {
    "hostNameRange" : "<HOST[0001-0002]>",
    "hostTemplateRefName" : "<changeme>"
  } ],
  "variables" : [ {
    "name" : "HDFS-1-NAMENODE-BASE-dfs_name_dir_list",
    "value" : "/dfs/nn"
  }, {
    "name" : "HDFS-1-SECONDARYNAMENODE-BASE-fs_checkpoint_dir_list",
    "value" : "/dfs/snn"
  }, {
    "name" : "HIVE-1-hive_metastore_database_host",
    "value" : "myCluster-1.myDomain.com"
  }, {
    "name" : "HIVE-1-hive_metastore_database_name",
    "value" : "hive1"
  }, {
    "name" : "HIVE-1-hive_metastore_database_password",
    "value" : "<changeme>"
  }, {
    "name" : "HIVE-1-hive_metastore_database_port",
    "value" : "3306"
  }, {
    "name" : "HIVE-1-hive_metastore_database_type",
    "value" : "mysql"
```

```
    }, {
      "name" : "HIVE-1-hive_metastore_database_user",
      "value" : "hive1"
    }, {
      "name" : "HUE-1-database_host",
      "value" : "myCluster-1.myDomain.com"
    }, {
      "name" : "HUE-1-database_name",
      "value" : "hueserver0be88b55f5dedbf7bc74d61a86c0253e"
    }, {
      "name" : "HUE-1-database_password",
      "value" : "<changeme>"
    }, {
      "name" : "HUE-1-database_port",
      "value" : "3306"
    }, {
      "name" : "HUE-1-database_type",
      "value" : "mysql"
    }, {
      "name" : "HUE-1-database_user",
      "value" : "hueserver0be88b5"
    }, {
      "name" : "IMPALA-1-IMPALAD-BASE-scratch_dirs",
      "value" : "/impala/impalad"
    }, {
      "name" : "KUDU-1-KUDU_MASTER-BASE-fs_data_dirs",
      "value" : "/var/lib/kudu/master"
    }, {
      "name" : "KUDU-1-KUDU_MASTER-BASE-fs_wal_dir",
      "value" : "/var/lib/kudu/master"
    }, {
      "name" : "KUDU-1-KUDU_TSERVER-BASE-fs_data_dirs",
      "value" : "/var/lib/kudu/tserver"
    }, {
      "name" : "KUDU-1-KUDU_TSERVER-BASE-fs_wal_dir",
      "value" : "/var/lib/kudu/tserver"
    }, {
      "name" : "MAPREDUCE-1-JOBTRACKER-BASE-jobtracker_mapred_local_dir_list",
      "value" : "/mapred/jt"
    }, {
      "name" : "MAPREDUCE-1-TASKTRACKER-BASE-tasktracker_mapred_local_dir_list",
      "value" : "/mapred/local"
    }, {
      "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_host",
      "value" : "myCluster-1.myDomain.com:3306"
    }, {
      "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_name",
      "value" : "oozieserver0be88b55f5dedbf7bc74d61a86c0253e"
    }, {
      "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_password",
      "value" : "<changeme>"
    }, {
      "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_type",
      "value" : "mysql"
    }, {
      "name" : "OOZIE-1-OOZIE_SERVER-BASE-oozie_database_user",
      "value" : "oozieserver0be88"
    }, {
      "name" : "YARN-1-NODEMANAGER-BASE-yarn_nodemanager_local_dirs",
      "value" : "/yarn/nm"
    }, {
      "name" : "YARN-1-NODEMANAGER-BASE-yarn_nodemanager_log_dirs",
      "value" : "/yarn/container-logs"
    } ]
  }
```

To modify the template:

1. Update the `hosts` section.

   If you have host templates defined in the source cluster, they appear in the `hostTemplates` section of the JSON template. For hosts that do not use host templates, the export process creates host templates based on role

assignments to facilitate creating the new cluster. In either case, you must match the items in the `hostTemplates` section with the `hosts` sections in the `instantiator` section.

Here is a sample of the `hostTemplates` section from the same JSON file as the `instantiator` section, above:

```
 "hostTemplates" : [ {
    "refName" : "HostTemplate-0-from-myCluster-1.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-GATEWAY-BASE",
"HBASE-1-HBASETHRIFTSERVER-BASE", "HBASE-1-MASTER-BASE", "HDFS-1-BALANCER-BASE",
"HDFS-1-GATEWAY-BASE", "HDFS-1-NAMENODE-BASE", "HDFS-1-NFSGATEWAY-BASE",
"HDFS-1-SECONDARYNAMENODE-BASE", "HIVE-1-GATEWAY-BASE", "HIVE-1-HIVEMETASTORE-BASE",
"HIVE-1-HIVESERVER2-BASE", "HUE-1-HUE_LOAD_BALANCER-BASE", "HUE-1-HUE_SERVER-BASE",
"IMPALA-1-CATALOGSERVER-BASE", "IMPALA-1-STATESTORE-BASE", "KAFKA-1-KAFKA_BROKER-BASE",
 "KS_INDEXER-1-HBASE_INDEXER-BASE", "KUDU-1-KUDU_MASTER-BASE", "MAPREDUCE-1-GATEWAY-BASE",
 "MAPREDUCE-1-JOBTRACKER-BASE", "OOZIE-1-OOZIE_SERVER-BASE", "SOLR-1-SOLR_SERVER-BASE",
 "SPARK_ON_YARN-1-GATEWAY-BASE", "SPARK_ON_YARN-1-SPARK_YARN_HISTORY_SERVER-BASE",
"SQOOP-1-SQOOP_SERVER-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-GATEWAY-BASE",
"YARN-1-JOBHISTORY-BASE", "YARN-1-RESOURCEMANAGER-BASE", "ZOOKEEPER-1-SERVER-BASE" ]
  }, {
    "refName" : "HostTemplate-1-from-myCluster-4.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE"
 ]
  }, {
    "refName" : "HostTemplate-2-from-myCluster-[2-3].myDomain.com",
    "cardinality" : 2,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KAFKA-1-KAFKA_BROKER-BASE", "KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
 "SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE"
 ]
  } ]
```

The value of `cardinality` indicates how many hosts are assigned to the host template in the source cluster.

The value of `roleConfigGroupsRefNames` indicates which role groups are assigned to the host(s).

Do the following for each host template in the `hostTemplates` section:

1. Locate the entry in the `hosts` section of the `instantiator` where you want the roles to be installed.
2. Copy the value of the `refName` to the value for `hostTemplateRefName`.
3. Enter the hostname in the new cluster as the value for `hostName`. Some host sections might instead use `hostNameRange` for clusters with multiple hosts that have the same set of roles. Indicate a range of hosts by using one of the following:
   - Brackets; for example, `myhost[1-4].foo.com`
   - A comma-delimited string of hostnames; for example, `host-1.domain, host-2.domain, host-3.domain`

Here is an example of the `hostTemplates` and the `hosts` section of the `instantiator` completed correctly:

```
"hostTemplates" : [ {
    "refName" : "HostTemplate-0-from-myCluster-1.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-GATEWAY-BASE",
"HBASE-1-HBASETHRIFTSERVER-BASE", "HBASE-1-MASTER-BASE", "HDFS-1-BALANCER-BASE",
"HDFS-1-GATEWAY-BASE", "HDFS-1-NAMENODE-BASE", "HDFS-1-NFSGATEWAY-BASE",
"HDFS-1-SECONDARYNAMENODE-BASE", "HIVE-1-GATEWAY-BASE", "HIVE-1-HIVEMETASTORE-BASE",
"HIVE-1-HIVESERVER2-BASE", "HUE-1-HUE_LOAD_BALANCER-BASE", "HUE-1-HUE_SERVER-BASE",
"IMPALA-1-CATALOGSERVER-BASE", "IMPALA-1-STATESTORE-BASE", "KAFKA-1-KAFKA_BROKER-BASE",
 "KS_INDEXER-1-HBASE_INDEXER-BASE", "KUDU-1-KUDU_MASTER-BASE", "MAPREDUCE-1-GATEWAY-BASE",
 "MAPREDUCE-1-JOBTRACKER-BASE", "OOZIE-1-OOZIE_SERVER-BASE", "SOLR-1-SOLR_SERVER-BASE",
 "SPARK_ON_YARN-1-GATEWAY-BASE", "SPARK_ON_YARN-1-SPARK_YARN_HISTORY_SERVER-BASE",
"SQOOP-1-SQOOP_SERVER-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-GATEWAY-BASE",
"YARN-1-JOBHISTORY-BASE", "YARN-1-RESOURCEMANAGER-BASE", "ZOOKEEPER-1-SERVER-BASE" ]
```

```
  }, {
    "refName" : "HostTemplate-1-from-myCluster-4.myDomain.com",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
"SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE"
 ]
  }, {
    "refName" : "HostTemplate-2-from-myCluster-[2-3].myDomain.com",
    "cardinality" : 2,
    "roleConfigGroupsRefNames" : [ "FLUME-1-AGENT-BASE", "HBASE-1-REGIONSERVER-BASE",
"HDFS-1-DATANODE-BASE", "HIVE-1-GATEWAY-BASE", "IMPALA-1-IMPALAD-BASE",
"KAFKA-1-KAFKA_BROKER-BASE", "KUDU-1-KUDU_TSERVER-BASE", "MAPREDUCE-1-TASKTRACKER-BASE",
 "SPARK_ON_YARN-1-GATEWAY-BASE", "SQOOP_CLIENT-1-GATEWAY-BASE", "YARN-1-NODEMANAGER-BASE"
 ]
  } ],
  "instantiator" : {
    "clusterName" : "myCluster_new",
    "hosts" : [ {
      "hostName" : "myNewCluster-1.myDomain.com",
      "hostTemplateRefName" : "HostTemplate-0-from-myCluster-1.myDomain.com",
      "roleRefNames" : [ "HDFS-1-NAMENODE-c975a0b51fd36e914896cd5e0adb1b5b" ]
    }, {
      "hostName" : "myNewCluster-5.myDomain.com",
      "hostTemplateRefName" : "HostTemplate-1-from-myCluster-4.myDomain.com"
    }, {
      "hostNameRange" : "myNewCluster-[3-4].myDomain.com",
      "hostTemplateRefName" : "HostTemplate-2-from-myCluster-[2-3].myDomain.com"
    } ],
```

2. For host sections that have a `roleRefNames` line, determine the role type and assign the appropriate host for the role. If there are multiple instances of a role, you must select the correct hosts. To determine the role type, search the template file for the value of `roleRefNames`.

   For example: For a role ref named `HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e`, if you search for that string, you find a section similar to the following:

```
"roles": [
{
"refName": "HDFS-1-NAMENODE-0be88b55f5dedbf7bc74d61a86c0253e",
"roleType": "NAMENODE"
}
]
```

   In this case, the role type is `NAMENODE`.

3. Modify the `variables` section. This section contains various properties from the source cluster. You can change any of these values to be different in the new cluster, or you can leave the values as copied from the source. For any values shown as <changeme>, you must provide the correct value.

   Note: Many of these variables contain information about databases used by the Hive Metastore and other CDH components. Change the values of these variables to match the databases configured for the new cluster.

4. Enter the internal name of the new cluster on the line with `"clusterName" : "<changeme>"`. For example:

```
"clusterName" : "QE_test_cluster"
```

5. (Optional) Change the display name for the cluster. Edit the line that begins with `"displayName"` (near the top of the JSON file); for example:

```
"displayName" : "myNewCluster",
```

Importing the Template to a New Cluster

To import the cluster template:

1. Log in to the Cloudera Manager server as root.
2. Run the following command to import the template. If you have remote repository URLS configured in the source cluster, append the command with `?addRepositories=true`.

```
curl -X POST -H "Content-Type: application/json" -d
      @path_to_template/template_filename.json
http://admin_user:admin_password@cloudera_manager_url:cloudera_manager_port/api/v12/cm/importClusterTemplate
```

You should see a response similar to the following:

```
{
  "id" : 17,
  "name" : "ClusterTemplateImport",
  "startTime" : "2016-03-09T23:44:38.491Z",
  "active" : true,
  "children" : {
    "items" : [ ]
  }
```

Examples:

```
curl -X POST -H "Content-Type: application/json" -d @myTemplate.json
http://admin:admin@myNewCluster-1.mydomain.com:7182/api/v12/cm/importClusterTemplate
```

```
curl -X POST -H "Content-Type: application/json" -d @myTemplate.json
http://admin:admin@myNewCluster-1.mydomain.com:7182/api/v12/cm/importClusterTemplate?addRepositories=true
```

If there is no response, or you receive an error message, the JSON file may be malformed, or the template may have invalid hostnames or invalid references. Inspect the JSON file, correct any errors, and then re-run the command.

3. Open Cloudera Manager for the new cluster in a web browser and click the Cloudera Manager logo to go to the home page.
4. Click the **All Recent Commands** tab.

   If the import is proceeding, you should see a link labeled **Import Cluster Template**. Click the link to view the progress of the import.

   If any of the commands fail, correct the problem and click **Retry**. You may need to edit some properties in Cloudera Manager.

After you import the template, Cloudera Manager applies the Autoconfiguration rules that set properties such as memory and CPU allocations for various roles. If the new cluster has different hardware or operational requirements, you may need to modify these values.

Sample Python Code

You can perform the steps to export and import a cluster template programmatically using a client written in Python or other languages. (You can also use the `curl` commands provided above.)

Python export example:

```
resource = ApiResource("myCluster-1.myDomain.com", 7180, "admin", "admin", version=12)
cluster = resource.get_cluster("Cluster1");
template = cluster.export(False)
pprint(template)
```

Python import example:

```
resource = ApiResource("localhost", 8180, "admin", "admin", version=12)
 with open('~/cluster-template.json') as data_file:
   data = json.load(data_file)
template = ApiClusterTemplate(resource).from_json_dict(data, resource)
cms = ClouderaManager(resource)
cms.import_cluster_template(template)
```

Deploying Clients

Client configuration files are generated automatically by Cloudera Manager based on the services you install.

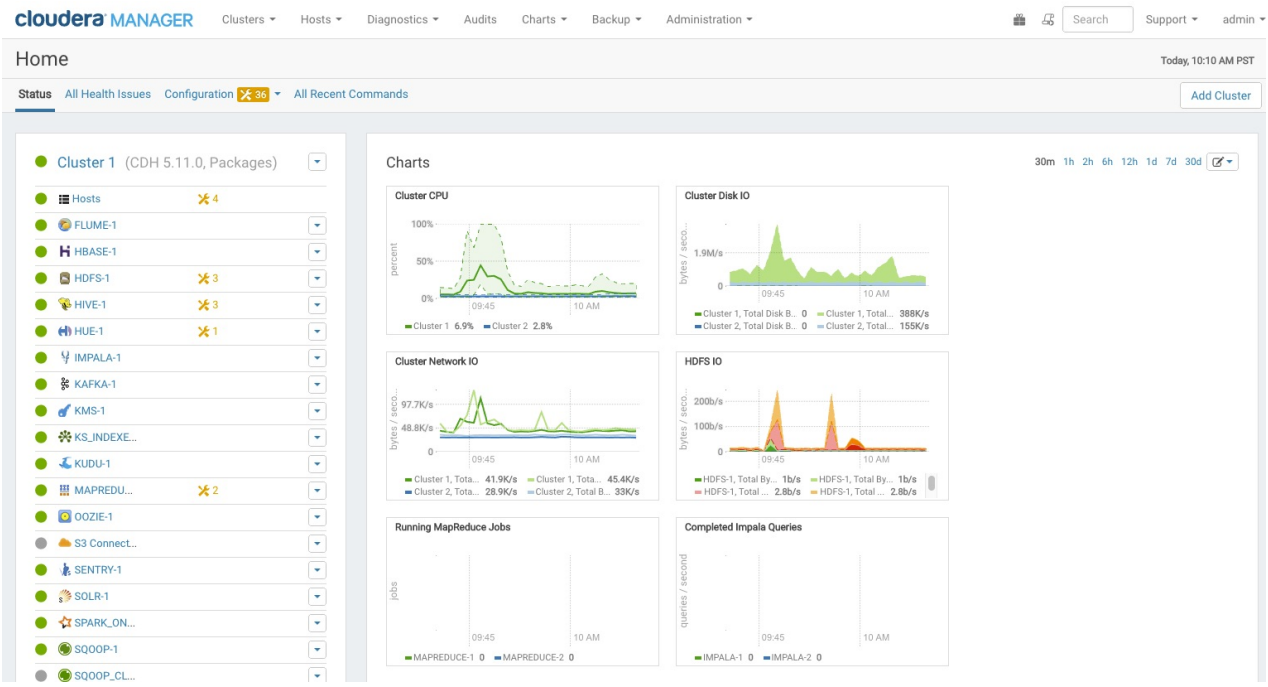
Cloudera Manager deploys these configurations automatically at the end of the installation workflow. You can also download the client configuration files to deploy them manually.

If you modify the configuration of your cluster, you may need to redeploy the client configuration files. If a service's status is "Client configuration redeployment required," you need to redeploy those files.

See Client Configuration Files for information on downloading client configuration files, or redeploying them through Cloudera Manager.

Testing the Installation

To begin testing, start the Cloudera Manager Admin Console. Once you've logged in, the Home page should look something like this:



On the left side of the screen is a list of services currently running with their status information. All the services should be running with **Good Health**. You can click each service to view more detailed information about each service. You can also test your installation by either checking each Host's heartbeats, running a MapReduce job, or interacting with the cluster with an existing Hue application.

Checking Host Heartbeats

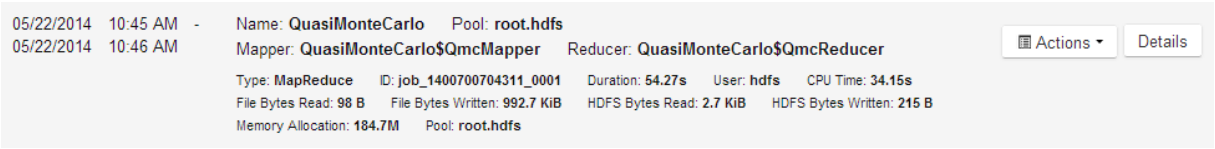
One way to check whether all the Agents are running is to look at the time since their last heartbeat. You can do this by clicking the **Hosts** tab where you can see a list of all the Hosts along with the value of their **Last Heartbeat**. By default, every Agent must heartbeat successfully every 15 seconds. A recent value for the **Last Heartbeat** means that the Server and Agents are communicating successfully.

Running a MapReduce Job

1. Log into a host in the cluster.
2. Run the Hadoop PiEstimator example using one of the following commands:
   - **Parcel** - `sudo -u hdfs hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100`
   - **Package** - `sudo -u hdfs hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100`

   or create and run the WordCount v1.0 application described in Hadoop Tutorial.
3. Depending on whether your cluster is configured to run MapReduce jobs on the YARN or MapReduce service, view the results of running the job by selecting one of the following from the top navigation bar in the Cloudera Manager Admin Console :
   - **Clusters** > ***ClusterName*** > **yarn Applications**
   - **Clusters** > ***ClusterName*** > **mapreduce Activities**

   If you run the PiEstimator job on the YARN service (the default) you will see an entry like the following in **yarn Applications**:



Testing with Hue

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

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

For more information, see the Hue User Guide.

Uninstalling Cloudera Manager and Managed Software

Use the following instructions to uninstall the Cloudera Manager Server, Agents, managed software, and databases.

Uninstalling Cloudera Manager and Managed Software

Follow the steps in this section to remove software and data.

Record User Data Paths

The user data paths listed Remove User Data on page 200, `/var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/zookeeper` *`data_drive_path`*`/dfs` *`data_drive_path`*`/mapred` *`data_drive_path`*`/yarn`, are the default settings. However, at some point they may have been reconfigured in Cloudera Manager. If you want to remove all user data from the cluster and have changed the paths, either when you installed CDH and managed services or at some later time, note the location of the paths by checking the configuration in each service.

Stop all Services

1. For each cluster managed by Cloudera Manager:
   a. On the **Home** > **Status** tab, click

   to the right of the cluster name and select **Stop**.
   b. Click **Stop** in the confirmation screen. The **Command Details** window shows the progress of stopping services. When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.
   c. On the **Home** > **Status** tab, click

   to the right of the Cloudera Management Service entry and select **Stop**. The **Command Details** window shows the progress of stopping services. When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.
2. a. Do one of the following:
   - 1. Select **Clusters** > **Cloudera Management Service**.
     2. Select **Actions** > **Stop**.
   - 1. On the **Home** > **Status** tab, click

     to the right of **Cloudera Management Service** and select **Stop**.

   b. Click **Stop** to confirm. The **Command Details** window shows the progress of stopping the roles.
   c. When **Command completed with *n*/*n* successful subcommands** appears, the task is complete. Click **Close**.

.

Deactivate and Remove Parcels

If you installed using packages, skip this step and go to Uninstall the Cloudera Manager Server on page 198; you will remove packages in Uninstall Cloudera Manager Agent and Managed Software on page 198. If you installed using parcels remove them as follows:

1. Click the parcel indicator in the main navigation bar.
2. In the **Location** selector on the left, select **All Clusters**.
3. For each activated parcel, select **Actions** > **Deactivate**. When this action has completed, the parcel button changes to **Activate**.
4. For each activated parcel, select **Actions** > **Remove from Hosts**. When this action has completed, the parcel button changes to **Distribute**.
5. For each activated parcel, select **Actions** > **Delete**. This removes the parcel from the local parcel repository.

There may be multiple parcels that have been downloaded and distributed, but that are not active. If this is the case, you should also remove those parcels from any hosts onto which they have been distributed, and delete the parcels from the local repository.

Delete the Cluster

On the **Home** page, Click the drop-down list next to the cluster you want to delete and select **Delete**.

Uninstall the Cloudera Manager Server

The commands for uninstalling the Cloudera Manager Server depend on the method you used to install it. Refer to steps below that correspond to the method you used to install the Cloudera Manager Server.

- **If you used the cloudera-manager-installer.bin file** - Run the following command on the Cloudera Manager Server host:

```
$ sudo /usr/share/cmf/uninstall-cloudera-manager.sh
```

- **If you did not use the cloudera-manager-installer.bin file** - If you installed the Cloudera Manager Server using a different installation method such as Puppet, run the following commands on the Cloudera Manager Server host.

  1. Stop the Cloudera Manager Server and its database:

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

  2. Uninstall the Cloudera Manager Server and its database. This process described also removes the embedded PostgreSQL database software, if you installed that option. If you did not use the embedded PostgreSQL database, omit the `cloudera-manager-server-db` steps.

     RHEL systems:

```
sudo yum remove cloudera-manager-server
sudo yum remove cloudera-manager-server-db-2
```

     SLES systems:

```
sudo zypper -n rm --force-resolution cloudera-manager-server
sudo zypper -n rm --force-resolution cloudera-manager-server-db-2
```

     Debian/Ubuntu systems:

```
sudo apt-get remove cloudera-manager-server
sudo apt-get remove cloudera-manager-server-db-2
```

Uninstall Cloudera Manager Agent and Managed Software

Do the following on all Agent hosts:

1. Stop the Cloudera Manager Agent.

   RHEL-compatible 7 and higher

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

   All other RHEL/SLES systems:

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

   Debian/Ubuntu systems:

```
$ sudo /usr/sbin/service cloudera-scm-agent hard_stop
```

2. Uninstall software:

OS	Parcel Install	Package Install
RHEL	`$ sudo yum remove 'cloudera-manager-*'`	• **CDH 5** `$ sudo yum remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper`
SLES	`$ sudo zypper remove 'cloudera-manager-*'`	• **CDH 5** `$ sudo zypper remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper`
Debian/Ubuntu	`$ sudo apt-get purge 'cloudera-manager-*'`	• **CDH 5** `$ sudo apt-get purge 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 whirr hue-common oozie-client solr solr-doc sqoop2-client zookeeper`

3. Run the `clean` command:

RHEL

```
$ sudo yum clean all
```

SLES

```
$ sudo zypper clean
```

Debian/Ubuntu

```
$ sudo apt-get clean
```

Remove Cloudera Manager and User Data

Kill Cloudera Manager and Managed Processes

On all Agent hosts, kill any running Cloudera Manager and managed processes:

```
$ for u in cloudera-scm flume hadoop hdfs hbase hive httpfs hue impala llama mapred
oozie solr spark sqoop sqoop2 yarn zookeeper; do sudo kill $(ps -u $u -o pid=); done
```

> **Note:** This step should not be necessary if you stopped all the services and the Cloudera Manager Agent correctly.

Remove Cloudera Manager Data

If you are uninstalling on RHEL, run the following commands on all Agent hosts to permanently remove Cloudera Manager data. If you want to be able to access any of this data in the future, you must back it up before removing it. If you used an embedded PostgreSQL database, that data is stored in `/var/lib/cloudera-scm-server-db`.

```
$ sudo umount cm_processes
$ sudo rm -Rf /usr/share/cmf /var/lib/cloudera* /var/cache/yum/cloudera*
/var/log/cloudera* /var/run/cloudera*
```

Remove the Cloudera Manager Lock File

On all Agent hosts, run this command to remove the Cloudera Manager lock file:

```
$ sudo rm /tmp/.scm_prepare_node.lock
```

Remove User Data

This step permanently removes all user data. To preserve the data, copy it to another cluster using the `distcp` command before starting the uninstall process. On all Agent hosts, run the following commands:

```
$ sudo rm -Rf /var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/lib/navigator
/var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/zookeeper
```

Run the following command on each data drive on all Agent hosts (adjust the paths for the data drives on each host):

```
$ sudo rm -Rf data_drive_path/dfs data_drive_path/mapred data_drive_path/yarn
```

> **Note:** For additional information about uninstalling CDH, including clean-up of CDH files, see Uninstalling CDH Components on page 420.

Stop and Remove External Databases

If you chose to store Cloudera Manager or user data in an external database, see the database vendor documentation for details on how to remove the databases.

Uninstalling a CDH Component From a Single Host

The following procedure removes CDH software components from a single host that is managed by Cloudera Manager.

1. In the Cloudera Manager Administration Console, select the **Hosts** tab.

   A list of hosts in the cluster displays.

2. Select the host where you want to uninstall CDH software.
3. Click the **Actions for Selected** button and select **Remove From Cluster**.

   Cloudera Manager removes the roles and host from the cluster.

4. (Optional) Manually delete the `krb5.conf` file used by Cloudera Manager.

Installing the Cloudera Navigator Data Management Component

The Cloudera Navigator data management component is implemented in two distinct roles—Navigator Audit Server, and Navigator Metadata Server—that run on the Cloudera Management Service. These roles can be added during the initial Cloudera Manager installation, or added later to an existing Cloudera Manager cluster.

Important: Cloudera Navigator Data Management requires a Cloudera Enterprise license. This feature is not available in Cloudera Express. See Managing Licenses for details.

The steps on this page are for installing Cloudera Navigator as part of a new Cloudera Manager cluster installation and for adding the service to an existing cluster. For information about upgrading an existing deployment, see Upgrading the Cloudera Navigator Data Management Component.

Note: See the product compatibility matrix for information on compatible Cloudera Navigator and Cloudera Manager versions.

Minimum Recommended Memory and Disk Space

Resource	Navigator Audit Server	Navigator Metadata Server
Memory	Varies, but requires less than Navigator Metadata Server	40 GB total
Java heap size	2 – 3 GB	10–20 GB (initial setup)
OS buffer cache	20 GB	20 GB (initial setup). Increase by 20-GB increments over time as needed.
Disk	Multiple hundreds of GB. Depends on cluster size and audit volumes generated.	200 GB (SSD recommended)
Default path	None. Location of the Cloudera Navigator database.	`/var/lib/cloudera-scm-navigator`

Navigator Metadata Server and Navigator Audit Server have different recommended configurations that you should consider when you plan your deployment. For initial installation, keep the following in mind:

- **Navigator Audit Server Memory and Disk Requirements**—For Navigator Audit Server, a Java heap size of 2-3 GB (gigabytes) is usually sufficient (memory typically does not pose any issues). For Navigator Audit Server, it is the database configuration that can affect performance and so must be configured properly. Because Navigator Audit Server may need to push millions of rows of audit data daily (depending on the cluster size, number of services, and other factors), Cloudera recommends:
  - Set up the database on the same host as the Navigator Audit Server to minimize latency.
  - Monitor the database workload over time and tune as needed.
- **Navigator Metadata Server Memory and Disk Requirements**—Navigator Metadata Server relies on an embedded Solr instance for its Search capability. The Solr indexes are saved locally to the host's hard-disk drive and typically consume only tens of GBs of disk space, so allocating ~200 GBs for the data is usually sufficient. For Navigator Metadata Server disk, Cloudera recommends:

- Mount SSD drives on the host where the Solr index will be located, for fastest I/O.
- Use the Purge function once the system is up and running to keep the hard-disk drive consumption at that location in check.

Bottlenecks that may emerge for Navigator Metadata Server are typically associated with I/O and memory (not CPU). Memory includes Java heap size and available RAM that can be used for the OS buffer cache setting. For Navigator Metadata Server RAM, Cloudera recommends:

- Set Java heap size to 10-20 GB, which should be sufficient for initial setup.
- Increase the OS buffer cache by 20 GB to improve performance if necessary, depending on the cluster activity.

See Memory Sizing Considerations for Navigator Metadata Server for more information.

Configuring a Database for Cloudera Navigator

During the Cloudera Navigator installation process, you must select a database to store audit events and policy, role, and audit report metadata. You can choose the embedded PostgreSQL database, or you can choose an external database such as Oracle or MySQL (see Cloudera Navigator Supported Databases for other supported database systems).

For production environments, Cloudera recommends using an external database rather than the embedded PostgreSQL database. In addition, the database must be setup and running before you begin the installation process. For more information, see Cloudera Manager and Managed Service Datastores on page 68.

Adding Cloudera Navigator Roles During the Cloudera Manager Installation Process

Cloudera Manager Required Role: Full Administrator

1. Install Cloudera Manager as detailed in Cloudera Manager Deployment on page 54.
2. On the first page of the Cloudera Manager installation wizard, choose one of the license options that supports Cloudera Navigator:
   - Cloudera Enterprise Enterprise Data Hub Edition Trial
   - Cloudera Enterprise, and upload the license:
     1. Click **Upload License**.
     2. Click the document icon to the left of the **Select a License File** text field.
     3. Go to the location of your license file, click the file, and click **Open**.
     4. Click **Upload**.
   - Flex Edition
   - Enterprise Data Hub Edition
3. Click **Continue** to proceed with the installation.
4. In the first page of the **Add Services** procedure, click the **Include Cloudera Navigator** checkbox.
5. To use external databases, enter the Cloudera Navigator Audit Server and Metadata Server database properties in the **Database Setup** page.

Adding Cloudera Navigator Data Management Roles to an Existing Cloudera Manager Cluster

If the Cloudera Manager cluster has sufficient resources, you can add instances of either Cloudera Navigator roles to the cluster at any time. See the appropriate topic from the Administration Guide for details:

- Adding the Navigator Audit Server Role
- Adding the Navigator Metadata Server Role

Cloudera Navigator Data Management Documentation

Other topics related to configuring, upgrading, managing, and using Cloudera Navigator Data Management component are listed in the following table.

FAQ	Cloudera Navigator Frequently Asked Questions answers common questions about Cloudera Navigator data management component and how it interacts with other Cloudera products and cluster components.
Introduction	Cloudera Navigator Data Management Overview provides an overview for data stewards, governance and compliance teams, data engineers, and administrators. Includes Getting Started with Cloudera Navigator, an overview of the Cloudera Navigator console (the UI) and the Cloudera Navigator APIs.
User Guide	Cloudera Data Management guide shows data stewards, compliance officers, and other business users how to use Cloudera Navigator for data governance, compliance, data stewardship, and other tasks.
Upgrade	Upgrading the Cloudera Navigator Data Management Component
Administration	Cloudera Navigator Data Management Component Administration
	Cloudera Navigator Data Management User Roles
Security	Configuring Authentication for Cloudera Navigator Data Management
	Configuring TLS/SSL for Cloudera Navigator Metadata Server
Release Notes	Cloudera Navigator Data Management Release Notes

Installing Cloudera Navigator Key Trustee Server

Important: Before installing Cloudera Navigator Key Trustee Server, see Encrypting Data at Rest for important considerations.

When the Key Trustee Server role is created it is tightly bound to the identity of the host on which it is installed. Moving the role to a different host, changing the host name, or changing the IP of the host is *not* supported

You can install Navigator Key Trustee Server using Cloudera Manager with parcels or using the command line with packages. See Parcels on page 32 for more information on parcels.

Note: If you are using or planning to use Key Trustee Server in conjunction with a CDH cluster, Cloudera strongly recommends using Cloudera Manager to install and manage Key Trustee Server to take advantage of Cloudera Manager's robust deployment, management, and monitoring capabilities.

Prerequisites

See Data at Rest Encryption Requirements for more information about encryption and Key Trustee Server requirements.

Setting Up an Internal Repository

You must create an internal repository to install or upgrade the Cloudera Navigator data encryption components. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see the following topics:

- Creating and Using a Parcel Repository for Cloudera Manager on page 168
- Creating and Using a Package Repository for Cloudera Manager on page 171

Installing Key Trustee Server

> **Important:** This feature is available only with a Cloudera Enterprise license. It is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see Managing Licenses.

Installing Key Trustee Server Using Cloudera Manager

> **Note:** These instructions apply to using Cloudera Manager only. To install Key Trustee Server using packages, skip to Installing Key Trustee Server Using the Command Line on page 204.

If you are installing Key Trustee Server for use with HDFS Transparent Encryption, the **Set up HDFS Data At Rest Encryption** wizard installs and configures Key Trustee Server. See Enabling HDFS Encryption Using the Wizard for instructions.

1. **(Recommended)** Create a new cluster in Cloudera Manager containing only the host that Key Trustee Server will be installed on. Cloudera recommends that each cluster use its own KTS instance. Although sharing a single KTS across clusters is technically possible, it is *neither approved nor supported* for security reasons—specifically, the increased security risks associated with single point of failure for encryption keys used by multiple clusters. For a better understanding of additional security reasons for this recommendation, see Data at Rest Encryption Reference Architecture. See Adding and Deleting Clusters for instructions on how to create a new cluster in Cloudera Manager.

   > **Important:** The **Add Cluster** wizard prompts you to install CDH and other cluster services. To exit the wizard without installing CDH, select a version of CDH to install and continue. When the installation begins, click the Cloudera Manager logo in the upper left corner and confirm you want to exit the wizard. This allows you to create the dedicated cluster with the Key Trustee Server hosts without installing CDH or other services that are not required for Key Trustee Server.

2. Add the internal parcel repository you created in Setting Up an Internal Repository on page 203 to Cloudera Manager following the instructions in Configuring Cloudera Manager Server Parcel Settings on page 40.
3. Download, distribute, and activate the Key Trustee Server parcel on the cluster containing the Key Trustee Server host, following the instructions in Managing Parcels on page 34.

   > **Important:** The KEYTRUSTEE parcel in Cloudera Manager is *not* the Key Trustee Server parcel; it is the Key Trustee KMS parcel. The parcel name for Key Trustee Server is KEYTRUSTEE_SERVER.

   After you activate the Key Trustee Server parcel, Cloudera Manager prompts you to restart the cluster. Click the **Close** button to ignore this prompt. You *do not* need to restart the cluster after installing Key Trustee Server.

After installing Key Trustee Server using Cloudera Manager, continue to Securing Key Trustee Server Host on page 206.

Installing Key Trustee Server Using the Command Line

> **Note:** These instructions apply to package-based installations using the command line only. To install Key Trustee Server using Cloudera Manager, see Installing Key Trustee Server Using Cloudera Manager on page 204.
>
> If you are using or planning to use Key Trustee Server in conjunction with a CDH cluster, Cloudera strongly recommends using Cloudera Manager to install and manage Key Trustee Server to take advantage of Cloudera Manager's robust deployment, management, and monitoring capabilities.

1. **Install the EPEL Repository**

   Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the `epel-release` package:

1. Copy the URL for the `epel-release-<version>.noarch` 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
```

2. **(RHEL 7 Only) Enable the `extras` Repository**

Key Trustee Server requires the `python-flask` package. For RHEL 6, this package is provided in the EPEL repository. For RHEL 7, it is provided in the RHEL `extras` repository. To enable this repository, run the following command:

```
$ sudo subscription-manager repos --enable=rhel-7-server-extras-rpms
```

3. **Install the PostgreSQL 9.3 Repository**

Note: Cloudera Navigator Key Trustee Server currently supports only PostgreSQL version 9.3. If you have a different version of PostgreSQL installed on the Key Trustee Server host, remove it before proceeding or select a different host on which to install Key Trustee Server.

To install the PostgreSQL 9.3 repository, run the following command:

```
$ sudo yum install
http://yum.postgresql.org/9.3/redhat/rhel-6-x86_64/pgdg-redhat93-9.3-3.noarch.rpm
```

Important: If you are using CentOS, add the following line to the CentOS `base` repository:

```
exclude=python-psycopg2*
```

By default, the `base` repository is located at `/etc/yum.repos.d/CentOS-Base.repo`. If you have an internal mirror of the `base` repository, update the correct file for your environment.

4. **Install the Cloudera Repository**

Add the internal repository you created. See Modifying Clients to Find the Repository on page 173 for more information.

Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
```

5. **Install the CDH Repository**

Key Trustee Server and Key HSM depend on the `bigtop-utils` package, which is included in the CDH repository. For instructions on adding the CDH repository, see To add the CDH repository. To create a local CDH repository, see Creating a Local Yum Repository on page 222 for instructions.

6. **Install NTP**

   The Network Time Protocol (NTP) service synchronizes system time. Cloudera recommends using NTP to ensure that timestamps in system logs, cryptographic signatures, and other auditable events are consistent across systems. Install and start NTP with the following commands:

```
$ sudo yum install ntp
$ sudo service ntpd start
## For RHEL/CentOS 7, use 'sudo systemctl start ntpd' instead ##
```

7. **Install Key Trustee Server**

   Run the following command to install the Key Trustee Server:

```
$ sudo yum install keytrustee-server
```

   Installing the Key Trustee Server also installs required dependencies, including PostgreSQL 9.3. After the installation completes, confirm that the PostgreSQL version is 9.3 by running the command `createuser -V`.

8. **Configure Services to Start at Boot**

   Ensure that `ntpd`, `keytrustee-db`, and `keytrusteed` start automatically at boot:

```
$ sudo chkconfig ntpd on
$ sudo chkconfig keytrustee-db on
$ sudo chkconfig keytrusteed on
```

   The `chkconfig` command provides no output if successful.

   Note: The `/etc/init.d/postgresql` script does not work when the PostgreSQL database is started by Key Trustee Server, and cannot be used to monitor the status of the database. Use `/etc/init.d/keytrustee-db` instead.

   After installing Key Trustee Server, continue to Securing Key Trustee Server Host on page 206.

Securing Key Trustee Server Host

Cloudera strongly recommends securing the Key Trustee Server host to protect against unauthorized access to Key Trustee Server. Red Hat provides security guides for RHEL:

- RHEL 6 Security Guide
- RHEL 7 Security Guide

Cloudera also recommends configuring the Key Trustee Server host to allow network communication only over certain ports.

You can use the following examples to create `iptables` rules for an EDH cluster. Add any other ports required by your environment, subject to your organization security policies. Note that in this example port 5432 is the database port for the Key Trustee database on legacy machines (prior to release 5.5). Port 11371 is the current port on which Key Trustee communicates, and port 11381 is the database port. Exercise caution if blocking other ports, as this can cause a disruption in service. See Ports Used by Cloudera Manager and Cloudera Navigator on page 17 for details about ports used with the Key Trustee Server.

```
# Flush iptables
iptables -F
iptables -X

# Allow unlimited traffic on loopback (localhost) connection
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Allow established, related connections
```

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Open all Cloudera Manager ports to allow Key Trustee Server to work properly

iptables -A INPUT -p tcp -m tcp --dport 5432 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 11371 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 11381 -j ACCEPT

# Drop all other connections
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

# Save iptables rules so that they're loaded if the system is restarted
sed 's/IPTABLES_SAVE_ON_STOP="no"/IPTABLES_SAVE_ON_STOP="yes"/' -i
/etc/sysconfig/iptables-config
sed 's/IPTABLES_SAVE_ON_RESTART="no"/IPTABLES_SAVE_ON_RESTART="yes"/' -i
/etc/sysconfig/iptables-config
```

Leveraging Native Processor Instruction Sets

AES-NI

The Advanced Encryption Standard New Instructions (AES-NI) instruction set is designed to improve the speed of encryption and decryption using AES. Some newer processors come with AES-NI, which can be enabled on a per-server basis. If you are uncertain whether AES-NI is available on a device, run the following command to verify:

```
$ grep -o aes /proc/cpuinfo
```

To determine whether the AES-NI kernel module is loaded, run the following command:

```
$ sudo lsmod | grep aesni
```

If the CPU supports AES-NI but the kernel module is not loaded, see your operating system documentation for instructions on installing the `aesni-intel` module.

Intel RDRAND

The Intel RDRAND instruction set, along with its underlying Digital Random Number Generator (DRNG), is useful for generating keys for cryptographic protocols without using `haveged`.

To determine whether the CPU supports RDRAND, run the following command:

```
$ grep -o rdrand /proc/cpuinfo
```

To enable RDRAND, install `rng-tools` version 4 or higher:

1. Download the source code:

```
$ sudo wget
http://downloads.sourceforge.net/project/gkernel/rng-tools/4/rng-tools-4.tar.gz
```

2. Extract the source code:

```
tar xvfz rng-tools-4.tar.gz
```

3. Enter the `rng-tools-4` directory:

```
$ cd rng-tools-4
```

4. Run `./configure`.

5. Run `make`.
6. Run `make install`.

Start `rngd` with the following command:

```
$ sudo rngd --no-tpm=1 -o /dev/random
```

Initializing Key Trustee Server

After installing Key Trustee Server, you must initialize it before it is operational. Continue to Initializing Standalone Key Trustee Server or Cloudera Navigator Key Trustee Server High Availability for instructions.

Installing Cloudera Navigator Key HSM

Important: Before installing Cloudera Navigator Key HSM, see Encrypting Data at Rest for important considerations.

Cloudera Navigator Key HSM is a universal hardware security module (HSM) driver that translates between the target HSM platform and Cloudera Navigator Key Trustee Server.

With Navigator Key HSM, you can use a Key Trustee Server to securely store and retrieve encryption keys and other secure objects, without being limited solely to a hardware-based platform.

Prerequisites

You must install Key HSM on the same host as Key Trustee Server. See Data at Rest Encryption Requirements for more information about encryption and Key HSM requirements.

Setting Up an Internal Repository

You must create an internal repository to install or upgrade Cloudera Navigator Key HSM. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see Creating and Using a Package Repository for Cloudera Manager on page 171.

Installing Navigator Key HSM

Important: If you have implemented Key Trustee Server high availability, install and configure Key HSM on each Key Trustee Server host.

1. **Set up the Key HSM Repository**

   Download the Key HSM tarball and create a local Key HSM repository with the files from the tarball. See Setting Up an Internal Repository on page 208 above for more information.

2. **Install the Key HSM repository**

   Add the local Key HSM repository you created in Step 1. See Modifying Clients to Find the Repository on page 173 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
```

3. **Install the CDH Repository**

Key Trustee Server and Key HSM depend on the `bigtop-utils` package, which is included in the CDH repository. For instructions on adding the CDH repository, see To add the CDH repository. To create a local CDH repository, see Creating a Local Yum Repository on page 222 for instructions.

4. **Install Navigator Key HSM**

   Install the Navigator Key HSM package using `yum`:

```
$ sudo yum install keytrustee-keyhsm
```

   Cloudera Navigator Key HSM is installed to the `/usr/share/keytrustee-server-keyhsm` directory by default.

Installing Key Trustee KMS

> **Important:**
>
> Following these instructions installs the required software to add the Key Trustee KMS service to your cluster; this enables you to use Cloudera Navigator Key Trustee Server as the underlying keystore for HDFS Transparent Encryption. This *does not* install Key Trustee Server. See Installing Cloudera Navigator Key Trustee Server on page 203 for instructions on installing Key Trustee Server. You must install Key Trustee Server before installing and using Key Trustee KMS.
>
> Also, when the Key Trustee KMS role is created, it is tightly bound to the identity of the host on which it is installed. Moving the role to a different host, changing the host name, or changing the IP of the host is *not* supported.

Key Trustee KMS is a custom Key Management Server (KMS) that uses Cloudera Navigator Key Trustee Server as the underlying keystore, instead of the file-based Java KeyStore (JKS) used by the default Hadoop KMS.

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.

> **Important:** 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. For additional details about installing JCE, refer to Step 3: If you are Using AES-256 Encryption, Install the JCE Policy File.

The **KMS (Navigator Key Trustee)** service in Cloudera Manager 5.3 is renamed to **Key Trustee KMS** in Cloudera Manager 5.4.

Setting Up an Internal Repository

You must create an internal repository to install Key Trustee KMS. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see Creating and Using a Parcel Repository for Cloudera Manager on page 168 if you are using parcels, or Creating and Using a Package Repository for Cloudera Manager on page 171 if you are using packages.

Installing Key Trustee KMS Using Parcels

1. Go to **Hosts** > **Parcels**.
2. Click **Configuration** and add your internal repository to the **Remote Parcel Repository URLs** section. See Configuring the Cloudera Manager Server to Use the Parcel URL for Hosted Repositories on page 170 for more information.
3. Download, distribute, and activate the Key Trustee KMS parcel. See Managing Parcels on page 34 for detailed instructions on using parcels to install or upgrade components.

> **Note:** The KEYTRUSTEE_SERVER parcel in Cloudera Manager is *not* the Key Trustee KMS parcel; it is the Key Trustee Server parcel. The parcel name for Key Trustee KMS is KEYTRUSTEE.

Installing Key Trustee KMS Using Packages

1. After Setting Up an Internal Repository on page 209, configure the Key Trustee KMS host to use the repository. See Modifying Clients to Find the Repository on page 173 for more information.
2. Because the `keytrustee-keyprovider` package depends on the `hadoop-kms` package, you must add the CDH repository. See To add the CDH repository for instructions. If you want to create an internal CDH repository, see Creating a Local Yum Repository on page 222.
3. Install the `keytrustee-keyprovider` package using the appropriate command for your operating system:
   - **RHEL-compatible**

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

   - **SLES**

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

   - **Ubuntu or Debian**

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

Post-Installation Configuration

For instructions on installing Key Trustee Server and configuring Key Trustee KMS to use Key Trustee Server, see the following topics:

- Installing Cloudera Navigator Key Trustee Server on page 203
- Enabling HDFS Encryption Using the Wizard

Installing Navigator HSM KMS Backed by Thales HSM

> **Important:** Following these instructions installs the required software to add the Navigator HSM KMS backed by Thales HSM to your cluster; this enables you to use a supported Thales HSM as the underlying keystore for HDFS Transparent Encryption.

HSM KMS backed by Thales HSM is a custom Key Management Server (KMS) that uses a supported Thales HSM as the underlying keystore, instead of the file-based Java KeyStore (JKS) used by the default Hadoop KMS.

> **Important:** HSM KMS backed by Thales HSM is supported only in Cloudera Manager deployments. You can install the software using parcels or packages, but running HSM KMS backed by Thales HSM outside of Cloudera Manager is not supported.

Client Prerequisites

Navigator HSM KMS backed by Thales HSM is supported on Thales HSMs only. The Thales HSM client must be installed first.

The following Thales nShield Connect software and firmware are required:

- Server version: 3.67.11cam4

- Firmware: 2.65.2
- Security World version: 12.30

Before performing the Thales HSM setup, run the `nfkminfo` command to verify that Thales HSM is configured correctly.

```
$ sudo /opt/nfast/bin/nfkminfo
          World generation 2
          state      0x1727 Initialised Usable Recovery !PINRecovery !ExistingClient
                     RTC NVRAM FTO !AlwaysUseStrongPrimes SEEDebug
```

If state reports `!Usable` instead of `Usable`, then configure the Thales HSM before continuing. See the Thales product documentation for details about how to configure the Thales client.

Run the following command to manually add the KMS user to the `nfast` group:

```
usermod -a -G nfast kms
```

If you do not manually add the KMS user, installation can fail.

Setting Up an Internal Repository

You must create an internal repository to install Navigator HSM KMS backed by Thales HSM. 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 on page 168 if you are using parcels, or Creating and Using a Package Repository for Cloudera Manager on page 171 if you are using packages.

Installing Navigator HSM KMS Backed by Thales HSM Using Parcels

1. Go to **Hosts** > **Parcels**.
2. Click **Configuration** and add your internal repository to the **Remote Parcel Repository URLs** section. See Configuring the Cloudera Manager Server to Use the Parcel URL for Hosted Repositories on page 170 for more information.
3. Download, distribute, and activate the Navigator HSM KMS parcel. See Managing Parcels on page 34 for detailed instructions on using parcels to install or upgrade components.

> **Note:** The KEYTRUSTEE_SERVER parcel in Cloudera Manager is *not* the Key Trustee KMS parcel; it is the Key Trustee Server parcel. The parcel name for Navigator HSM KMS backed by Thales HMS is KEYTRUSTEE.

Installing Navigator HSM KMS Backed by Thales HSM Using Packages

1. After Setting Up an Internal Repository on page 211, configure the Navigator KMS Services backed by Thales HSM host to use the repository. See Modifying Clients to Find the Repository on page 173 for more information.
2. Because the `keytrustee-keyprovider` package depends on the `hadoop-kms` package, you must add the CDH repository. See To add the CDH repository for instructions. If you want to create an internal CDH repository, see Creating a Local Yum Repository on page 222.
3. Install the `keytrustee-keyprovider` package using the appropriate command for your operating system:

> **Important:** When installing via packages, be sure to install on each and every host on which you wish to run the HSM KMS service.

- **RHEL-compatible**

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

Post-Installation Configuration

For instructions on configuring HSM KMS, see Enabling HDFS Encryption Using the Wizard.

Installing Navigator HSM KMS Backed by Luna HSM

Important: Following these instructions installs the required software to add the Navigator KMS Services backed by Luna HSM to your cluster; this enables you to use a supported Luna HSM as the underlying keystore for HDFS Transparent Encryption.

Navigator HSM KMS backed by Luna HSM is a custom Key Management Server (KMS) that uses a supported Luna HSM as the underlying keystore, instead of the file-based Java KeyStore (JKS) used by the default Hadoop KMS.

Important: Navigator HSM KMS backed by Luna HSM is supported only in Cloudera Manager deployments. You can install the software using parcels or packages, but running Navigator HSM KMS backed by Luna HSM outside of Cloudera Manager is not supported.

Client Prerequisites

Navigator HSM KMS backed by Luna HSM is supported on Luna HSMs only. The Luna HSM client must be installed first.

For details about the required Luna software and firmware, refer to CDH 5 and Cloudera Manager 5 Requirements and Supported Versions, and scroll to the section "Navigator HSM KMS: Recommended Hardware and Supported Distributions".

Before performing the Luna HSM KMS setup, run the `vt1 verify` command (located at `/usr/safenet/lunaclient/bin/vtl`) to verify that the Luna HSM is configured correctly. See the Luna product documentation for details about how to configure the Luna HSM client.

Setting Up an Internal Repository

You must create an internal repository to install Navigator HSM KMS backed by Luna HSM. 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 on page 168 if you are using parcels, or Creating and Using a Package Repository for Cloudera Manager on page 171if you are using packages.

Installing Navigator HSM KMS Backed by Luna HSM Using Parcels

1. Go to **Hosts** > **Parcels**.
2. Click **Configuration** and add your internal repository to the **Remote Parcel Repository URLs** section. See Configuring the Cloudera Manager Server to Use the Parcel URL for Hosted Repositories on page 170 for more information.
3. Download, distribute, and activate the Navigator HSM KMS parcel. See Managing Parcels on page 34 for detailed instructions on using parcels to install or upgrade components.

Note: The KEYTRUSTEE_SERVER parcel in Cloudera Manager is *not* the Key Trustee KMS parcel; it is the Key Trustee Server parcel. The parcel name for Navigator HSM KMS backed by Luna HSM is KEYTRUSTEE.

Installing Navigator HSM KMS Backed by Luna HSM Using Packages

1. After Setting Up an Internal Repository on page 212, configure the Navigator HSM KMS backed by Luna HSM host to use the repository. See Modifying Clients to Find the Repository on page 173 for more information.
2. Because the `keytrustee-keyprovider` package depends on the `hadoop-kms` package, you must add the CDH repository. See To add the CDH repository for instructions. If you want to create an internal CDH repository, see Creating a Local Yum Repository on page 222.

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

> **Important:** When installing via packages, be sure to install on each and every host on which you wish to run the HSM KMS service.

- **RHEL-compatible**

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

Post-Installation Configuration

For instructions on configuring HSM KMS, see Enabling HDFS Encryption Using the Wizard.

Installing Cloudera Navigator Encrypt

> **Important:** Before installing Cloudera Navigator Encrypt, see Encrypting Data at Rest and the Table 14 for important considerations.

Prerequisites

See Data at Rest Encryption Requirements for more information about encryption and Navigator Encrypt requirements.

Setting Up an Internal Repository

You must create an internal repository to install or upgrade Navigator Encrypt. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see Creating and Using a Package Repository for Cloudera Manager on page 171.

Installing Navigator Encrypt (RHEL-Compatible)

> **Note:** For details about supported Linux Operating Systems, refer to the Table 14.

1. **Install the Cloudera Repository**

   Add the internal repository you created. See Modifying Clients to Find the Repository on page 173 for more information.

   Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

2. **Install 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 Kernel Libraries**

   For Navigator Encrypt to run as a kernel module, you must download and install the kernel development headers. Each kernel module is compiled specifically for the underlying kernel version. Running as a kernel module allows Navigator Encrypt to provide high performance and completely transparency to user-space applications.

   To determine your current kernel version, run `uname -r`.

   To install the development headers for your current kernel version, run:

```
$ sudo yum install kernel-headers-$(uname -r) kernel-devel-$(uname -r)
```

   For OL with the Unbreakable Enterprise Kernel (UEK), run:

```
$ sudo yum install kernel-uek-headers-$(uname -r) kernel-uek-devel-$(uname -r)
```

   Note: For UEK3, you do not need to install `kernel-uek-headers-*`

   If yum cannot find these packages, it displays an error similar to the following:

```
Unable to locate package <packagename>.
```

   In this case, do one of the following to proceed:

   - Find and install the kernel headers package by using a tool like RPM Pbone .
   - Upgrade your kernel to the latest version. If you upgrade the kernel, you must reboot after upgrading and select the kernel from the grub menu to make it active.

4. **(RHEL or CentOS Only) Manually Install dkms**

   Because of a broken dependency in all versions of RHEL or CentOS, you must manually install the dkms package:

```
$ sudo yum install
http://repository.it4i.cz/mirrors/repoforge/redhat/el6/en/x86_64/rpmforge/RPMS/dkms-2.1.1.2-1.el6.rf.noarch.rpm
```

   Note: This link is provided as an example for RHEL 6 only. For other versions, be sure to use the correct URL.

5. **Install Navigator Encrypt**

   Install the Navigator Encrypt client using the yum package manager:

```
$ sudo yum install navencrypt
```

If you attempt to install Navigator Encrypt with incorrect or missing kernel headers, you see a message like the following:

```
Building navencryptfs 3.8.0 DKMS kernel module...

###################### BUILDING ERROR ######################


Creating symlink /var/lib/dkms/navencryptfs/3.8.0/source ->
                 /usr/src/navencryptfs-3.8.0

DKMS: add completed.
Error! echo
Your kernel headers for kernel 3.10.0-229.4.2.el7.x86_64 cannot be found at
/lib/modules/3.10.0-229.4.2.el7.x86_64/build or
/lib/modules/3.10.0-229.4.2.el7.x86_64/source.

###################### BUILDING ERROR ######################

Failed installation of navencryptfs 3.8.0 DKMS kernel module !
```

To recover, see Navigator Encrypt Kernel Module Setup.

Installing Navigator Encrypt (SLES)

1. **Install the Cloudera Repository**

   Add the internal repository you created. See Modifying Clients to Find the Repository on page 173 for more information.

   Import the GPG key by running the following command:

```
$ sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

2. **Install NTP**

   The Network Time Protocol (NTP) service synchronizes system time. Cloudera recommends using NTP to ensure that timestamps in system logs, cryptographic signatures, and other auditable events are consistent across systems. Install and start NTP with the following commands:

   - **SLES 11**

```
$ sudo zypper install ntp
# /etc/init.d/ntp start
```

   - **SLES 12**

```
$ sudo zypper install ntp
# service ntpd start
```

3. **Install the Kernel Module Package and Navigator Encrypt Client**

   Install the kernel module package (KMP) and Navigator Encrypt client with `zypper`:

```
$ sudo zypper install cloudera-navencryptfs-kmp-<kernel_flavor>
$ sudo zypper install navencrypt
```

   Replace *<kernel_flavor>* with the kernel flavor for your system. Navigator Encrypt supports the `default`, `xen`, and `ec2` kernel flavors.

4. **Enable Unsupported Modules**

Edit `/etc/modprobe.d/unsupported-modules` and set `allow_unsupported_modules` to `1`. For example:

```
#
# Every kernel module has a flag 'supported'. If this flag is not set loading
# this module will taint your kernel. You will not get much help with a kernel
# problem if your kernel is marked as tainted. In this case you firstly have
# to avoid loading of unsupported modules.
#
# Setting allow_unsupported_modules 1 enables loading of unsupported modules
# by modprobe, setting allow_unsupported_modules 0 disables it. This can
# be overridden using the --allow-unsupported-modules command line switch.
allow_unsupported_modules 1
```

5. **(SLES 12 SP2 only) Run `systemctl daemon-reload`**

   Due to changes in SLES 12 SP2, you must run the following command after installing Navigator Encrypt:

```
$ sudo systemctl daemon-reload
```

Installing Navigator Encrypt (Debian or Ubuntu)

1. **Install the Cloudera Repository**

   Add the internal repository you created. See Modifying Clients to Find the Repository on page 173 for more information.

   - **Ubuntu**

```
$ echo "deb http://repo.example.com/path/to/ubuntu/stable $DISTRIB_CODENAME main" | sudo
 tee -a /etc/apt/sources.list
```

   - **Debian**

```
$ echo "deb http://repo.example.com/path/to/debian/stable $DISTRIB_CODENAME main" | sudo
 tee -a /etc/apt/sources.list
```

   Import the GPG key by running the following command:

```
$ wget -O - http://repo.example.com/path/to/gpg_gazzang.asc | apt-key add -
```

   Update the repository index with `apt-get update`.

2. **Install NTP**

   The Network Time Protocol (NTP) service synchronizes system time. Cloudera recommends using NTP to ensure that timestamps in system logs, cryptographic signatures, and other auditable events are consistent across systems. Install and start NTP with the following commands:

```
$ sudo apt-get install ntp
$ sudo /etc/init.d/ntp start
```

3. **Install Kernel Headers**

   Determine your kernel version by running `uname -r`, and install the appropriate headers:

```
$ sudo apt-get install linux-headers-$(uname -r)
```

4. **Install the Navigator Encrypt Client**

   Install Navigator Encrypt:

```
$ sudo apt-get install navencrypt
```

Post Installation

To ensure that Navigator Encrypt and NTP start after a reboot, add them to the start order with `chkconfig`:

```
$ sudo chkconfig --level 235 navencrypt-mount on
$ sudo chkconfig --level 235 ntpd on
```

AES-NI and RDRAND

The Advanced Encryption Standard New Instructions (AES-NI) instruction set is designed to improve the speed of encryption and decryption using AES. Some newer processors come with AES-NI, which can be enabled on a per-server basis.

Both the eCryptfs and dm-crypt back ends for Navigator Encrypt can automatically detect and use AES-NI if it is available. If you are uncertain whether AES-NI is available on a device, run the following command to verify:

```
$ grep -o aes /proc/cpuinfo
```

To determine whether the AES-NI kernel module is loaded, run the following command:

```
$ sudo lsmod | grep aesni
```

If the CPU supports AES-NI but the kernel module is not loaded, see your operating system documentation for instructions on installing the `aesni-intel` module.

Navigator Encrypt needs a source of random numbers if it is using dm-crypt as its back end. Use `rng-tools` version 4 or higher to seed the system's entropy pool, using the RDRAND instruction. To install and start `rngd`:

1. Download the source code:

```
$ sudo wget
http://downloads.sourceforge.net/project/gkernel/rng-tools/4/rng-tools-4.tar.gz
```

2. Extract the source code:

```
tar xvfz rng-tools-4.tar.gz
```

3. Enter the `rng-tools-4` directory:

```
$ cd rng-tools-4
```

4. Run `./configure`
5. Run `make`
6. Run `make install`

Once you have installed `rng-tools`, start the `rngd` daemon by running the following command as root:

```
$ sudo rngd --no-tpm=1 -o /dev/random
```

Setting Up TLS for Navigator Encrypt Clients

Transport Layer Security (TLS) certificates are used to secure communication with Navigator Encrypt. Cloudera strongly recommends using certificates signed by a trusted Certificate Authority (CA).

If the TLS certificate is signed by an unrecognized CA, such as an internal CA, then you must add the root certificate to the host certificate truststore of each Navigator Encrypt client. Be aware that Navigator Encrypt uses the operating system's truststore, which is distinct from the JDK truststore used by Cloudera Manager.

To set up TLS certificates on a Navigator Encrypt client:

1. If not already installed, install the CA-certificates:

```
yum install ca-certificates
```

2. Enable the dynamic CA configuration feature:

```
update-ca-trust enable
```

3. Copy the root certificate into the host certificate truststore:

```
cp /path/to/root.pem /etc/pki/ca-trust/source/anchors/
```

4. Update the host certificate truststore:

```
update-ca-trust
```

Example

```
[root@navencrypt-1 ~]# service navencrypt-mount stop
Stopping navencrypt directories
 * Umounting /dev/nvtest/test1 ...                              [  OK  ]
 * Umounting /dev/nvtest/test2 ...                              [  OK  ]
 * Unloading module ...                                         [  OK  ]

[root@navencrypt-1 ~]# update-ca-trust enable
[root@navencrypt-1 ~]# cp dd-1.lab.usa.company.com.pem /etc/pki/ca-trust/source/anchors/
[root@navencrypt-1 ~]# update-ca-trust

[root@navencrypt-1 ~]# service navencrypt-mount start
Starting navencrypt directories
 * Mounting '/dev/nvtest/test1'                                 [  OK  ]
 * Mounting '/dev/nvtest/test2'                                 [  OK  ]
```

Entropy Requirements

Many cryptographic operations, such as those used with TLS or HDFS encryption, require a sufficient level of system entropy to ensure randomness; likewise, Navigator Encrypt needs a source of random numbers to ensure good performance. Hence, you need to make sure that the hosts running Navigator Encrypt (as well as Key Trustee Server, Key Trustee KMS) and have sufficient entropy to perform cryptographic operations.

You can check the available entropy on a Linux system by running the following command:

```
$ cat /proc/sys/kernel/random/entropy_avail
```

The output displays the entropy currently available. Check the entropy several times to determine the state of the entropy pool on the system. If the entropy is consistently low (500 or less), you must increase it by installing `rng-tools` version 4 or higher, and starting the `rngd` service.

Install rng_tools Using Package Manager

If version 4 or higher of the `rng-tools` package is available from the local package manager (`yum`), then install it directly from the package manager. If the appropriate version of `rng-tools` is unavailable, see Building rng-tools From Source on page 219.

> **Note:** If you're using RHEL 6.7 and later, or recent versions of Ubuntu, Debian, and SLES, then package manager should provide version 4.x or higher. Be sure to check the version of `rng-tools` provided by your package manager before installation to determine whether or not you need to build from source instead.

Run the following commands on RHEL 6-compatible systems:

```
$ sudo yum install rng-tools
$ sudo service rngd start
$ sudo chkconfig rngd on
```

For RHEL 7, run the following commands:

```
$ sudo yum install rng-tools
$ cp /usr/lib/systemd/system/rngd.service /etc/systemd/system/
$ systemctl daemon-reload
$ systemctl start rngd
$ systemctl enable rngd
```

Building rng-tools From Source

If you are unable to install `rng-tools` using package manager, then build from source.

> **Note:** If your package manager only offers an older version (3.x or earlier), then you must build from source.

To install and start `rngd` and build from source:

1. Download the source code:

```
$ sudo wget
http://downloads.sourceforge.net/project/gkernel/rng-tools/4/rng-tools-4.tar.gz
```

2. Extract the source code:

```
tar xvfz rng-tools-4.tar.gz
```

3. Enter the `rng-tools-4` directory:

```
$ cd rng-tools-4
```

4. Run `./configure`
5. Run `make`
6. Run `make install`

After you have installed `rng-tools`, start the `rngd` daemon by running the following command as root:

```
$ sudo rngd --no-tpm=1 -o /dev/random
```

For improved performance, Cloudera recommends configuring Navigator Encrypt to read directly from `/dev/random` instead of `/dev/urandom`.

To configure Navigator Encrypt to use `/dev/random` as an entropy source, add `--use-random` to the `navencrypt-prepare` command when you are setting up Navigator Encrypt.

Uninstalling and Reinstalling Navigator Encrypt

Uninstalling Navigator Encrypt

For RHEL-compatible OSes:

```
$ sudo yum remove navencrypt
$ sudo yum remove navencrypt-kernel-module
```

These commands remove the software itself. On RHEL-compatible OSes, the `/etc/navencrypt` directory is not removed as part of the uninstallation. Remove it manually if required.

Reinstalling Navigator Encrypt

After uninstalling Navigator Encrypt, repeat the installation instructions for your distribution in Installing Cloudera Navigator Encrypt on page 213.

When Navigator Encrypt is uninstalled, the configuration files and directories located in `/etc/navencrypt` are not removed. Consequently, you do not need to use the `navencrypt register` command during reinstallation. If you no longer require the previous installation configuration information in the directory `/etc/navencrypt`, you can remove its contents.

Installing and Deploying CDH Using the Command Line

Before You Install CDH 5 on a Cluster

Important:

- Use the `service` command to start, stop, and restart CDH components, 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).
- On SLES 11 platforms, do not install or try to use the IBM Java version bundled with the SLES distribution; Hadoop will not run correctly with that version. Install the Oracle JDK following directions under Java Development Kit Installation.
- If you are migrating from MapReduce v1 (MRv1) to MapReduce v2 (MRv2, YARN), see Migrating from MapReduce (MRv1) to MapReduce (MRv2) on page 236 for important information and instructions.

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

1. Verify you are using a supported operating system for CDH 5. See CDH and Cloudera Manager Supported Operating Systems.
2. If you haven't already done so, install the Oracle Java Development Kit. For instructions and recommendations, see Java Development Kit Installation.

Scheduler Defaults

Note the following differences between MRv1 (MapReduce) and MRv2 (YARN).

- MRv1 (MapReduce v1):
  - Cloudera Manager and CDH 5 set the default to FIFO.

  FIFO is set as the default for backward-compatibility purposes, but Cloudera recommends Fair Scheduler. Capacity Scheduler is also available.
- MRv2 (YARN):
  - Cloudera Manager and CDH 5 set the default to Fair Scheduler.

  Cloudera recommends Fair Scheduler because Impala and Llama are optimized for it. FIFO and Capacity Scheduler are also available.

High Availability

In CDH 5, you can configure high availability both for the NameNode and the JobTracker or ResourceManager.

- For more information and instructions on setting up a new HA configuration, see High Availability.

Important:

If you configure HA for the NameNode, do not install `hadoop-hdfs-secondarynamenode`. After completing the HDFS HA software configuration, follow the installation instructions in Deploying HDFS High Availability.

Creating a Local Yum Repository

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

This section explains how to set up a local yum repository to install CDH on the machines in your cluster. There are a number of reasons you might want to do this, for example:

- The machines in your cluster do not have Internet access. You can still use yum to do an installation on those machines by creating a local yum repository.
- You may want to keep a stable local repository to ensure that any new installations (or re-installations on existing cluster members) use exactly the same bits.
- Using a local repository may be the most efficient way to distribute the software to the cluster members.

To set up your own internal mirror, follow the steps below. You need an Internet connection for the steps that require you to download packages and create the repository itself. You also need an Internet connection to download updated RPMs to your local repository.

1. Download the repo file. Click the link for your RHEL or CentOS system in the table, find the appropriate repo file, and save in `/etc/yum.repos.d/`.

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

2. Install a web server such as `apache/lighttpd` on the machine that hosts the RPMs. The default configuration should work. HTTP access must be allowed to pass through any firewalls between this server and the Internet connection.
3. On the server with the web server, install the RPM packages, `yum-utils` and `createrepo`, if not already installed. The `yum-utils` package includes the `reposync` command, which is required to create the local Yum repository.

```
sudo yum install yum-utils createrepo
```

4. On the same computer as in the previous steps, download the yum repository into a temporary location. On RHEL/CentOS 6, you can use a command such as:

```
reposync -r cloudera-cdh5
```

   You can replace with any alpha-numeric string. It will be the name of your local repository, used in the header of the repo file other systems use to connect to your repository. You can now disconnect your server from the Internet.

5. Put all the RPMs into a directory served by your web server, such as `/var/www/html/cdh/5/RPMS/noarch/` (or `x86_64` or `i386` instead of `noarch`). The directory structure `5/RPMS/noarch` is required. Make sure you can remotely access the files in the directory using HTTP, using a URL similar to `http://<yourwebserver>/cdh/5/RPMS/`).
6. On your web server, issue the following command from the `5/` subdirectory of your RPM directory:

```
createrepo .
```

This creates or update the metadata required by the `yum` command to recognize the directory as a repository. The command creates a new directory called `repodata`. If necessary, adjust the permissions of files and directories in your entire repository directory to be readable by the web server user.

7. Edit the repo file you downloaded in step 1 and replace the line starting with `baseurl=` or `mirrorlist=` with `baseurl=http://<yourwebserver>/cdh/5/`, using the URL from step 5. Save the file back to `/etc/yum.repos.d/`.
8. While disconnected from the Internet, issue the following commands to install CDH from your local `yum` repository.

Example:

```
yum update
yum install hadoop
```

Once you have confirmed that your internal mirror works, you can distribute this modified repo file to any system which can connect to your repository server. Those systems can now install CDH from your local repository without Internet access. Follow the instructions under Installing the Latest CDH 5 Release on page 223, starting at Step 2 (you have already done Step 1).

Installing the Latest CDH 5 Release

CDH 5 Installation Options

There are multiple ways to install CDH 5:

- **Managed Deployment**: automatically install CDH 5 with a Cloudera Manager Deployment on page 54. This is the simplest and preferred method.
- **Unmanaged Deployment**:
  - Manually install the CDH 5 package or repository: either add the CDH 5 repository *OR* build your own CDH 5 repository.
  - Manually install the CDH 5 tarball. See "Package and Tarball Binaries" below.

Note: Cloudera recommends installing CDH 5 and dependencies with Cloudera Manager.

Package and Tarball Binaries

Installing from Packages

- To install and deploy YARN, see Deploying MapReduce v2 (YARN) on a Cluster.
- To install and deploy MRv1, see Deploying MapReduce v1 (MRv1) on a Cluster.

Installing from a Tarball

- The CDH 5 tarball deploys YARN and includes the MRv1 binaries. There is no separate tarball for MRv1. The MRv1 scripts are in the directory, `bin-mapreduce1`, and examples are in `examples-mapreduce1`.

Before You Begin Installing CDH 5 Manually

- To migrate from MRv1 to YARN, see Migrating from MapReduce (MRv1) to MapReduce (MRv2) on page 236.
- For a list of supported operating systems, see CDH and Cloudera Manager Supported Operating Systems.
- Installing CDH 5 requires `sudo` privileges. If necessary, use root user (superuser) to configure `sudo` privileges.
- CDH5 requires the Oracle Java Development Kit (JDK). See Java Development Kit Installation.
- In CDH 5, both the NameNode and Resource Manager (or Job Tracker) can be configured for High Availability
- Use the `service (8)` command to start and stop services rather than running scripts in `/etc/init.d` directly.

Note: Running Services

Use the `service` command to start, stop, and restart CDH components, 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).

Steps to Install CDH 5 Manually

Step 1: Add or Build the CDH 5 Repository

- To install CDH 5 on a RHEL system, download packages with `yum` or use a web browser.
- To install CDH 5 on a SLES system, download packages with `zypper` or YaST or use a web browser.
- To install CDH 5 on an Ubuntu or Debian system, download packages with `apt` or use a web browser.

On RHEL-compatible Systems

Use one of the following methods to install CDH 5 on RHEL-compatible systems.

- Add the CDH 5 repository *OR*
- Build a Yum Repository

Do this on all the systems in the cluster.

To add the CDH 5 repository:

Download the repo file. Click the link for your RHEL or CentOS system in the table, find the appropriate repo file, and save in `/etc/yum.repos.d/`.

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

Continue with Step 2: Optionally Add a Repository Key on page 227. Then choose Step 3: Install CDH 5 with YARN on page 228, or Step 4: Install CDH 5 with MRv1 on page 229; or do both steps to install both implementations.

Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

***OR:* To build a Yum repository:**

Follow the instructions at Creating a Local Yum Repository to create your own `yum` repository:

- Download the appropriate repo file
- Create the repo
- Distribute the repo and set up a web server.

Continue with Step 2: Optionally Add a Repository Key on page 227. Then choose Step 3: Install CDH 5 with YARN on page 228, or Step 4: Install CDH 5 with MRv1 on page 229; or do both steps to install both implementations.

Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo yum clean all
```

On SLES Systems

Use one of the following methods to download the CDH 5 repository or package on SLES systems.

- Add the CDH 5 repository *OR*
- Build a SLES Repository

To add the CDH 5 repository:

1. Run the following command:

```
$ sudo zypper addrepo -f
http://archive.cloudera.com/cdh5/sles/12/x86_64/cdh/cloudera-cdh.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

Continue with Step 2: Optionally Add a Repository Key on page 227. Then choose Step 3: Install CDH 5 with YARN on page 228, or Step 4: Install CDH 5 with MRv1 on page 229; or do both steps to install both implementations.

Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

***OR:* To build a SLES repository:**

If you want to create your own SLES repository, create a mirror of the CDH SLES directory by following these instructions that explain how to create a SLES repository from the mirror.

Continue with Step 2: Optionally Add a Repository Key on page 227. Then choose Step 3: Install CDH 5 with YARN on page 228, or Step 4: Install CDH 5 with MRv1 on page 229; or do both steps to install both implementations.

Note: Clean repository cache.

Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:

```
sudo zypper clean --all
```

On Ubuntu or Debian Systems

Use one of the following methods to download the CDH 5 repository or package.

- Add the CDH 5 repository *OR*
- Build a Debian Repository

To add the CDH 5 repository:

- Download the appropriate `cloudera.list` file by issuing one of the following commands. You can use another HTTP client if `wget` is not available, but the syntax may be different.

> **Important: Ubuntu 14.04 (Trusty)**
>
> For Ubuntu Trusty systems, you must perform an extra step after adding the repository. See "Additional Step for Trusty Ubuntu Trusty and Debian Jessie" below.

OS Version	Command
Debian 8 **Jessie**	`$ sudo wget 'https://archive.cloudera.com/cdh5/debian/jessie/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list`
Debian 7 **Wheezy**	`$ sudo wget 'https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list`
Ubuntu 16 **Xenial**	`$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/xenial/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list`
Ubuntu 14 **Trusty**	`$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list`
Ubuntu 12 **Precise**	`$ sudo wget 'https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/cloudera.list' \ -O /etc/apt/sources.list.d/cloudera.list`

> **Note: Clean repository cache.**
>
> Before proceeding, clean cached packages and headers to ensure your system repos are up-to-date:
>
> ```
> sudo apt-get update
> ```

Additional step for Ubuntu Trusty and Debian Jessie

This step ensures that you get the right ZooKeeper package for the current CDH release. You need to prioritize the Cloudera repository you have just added, such that you install the CDH version of ZooKeeper rather than the version that is bundled with Ubuntu Trusty or Debian Jessie.

To do this, create a file at `/etc/apt/preferences.d/cloudera.pref` with the following contents:

```
Package: *
Pin: release o=Cloudera, l=Cloudera
Pin-Priority: 501
```

> **Note:** You *do not* need to run `apt-get update` after creating this file.

Continue with Step 2: Optionally Add a Repository Key on page 227. Then choose Step 3: Install CDH 5 with YARN on page 228, or Step 4: Install CDH 5 with MRv1 on page 229; or do both steps to install both implementations.

***OR:* To build a Debian repository:**

If you want to create your own `apt` repository, create a mirror of the CDH Debian directory and then create an apt repository from the mirror.

Continue with Step 2: Optionally Add a Repository Key on page 227. Then choose Step 3: Install CDH 5 with YARN on page 228, or Step 4: Install CDH 5 with MRv1 on page 229; or do both steps to install both implementations.

Step 2: Optionally Add a Repository Key

Before installing YARN or MRv1: (Optionally) add a repository key on each system in the cluster. Add the Cloudera Public GPG Key to your repository by executing one of the following commands:

- **For RHEL/CentOS/Oracle 5 systems:**

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

- **For RHEL/CentOS/Oracle 6 systems:**

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

- **For RHEL/CentOS/Oracle 7 systems:**

```
sudo rpm --import
https://archive.cloudera.com/cdh5/redhat/7/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- **For all SLES systems:**

```
sudo rpm --import
https://archive.cloudera.com/cdh5/sles/12/x86_64/cdh/RPM-GPG-KEY-cloudera
```

- **For Ubuntu or Debian systems:**

OS Version	Command
Debian 8 **Jessie**	`$ wget https://archive.cloudera.com/cdh5/debian/jessie/amd64/cdh/archive.key -O archive.key` `$ sudo apt-key add archive.key`
Debian 7 **Wheezy**	`$ wget https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh/archive.key -O archive.key` `$ sudo apt-key add archive.key`
Ubuntu 16 **Xenial**	`$ wget https://archive.cloudera.com/cdh5/ubuntu/xenial/amd64/cdh/archive.key -O archive.key` `$ sudo apt-key add archive.key`
Ubuntu 14 **Trusty**	`$ wget https://archive.cloudera.com/cdh5/ubuntu/trusty/amd64/cdh/archive.key -O archive.key` `$ sudo apt-key add archive.key`
Ubuntu 12 **Precise**	`$ wget https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh/archive.key -O archive.key` `$ sudo apt-key add archive.key`

This key enables you to verify that you are downloading genuine packages.

Step 3: Install CDH 5 with YARN

> **Note:** Skip this step if you intend to use *only* MRv1. Directions for installing MRv1 are in Step 4.

To install CDH 5 with YARN:

> **Note:** When configuring HA for the NameNode, do not install `hadoop-hdfs-secondarynamenode`. After completing the HA software configuration, follow the installation instructions under Deploying HDFS High Availability.

1. Install and deploy ZooKeeper.

   > **Important:** Cloudera recommends that you install (or update) and start a ZooKeeper cluster before proceeding. This is a **requirement** if you are deploying high availability (HA) for the NameNode.

   Follow instructions under ZooKeeper Installation.

2. Install each type of daemon package on the appropriate systems(s), as follows.

Where to install	Install commands
Resource Manager host (analogous to MRv1 JobTracker) running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-yarn-resourcemanager`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-yarn-resourcemanager`
Ubuntu or Debian	`sudo apt-get update; sudo apt-get install hadoop-yarn-resourcemanager`
NameNode host running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-hdfs-namenode`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode`
Ubuntu or Debian	`sudo apt-get install hadoop-hdfs-namenode`
Secondary NameNode host (if used) running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode`
Ubuntu or Debian	`sudo apt-get install hadoop-hdfs-secondarynamenode`
All cluster hosts except the Resource Manager running:	

Where to install	Install commands
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce`
Ubuntu or Debian	`sudo apt-get install hadoop-yarn-nodemanager hadoop-hdfs-datanode hadoop-mapreduce`
One host in the cluster running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver`
Ubuntu or Debian	`sudo apt-get install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver`
All client hosts running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-client`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-client`
Ubuntu or Debian	`sudo apt-get install hadoop-client`

Note: The `hadoop-yarn` and `hadoop-hdfs` packages are installed on each system automatically as dependencies of the other packages.

Step 4: Install CDH 5 with MRv1

Note: If installing both MRv1 and YARN, do not install packages that you already installed in Step 3: Install CDH 5 with YARN on page 228. If installing YARN *only*, skip this step and go to Step 3: Install CDH 5 with YARN on page 228.

Important: Before proceeding, you need to decide:

- Whether to configure High Availability (HA) for the NameNode or JobTracker; see the High Availability for more information and instructions.
- Where to deploy the NameNode, Secondary NameNode, and JobTracker daemons. As a general rule:
  - The NameNode and JobTracker run on the same "master" host unless the cluster is large (more than a few tens of nodes), and the master host (or hosts) should not run the Secondary NameNode (if used), DataNode or TaskTracker services.
  - In a large cluster, it is especially important that the Secondary NameNode (if used) runs on a separate machine from the NameNode.
  - Each node in the cluster **except the master host(s)** should run the DataNode and TaskTracker services.

If you decide to configure HA for the NameNode, do not install `hadoop-hdfs-secondarynamenode`. After completing the HA software configuration, follow the installation instructions under Deploying HDFS High Availability.

First, install and deploy ZooKeeper.

Important: Cloudera recommends that you install (or update) and start a ZooKeeper cluster before proceeding. This is a **requirement** if you are deploying high availability (HA) for the NameNode or JobTracker.

Follow instructions under ZooKeeper Installation. Make sure you create the `myid` file in the data directory, as instructed, if you are starting a ZooKeeper ensemble after a fresh install.

Next, install packages.

Install each type of daemon package on the appropriate systems(s), as follows.

Note: Ubuntu systems may try to start the service immediately after you install it. This should fail harmlessly, but you can find information at askubuntu on how to prevent this.

Where to install	Install commands
JobTracker host running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-jobtracker`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-jobtracker`
Ubuntu or Debian	`sudo apt-get update; sudo apt-get install hadoop-0.20-mapreduce-jobtracker`
NameNode host running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-hdfs-namenode`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-hdfs-namenode`
Ubuntu or Debian	`sudo apt-get install hadoop-hdfs-namenode`

Where to install	Install commands
Secondary NameNode host (if used) running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-hdfs-secondarynamenode`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-hdfs-secondarynamenode`
Ubuntu or Debian	`sudo apt-get install hadoop-hdfs-secondarynamenode`
All cluster hosts except the JobTracker, NameNode, and Secondary (or Standby) NameNode hosts running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode`
Ubuntu or Debian	`sudo apt-get install hadoop-0.20-mapreduce-tasktracker hadoop-hdfs-datanode`
All client hosts running:	
RHEL/CentOS compatible	`sudo yum clean all; sudo yum install hadoop-client`
SLES	`sudo zypper clean --all; sudo zypper install hadoop-client`
Ubuntu or Debian	`sudo apt-get install hadoop-client`

Step 5: (Optional) Install LZO

This section explains how to install LZO (Lempel–Ziv–Oberhumer) compression. For more information, see Choosing and Configuring Data Compression

Note: If upgrading (rather than installing for the first time), remove the old LZO version first. For example, on a RHEL system:

```
yum remove hadoop-lzo
```

1. Add the repository on each host in the cluster. Follow the instructions for your OS version:

For OS Version	Do this
RHEL/CentOS/Oracle 5	Go to this link and save the file in the `/etc/yum.repos.d/` directory.
RHEL/CentOS/Oracle 6	Go to this link and save the file in the `/etc/yum.repos.d/` directory.
RHEL/CentOS/Oracle 7	Go to this link and save the file in the `/etc/yum.repos.d/` directory.
SLES	1. Run the following command:

For OS Version	Do this

	$ sudo zypper addrepo -f https://archive.cloudera.com/gplextras5/sles/12/x86_64/gplextras/cloudera-gplextras5.repo
	2. Update your system package index by running:
	$ sudo zypper refresh

Ubuntu or Debian	Go to this link and save the file as /etc/apt/sources.list.d/gplextras.list. **Important:** Make sure you do not let the file name default to cloudera.list, as that will overwrite your existing cloudera.list.

2. Install the package on each host as follows:

For OS version	Install commands
RHEL/CentOS compatible	sudo yum install hadoop-lzo
SLES	sudo zypper install hadoop-lzo
Ubuntu or Debian	sudo apt-get install hadoop-lzo

3. Continue with installing and deploying CDH. As part of the deployment, you will need to do some additional configuration for LZO, as shown under Configuring LZO on page 262.

Important: Be sure to do this configuration *after* you have copied the default configuration files to a custom location and set alternatives to point to it.

Step 6: Deploy CDH and Install Components

Proceed with:

- deploying CDH 5
- installing components.

Installing an Earlier CDH 5 Release

Follow these instructions to install a CDH 5 release that is **earlier than the current CDH 5 release**.

A common reason for doing this would be that you need to add new nodes to an existing cluster that is not running the most recent version of CDH 5. For example your cluster might be running CDH 5.0.1 when the most recent release is CDH 5.1.0; in this case, you will want to install CDH 5.0.1 on the new nodes, not CDH 5.1.0. These instructions are tailored for a fresh install (rather than an upgrade), in a cluster not being managed by Cloudera Manager,

Warning:

Do not attempt to use these instructions to roll your cluster back to a previous release. Use them only to expand an existing cluster that you do not want to upgrade to the latest release, or to create a new cluster running a version of CDH 5 that is earlier than the current CDH 5 release.

Downloading and Installing an Earlier Release

Choose your Linux version and proceed as follows to install an earlier release:

- On RHEL-compatible systems
- On SLES systems
- On Ubuntu and Debian systems

On RHEL-compatible systems

Step 1. Download and save the Yum repo file

Click the entry in the table below that matches your RHEL or CentOS system, go to the repo file for your system and save it in the `/etc/yum.repos.d/` directory.

For OS Version	Click this Link
RHEL/CentOS/Oracle 5	Red Hat/CentOS/Oracle 5 link
RHEL/CentOS 6 (64-bit)	Red Hat/CentOS 6 link

Step 2. Edit the repo file

Open the repo file you have just saved and change the `5` at the end of the line that begins `baseurl=` to the version number you want.

For example, if you have saved the file for Red Hat 6, it will look like this when you open it for editing:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/
gpgkey = https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera

gpgcheck = 1
```

If you want to install CDH 5.0.1, for example, change
`baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/` to

`baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.0.1/`

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.0.1/
gpgkey = https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/RPM-GPG-KEY-cloudera

gpgcheck = 1
```

Step 3: Proceed with the installation

1. Go to http://www.cloudera.com/content/cloudera/en/documentation.html.
2. Use the `Select Version` scroller to find the release you want, for example, select CDH and `5.0.x`
3. Find the CDH Installation Guide for your release.
4. Follow the instructions for RHEL on the "Installing CDH 5" page, starting with the instructions for optionally adding a repository key. (This comes immediately before the steps for installing CDH 5 with MRv1 or YARN, and is usually Step 2.)

On SLES systems

Step 1. Add the Cloudera repo

1. Run the following command:

```
$ sudo zypper addrepo -f
https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/cloudera-cdh5.repo
```

2. Update your system package index by running:

```
$ sudo zypper refresh
```

Step 2. Edit the repo file

Open the repo file that you have just added to your system and change the 5 at the end of the line that begins `baseurl=` to the version number you want.

The file should look like this when you open it for editing:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/
gpgkey = https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

If you want to install CDH5.0.1, for example, change
`baseurl=https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/` to

`baseurl= https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.0.1/`

In this example, the resulting file should look like this:

```
[cloudera-cdh5]
name=Cloudera's Distribution for Hadoop, Version 5
baseurl=https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5.0.1/
gpgkey = https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

Step 3: Proceed with the installation

1. Go to http://www.cloudera.com/content/cloudera/en/documentation.html.
2. Use the `Select a Product Version` scroller to find the release you want, for example `CDH 5.0.x`
3. Find the CDH Installation Guide for your release.
4. Follow the instructions for SLES on the "Installing CDH 5" page, starting with the instructions for optionally adding a repository key. (This comes immediately before the steps for installing CDH 5 with MRv1 or YARN, and is usually Step 2.)

On Ubuntu and Debian systems

Proceed as follows to add the Cloudera repo for your operating-system version and the Cloudera release you need.

Step 1: Create the repo File

Create a new file `/etc/apt/sources.list.d/cloudera.list` with the following contents:

- For Ubuntu systems:

```
deb [arch=amd64] https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5
contrib deb-src https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5
contrib
```

- For Debian systems:

```
deb https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib deb-src
 https://archive.cloudera.com/cdh5/ <OS-release-arch> <RELEASE>-cdh5 contrib
```

where: <OS-release-arch> is `debian/wheezy/amd64/cdh` or `ubuntu/precise/amd64/cdh`, and <RELEASE> is the name of your distribution, which you can find by running `lsb_release -c`.

Now replace `-cdh5` near the end of each line (before `contrib`) with the CDH release you need to install. Here are some examples using CDH5.0.1:

For 64-bit Ubuntu Precise:

```
deb [arch=amd64] https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh
precise-cdh5.0.1 contrib
deb-src https://archive.cloudera.com/cdh5/ubuntu/precise/amd64/cdh precise-cdh5.0.1
contrib
```

For Debian Wheezy:

```
deb https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.0.1 contrib
deb-src https://archive.cloudera.com/cdh5/debian/wheezy/amd64/cdh wheezy-cdh5.0.1 contrib
```

Step 2: Proceed with the installation

1. Go to http://www.cloudera.com/content/cloudera/en/documentation.html.
2. Use the `Select a Product Version` scroller to find the release you want, for example `CDH 5.0.x`
3. Find the CDH Installation Guide for your release.
4. Follow the instructions for Ubuntu or Debian on the "Installing CDH 5" page, starting with the instructions for optionally adding a repository key. (This comes immediately before the steps for installing CDH5 with MRv1 or YARN, and is usually Step 2.)

CDH 5 and MapReduce

CDH 5 supports two versions of the MapReduce computation framework: MRv1 and MRv2. The default installation in CDH 5 is MapReduce (MRv2) built on the YARN framework. In this document, Cloudera refers to MapReduce (MRv2) as YARN. You can use the instructions later in this section to install MRv1, YARN, or both implementations.

Important: 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.

YARN (MRv2)

The MapReduce v2 (MRv2) or YARN architecture splits the two primary responsibilities of the JobTracker — resource management and job scheduling/monitoring — into separate daemons: a global ResourceManager and per-application ApplicationMasters. With YARN, the ResourceManager and per-host NodeManagers form the data-computation framework. The ResourceManager service effectively replaces the functions of the JobTracker, and NodeManagers run on worker hosts instead of TaskTracker daemons. The per-application ApplicationMaster is, in effect, a framework-specific library and negotiates resources from the ResourceManager and works with the NodeManagers to run and monitor the tasks. For details of this architecture, see Apache Hadoop NextGen MapReduce (YARN).

See also Migrating from MapReduce (MRv1) to MapReduce (MRv2) on page 236.

Migrating from MapReduce (MRv1) to MapReduce (MRv2)

This is a guide to migrating from Apache MapReduce 1 (MRv1) to MapReduce (MRv2) (or YARN).

Introduction

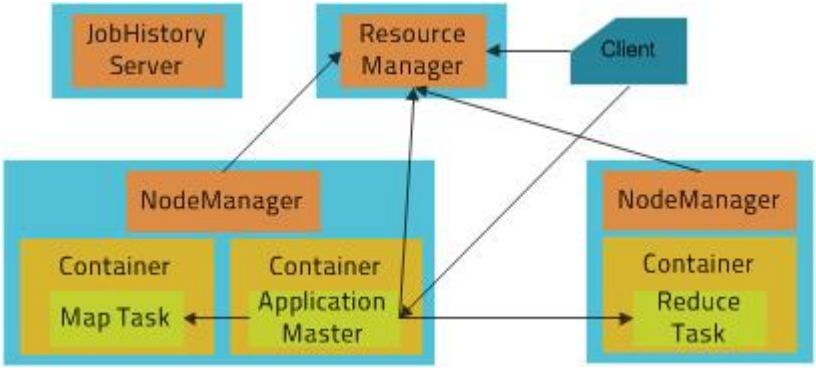
MapReduce 2, or Next Generation MapReduce, is a long needed upgrade to the way that scheduling, resource management, and execution occur in Hadoop. At their core, the improvements separate cluster resource management capabilities from MapReduce-specific logic. They enable Hadoop to share resources dynamically between MapReduce and other parallel processing frameworks, such as Impala, allow more sensible and finer-grained resource configuration for better cluster utilization, and permit it to scale to accommodate more and larger jobs.

This document provides a guide to both the architectural and user-facing changes, so that both cluster operators and MapReduce programmers can easily make the transition.

Terminology and Architecture

MapReduce 1 (MapReduce MRv1) has been split into two components. The cluster resource management capabilities have become YARN (Yet Another Resource Negotiator), while the MapReduce-specific capabilities remain MapReduce. In the MapReduce MRv1 architecture, the cluster was managed by a service called the JobTracker. TaskTracker services lived on each host and would launch tasks on behalf of jobs. The JobTracker would serve information about completed jobs.

In MapReduce MRv2, the functions of the JobTracker have been split between three services. The ResourceManager is a persistent YARN service that receives and runs applications (a MapReduce job is an application) on the cluster. It contains the scheduler, which, as previously, is pluggable. The MapReduce-specific capabilities of the JobTracker have been moved into the MapReduce ApplicationMaster, one of which is started to manage each MapReduce job and terminated when the job completes. The JobTracker function of serving information about completed jobs has been moved to the JobHistory Server. The TaskTracker has been replaced with the NodeManager, a YARN service that manages resources and deployment on a host. The TaskTracker is responsible for launching containers, each of which can house a map or reduce task.



The new architecture has its advantages. First, by breaking up the JobTracker into a few different services, it avoids many of the scaling issues faced by MapReduce in Hadoop 1. More importantly, it makes it possible to run frameworks other than MapReduce on a Hadoop cluster. For example, Impala can also run on YARN and share resources with MapReduce.

For MapReduce Programmers: Writing and Running Jobs

Nearly all jobs written for MRv1 can run without any modifications on an MRv2 cluster.

Java API Compatibility

MRv2 supports both the old (`mapred`) and new (`mapreduce`) MapReduce APIs used for MRv1, with a few caveats. The difference between the old and new APIs, which concerns user-facing changes, should not be confused with the

difference between MRv1 and MRv2, which concerns changes to the underlying framework. CDH 4 and CDH 5 both support the new and old MapReduce APIs.

In general, applications that use `@Public`/`@Stable` APIs are binary-compatible from CDH 4, meaning that compiled binaries should be able to run without modifications on the new framework. Source compatibility might be broken for applications that use a few obscure APIs that are technically public, but rarely needed and primarily exist for internal use. These APIs are detailed below. Source incompatibility means that code changes are required to compile. Source incompatibility is orthogonal to binary compatibility. Binaries for an application that is binary-compatible, but not source-compatible, continue to run on the new framework, but you must update your code to regenerate those binaries.

	Binary Incompatibilities	Source Incompatibilities
CDH 4 MRv1 to CDH 5 MRv1	None	None
CDH 4 MRv1 to CDH 5 MRv2	None	Rare
CDH 5 MRv1 to CDH 5 MRv2	None	Rare

The following are the known source incompatibilities:

- `KeyValueLineRecordReader#getProgress` and `LineRecordReader#getProgress` now throw IOExceptions in both the old and new APIs. Their superclass method, `RecordReader#getProgress`, already did this, but source compatibility will be broken for the rare code that used it without a `try/catch` block.
- `FileOutputCommitter#abortTask` now throws an `IOException`. Its superclass method always did this, but source compatibility will be broken for the rare code that used it without a `try/catch` block. This was fixed in CDH 4.3 MRv1 to be compatible with MRv2.
- `Job#getDependentJobs`, an API marked `@Evolving`, now returns a `List` instead of an `ArrayList`.

Compiling Jobs Against MRv2

If you are using Maven, compiling against MRv2 requires including the same artifact, `hadoop-client`. Changing the version to Hadoop 2 version (for example, using 2.2.0-cdh5.0.0 instead of 2.0.0-mr1-cdh4.3.0) should be enough. If you are not using Maven, compiling against all the Hadoop JARs is recommended. A comprehensive list of Hadoop Maven artifacts is available at: Using the CDH 5 Maven Repository.

If you want your job to run against both MRv1 and MRv2, compile it against MRv2.

Job Configuration

As in MRv1, job configuration options can be specified on the command line, in Java code, or in the `mapred-site.xml` on the client machine in the same way they previously were. The vast majority of job configuration options that were available in MRv1 work in MRv2 as well. For consistency and clarity, many options have been given new names. The older names are deprecated, but will still work for the time being. The exceptions to this are `mapred.child.ulimit` and all options relating to JVM reuse, as these are no longer supported.

Submitting and Monitoring Jobs

The MapReduce command line interface remains entirely compatible. Use of the Hadoop command line tool to run MapReduce related commands (`pipes`, `job`, `queue`, `classpath`, `historyserver`, `distcp`, `archive`) is deprecated, but still works. The `mapred` command line tool is preferred for these commands.

Selecting Appropriate JAR files for Your Jobs

The following table shows the names and locations of the JAR files used in MRv1 and the corresponding names and locations in YARN:

Name	MapReduce MRv1 location	YARN location
Streaming	`/usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.0.0-mr1-cdh<version>.jar`	`/usr/lib/hadoop-mapreduce/hadoop-streaming.jar`

Name	MapReduce MRv1 location	YARN location
Rumen	N/A	`/usr/lib/hadoop-mapreduce/hadoop-rumen.jar`
Hadoop Examples	`/usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar`	`/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar`
DistCp v1	`/usr/lib/hadoop-0.20-mapreduce/hadoop-tools.jar`	`/usr/lib/hadoop-mapreduce/hadoop-extras.jar`
DistCp v2	N/A	`/usr/lib/hadoop-mapreduce/hadoop-distcp.jar`
Hadoop archives	`/usr/lib/hadoop-0.20-mapreduce/hadoop-tools.jar`	`/usr/lib/hadoop-mapreduce/hadoop-archives.jar`

Requesting Resources

A MapReduce job submission includes the amount of resources to reserve for each map and reduce task. As in MapReduce 1, the amount of memory requested is controlled by the `mapreduce.map.memory.mb` and `mapreduce.reduce.memory.mb` properties.

MapReduce 2 adds additional parameters that control how much processing power to reserve for each task as well. The `mapreduce.map.cpu.vcores` and `mapreduce.reduce.cpu.vcores` properties express how much parallelism a map or reduce task can take advantage of. These should remain at their default value of 1 unless your code is explicitly spawning extra compute-intensive threads.

Note:

As of CDH 5.4.0, configuring MapReduce jobs is simpler than before. Instead of having to set both the heap size (`mapreduce.map.java.opts` or `mapreduce.reduce.java.opts`) and the container size (`mapreduce.map.memory.mb` or `mapreduce.reduce.memory.mb`), you can now choose to set only one of them. CDH infers the other value from `mapreduce.job.heap.memory-mb.ratio`. If you do not specify either value, container size defaults to 1 GiB and CDH infers the heap size.

The impact on user jobs is as follows: for jobs that do not set heap size, this increases the JVM size from 200 MB to a default 820 MB. This should be fine for most jobs, but streaming tasks might need more memory because their Java process takes their total usage over the container size. Even in that case, this would likely happen only for those tasks relying on aggressive garbage collection to keep the heap under 200 MB.

For Administrators: Configuring and Running MRv2 Clusters

Configuration Migration

Since MapReduce 1 functionality has been split into two components, MapReduce cluster configuration options have been split into YARN configuration options, which go in `yarn-site.xml`, and MapReduce configuration options, which go in `mapred-site.xml`. Many have been given new names to reflect the shift. As JobTrackers and TaskTrackers no longer exist in MRv2, all configuration options pertaining to them no longer exist, although many have corresponding options for the ResourceManager, NodeManager, and JobHistoryServer.

A minimal configuration required to run MRv2 jobs on YARN is:

- `yarn-site.xml` configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>you.hostname.com</value>
  </property>
```

```
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

- `mapred-site.xml` configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

See Deploying MapReduce v2 (YARN) on a Cluster on page 263 for instructions for a full deployment.

Resource Configuration

One of the larger changes in MRv2 is the way that resources are managed. In MRv1, each host was configured with a fixed number of map slots and a fixed number of reduce slots. Under YARN, there is no distinction between resources available for maps and resources available for reduces - all resources are available for both. Second, the notion of slots has been discarded, and resources are now configured in terms of amounts of memory (in megabytes) and CPU (in "virtual cores", which are described below). Resource configuration is an inherently difficult topic, and the added flexibility that YARN provides in this regard also comes with added complexity. Cloudera Manager will pick sensible values automatically, but if you are setting up your cluster manually or just interested in the details, read on.

Configuring Memory Settings for YARN and MRv2

The memory configuration for YARN and MRv2 memory is important to get the best performance from your cluster. Several different settings are involved. The table below shows the default settings, as well as the settings that Cloudera recommends, for each configuration option. See Managing YARN (MRv2) and MapReduce (MRv1) for more configuration specifics; and, for detailed tuning advice with sample configurations, see Tuning YARN.

Table 20: YARN and MRv2 Memory Configuration

Cloudera Manager Property Name	CDH Property Name	Default Configuration	Cloudera Tuning Guidelines
Container Memory Minimum	`yarn.scheduler.minimum-allocation-mb`	1 GB	0
Container Memory Maximum	`yarn.scheduler.maximum-allocation-mb`	64 GB	amount of memory on largest host
Container Memory Increment	`yarn.scheduler.increment-allocation-mb`	512 MB	Use a fairly large value, such as 128 MB
Container Memory	`yarn.nodemanager.resource.memory-mb`	8 GB	8 GB
Map Task Memory	`mapreduce.map.memory.mb`	1 GB	1 GB
Reduce Task Memory	`mapreduce.reduce.memory.mb`	1 GB	1 GB
Map Task Java Opts Base	`mapreduce.map.java.opts`	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m
Reduce Task Java Opts Base	`mapreduce.reduce.java.opts`	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m

Cloudera Manager Property Name	CDH Property Name	Default Configuration	Cloudera Tuning Guidelines
ApplicationMaster Memory	yarn.app.mapreduce.am.resource.mb	1 GB	1 GB
ApplicationMaster Java Opts Base	yarn.app.mapreduce.am.command-opts	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m

Resource Requests

From the perspective of a developer requesting resource allocations for a job's tasks, nothing needs to be changed. Map and reduce task memory requests still work and, additionally, tasks that will use multiple threads can request more than 1 core with the `mapreduce.map.cpu.vcores` and `mapreduce.reduce.cpu.vcores properties`.

Configuring Host Capacities

In MRv1, the `mapred.tasktracker.map.tasks.maximum` and `mapred.tasktracker.reduce.tasks.maximum` properties dictated how many map and reduce slots each TaskTracker had. These properties no longer exist in YARN. Instead, YARN uses `yarn.nodemanager.resource.memory-mb` and `yarn.nodemanager.resource.cpu-vcores`, which control the amount of memory and CPU on each host, both available to both maps and reduces. If you were using Cloudera Manager to configure these automatically, Cloudera Manager will take care of it in MRv2 as well. If configuring these manually, simply set these to the amount of memory and number of cores on the machine after subtracting out resources needed for other services.

Virtual Cores

To better handle varying CPU requests, YARN supports virtual cores (vcores) , a resource meant to express parallelism. The "virtual" in the name is somewhat misleading - on the NodeManager, vcores should be configured equal to the number of physical cores on the machine. Tasks should be requested with vcores equal to the number of cores they can saturate at once. Currently vcores are very coarse - tasks will rarely want to ask for more than one of them, but a complementary axis that represents processing power may be added in the future to enable finer-grained resource configuration.

Rounding Request Sizes

Also noteworthy are the `yarn.scheduler.minimum-allocation-mb`, `yarn.scheduler.minimum-allocation-vcores`, `yarn.scheduler.increment-allocation-mb`, and `yarn.scheduler.increment-allocation-vcores` properties, which default to 1024, 1, 512, and 1 respectively. If tasks are submitted with resource requests lower than the minimum-allocation values, their requests will be set to these values. If tasks are submitted with resource requests that are not multiples of the increment-allocation values, their requests will be rounded up to the nearest increments.

To make all of this more concrete, let's use an example. Each host in the cluster has 24 GB of memory and 6 cores. Other services running on the hosts require 4 GB and 1 core, so we set `yarn.nodemanager.resource.memory-mb` to 20480 and `yarn.nodemanager.resource.cpu-vcores` to 5. If you leave the map and reduce task defaults of 1024 MB and 1 virtual core intact, you will have at most 5 tasks running at the same time. If you want each of your tasks to use 5 GB, set their `mapreduce.(map|reduce).memory.mb` to 5120, which would limit you to 4 tasks running at the same time.

Scheduler Configuration

Cloudera recommends using the Fair Scheduler in MRv2. (FIFO and Capacity Scheduler are also available.) Fair Scheduler allocation files require changes in light of the new way that resources work. The `minMaps, maxMaps, minReduces,` and `maxReduces` queue properties have been replaced with a `minResources` property and a `maxProperties`. Instead of taking a number of slots, these properties take a value like "1024 MB, 3 vcores". By default, the MRv2 Fair Scheduler will attempt to equalize memory allocations in the same way it attempted to equalize slot allocations in MRv1. The MRv2 Fair Scheduler contains a number of new features including hierarchical queues and fairness based on multiple resources.

For further information on tuning and resource management, see Tuning YARN and YARN (MRv2) and MapReduce (MRv1) Schedulers.

Administration Commands

The `jobtracker` and `tasktracker` commands, which start the JobTracker and TaskTracker, are no longer supported because these services no longer exist. They are replaced with `yarn resourcemanager` and `yarn nodemanager`, which start the ResourceManager and NodeManager respectively. `hadoop mradmin` is no longer supported. Instead, `yarn rmadmin` should be used. The new admin commands mimic the functionality of the MRv1 names, allowing hosts, queues, and ACLs to be refreshed while the ResourceManager is running.

Security

The following section outlines the additional changes needed to migrate a secure cluster.

New YARN Kerberos service principals should be created for the ResourceManager and NodeManager, using the pattern used for other Hadoop services, that is, yarn@*HOST*. The `mapred` principal should still be used for the JobHistory Server. If you are using Cloudera Manager to configure security, this will be taken care of automatically.

As in MRv1, a configuration must be set to have the user that submits a job own its task processes. The equivalent of the MRv1 LinuxTaskController is the LinuxContainerExecutor. In a secure setup, NodeManager configurations should set `yarn.nodemanager.container-executor.class` to `org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor`. Properties set in the `taskcontroller.cfg` configuration file should be migrated to their analogous properties in the `container-executor.cfg` file.

In secure setups, configuring `hadoop-policy.xml` allows administrators to set up access control lists on internal protocols. The following is a table of MRv1 options and their MRv2 equivalents:

MRv1	MRv2	Comment
`security.task.umbilical.protocol.acl`	`security.job.task.protocol.acl`	As in MRv1, this should never be set to anything other than *
`security.inter.tracker.protocol.acl`	`security.resourcetracker.protocol.acl`	
`security.job.submission.protocol.acl`	`security.applicationclient.protocol.acl`	
`security.admin.operations.protocol.acl`	`security.resourcemanager-administration.protocol.acl`	
	`security.applicationmaster.protocol.acl`	No MRv1 equivalent
	`security.containermanagement.protocol.acl`	No MRv1 equivalent
	`security.resourcelocalizer.protocol.acl`	No MRv1 equivalent
	`security.job.client.protocol.acl`	No MRv1 equivalent

Queue access control lists (ACLs) are now placed in the Fair Scheduler configuration file instead of the JobTracker configuration. A list of users and groups that can submit jobs to a queue can be placed in `aclSubmitApps` in the queue's configuration. The queue administration ACL is no longer supported, but will be in a future release.

Ports

The following is a list of default ports used by MRv2 and YARN, as well as the configuration properties used to configure them.

Port	Use	Property
8032	ResourceManager Client RPC	`yarn.resourcemanager.address`
8030	ResourceManager Scheduler RPC (for ApplicationMasters)	`yarn.resourcemanager.scheduler.address`

Port	Use	Property
8033	ResourceManager Admin RPC	`yarn.resourcemanager.admin.address`
8088	ResourceManager Web UI and REST APIs	`yarn.resourcemanager.webapp.address`
8031	ResourceManager Resource Tracker RPC (for NodeManagers)	`yarn.resourcemanager.resource-tracker.address`
8040	NodeManager Localizer RPC	`yarn.nodemanager.localizer.address`
8042	NodeManager Web UI and REST APIs	`yarn.nodemanager.webapp.address`
10020	Job History RPC	`mapreduce.jobhistory.address`
19888	Job History Web UI and REST APIs	`mapreduce.jobhistory.webapp.address`
13562	Shuffle HTTP	`mapreduce.shuffle.port`

Note: You can set `yarn.resourcemanager.hostname.id` for each ResourceManager instead of setting the ResourceManager values; this will cause YARN to use the default ports on those hosts.

High Availability

YARN supports ResourceManager HA to make a YARN cluster highly-available; the underlying architecture of active-standby pair is similar to JobTracker HA in MRv1. A major improvement over MRv1 is: in YARN, the completed tasks of in-flight MapReduce jobs are not re-run on recovery after the ResourceManager is restarted or failed over. Further, the configuration and setup has also been simplified. The main differences are:

1. Failover controller has been moved from a separate ZKFC daemon to be a part of the ResourceManager itself. So, there is no need to run an additional daemon.
2. Clients, applications, and NodeManagers do not require configuring a proxy-provider to talk to the active ResourceManager.

Below is a table with HA-related configurations used in MRv1 and their equivalents in YARN:

MRv1	YARN / MRv2	Comment
`mapred.jobtrackers.`*`name`*	`yarn.resourcemanager.ha.rm-ids`	
`mapred.ha.jobtracker.id`	`yarn.resourcemanager.ha.id`	Unlike in MRv1, this must be configured in YARN.
`mapred.jobtracker.`*`rpc-address.name.id`*	(See Configuring YARN (MRv2) ResourceManager High Availability Using the Command Line	YARN/ MRv2 has different RPC ports for different functionalities. Each port-related configuration must be suffixed with an id. There is no *name* component in YARN.
`mapred.ha.jobtracker.`*`rpc-address.name.id`*	`yarn.resourcemanager.ha.admin.address`	
`mapred.ha.fencing.methods`	`yarn.resourcemanager.ha.fencer`	Not required to be specified
`mapred.client.failover.*`	None	Not required
	`yarn.resourcemanager.ha.enabled`	Enable HA

MRv1	YARN / MRv2	Comment
mapred.jobtracker.restart.recover	yarn.resourcemanager.recovery.enabled	Enable recovery of jobs after failover
	yarn.resourcemanager.store.class	org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore
mapred.ha.automatic-failover.enabled	yarn.resourcemanager.ha.automatic-failover.enabled	Enable automatic failover
mapred.ha.zkfc.port	yarn.resourcemanager.ha.automatic-failover.port	
mapred.job.tracker	yarn.resourcemanager.cluster.id	Cluster name

Upgrading an MRv1 Installation Using Cloudera Manager

See Importing MapReduce Configurations to YARN for instructions.

Upgrading an MRv1 Installation Using the Command Line

1. Uninstall the following packages: hadoop-0.20-mapreduce, hadoop-0.20-mapreduce-jobtracker, hadoop-0.20-mapreduce-tasktracker, hadoop-0.20-mapreduce-zkfc, hadoop-0.20-mapreduce-jobtrackerha.
2. Install the following additional packages : hadoop-yarn, hadoop-mapreduce, hadoop-mapreduce-historyserver, hadoop-yarn-resourcemanager, hadoop-yarn-nodemanager.
3. Look at all the service configurations placed in mapred-site.xml and replace them with their corresponding YARN configuration. Configurations starting with yarn should be placed inside yarn-site.xml, not mapred-site.xml. Refer to Resource Configuration for best practices on how to convert TaskTracker slot capacities (mapred.tasktracker.map.tasks.maximum and mapred.tasktracker.reduce.tasks.maximum) to NodeManager resource capacities (yarn.nodemanager.resource.memory-mb and yarn.nodemanager.resource.cpu-vcores), as well as how to convert configurations in the Fair Scheduler allocations file, fair-scheduler.xml.
4. Start the ResourceManager, NodeManagers, and the JobHistoryServer.

Web UI

In MRv1, the JobTracker Web UI served detailed information about the state of the cluster and the jobs (recent and current) running on it. It also contained the job history page, which served information from disk about older jobs.

The MRv2 Web UI provides the same information structured in the same way, but has been revamped with a new look and feel. The ResourceManager's UI, which includes information about running applications and the state of the cluster, is now located by default at <ResourceManager host>:8088. The JobHistory UI is now located by default at <JobHistoryServer host>:19888. Jobs can be searched and viewed there just as they could in MRv1.

Because the ResourceManager is meant to be agnostic to many of the concepts in MapReduce, it cannot host job information directly. Instead, it proxies to a Web UI that can. If the job is running, this proxy is the relevant MapReduce ApplicationMaster; if the job has completed, then this proxy is the JobHistoryServer. Thus, the user experience is similar to that of MRv1, but the information is now coming from different places.

Summary of Configuration Changes

The following tables summarize the changes in configuration parameters between MRv1 and MRv2.

JobTracker Properties and ResourceManager Equivalents

MRv1	YARN / MRv2
mapred.jobtracker.taskScheduler	yarn.resourcemanager.scheduler.class
mapred.jobtracker.completeuserjobs.maximum	yarn.resourcemanager.max-completed-applications

MRv1	YARN / MRv2
mapred.jobtracker.restart.recover	yarn.resourcemanager.recovery.enabled
mapred.job.tracker	yarn.resourcemanager.hostname or all of the following: yarn.resourcemanager.address yarn.resourcemanager.scheduler.address yarn.resourcemanager.resource-tracker.address yarn.resourcemanager.admin.address
mapred.job.tracker.http.address	yarn.resourcemanager.webapp.address or yarn.resourcemanager.hostname
mapred.job.tracker.handler.count	yarn.resourcemanager.resource-tracker.client.thread-count
mapred.hosts	yarn.resourcemanager.nodes.include-path
mapred.hosts.exclude	yarn.resourcemanager.nodes.exclude-path
mapred.cluster.max.map.memory.mb	yarn.scheduler.maximum-allocation-mb
mapred.cluster.max.reduce.memory.mb	yarn.scheduler.maximum-allocation-mb
mapred.acls.enabled	yarn.acl.enable
mapreduce.cluster.acls.enabled	yarn.acl.enable

JobTracker Properties and JobHistoryServer Equivalents

MRv1	YARN / MRv2	Comment
mapred.job.tracker.retiredjobs.cache.size	mapreduce.jobhistory.joblist.cache.size	
mapred.job.tracker.jobhistory.lru.cache.size	mapreduce.jobhistory.loadedjobs.cache.size	
mapred.job.tracker.history.completed.location	mapreduce.jobhistory.done-dir	Local FS in MR1; stored in HDFS in MR2
hadoop.job.history.user.location	mapreduce.jobhistory.done-dir	
hadoop.job.history.location	mapreduce.jobhistory.done-dir	

JobTracker Properties and MapReduce ApplicationMaster Equivalents

MRv1	YARN / MRv2	Comment
mapreduce.jobtracker.staging.root.dir	yarn.app.mapreduce.am.staging-dir	Now configurable per job

TaskTracker Properties and NodeManager Equivalents

MRv1	YARN / MRv2
mapred.tasktracker.map.tasks.maximum	yarn.nodemanager.resource.memory-mb and yarn.nodemanager.resource.cpu-vcores
mapred.tasktracker.reduce.tasks.maximum	yarn.nodemanager.resource.memory-mb and yarn.nodemanager.resource.cpu-vcores
mapred.tasktracker.expiry.interval	yarn.nm.liveliness-monitor.expiry-interval-ms

MRv1	YARN / MRv2
mapred.tasktracker.resourcecalculatorplugin	yarn.nodemanager.container-monitor.resource-calculator.class
mapred.tasktracker.taskmemorymanager.monitoring-interval	yarn.nodemanager.container-monitor.interval-ms
mapred.tasktracker.tasks.sleeptime-before-sigkill	yarn.nodemanager.sleep-delay-before-sigkill.ms
mapred.task.tracker.task-controller	yarn.nodemanager.container-executor.class
mapred.local.dir	yarn.nodemanager.local-dirs
mapreduce.cluster.local.dir	yarn.nodemanager.local-dirs
mapred.disk.healthChecker.interval	yarn.nodemanager.disk-health-checker.interval-ms
mapred.healthChecker.script.path	yarn.nodemanager.health-checker.script.path
mapred.healthChecker.interval	yarn.nodemanager.health-checker.interval-ms
mapred.healthChecker.script.timeout	yarn.nodemanager.health-checker.script.timeout-ms
mapred.healthChecker.script.args	yarn.nodemanager.health-checker.script.opts
local.cache.size	yarn.nodemanager.localizer.cache.target-size-mb
mapreduce.tasktracker.cache.local.size	yarn.nodemanager.localizer.cache.target-size-mb

TaskTracker Properties and Shuffle Service Equivalents

The table that follows shows TaskTracker properties and their equivalents in the auxiliary shuffle service that runs inside NodeManagers.

MRv1	YARN / MRv2
tasktracker.http.threads	mapreduce.shuffle.max.threads
mapred.task.tracker.http.address	mapreduce.shuffle.port
mapred.tasktracker.indexcache.mb	mapred.tasktracker.indexcache.mb

Per-Job Configuration Properties

Many of these properties have new names in MRv2, but the MRv1 names will work for all properties except mapred.job.restart.recover.

MRv1	YARN / MRv2	Comment
io.sort.mb	mapreduce.task.io.sort.mb	MRv1 name still works
io.sort.factor	mapreduce.task.io.sort.factor	MRv1 name still works
io.sort.spill.percent	mapreduce.task.io.sort.spill.percent	MRv1 name still works
mapred.map.tasks	mapreduce.job.maps	MRv1 name still works
mapred.reduce.tasks	mapreduce.job.reduces	MRv1 name still works
mapred.job.map.memory.mb	mapreduce.map.memory.mb	MRv1 name still works
mapred.job.reduce.memory.mb	mapreduce.reduce.memory.mb	MRv1 name still works
mapred.map.child.log.level	mapreduce.map.log.level	MRv1 name still works
mapred.reduce.child.log.level	mapreduce.reduce.log.level	MRv1 name still works
mapred.inmem.merge.threshold	mapreduce.reduce.shuffle.merge.inmem.threshold	MRv1 name still works
mapred.job.shuffle.merge.percent	mapreduce.reduce.shuffle.merge.percent	MRv1 name still works
mapred.job.shuffle.input.buffer.percent	mapreduce.reduce.shuffle.input.buffer.percent	MRv1 name still works

MRv1	YARN / MRv2	Comment
mapred.job.reduce.input.buffer.percent	mapreduce.reduce.input.buffer.percent	MRv1 name still works
mapred.map.tasks.speculative.execution	mapreduce.map.speculative	Old one still works
mapred.reduce.tasks.speculative.execution	mapreduce.reduce.speculative	MRv1 name still works
mapred.min.split.size	mapreduce.input.fileinputformat.split.minsize	MRv1 name still works
keep.failed.task.files	mapreduce.task.files.preserve.failedtasks	MRv1 name still works
mapred.output.compress	mapreduce.output.fileoutputformat.compress	MRv1 name still works
mapred.map.output.compression.codec	mapreduce.map.output.compress.codec	MRv1 name still works
mapred.compress.map.output	mapreduce.map.output.compress	MRv1 name still works
mapred.output.compression.type	mapreduce.output.fileoutputformat.compress.type	MRv1 name still works
mapred.userlog.limit.kb	mapreduce.task.userlog.limit.kb	MRv1 name still works
jobclient.output.filter	mapreduce.client.output.filter	MRv1 name still works
jobclient.completion.poll.interval	mapreduce.client.completion.pollinterval	MRv1 name still works
jobclient.progress.monitor.poll.interval	mapreduce.client.progressmonitor.pollinterval	MRv1 name still works
mapred.task.profile	mapreduce.task.profile	MRv1 name still works
mapred.task.profile.maps	mapreduce.task.profile.maps	MRv1 name still works
mapred.task.profile.reduces	mapreduce.task.profile.reduces	MRv1 name still works
mapred.line.input.format.linespermap	mapreduce.input.lineinputformat.linespermap	MRv1 name still works
mapred.skip.attempts.to.start.skipping	mapreduce.task.skip.start.attempts	MRv1 name still works
mapred.skip.map.auto.incr.proc.count	mapreduce.map.skip.proc.count.autoincr	MRv1 name still works
mapred.skip.reduce.auto.incr.proc.count	mapreduce.reduce.skip.proc.count.autoincr	MRv1 name still works
mapred.skip.out.dir	mapreduce.job.skip.outdir	MRv1 name still works
mapred.skip.map.max.skip.records	mapreduce.map.skip.maxrecords	MRv1 name still works
mapred.skip.reduce.max.skip.groups	mapreduce.reduce.skip.maxgroups	MRv1 name still works
job.end.retry.attempts	mapreduce.job.end-notification.retry.attempts	MRv1 name still works
job.end.retry.interval	mapreduce.job.end-notification.retry.interval	MRv1 name still works
job.end.notification.url	mapreduce.job.end-notification.url	MRv1 name still works
mapred.merge.recordsBeforeProgress	mapreduce.task.merge.progress.records	MRv1 name still works
mapred.job.queue.name	mapreduce.job.queuename	MRv1 name still works
mapred.reduce.slowstart.completed.maps	mapreduce.job.reduce.slowstart.completedmaps	MRv1 name still works
mapred.map.max.attempts	mapreduce.map.maxattempts	MRv1 name still works
mapred.reduce.max.attempts	mapreduce.reduce.maxattempts	MRv1 name still works
mapred.reduce.parallel.copies	mapreduce.reduce.shuffle.parallelcopies	MRv1 name still works
mapred.task.timeout	mapreduce.task.timeout	MRv1 name still works
mapred.max.tracker.failures	mapreduce.job.maxtaskfailures.per.tracker	MRv1 name still works
mapred.job.restart.recover	mapreduce.am.max-attempts	

MRv1	YARN / MRv2	Comment
mapred.combine.recordsBeforeProgress	mapreduce.task.combine.progress.records	MRv1 name should still work - see MAPREDUCE-5130

Miscellaneous Properties

MRv1	YARN / MRv2
mapred.heartbeats.in.second	yarn.resourcemanager.nodemanagers.heartbeat-interval-ms
mapred.userlog.retain.hours	yarn.log-aggregation.retain-seconds

MRv1 Properties that have no MRv2 Equivalents

MRv1	Comment
mapreduce.tasktracker.group	
mapred.child.ulimit	
mapred.tasktracker.dns.interface	
mapred.tasktracker.dns.nameserver	
mapred.tasktracker.instrumentation	NodeManager does not accept instrumentation
mapred.job.reuse.jvm.num.tasks	JVM reuse no longer supported
mapreduce.job.jvm.numtasks	JVM reuse no longer supported
mapred.task.tracker.report.address	No need for this, as containers do not use IPC with NodeManagers, and ApplicationMaster ports are chosen at runtime
mapreduce.task.tmp.dir	No longer configurable. Now always tmp/ (under container's local dir)
mapred.child.tmp	No longer configurable. Now always tmp/ (under container's local dir)
mapred.temp.dir	
mapred.jobtracker.instrumentation	ResourceManager does not accept instrumentation
mapred.jobtracker.plugins	ResourceManager does not accept plugins
mapred.task.cache.level	
mapred.queue.names	These go in the scheduler-specific configuration files
mapred.system.dir	
mapreduce.tasktracker.cache.local.numberdirectories	
mapreduce.reduce.input.limit	
io.sort.record.percent	Tuned automatically (MAPREDUCE-64)
mapred.cluster.map.memory.mb	Not necessary; MRv2 uses resources instead of slots
mapred.cluster.reduce.memory.mb	Not necessary; MRv2 uses resources instead of slots
mapred.max.tracker.blacklists	
mapred.jobtracker.maxtasks.per.job	Related configurations go in scheduler-specific configuration files

MRv1	Comment
mapred.jobtracker.taskScheduler.maxRunningTasksPerJob	Related configurations go in scheduler-specific configuration files
io.map.index.skip	
mapred.user.jobconf.limit	
mapred.local.dir.minspacestart	
mapred.local.dir.minspacekill	
hadoop.rpc.socket.factory.class.JobSubmissionProtocol	
mapreduce.tasktracker.outofband.heartbeat	Always on
mapred.jobtracker.job.history.block.size	

Deploying CDH 5 on a Cluster

Note: Do the tasks in this section after installing the latest version of CDH; see Installing the Latest CDH 5 Release on page 223.

To deploy CDH 5 on a cluster, do the following:

1. Configuring Dependencies Before Deploying CDH on a Cluster on page 248
2. Deploying HDFS on a Cluster on page 252
3. Deploy YARN with MapReduce v2 (YARN) *or* MapReduce v1 (MRv1)

See also:

- Configuring Hadoop Daemons to Run at Startup on page 272
- Optimizing Performance in CDH
- Configuring Centralized Cache Management in HDFS
- Managing HDFS Snapshots
- Configuring an NFSv3 Gateway Using the Command Line

Configuring Dependencies Before Deploying CDH on a Cluster

This section explains the tasks you must perform before deploying CDH on a cluster.

Enable an NTP Service

CDH requires that you configure a Network Time Protocol (NTP) service on each machine in your cluster. Most operating systems include the `ntpd` service for time synchronization.

RHEL 7 compatible operating systems use `chronyd` by default instead of `ntpd`. If `chronyd` is running (on any OS), Cloudera Manager uses it to determine whether the host clock is synchronized. Otherwise, Cloudera Manager uses `ntpd`.

Note: If you are using `ntpd` to synchronize your host clocks, but `chronyd` is also running, Cloudera Manager relies on `chronyd` to verify time synchronization, even if it is not synchronizing properly. This can result in Cloudera Manager reporting clock offset errors, even though the time is correct.

To fix this, either configure and use `chronyd` or disable it and remove it from the hosts.

To use `ntpd` for time synchronization:

1. Install the `ntp` package:

- RHEL compatible:

```
yum install ntp
```

- SLES:

```
zypper install ntp
```

- Ubuntu:

```
apt-get install ntp
```

2. Edit the `/etc/ntp.conf` file to add NTP servers, as in the following example.

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
```

3. Start the `ntpd` service:

- RHEL 7 Compatible:

```
sudo systemctl start ntpd
```

- RHEL 6 Compatible, SLES, Ubuntu:

```
sudo service ntpd start
```

4. Configure the `ntpd` service to run at boot:

- RHEL 7 Compatible:

```
sudo systemctl enable ntpd
```

- RHEL 6 Compatible, SLES, Ubuntu:

```
chkconfig ntpd on
```

5. Synchronize the system clock to the NTP server:

```
ntpdate -u <ntp_server>
```

6. Synchronize the hardware clock to the system clock:

```
hwclock --systohc
```

Configuring Network Names

This page is for manual CDH installations only. Cloudera Manager users should disregard.

Important: CDH requires IPv4. IPv6 is not supported.

Tip: When bonding, use the `bond0` IP address as it represents all aggregated links.

Configure each host in the cluster as follows to ensure that all members can communicate with each other:

1. Set the hostname to a unique name (not `localhost`).

```
sudo hostnamectl set-hostname foo-1.example.com
```

2. Edit `/etc/hosts` with the IP address and fully qualified domain name (FQDN) of each host in the cluster. You can add the unqualified name as well.

```
1.1.1.1  foo-1.example.com  foo-1
2.2.2.2  foo-2.example.com  foo-2
3.3.3.3  foo-3.example.com  foo-3
4.4.4.4  foo-4.example.com  foo-4
```

Important:

- The canonical name of each host in `/etc/hosts` **must** be the FQDN (for example `myhost-1.mynet.myco.com`), not the unqualified hostname (for example `myhost-1`). The canonical name is the first entry after the IP address.
- Do not use aliases, either in `/etc/hosts` or in configuring DNS.

3. Edit `/etc/sysconfig/network` with the FQDN of this host only:

```
HOSTNAME=foo-1.example.com
```

4. Verify that each host consistently identifies to the network:

   a. Run `uname -a` and check that the hostname matches the output of the `hostname` command.

   b. Run `/sbin/ifconfig` and note the value of `inet addr` in the `eth0` (or `bond0`) entry, for example:

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:A4:E8:97
          inet addr:172.29.82.176  Bcast:172.29.87.255  Mask:255.255.248.0
...
```

   c. Run `host -v -t A `hostname`` and verify that the output matches the `hostname` command.

   The IP address should be the same as reported by `ifconfig` for `eth0` (or `bond0`):

```
$ host -v -t A `hostname`
Trying "foo-1.example.com"
...
;; ANSWER SECTION:
foo-1.example.com. 60 IN A 172.29.82.176
```

5. For YARN: ensure `conf/core-site.xml` and `conf/yarn-site.xml` have the **hostnames** (not the IP addresses) of the NameNode, the ResourceManager, and ResourceManager Scheduler. See Customizing Configuration Files and Deploying MapReduce v2 (YARN) on a Cluster.
6. For MRv1: ensure `conf/core-site.xml` and `conf/mapred-site.xml` have the **hostnames** (not the IP addresses) of the NameNode and the JobTracker. These can be FQDNs (`foo-1.example.com`) or unqualified hostnames (`foo-1`). See Customizing Configuration Files and Deploying MapReduce v1 (MRv1) on a Cluster.
7. Ensure components that depend on a client-server relationship (Oozie, HBase, ZooKeeper) are properly configured:
   - Oozie Installation
   - HBase Installation
   - ZooKeeper Installation

Disabling SELinux

> **Note:** Cloudera Enterprise, with the exception of Cloudera Navigator Encrypt, is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in `enforcing` mode. Cloudera is not responsible for SELinux policy development, support, or enforcement. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.
>
> If you are using SELinux in `enforcing` mode, Cloudera Support can request that you disable SELinux or change the mode to `permissive` to rule out SELinux as a factor when investigating reported issues.

Security-Enhanced Linux (SELinux) allows you to set access control through policies. If you are having trouble deploying CDH with your policies, set SELinux in permissive mode on each host before you deploy CDH on your cluster.

To set the SELinux mode, perform the following steps on each host.

1. Check the SELinux state:

```
getenforce
```

2. If the output is either `Permissive` or `Disabled`, you can skip this task and go to Disabling the Firewall on page 251. If the output is `enforcing`, continue to the next step.
3. Open the `/etc/selinux/config` file (in some systems, the `/etc/sysconfig/selinux` file).
4. Change the line `SELINUX=enforcing` to `SELINUX=permissive`.
5. Save and close the file.
6. Restart your system or run the following command to disable SELinux immediately:

```
setenforce 0
```

After you have installed and deployed CDH, you can re-enable SELinux by changing `SELINUX=permissive` back to `SELINUX=enforcing` in `/etc/selinux/config` (or `/etc/sysconfig/selinux`), and then running the following command to immediately switch to `enforcing` mode:

```
setenforce 1
```

If you are having trouble getting Cloudera Software working with SELinux, contact your OS vendor for support. Cloudera is not responsible for developing or supporting SELinux policies.

Disabling the Firewall

To disable the firewall on each host in your cluster, perform the following steps on each host.

1. For `iptables`, save the existing rule set:

```
sudo iptables-save > ~/firewall.rules
```

2. Disable the firewall:
   - RHEL 7 compatible:

```
sudo systemctl disable firewalld
sudo systemctl stop firewalld
```

   - RHEL 6 compatible:

```
sudo chkconfig iptables off
sudo service iptables stop
```

- SLES:

```
sudo chkconfig SuSEfirewall2_setup off
sudo chkconfig SuSEfirewall2_init off
sudo rcSuSEfirewall2 stop
```

- Ubuntu:

```
sudo service ufw stop
```

Deploying HDFS on a Cluster

Important:

For instructions for configuring High Availability (HA) for the NameNode, see HDFS High Availability. For instructions on using HDFS Access Control Lists (ACLs), see HDFS Extended ACLs.

Proceed as follows to deploy HDFS on a cluster. Do this for all clusters, whether you are deploying MRv1 or YARN:

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

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).

Copying the Hadoop Configuration and Setting Alternatives

To customize the Hadoop configuration:

1. Copy the default configuration to your custom directory:

```
$ sudo cp -r /etc/hadoop/conf.empty /etc/hadoop/conf.my_cluster
```

You can call this configuration anything you like; in this example, it's called `my_cluster`.

Important:

When performing the configuration tasks in this section, and when you go on to deploy MRv1 or YARN, edit the configuration files in this custom directory. Do not create your custom configuration in the default directory `/etc/hadoop/conf.empty`.

2. CDH uses the `alternatives` setting to determine which Hadoop configuration to use. Set `alternatives` to point to your custom directory, as follows.

To manually set the configuration on RHEL-compatible systems:

```
$ sudo alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.my_cluster
 50
$ sudo alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

To manually set the configuration on Ubuntu and SLES systems:

```
$ sudo update-alternatives --install /etc/hadoop/conf hadoop-conf
/etc/hadoop/conf.my_cluster 50
$ sudo update-alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

This tells CDH to use the configuration in `/etc/hadoop/conf.my_cluster`.

You can display the current `alternatives` setting as follows.

To display the current setting on RHEL-compatible systems:

```
sudo alternatives --display hadoop-conf
```

To display the current setting on Ubuntu, Debian, and SLES systems:

```
sudo update-alternatives --display hadoop-conf
```

You should see output such as the following:

```
hadoop-conf - status is auto.
link currently points to /etc/hadoop/conf.my_cluster
/etc/hadoop/conf.my_cluster - priority 50
/etc/hadoop/conf.empty - priority 10
Current `best' version is /etc/hadoop/conf.my_cluster.
```

Because the configuration in `/etc/hadoop/conf.my_cluster` has the highest priority (50), that is the one CDH will use. For more information on alternatives, see the `update-alternatives(8)` man page on Ubuntu and SLES systems or the `alternatives(8)` man page On Red Hat-compatible systems.

Customizing Configuration Files

The following tables show the most important properties that you must configure for your cluster.

Note:

For information on other important configuration properties, and the configuration files, see the Apache Cluster Setup page.

Property	Configuration File	Description
`fs.defaultFS`	`core-site.xml`	Note: `fs.default.name` is deprecated. Specifies the NameNode and the default file system, in the form `hdfs://<namenode host>:<namenode port>/`. The default value is `file///`. The default file system is used to resolve relative paths; for example, if `fs.default.name` or `fs.defaultFS` is set to `hdfs://mynamenode/`, the relative URI `/mydir/myfile` resolves to `hdfs://mynamenode/mydir/myfile`.

Property	Configuration File	Description
		Note: for the cluster to function correctly, the <namenode> part of the string **must** be the hostname (for example mynamenode), or the HA-enabled logical URI, not the IP address.
dfs.permissions.superusergroup	hdfs-site.xml	Specifies the UNIX group containing users that will be treated as superusers by HDFS. You can stick with the value of 'hadoop' or pick your own group depending on the security policies at your site.

Sample Configuration

core-site.xml:

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://namenode-host.company.com:8020</value>
</property>
```

hdfs-site.xml:

```
<property>
  <name>dfs.permissions.superusergroup</name>
  <value>hadoop</value>
</property>
```

Configuring Local Storage Directories

You need to specify, create, and assign the correct permissions to the local directories where you want the HDFS daemons to store data. You specify the directories by configuring the following two properties in the hdfs-site.xml file.

Property	Configuration File Location	Description
dfs.name.dir or dfs.namenode.name.dir	hdfs-site.xml on the NameNode	This property specifies the URIs of the directories where the NameNode stores its metadata and edit logs. Cloudera recommends that you specify at least two directories. One of these should be located on an NFS mount point, unless you will be using a HDFS HA configuration.
dfs.data.dir or dfs.datanode.data.dir	hdfs-site.xml on each DataNode	This property specifies the URIs of the directories where the DataNode stores blocks. Cloudera recommends that you configure the disks on the DataNode in a JBOD configuration, mounted at /data/1/ through /data/N, and configure dfs.data.dir or dfs.datanode.data.dir to specify file:///data/1/dfs/dn through file:///data/N/dfs/dn/.

> **Note:**
>
> `dfs.data.dir` and `dfs.name.dir` are deprecated; you should use `dfs.datanode.data.dir` and `dfs.namenode.name.dir` instead, though `dfs.data.dir` and `dfs.name.dir` will still work.

Sample configuration:

hdfs-site.xml on the NameNode:

```
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///data/1/dfs/nn,file:///nfsmount/dfs/nn</value>
</property>
```

hdfs-site.xml on each DataNode:

```
<property>
  <name>dfs.datanode.data.dir</name>

<value>file:///data/1/dfs/dn,file:///data/2/dfs/dn,file:///data/3/dfs/dn,file:///data/4/dfs/dn</value>
</property>
```

After specifying these directories as shown above, you must create the directories and assign the correct file permissions to them on each node in your cluster.

In the following instructions, local path examples are used to represent Hadoop parameters. Change the path examples to match your configuration.

Local directories:

- The `dfs.name.dir` or `dfs.namenode.name.dir` parameter is represented by the `/data/1/dfs/nn` and `/nfsmount/dfs/nn` path examples.
- The `dfs.data.dir` or `dfs.datanode.data.dir` parameter is represented by the `/data/1/dfs/dn`, `/data/2/dfs/dn`, `/data/3/dfs/dn`, and `/data/4/dfs/dn` examples.

To configure local storage directories for use by HDFS:

1. On a NameNode host: create the `dfs.name.dir` or `dfs.namenode.name.dir` local directories:

```
$ sudo mkdir -p /data/1/dfs/nn /nfsmount/dfs/nn
```

> **Important:**
>
> If you are using High Availability (HA), you should **not** configure these directories on an NFS mount; configure them on local storage.

2. On all DataNode hosts: create the `dfs.data.dir` or `dfs.datanode.data.dir` local directories:

```
$ sudo mkdir -p /data/1/dfs/dn /data/2/dfs/dn /data/3/dfs/dn /data/4/dfs/dn
```

3. Configure the owner of the `dfs.name.dir` or `dfs.namenode.name.dir` directory, and of the `dfs.data.dir` or `dfs.datanode.data.dir` directory, to be the `hdfs` user:

```
$ sudo chown -R hdfs:hdfs /data/1/dfs/nn /nfsmount/dfs/nn /data/1/dfs/dn /data/2/dfs/dn
 /data/3/dfs/dn /data/4/dfs/dn
```

> **Note:**
>
> For a list of the users created when you install CDH, see Hadoop Users in Cloudera Manager and CDH.

Here is a summary of the correct owner and permissions of the local directories:

Directory	Owner	Permissions (see Footnote 1)
`dfs.name.dir` or `dfs.namenode.name.dir`	`hdfs:hdfs`	drwx------
`dfs.data.dir` or `dfs.datanode.data.dir`	`hdfs:hdfs`	drwx------

Footnote: 1 The Hadoop daemons automatically set the correct permissions for you on `dfs.data.dir` or `dfs.datanode.data.dir`. But in the case of `dfs.name.dir` or `dfs.namenode.name.dir`, permissions are currently incorrectly set to the file-system default, usually drwxr-xr-x (755). Use the chmod command to reset permissions for these `dfs.name.dir` or `dfs.namenode.name.dir` directories to drwx------ (700); for example:

```
$ sudo chmod 700 /data/1/dfs/nn /nfsmount/dfs/nn
```

or

```
$ sudo chmod go-rx /data/1/dfs/nn /nfsmount/dfs/nn
```

> **Note:**
>
> If you specified nonexistent directories for the `dfs.data.dir` or `dfs.datanode.data.dir` property in the `hdfs-site.xml` file, CDH 5 will shut down. (In previous releases, CDH silently ignored nonexistent directories for `dfs.data.dir`.)

Configuring DataNodes to Tolerate Local Storage Directory Failure

By default, the failure of a single `dfs.data.dir` or `dfs.datanode.data.dir` will cause the HDFS DataNode process to shut down, which results in the NameNode scheduling additional replicas for each block that is present on the DataNode. This causes needless replications of blocks that reside on disks that have not failed.

To prevent this, you can configure DataNodes to tolerate the failure of `dfs.data.dir` or `dfs.datanode.data.dir` directories; use the `dfs.datanode.failed.volumes.tolerated` parameter in `hdfs-site.xml`. For example, if the value for this parameter is 3, the DataNode will only shut down after four or more data directories have failed. This value is respected on DataNode startup; in this example the DataNode will start up as long as no more than three directories have failed.

> **Note:**
>
> It is important that `dfs.datanode.failed.volumes.tolerated` not be configured to tolerate too many directory failures, as the DataNode will perform poorly if it has few functioning data directories.

Formatting the NameNode

Before starting the NameNode for the first time you need to format the file system.

Important:

- Make sure you format the NameNode as user `hdfs`.
- If you are re-formatting the NameNode, keep in mind that this invalidates the DataNode storage locations, so you should remove the data under those locations after the NameNode is formatted.

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

Note:

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

You'll get a confirmation prompt; for example:

```
Re-format filesystem in /data/namedir ? (Y or N)
```

Note: Respond with an **upper-case** Y; if you use lower case, the process will abort.

Configuring a Remote NameNode Storage Directory

You should configure the NameNode to write to multiple storage directories, including one remote NFS mount. To keep NameNode processes from hanging when the NFS server is unavailable, configure the NFS mount as a `soft` mount (so that I/O requests that time out fail rather than hang), and set other options as follows:

```
tcp,soft,intr,timeo=10,retrans=10
```

These options configure a soft mount over TCP; transactions will be retried ten times (`retrans=10`) at 1-second intervals (`timeo=10`) before being deemed to have failed.

Example:

```
mount -t nfs -o tcp,soft,intr,timeo=10,retrans=10, <server>:<export> <mount_point>
```

where `<server>` is the remote host, `<export>` is the exported file system, and `<mount_point>` is the local mount point.

Note:

Cloudera recommends similar settings for shared HA mounts, as in the example that follows.

Example for HA:

```
mount -t nfs -o tcp,soft,intr,timeo=50,retrans=12, <server>:<export> <mount_point>
```

Note that in the HA case `timeo` should be set to 50 (five seconds), rather than 10 (1 second), and `retrans` should be set to 12, giving an overall timeout of 60 seconds.

For more information, see the man pages for `mount` and `nfs`.

Configuring Remote Directory Recovery

You can enable the `dfs.namenode.name.dir.restore` option so that the NameNode will attempt to recover a previously failed NameNode storage directory on the next checkpoint. This is useful for restoring a remote storage directory mount that has failed because of a network outage or intermittent NFS failure.

Configuring the Secondary NameNode

Important:

The Secondary NameNode does not provide failover or High Availability (HA). If you intend to configure HA for the NameNode, skip this section: do not install or configure the Secondary NameNode (the Standby NameNode performs checkpointing). After completing the HA software configuration, follow the installation instructions under Deploying HDFS High Availability.

In non-HA deployments, configure a Secondary NameNode that will periodically merge the EditLog with the FSImage, creating a new FSImage which incorporates the changes which were in the EditLog. This reduces the amount of disk space consumed by the EditLog on the NameNode, and also reduces the restart time for the Primary NameNode.

A standard Hadoop cluster (not a Hadoop Federation or HA configuration), can have only one Primary NameNode plus one Secondary NameNode. On production systems, the Secondary NameNode should run on a different machine from the Primary NameNode to improve scalability (because the Secondary NameNode does not compete with the NameNode for memory and other resources to create the system snapshot) and durability (because the copy of the metadata is on a separate machine that is available if the NameNode hardware fails).

Configuring the Secondary NameNode on a Separate Machine

To configure the Secondary NameNode on a separate machine from the NameNode, proceed as follows.

1. Add the name of the machine that will run the Secondary NameNode to the `masters` file.
2. Add the following property to the `hdfs-site.xml` file:

```
<property>
  <name>dfs.namenode.http-address</name>
  <value><namenode.host.address>:50070</value>
  <description>
    The address and the base port on which the dfs NameNode Web UI will listen.
  </description>
</property>
```

Note:

- `dfs.http.address` is deprecated; use `dfs.namenode.http-address`.
- In most cases, you should set `dfs.namenode.http-address` to a routable IP address with port 50070. However, in some cases such as Amazon EC2, when the NameNode should bind to multiple local addresses, you may want to set `dfs.namenode.http-address` to `0.0.0.0:50070` *on the NameNode machine only*, and set it to a real, routable address on the Secondary NameNode machine. The different addresses are needed in this case because HDFS uses `dfs.namenode.http-address` for two different purposes: it defines both the address the NameNode binds to, and the address the Secondary NameNode connects to for checkpointing. Using `0.0.0.0` on the NameNode allows the NameNode to bind to all its local addresses, while using the externally-routable address on the Secondary NameNode provides the Secondary NameNode with a real address to connect to.

For more information, see Multi-host SecondaryNameNode Configuration.

More about the Secondary NameNode

- The NameNode stores the HDFS metadata information in RAM to speed up interactive lookups and modifications of the metadata.
- For reliability, this information is flushed to disk periodically. To ensure that these writes are not a speed bottleneck, only the list of modifications is written to disk, not a full snapshot of the current filesystem. The list of modifications is appended to a file called `edits`.
- Over time, the `edits` log file can grow quite large and consume large amounts of disk space.
- When the NameNode is restarted, it takes the HDFS system state from the `fsimage` file, then applies the contents of the `edits` log to construct an accurate system state that can be loaded into the NameNode's RAM. If you restart a large cluster that has run for a long period with no Secondary NameNode, the `edits` log may be quite large, and so it can take some time to reconstruct the system state to be loaded into RAM.

When the Secondary NameNode is configured, it periodically constructs a checkpoint by compacting the information in the edits log and merging it with the most recent `fsimage` file; it then clears the edits log. So, when the NameNode restarts, it can use the latest checkpoint and apply the contents of the smaller edits log. The interval between checkpoints is determined by the checkpoint period (`dfs.namenode.checkpoint.period`) or the number of edit transactions (`dfs.namenode.checkpoint.txns`). The default checkpoint period is one hour, and the default number of edit transactions before a checkpoint is 1,000,000. The SecondaryNameNode will checkpoint in an hour if there have not been 1,000,000 edit transactions within the hour; it will checkpoint after 1,000,000 transactions have been committed if they were committed in under one hour.

Secondary NameNode Parameters

The behavior of the Secondary NameNode is controlled by the following parameters in `hdfs-site.xml`.

- `dfs.namenode.checkpoint.check.period`
- `dfs.namenode.checkpoint.txns`
- `dfs.namenode.checkpoint.dir`
- `dfs.namenode.checkpoint.edits.dir`
- `dfs.namenode.num.checkpoints.retained`

See https://archive.cloudera.com/cdh5/cdh/5/hadoop/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml for details.

Enabling Trash

The Hadoop trash feature helps prevent accidental deletion of files and directories. If trash is enabled and a file or directory is deleted using the Hadoop shell, the file is moved to the `.Trash` directory in the user's home directory instead of being deleted. Deleted files are initially moved to the `Current` sub-directory of the `.Trash` directory, and their original path is preserved. If trash checkpointing is enabled, the `Current` directory is periodically renamed using a timestamp. Files in `.Trash` are permanently removed after a user-configurable time delay. Files and directories in the trash can be restored simply by moving them to a location outside the `.Trash` directory.

Important:

- The trash feature is disabled by default. Cloudera recommends that you enable it on all production clusters.
- The trash feature works by default only for files and directories deleted using the Hadoop shell. Files or directories deleted programmatically using other interfaces (WebHDFS or the Java APIs, for example) are not moved to trash, even if trash is enabled, unless the program has implemented a call to the trash functionality. (Hue, for example, implements trash as of CDH 4.4.)

  Users can bypass trash when deleting files using the shell by specifying the `-skipTrash` option to the `hadoop fs -rm -r` command. This can be useful when it is necessary to delete files that are too large for the user's quota.

Trash is configured with the following properties in the `core-site.xml` file:

CDH Parameter	Value	Description
`fs.trash.interval`	*minutes or* 0	The number of minutes after which a trash checkpoint directory is deleted. This option can be configured both on the server and the client. • If trash is enabled on the server configuration, then the value configured on the server is used and the client configuration is ignored. • If trash is disabled in the server configuration, then the client side configuration is checked. • If the value of this property is zero (the default), then the trash feature is disabled.
`fs.trash.checkpoint.interval`	*minutes or* 0	The number of minutes between trash checkpoints. Every time the checkpointer runs on the NameNode, it creates a new checkpoint of the "Current" directory and removes checkpoints older than `fs.trash.interval` minutes. This value should be smaller than or equal to `fs.trash.interval`. This option is configured on the server. If configured to zero (the default), then the value is set to the value of `fs.trash.interval`.

For example, to enable trash so that files deleted using the Hadoop shell are not deleted for 24 hours, set the value of the `fs.trash.interval` property in the server's `core-site.xml` file to a value of `1440`.

Note:

The period during which a file remains in the trash starts when the file is moved to the trash, not when the file is last modified.

Configuring Storage Balancing for DataNodes

You can configure HDFS to distribute writes on each DataNode in a manner that balances out available storage among that DataNode's disk volumes.

By default a DataNode writes new block replicas to disk volumes solely on a round-robin basis. You can configure a volume-choosing policy that causes the DataNode to take into account how much space is available on each volume when deciding where to place a new replica.

You can configure

- how much DataNode volumes are allowed to differ in terms of bytes of free disk space before they are considered imbalanced, *and*
- what percentage of new block allocations will be sent to volumes with more available disk space than others.

To configure storage balancing, set the following properties in `hdfs-site.xml`.

Note: Keep in mind that if usage is markedly imbalanced among a given DataNode's storage volumes when you enable storage balancing, throughput on that DataNode will be affected initially, as writes are disproportionately directed to the under-utilized volumes.

Property	Value	Description
dfs.datanode.fsdataset.volume.choosing.policy	org.apache.hadoop.hdfs.server.datanode.fsdataset.AvailableSpaceVolumeChoosingPolicy	Enables storage balancing among the DataNode's volumes.
dfs.datanode.available-space-volume-choosing-policy.balanced-space-threshold	10737418240 (default)	The amount by which volumes are allowed to differ from each other in terms of bytes of free disk space before they are considered imbalanced. The default is 10737418240 (10 GB). If the free space on each volume is within this range of the other volumes, the volumes will be considered balanced and block assignments will be done on a pure round-robin basis.
dfs.datanode.available-space-volume-choosing-policy.balanced-space-preference-fraction	0.75 (default)	What proportion of new block allocations will be sent to volumes with more available disk space than others. The allowable range is 0.0-1.0, but set it in the range 0.5 - 1.0 (that is, 50-100%), since there should be no reason to prefer that volumes with less available disk space receive more block allocations.

Enabling WebHDFS

> **Note:**
>
> To configure HttpFs instead, see HttpFS Installation on page 338.

If you want to use WebHDFS, you must first enable it.

To enable WebHDFS:

Set the following property in `hdfs-site.xml`:

```
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
```

To enable numeric usernames in WebHDFS:

By default, WebHDFS supports the following username pattern:

```
^[A-Za-z_][A-Za-z0-9._-]*[$]?$
```

You can override the default username pattern by setting the `dfs.webhdfs.user.provider.user.pattern` property in `hdfs-site.xml`. For example, to allow numerical usernames, the property can be set as follows:

```
<property>
  <name>dfs.webhdfs.user.provider.user.pattern</name>
  <value>^[A-Za-z0-9_][A-Za-z0-9._-]*[$]?$</value>
</property>
```

> **Important:** The username pattern should be compliant with the requirements of the operating system in use. Hence, Cloudera recommends you use the default pattern and avoid modifying the `dfs.webhdfs.user.provider.user.pattern` property when possible.

Note:

- To use WebHDFS in a secure cluster, you must set additional properties to configure secure WebHDFS. For instructions, see the Cloudera Security guide.
- When you use WebHDFS in a high-availability (HA) configuration, you must supply the value of `dfs.nameservices` in the WebHDFS URI, rather than the address of a particular NameNode; for example:

  `hdfs dfs -ls webhdfs://nameservice1/`, *not*

  `hdfs dfs -ls webhdfs://server1.myent.myco.com:20101/`

Configuring LZO

If you have installed LZO, configure it as follows.

To configure LZO:

Set the following property in `core-site.xml`.

Note:

If you copy and paste the *value* string, make sure you remove the line-breaks and carriage returns, which are included below because of page-width constraints.

```
<property>
  <name>io.compression.codecs</name>

<value>org.apache.hadoop.io.compress.DefaultCodec,org.apache.hadoop.io.compress.GzipCodec,
org.apache.hadoop.io.compress.BZip2Codec,com.hadoop.compression.lzo.LzoCodec,
com.hadoop.compression.lzo.LzopCodec,org.apache.hadoop.io.compress.SnappyCodec</value>
</property>
```

For more information about LZO, see Using LZO Compression.

Start HDFS

To deploy HDFS now, proceed as follows.

1. Deploy the configuration.
2. Start HDFS.
3. Create the `/tmp` directory.

Deploy the configuration

To deploy your configuration to your entire cluster:

1. Push your custom directory (for example `/etc/hadoop/conf.my_cluster`) to each node in your cluster; for example:

```
$ scp -r /etc/hadoop/conf.my_cluster
myuser@myCDHnode-<n>.mycompany.com:/etc/hadoop/conf.my_cluster
```

2. Manually set `alternatives` on each node to point to that directory, as follows.

   To manually set the configuration on RHEL-compatible systems:

```
$ sudo alternatives --verbose --install /etc/hadoop/conf hadoop-conf
/etc/hadoop/conf.my_cluster 50
$ sudo alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

To manually set the configuration on Ubuntu and SLES systems:

```
$ sudo update-alternatives --install /etc/hadoop/conf hadoop-conf
/etc/hadoop/conf.my_cluster 50
$ sudo update-alternatives --set hadoop-conf /etc/hadoop/conf.my_cluster
```

For more information on alternatives, see the `update-alternatives(8)` man page on Ubuntu and SLES systems or the `alternatives(8)` man page On RHEL-compatible systems.

Start HDFS

Start HDFS on each node in the cluster, as follows:

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

Note:

This starts all the CDH services installed on the node. This is normally what you want, but you can start services individually if you prefer.

Create the /tmp Directory

Important:

If you do not create `/tmp` properly, with the right permissions as shown below, you may have problems with CDH components later. Specifically, if you do not create `/tmp` yourself, another process may create it automatically with restrictive permissions that will prevent your other applications from using it.

Create the `/tmp` directory after HDFS is up and running, and set its permissions to 1777 (`drwxrwxrwt`), as follows:

```
$ sudo -u hdfs hadoop fs -mkdir /tmp
$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp
```

Note:

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

Deploy YARN or MRv1

To to deploy MRv1 or YARN, and start HDFS services if you have not already done so, see

- Deploying MapReduce v2 (YARN) on a Cluster on page 263 *or*
- Deploying MapReduce v1 (MRv1) on a Cluster on page 269

Deploying MapReduce v2 (YARN) on a Cluster

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

This section describes configuration tasks for YARN clusters only, and is specifically tailored for administrators who have installed YARN from packages.

Important:

Do the following tasks after you have configured and deployed HDFS:

Note: Running Services

Use the `service` command to start, stop, and restart CDH components, 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).

About MapReduce v2 (YARN)

The default installation in CDH 5 is MapReduce 2.*x* (MRv2) built on the YARN framework. In this document we usually refer to this new version as **YARN**. The fundamental idea of MRv2's YARN architecture is to split up the two primary responsibilities of the JobTracker — resource management and job scheduling/monitoring — into separate daemons: a global ResourceManager (RM) and per-application ApplicationMasters (AM). With MRv2, the ResourceManager (RM) and per-host NodeManagers (NM), form the data-computation framework. The ResourceManager service effectively replaces the functions of the JobTracker, and NodeManagers run on worker hosts instead of TaskTracker daemons. The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to run and monitor the tasks. For details of the new architecture, see Apache Hadoop NextGen MapReduce (YARN).

See also Selecting Appropriate JAR files for Your Jobs on page 237.

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 performance and may result in an unstable cluster deployment.

- If you have installed YARN from packages, follow the instructions below to deploy it. (To deploy MRv1 instead, see Deploying MapReduce v1 (MRv1) on a Cluster.)
- If you have installed CDH 5 from tarballs, the default deployment is YARN. Keep in mind that the instructions on this page are tailored for a deployment following installation from packages.

Step 1: Configure Properties for YARN Clusters

Note:

Edit these files in the custom directory you created when you copied the Hadoop configuration. When you have finished, you will push this configuration to all the hosts in the cluster; see Step 5.

Property	Configuration File	Description
`mapreduce.framework.name`	`mapred-site.xml`	If you plan on running YARN, you must set this property to the value of `yarn`.

Sample Configuration:

mapred-site.xml:

```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

Step 2: Configure YARN daemons

Configure the following services: ResourceManager (on a dedicated host) and NodeManager (on every host where you plan to run MapReduce v2 jobs).

The following table shows the most important properties that you must configure for your cluster in `yarn-site.xml`

Property	Recommended value	Description
`yarn.nodemanager.aux-services`	`mapreduce_shuffle`	Shuffle service that needs to be set for Map Reduce applications.
`yarn.resourcemanager.hostname`	`resourcemanager.company.com`	The following properties will be set to their default ports on this host: `yarn.resourcemanager.address`, `yarn.resourcemanager.admin.address`, `yarn.resourcemanager.scheduler.address`, `yarn.resourcemanager.resource-tracker.address`, `yarn.resourcemanager.webapp.address`
`yarn.application.classpath`	`$HADOOP_CONF_DIR`, `$HADOOP_COMMON_HOME/*`, `$HADOOP_COMMON_HOME/lib/*`, `$HADOOP_HDFS_HOME/*`, `$HADOOP_HDFS_HOME/lib/*`, `$HADOOP_MAPRED_HOME/*`, `$HADOOP_MAPRED_HOME/lib/*`, `$HADOOP_YARN_HOME/*`, `$HADOOP_YARN_HOME/lib/*`	Classpath for typical applications.
`yarn.log.aggregation-enable`	`true`	

Next, you need to specify, create, and assign the correct permissions to the local directories where you want the YARN daemons to store data.

You specify the directories by configuring the following two properties in the `yarn-site.xml` file on all cluster hosts:

Property	Description
`yarn.nodemanager.local-dirs`	Specifies the URIs of the directories where the NodeManager stores its localized files. All of the files required for running a particular YARN application will be put here for the duration of the application run. Cloudera recommends that this property specify a directory on each of the JBOD mount points; for example, `file:///data/1/yarn/local` through `/data/N/yarn/local`.

Property	Description
yarn.nodemanager.log-dirs	Specifies the URIs of the directories where the NodeManager stores container log files. Cloudera recommends that this property specify a directory on each of the JBOD mount points; for example, file:///data/1/yarn/logs through file:///data/N/yarn/logs.
yarn.nodemanager.remote-app-log-dir	Specifies the URI of the directory where logs are aggregated. Set the value to *either* hdfs://*namenode-host.company.com*:8020/var/log/hadoop-yarn/apps, using the fully qualified domain name of your NameNode host, *or* hdfs:/var/log/hadoop-yarn/apps.

Here is an example configuration:

yarn-site.xml:

```
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>resourcemanager.company.com</value>
  </property>
  <property>
    <description>Classpath for typical applications.</description>
    <name>yarn.application.classpath</name>
    <value>
        $HADOOP_CONF_DIR,
        $HADOOP_COMMON_HOME/*,$HADOOP_COMMON_HOME/lib/*,
        $HADOOP_HDFS_HOME/*,$HADOOP_HDFS_HOME/lib/*,
        $HADOOP_MAPRED_HOME/*,$HADOOP_MAPRED_HOME/lib/*,
        $HADOOP_YARN_HOME/*,$HADOOP_YARN_HOME/lib/*
    </value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>

<value>file:///data/1/yarn/local,file:///data/2/yarn/local,file:///data/3/yarn/local</value>

  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>

<value>file:///data/1/yarn/logs,file:///data/2/yarn/logs,file:///data/3/yarn/logs</value>

  </property>
  <property>
    <name>yarn.log.aggregation-enable</name>
    <value>true</value>
  </property>
  <property>
    <description>Where to aggregate logs</description>
    <name>yarn.nodemanager.remote-app-log-dir</name>
    <value>hdfs://<namenode-host.company.com>:8020/var/log/hadoop-yarn/apps</value>
  </property>
```

After specifying these directories in the yarn-site.xml file, you must create the directories and assign the correct file permissions to them on each host in your cluster.

In the following instructions, local path examples are used to represent Hadoop parameters. Change the path examples to match your configuration.

To configure local storage directories for use by YARN:

1. Create the `yarn.nodemanager.local-dirs` local directories:

```
$ sudo mkdir -p /data/1/yarn/local /data/2/yarn/local /data/3/yarn/local
/data/4/yarn/local
```

2. Create the `yarn.nodemanager.log-dirs` local directories:

```
$ sudo mkdir -p /data/1/yarn/logs /data/2/yarn/logs /data/3/yarn/logs /data/4/yarn/logs
```

3. Configure the owner of the `yarn.nodemanager.local-dirs` directory to be the `yarn` user:

```
$ sudo chown -R yarn:yarn /data/1/yarn/local /data/2/yarn/local /data/3/yarn/local
/data/4/yarn/local
```

4. Configure the owner of the `yarn.nodemanager.log-dirs` directory to be the `yarn` user:

```
$ sudo chown -R yarn:yarn /data/1/yarn/logs /data/2/yarn/logs /data/3/yarn/logs
/data/4/yarn/logs
```

Here is a summary of the correct owner and permissions of the local directories:

Directory	Owner	Permissions
`yarn.nodemanager.local-dirs`	`yarn:yarn`	drwxr-xr-x
`yarn.nodemanager.log-dirs`	`yarn:yarn`	drwxr-xr-x

Step 3: Configure the JobHistory Server

If you have decided to run YARN on your cluster instead of MRv1, you should also run the MapReduce JobHistory Server. The following table shows the most important properties that you must configure in `mapred-site.xml`.

Property	Recommended value	Description
`mapreduce.jobhistory.address`	`historyserver.company.com:10020`	The address of the JobHistory Server `host:port`
`mapreduce.jobhistory.webapp.address`	`historyserver.company.com:19888`	The address of the JobHistory Server web application `host:port`

In addition, make sure proxying is enabled for the `mapred` user; configure the following properties in `core-site.xml`:

Property	Recommended value	Description
`hadoop.proxyuser.mapred.groups`	*	Allows the `mapred`user to move files belonging to users in these groups
`hadoop.proxyuser.mapred.hosts`	*	Allows the `mapred`user to move files belonging on these hosts

Step 4: Configure the Staging Directory

YARN requires a staging directory for temporary files created by running jobs. By default it creates `/tmp/hadoop-yarn/staging` with restrictive permissions that may prevent your users from running jobs. To forestall this, you should configure and create the staging directory yourself; in the example that follows we use `/user`:

1. Configure `yarn.app.mapreduce.am.staging-dir` in `mapred-site.xml`:

```
<property>
    <name>yarn.app.mapreduce.am.staging-dir</name>
    <value>/user</value>
</property>
```

2. Once HDFS is up and running, you will create this directory and a `history` subdirectory under it (see Step 8).

Alternatively, you can do the following:

1. Configure `mapreduce.jobhistory.intermediate-done-dir` and `mapreduce.jobhistory.done-dir` in `mapred-site.xml`.
2. Create these two directories.
3. Set permissions on `mapreduce.jobhistory.intermediate-done-dir` to 1777.
4. Set permissions on `mapreduce.jobhistory.done-dir` to 750.

If you configure `mapreduce.jobhistory.intermediate-done-dir` and `mapreduce.jobhistory.done-dir` as above, you can skip Step 8.

Step 5: If Necessary, Deploy your Custom Configuration to your Entire Cluster

Deploy the configuration on page 262 if you have not already done so.

Step 6: If Necessary, Start HDFS on Every Host in the Cluster

Start HDFS on page 262 if you have not already done so.

Step 7: If Necessary, Create the HDFS `/tmp` Directory

Create the /tmp Directory on page 263 if you have not already done so.

Important:

If you do not create `/tmp` with the permissions as shown below, you might have problems with CDH components later. Specifically, if you do not create `/tmp` yourself, another process might create it automatically with restrictive permissions that prevent your other applications from using it.

Step 8: Create the `history` Directory and Set Permissions

This is a subdirectory of the staging directory you configured in Step 4. In this example we're using `/user/history`. Create it and set permissions as follows:

```
sudo -u hdfs hadoop fs -mkdir -p /user/history
sudo -u hdfs hadoop fs -chmod -R 1777 /user/history
sudo -u hdfs hadoop fs -chown mapred:hadoop /user/history
```

Step 9: Start YARN and the MapReduce JobHistory Server

To start YARN, start the ResourceManager and NodeManager services:

Note:

Make sure you always start ResourceManager before starting NodeManager services.

On the ResourceManager system:

```
$ sudo service hadoop-yarn-resourcemanager start
```

On each NodeManager system (typically the same ones where DataNode service runs):

```
$ sudo service hadoop-yarn-nodemanager start
```

To start the MapReduce JobHistory Server

On the MapReduce JobHistory Server system:

```
$ sudo service hadoop-mapreduce-historyserver start
```

Step 10: Create a Home Directory for each MapReduce User

Create a home directory on the NameNode for each MapReduce user. For example:

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

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

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

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

Step 11: Configure the Hadoop Daemons to Run at Startup

See Configuring Hadoop Daemons to Run at Startup on page 272.

Deploying MapReduce v1 (MRv1) on a Cluster

This topic describes configuration and startup tasks for MRv1 clusters only.

Important:

- If you use Cloudera Manager, do not use these command-line instructions.
- Do not run MRv1 and YARN on the same set of hosts at the same time. This will degrade performance and may result in an unstable cluster deployment. To deploy YARN instead, see Deploying MapReduce v2 (YARN) on a Cluster on page 263. If you have installed CDH 5 from tarballs, the default deployment is YARN.

1. Make sure you have configured and deployed HDFS.
2. Configure the JobTracker's RPC server.
   a. Open the `mapred-site.xml` file in the custom directory you created when you copied the Hadoop configuration.
   b. Specify the hostname and (optionally) port of the JobTracker's RPC server, in the form <host><port>. The default value is `local`. With the default value, JobTracker runs on demand when you run a MapReduce job. Do not try to start the JobTracker yourself in this case. If you specify the host other than local, use the hostname (for example mynamenode) not the IP address.

      For example:

```
<property>
 <name>mapred.job.tracker</name>
 <value>jobtracker-host.company.com:8021</value>
</property>
```

> **Note:** For instructions on configuring a highly available JobTracker, see MapReduce (MRv1) JobTracker High Availability.

3. Configure local storage directories for use by MRv1 daemons.
   a. Open the `mapred-site.xml` file in the custom directory you created when you copied the Hadoop configuration.
   b. Edit the `mapred.local.dir` property to specify the directories where the TaskTracker will store temporary data and intermediate map output files while running MapReduce jobs. Cloudera recommends that you specify a directory on each of the JBOD mount points: `/data/1/mapred/local` through `/data/N/mapred/local`. For example:

```
<property>
 <name>mapred.local.dir</name>
 <value>/data/1/mapred/local,/data/2/mapred/local,/data/3/mapred/local</value>
</property>
```

   c. Create the `mapred.local.dir` local directories:

```
$ sudo mkdir -p /data/1/mapred/local /data/2/mapred/local /data/3/mapred/local
/data/4/mapred/local
```

   d. Configure the owner of the `mapred.local.dir` directory to be the `mapred` user:

```
$ sudo chown -R mapred:hadoop /data/1/mapred/local /data/2/mapred/local
/data/3/mapred/local /data/4/mapred/local
```

   .
   e. Set the permissions to `drwxr-xr-x`.
   f. Configure a health check script for DataNode processes.

      Because a TaskTracker that has few functioning local directories will not perform well, Cloudera recommends configuring a health script that checks if the DataNode process is running (if configured as described under Configuring DataNodes to Tolerate Local Storage Directory Failure, the DataNode will shut down after the configured number of directory failures). Here is an example health script that exits if the DataNode process is not running:

```
#!/bin/bash
if ! jps | grep -q DataNode ; then
 echo ERROR: datanode not up
fi
```

      In practice, the `dfs.data.dir` and `mapred.local.dir` are often configured on the same set of disks, so a disk failure will result in the failure of both a `dfs.data.dir` and `mapred.local.dir`.

      For more information, go to the section titled "Configuring the Node Health Check Script" in the Apache cluster setup documentation.

   g. Set the `mapreduce.jobtracker.restart.recover` property to `true`. This ensures that running jobs that fail because of a system crash or hardware failure are re-run when the JobTracker restarts. A recovered job has the following properties:

      - It will have the same job ID as when it was submitted.
      - It will run under the same user as the original job.
      - It will write to the same output directory as the original job, overwriting any previous output.
      - It will show as RUNNING on the JobTracker web page after you restart the JobTracker.

   h. Repeat for each TaskTracker.

4. Configure a health check script for DataNode processes.

Because a TaskTracker that has few functioning local directories will not perform well, Cloudera recommends configuring a health script that checks if the DataNode process is running (if configured as described under Configuring DataNodes to Tolerate Local Storage Directory Failure on page 256, the DataNode will shut down after the configured number of directory failures). The following is an example health script that exits if the DataNode process is not running:

```
#!/bin/bash
if ! jps | grep -q DataNode ; then
 echo ERROR: datanode not up
fi
```

For more information, go to the section titled "Configuring the Node Health Check Script" in the Apache cluster setup documentation.

5. Configure JobTracker recovery.

   Set the property `mapreduce.jobtracker.restart.recover` to `true` in `mapred-site.xml`.

   JobTracker ensures that running jobs that fail because of a system crash or hardware failure are re-run when the JobTracker restarts. A recovered job has the following properties:

   - It will have the same job ID as when it was submitted.
   - It will run under the same user as the original job.
   - It will write to the same output directory as the original job, overwriting any previous output.
   - It will show as RUNNING on the JobTracker web page after you restart the JobTracker.

6. Create MapReduce `/var` directories:

```
sudo -u hdfs hadoop fs -mkdir -p /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
sudo -u hdfs hadoop fs -chmod 1777 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
sudo -u hdfs hadoop fs -chown -R mapred /var/lib/hadoop-hdfs/cache/mapred
```

7. Verify the HDFS file structure:

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

You should see:

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

8. Create and configure the `mapred.system.dir` directory in HDFS. The HDFS directory specified by the `mapred.system.dir` parameter (by default `${hadoop.tmp.dir}/mapred/system` and configure it to be owned by the `mapred` user.

   To create the directory in its default location:

```
$ sudo -u hdfs hadoop fs -mkdir /tmp/mapred/system
$ sudo -u hdfs hadoop fs -chown mapred:hadoop /tmp/mapred/system
```

Important: If you create the `mapred.system.dir` directory in a different location, specify that path in the `conf/mapred-site.xml` file.

When starting up, MapReduce sets the permissions for the `mapred.system.dir` directory to drwx------, assuming the user `mapred` owns that directory.

9. Start MapReduce by starting the TaskTracker and JobTracker services.
   - On each TaskTracker system:

```
$ sudo service hadoop-0.20-mapreduce-tasktracker start
```

   - On the JobTracker system:

```
$ sudo service hadoop-0.20-mapreduce-jobtracker start
```

10. Create a home directory for each MapReduce user. On the NameNode, enter:

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

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

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

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

11. Set `HADOOP_MAPRED_HOME`.

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

Set this environment variable for each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop in an MRv1 installation.

12. Configure the Hadoop daemons to start automatically. For more information, see Configuring Hadoop Daemons to Run at Startup on page 272.

Configuring Hadoop Daemons to Run at Startup

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

Important:

Make sure you are not trying to run MRv1 and YARN on the same set of nodes at the same time. This is not recommended; it will degrade your performance and may result in an unstable MapReduce cluster deployment.

To start the Hadoop daemons at boot time and on restarts, enable their `init` scripts on the systems on which the services will run, using the `chkconfig` tool. See Configuring init to Start Core Hadoop System Services.

Non-core services can also be started at boot time; after you install the non-core components, see Configuring init to Start Non-core Hadoop System Services for instructions.

Installing CDH 5 Components

In a new installation, you should install and deploy CDH before proceeding to install the components listed below. See Installing the Latest CDH 5 Release on page 223 and Deploying CDH 5 on a Cluster on page 248.

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

CDH 5 Components

Use the following sections to install or upgrade CDH 5 components:

- Crunch Installation on page 273
- Flume Installation on page 275
- HBase Installation on page 282
- HCatalog Installation on page 312
- Hive Installation on page 331
- HttpFS Installation on page 338
- Hue Installation on page 341
- Impala Installation on page 317
- KMS Installation and Upgrade on page 362
- Mahout Installation on page 368
- Oozie Installation on page 370
- Pig Installation on page 386
- Search Installation on page 389
- Sentry Installation on page 395
- Snappy Installation on page 396
- Spark Installation on page 397
- Sqoop 1 Installation on page 398
- Sqoop 2 Installation on page 403
- Whirr Installation on page 411
- ZooKeeper Installation

See also the instructions for installing or updating LZO.

Crunch Installation

The Apache Crunch™ project develops and supports Java APIs that simplify the process of creating data pipelines on top of Apache Hadoop. The Crunch APIs are modeled after FlumeJava, which is the library that Google uses for building data pipelines on top of their own implementation of MapReduce.

The Apache Crunch Java library provides a framework for writing, testing, and running MapReduce pipelines. Its goal is to make pipelines that are composed of many user-defined functions simple to write, easy to test, and efficient to run. Running on top of Hadoop MapReduce and Apache Spark, the Apache Crunch library is a simple Java API for tasks like joining and data aggregation that are tedious to implement on plain MapReduce. The APIs are especially useful when processing data that does not fit naturally into relational model, such as time series, serialized object formats like protocol buffers or Avro records, and HBase rows and columns. For Scala users, there is the Scrunch API, which is built on top of the Java APIs and includes a REPL (read-eval-print loop) for creating MapReduce pipelines.

The following sections describe how to install Crunch:

Crunch Prerequisites

- A supported operating system.
- Oracle JDK.

Crunch Packaging

The packaging options for installing Crunch are:

- RPM packages
- Debian packages

There are two Crunch packages:

- `crunch`: provides all the functionality of crunch allowing users to create data pipelines over execution engines like MapReduce, Spark, and so on.
- `crunch-doc`: the documentation package.

> **Note:** Crunch is also available as a parcel, included with the CDH 5 parcel. If you install CDH 5 with Cloudera Manager, Crunch will be installed automatically.

Installing and Upgrading Crunch

To install the Crunch packages:

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install or upgrade Crunch on a Red Hat system:

```
$ sudo yum install crunch
```

To install or upgrade Crunch on a SLES system:

```
$ sudo zypper install crunch
```

To install or upgrade Crunch on an Ubuntu or Debian system:

```
$ sudo apt-get install crunch
```

To use the Crunch documentation:

The Crunch docs are bundled in a `crunch-doc` package that should be installed separately.

```
$ sudo apt-get install crunch-doc
```

The contents of this package are saved under `/usr/share/doc/crunch*`.

After a package installation, the Crunch jars can be found in `/usr/lib/crunch`.

If you installed CDH 5 through Cloudera Manager, the CDH 5 parcel includes Crunch and the jars are installed automatically as part of the CDH 5 installation. By default the jars will be found in `/opt/cloudera/parcels/CDH/lib/crunch`.

Crunch Documentation

For more information about Crunch, see the following documentation:

- Getting Started with Crunch
- Apache Crunch User Guide

Flume Installation

Apache Flume is a distributed, reliable, and available system for efficiently collecting, aggregating and moving large amounts of log data from many different sources to a centralized datastore.

> **Note:**
>
> To install Flume using Cloudera Manager, see Managing Flume.

Upgrading Flume

Use the instructions that follow to upgrade Flume.

> **Note: Running Services**
>
> Use the `service` command to start, stop, and restart CDH components, 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).

Upgrading Flume from an Earlier CDH 5 release

These instructions assume that you are upgrading Flume as part of an upgrade to the latest CDH 5 release, and have already performed the steps in Upgrading from an Earlier CDH 5 Release to the Latest Release.

To upgrade Flume from an earlier CDH 5 release, install the new version of Flume using one of the methods described below: Installing the Flume RPM or Debian Packages on page 276 or Installing the Flume Tarball on page 276.

> **Important: Configuration files**
>
> - If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
> - If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

Flume Packaging

The are the available packaging options for installing Flume:

- Tarball (.tar.gz)
- RPM packages
- Debian packages

Installing the Flume Tarball

The Flume tarball is a self-contained package containing everything needed to use Flume on a Unix-like system. To install Flume from the tarball, you unpack it in the appropriate directory.

> **Note:**
>
> The tarball does not come with any scripts suitable for running Flume as a service or daemon. This makes the tarball distribution appropriate for *ad hoc* installations and preliminary testing, but a more complete installation is provided by the binary RPM and Debian packages.

To install the Flume tarball on Linux-based systems:

1. Run the following commands, replacing the `(component_version)` with the current version numbers for Flume and CDH.

```
$ cd /usr/local/lib
$ sudo tar -zxvf <path_to_flume-ng-(Flume_version)-cdh(CDH_version).tar.gz>
$ sudo mv flume-ng-(Flume_version)-cdh(CDH_version) flume-ng
```

   For example,

```
$ cd /usr/local/lib
$ sudo tar -zxvf <path_to_flume-ng-1.4.0-cdh5.0.0.tar.gz>
$ sudo mv flume-ng-1.4.0-cdh5.0.0 flume-ng
```

2. To complete the configuration of a tarball installation, you must set your `PATH` variable to include the `bin/` subdirectory of the directory where you installed Flume. For example:

```
$ export PATH=/usr/local/lib/flume-ng/bin:$PATH
```

Installing the Flume RPM or Debian Packages

Installing the Flume RPM and Debian packages is more convenient than installing the Flume tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations
- Handle daemon startup and shutdown.

The Flume RPM and Debian packages consist of three packages:

- `flume-ng` — Everything you need to run Flume
- `flume-ng-agent` — Handles starting and stopping the Flume agent as a service
- `flume-ng-doc` — Flume documentation

All Flume installations require the common code provided by `flume-ng`.

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install Flume on Ubuntu and other Debian systems:

```
$ sudo apt-get install flume-ng
```

To install Flume On RHEL-compatible systems:

```
$ sudo yum install flume-ng
```

To install Flume on SLES systems:

```
$ sudo zypper install flume-ng
```

You might also want Flume to run automatically on start-up. To do this, install the Flume agent.

To install the Flume agent so Flume starts automatically on Ubuntu and other Debian systems:

```
$ sudo apt-get install flume-ng-agent
```

To install the Flume agent so Flume starts automatically on Red Hat-compatible systems:

```
$ sudo yum install flume-ng-agent
```

To install the Flume agent so Flume starts automatically on SLES systems:

```
$ sudo zypper install flume-ng-agent
```

To install the documentation:

To install the documentation on Ubuntu and other Debian systems:

```
$ sudo apt-get install flume-ng-doc
```

To install the documentation on RHEL-compatible systems:

```
$ sudo yum install flume-ng-doc
```

To install the documentation on SLES systems:

```
$ sudo zypper install flume-ng-doc
```

Flume Configuration

Flume 1.x provides a template configuration file for `flume.conf` called `conf/flume-conf.properties.template` and a template for `flume-env.sh` called `conf/flume-env.sh.template`.

1. Copy the Flume template property file `conf/flume-conf.properties.template` to `conf/flume.conf`, then edit it as appropriate.

```
$ sudo cp conf/flume-conf.properties.template conf/flume.conf
```

   This is where you define your sources, sinks, and channels, and the flow within an agent. By default, the properties file is configured to work out of the box using a sequence generator source, a logger sink, and a memory channel. For information on configuring agent flows in Flume 1.x, as well as more details about the supported sources, sinks and channels, see the documents listed under Viewing the Flume Documentation.

2. Optionally, copy the template flume-env.sh file `conf/flume-env.sh.template` to `conf/flume-env.sh`.

```
$ sudo cp conf/flume-env.sh.template conf/flume-env.sh
```

The `flume-ng` executable looks for a file named `flume-env.sh` in the `conf` directory, and sources it if it finds it. Some use cases for using `flume-env.sh` are to specify a bigger heap size for the Flume agent, or to specify debugging or profiling options using JAVA_OPTS when developing your own custom Flume NG components, such as sources and sinks. If you do not make any changes to this file, then you need not perform the copy as it is empty by default.

Verifying the Flume Installation

At this point, you should have everything necessary to run Flume, and the `flume-ng` command should be in your $PATH. You can test this by running:

```
$ flume-ng help
```

You should see something similar to this:

```
Usage: /usr/bin/flume-ng <command> [options]...

commands:
  help                  display this help text
  agent                 run a Flume agent
  avro-client           run an avro Flume client
  version               show Flume version info

global options:
  --conf,-c <conf>      use configs in <conf> directory
  --classpath,-C <cp>   append to the classpath
  --dryrun,-d           do not actually start Flume, just print the command
  --Dproperty=value     sets a JDK system property value

agent options:
  --conf-file,-f <file> specify a config file (required)
  --name,-n <name>      the name of this agent (required)
  --help,-h             display help text

avro-client options:
  --rpcProps,-P <file>  RPC client properties file with server connection params
  --host,-H <host>      hostname to which events will be sent (required)
  --port,-p <port>      port of the avro source (required)
  --dirname <dir>       directory to stream to avro source
  --filename,-F <file>  text file to stream to avro source [default: std input]
  --headerFile,-R <file> headerFile containing headers as key/value pairs on each new
line
  --help,-h             display help text

  Either --rpcProps or both --host and --port must be specified.

Note that if <conf> directory is specified, then it is always included first
in the classpath.
```

Note:

If Flume is not found and you installed Flume from a tarball, make sure that `$FLUME_HOME/bin` is in your $PATH.

Running Flume

If Flume is installed using an RPM or Debian package, or managed by Cloudera Manager, you can use the following commands to start, stop, and restart the Flume agent using `init` scripts:

```
$ sudo service flume-ng-agent <start | stop | restart>
```

You can also run the agent in the foreground directly by using the `flume-ng agent` command:

```
$ /usr/bin/flume-ng agent -c <config-dir> -f <config-file> -n <agent-name>
```

For example:

```
$ /usr/bin/flume-ng agent -c /etc/flume-ng/conf -f /etc/flume-ng/conf/flume.conf -n
agent
```

Files Installed by the Flume RPM and Debian Packages

Resource	Location	Notes
Configuration Directory	`/etc/flume-ng/conf`	
Configuration File	`/etc/flume-ng/conf/flume.conf`	This configuration will be picked-up by the flume agent startup script.
Template of User Customizable Configuration File	`/etc/flume-ng/conf/flume-conf.properties.template`	Contains a sample config. To use this configuration you should copy this file onto `/etc/flume-ng/conf/flume.conf` and then modify as appropriate
Template of User Customizable environment file	`/etc/flume-ng/conf/flume-env.sh.template`	If you want modify this file, copy it first and modify the copy
Daemon Log Directory	`/var/log/flume-ng`	Contains log files generated by flume agent
Default Flume Home	`/usr/lib/flume-ng`	Provided by RPMS and DEBS
Flume Agent startup script	`/etc/init.d/flume-ng-agent`	Provided by RPMS and DEBS
Recommended tar.gz Flume Home	`/usr/local/lib/flume-ng`	Recommended but installation dependent
Flume Wrapper Script	`/usr/bin/flume-ng`	Called by the Flume Agent startup script
Flume Agent configuration file	`/etc/default/flume-ng-agent`	Allows you to specify non-default values for the agent name and for the configuration file location

Supported Sources, Sinks, and Channels

The following tables list the only currently-supported sources, sinks, and channels. For more information, including information on developing custom components, see the documents listed under Viewing the Flume Documentation.

Sources

Type	Description	Implementation Class
avro	Avro Netty RPC event source. Listens on Avro port and receives events from external Avro streams.	AvroSource
netcat	Netcat style TCP event source. Listens on a given port and turns each line of text into an event.	NetcatSource
seq	Monotonically incrementing sequence generator event source	SequenceGeneratorSource

Type	Description	Implementation Class
exec	Run a long-lived Unix process and read from stdout.	ExecSource
syslogtcp	Reads syslog data and generates flume events. Creates a new event for a string of characters separated by carriage return (\n).	SyslogTcpSource
syslogudp	Reads syslog data and generates flume events. Treats an entire message as a single event.	SyslogUDPSource
org.apache.flume.source.avroLegacy.AvroLegacySource	Allows the Flume 1.x agent to receive events from Flume 0.9.4 agents over avro rpc.	AvroLegacySource
org.apache.flume.source.thriftLegacy.ThriftLegacySource	Allows the Flume 1.x agent to receive events from Flume 0.9.4 agents over thrift rpc.	ThriftLegacySource
org.apache.flume.source.StressSource	Mainly for testing purposes. Not meant for production use. Serves as a continuous source of events where each event has the same payload.	StressSource
org.apache.flume.source.scribe.ScribeSource	Scribe event source. Listens on Scribe port and receives events from Scribe.	ScribeSource
multiport_syslogtcp	Multi-port capable version of the SyslogTcpSource.	MultiportSyslogTCPSource
spooldir	Ingests data by placing files to be ingested into a "spooling" directory on disk.	SpoolDirectorySource
http	Accepts Flume events by HTTP POST and GET. GET should be used for experimentation only.	HTTPSource
org.apache.flume.source.jms.JMSSource	Reads messages from a JMS destination such as a queue or topic.	JMSSource
org.apache.flume.agent.embedded.EmbeddedSource	Used only by the Flume embedded agent. See Flume Developer Guide for more details.	EmbeddedSource
org.apache.flume.source.kafka.KafkaSource	Streams data from Kafka to Hadoop or from any Flume source to Kafka.	KafkaSource
org.apache.flume.source.taildir.TaildirSource	Watches specified files, and tails them in near real-time when it detects appends to these files. • This source is reliable and does not miss data, even when the tailing files rotate.	TaildirSource

Type	Description	Implementation Class
	• It periodically writes the last read position of each file in a position file using the JSON format. • If Flume is stopped or down for some reason, it can restart tailing from the position written in the existing position file. • It can add event headers to each tailing file group.	

Sinks

Type	Description	Implementation Class
logger	Log events at INFO level using configured logging subsystem (log4j by default)	LoggerSink
avro	Sink that invokes a pre-defined Avro protocol method for all events it receives (when paired with an avro source, forms tiered collection)	AvroSink
hdfs	Writes all events received to HDFS (with support for rolling, bucketing, HDFS-200 append, and more)	HDFSEventSink
file_roll	Writes all events received to one or more files.	RollingFileSink
org.apache.flume.hbase.HBaseSink	A simple sink that reads events from a channel and writes them synchronously to HBase. The AsyncHBaseSink is recommended. See Importing Data Into HBase.	HBaseSink
org.apache.flume.sink.hbase.AsyncHBaseSink	A simple sink that reads events from a channel and writes them asynchronously to HBase. This is the recommended HBase sink, but it does not support Kerberos. See Importing Data Into HBase.	AsyncHBaseSink
org.apache.flume.sink.solr.morphline.MorphlineSolrSink	Extracts and transforms data from Flume events, and loads it into Apache Solr servers. See the section on MorphlineSolrSink in the Flume User Guide listed under Viewing the Flume Documentation on page 282.	MorphlineSolrSink
org.apache.flume.sink.kafka.KafkaSink	Used to send data to Kafka from a Flume source. You can use the Kafka sink in addition to Flume sinks such as HBase or HDFS.	KafkaSink

Channels

Type	Description	Implementation Class
memory	In-memory, fast, non-durable event transport	MemoryChannel
jdbc	JDBC-based, durable event transport (Derby-based)	JDBCChannel
file	File-based, durable event transport	FileChannel
org.apache.flume.channel.kafka.KafkaChannel	Use the Kafka channel: • To write to Hadoop directly from Kafka without using a source. • To write to Kafka directly from Flume sources without additional buffering. • As a reliable and highly available channel for any source/sink combination.	KafkaChannel

Providing for Disk Space Usage

It's important to provide plenty of disk space for any Flume File Channel. The largest consumers of disk space in the File Channel are the data logs. You can configure the File Channel to write these logs to multiple data directories. The following space will be consumed by default in each data directory:

- Current log file (up to 2 GB)
- Last log file (up to 2 GB)
- Pending delete log file (up to 2 GB)

Events in the queue could cause many more log files to be written, each of them up 2 GB in size by default.

You can configure both the maximum log file size (`MaxFileSize`) and the directories the logs will be written to (`DataDirs`) when you configure the File Channel; see the File Channel section of the Flume User Guide for details.

Viewing the Flume Documentation

For additional Flume documentation, see the Flume User Guide and the Flume Developer Guide.

For additional information about Flume, see the Apache Flume wiki.

HBase Installation

Apache HBase provides large-scale tabular storage for Hadoop using the Hadoop Distributed File System (HDFS). Cloudera recommends installing HBase in a standalone mode before you try to run it on a whole cluster.

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

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).

Use the following sections to install, update, and configure HBase:

Next Steps

After installing and configuring HBase, check out the following topics about using HBase:

- Importing Data Into HBase
- Writing Data to HBase
- Reading Data from HBase

New Features and Changes for HBase in CDH 5

CDH 5.0.x and 5.1.x each include major upgrades to HBase. Each of these upgrades provides exciting new features, as well as things to keep in mind when upgrading from a previous version.

For new features and changes introduced in older CDH 5 releases, skip to CDH 5.1 HBase Changes or CDH 5.0.x HBase Changes.

CDH 5.4 HBase Changes

CDH 5.4 introduces HBase 1.0, which represents a major upgrade to HBase. This upgrade introduces new features and moves some features which were previously marked as experimental to fully supported status. This overview provides information about the most important features, how to use them, and where to find out more information. Cloudera appreciates your feedback about these features.

Highly-Available Read Replicas

CDH 5.4 introduces highly-available read replicas. Using read replicas, clients can request, on a per-read basis, a read result using a new consistency model, timeline consistency, rather than strong consistency. The read request is sent to the RegionServer serving the region, but also to any RegionServers hosting replicas of the region. The client receives the read from the fastest RegionServer to respond, and receives an indication of whether the response was from the primary RegionServer or from a replica. See HBase Read Replicas for more details.

MultiWAL Support

CDH 5.4 introduces support for writing multiple write-ahead logs (MultiWAL) on a given RegionServer, allowing you to increase throughput when a region writes to the WAL. See Configuring HBase MultiWAL Support.

Medium-Object (MOB) Storage

CDH 5.4 introduces a mechanism for storing objects between 100 KB and 10 MB in a default configuration, or *medium objects*, directly in HBase. Storing objects up to 50 MB is possible with additional configuration. Previously, storing these medium objects directly in HBase could degrade performance due to write amplification caused by splits and compactions.

MOB storage requires HFile V3.

doAs Impersonation for the Thrift Gateway

Prior to CDH 5.4, the Thrift gateway could be configured to authenticate to HBase on behalf of the client as a static user. A new mechanism, doAs Impersonation, allows the client to authenticate as any HBase user on a per-call basis for a higher level of security and flexibility.

Namespace Create Authorization

Prior to CDH 5.4, only global admins could create namespaces. Now, a Namespace Create authorization can be assigned to a user, who can then create namespaces.

Authorization to List Namespaces and Tables

Prior to CDH 5.4, authorization checks were not performed on list namespace and list table operations, so you could list the names of tany tables or namespaces, regardless of your authorization. In CDH 5.4, you are not able to list namespaces or tables you do not have authorization to access.

Crunch API Changes for HBase

In CDH 5.4, Apache Crunch adds the following API changes for HBase:

- `HBaseTypes.cells()` was added to support serializing HBase Cell objects.
- Each method of `HFileUtils` now supports `PCollection<C extends Cell>`, which includes both `PCollection<KeyValue>` and `PCollection<Cell>`, on their method signatures.
- `HFileTarget`, `HBaseTarget`, and `HBaseSourceTarget` each support any subclass of `Cell` as an output type. `HFileSource` and `HBaseSourceTarget` still return `KeyValue` as the input type for backward-compatibility with existing Crunch pipelines.

ZooKeeper 3.4 Is Required

HBase 1.0 requires ZooKeeper 3.4.

HBase API Changes for CDH 5.4

CDH 5.4.0 introduces HBase 1.0, which includes some major changes to the HBase APIs. Besides the changes listed above, some APIs have been deprecated in favor of new public APIs.

- The `HConnection` API is deprecated in favor of `Connection`.
- The HConnectionManager API is deprecated in favor of `ConnectionFactory`.
- The HTable API is deprecated in favor of `Table`.
- The `HTableAdmin` API is deprecated in favor of `Admin`.

HBase 1.0 API Example

```
Configuration conf = HBaseConfiguration.create();
try (Connection connection = ConnectionFactory.createConnection(conf)) {
  try (Table table = connection.getTable(TableName.valueOf(tablename)) {
    // use table as needed, the table returned is lightweight
  }
}
```

CDH 5.3 HBase Changes

CDH 5.4 introduces HBase 0.98.6, which represents a minor upgrade to HBase. CDH 5.3 provides `checkAndMutate(RowMutations)`, in addition to existing support for atomic `checkAndPut` as well as `checkAndDelete` operations on individual rows (HBASE-11796).

`SlabCache` Has Been Deprecated

`SlabCache`, which was marked as deprecated in CDH 5.2, has been removed in CDH 5.3. To configure the BlockCache, see Configuring the HBase BlockCache.

`checkAndMutate(RowMutations)` API

CDH 5.3 provides `checkAndMutate(RowMutations)`, in addition to existing support for atomic `checkAndPut` as well as `checkAndDelete` operations on individual rows (HBASE-11796).

CDH 5.2 HBase Changes

CDH 5.2 introduces HBase 0.98.6, which represents a minor upgrade to HBase. This upgrade introduces new features and moves some features which were previously marked as experimental to fully supported status. This overview

provides information about the most important features, how to use them, and where to find out more information. Cloudera appreciates your feedback about these features.

JAVA_HOME must be set in your environment.

HBase now requires JAVA_HOME to be set in your environment. If it is not set, HBase will fail to start and an error will be logged. If you use Cloudera Manager, this is set automatically. If you use CDH without Cloudera Manager, JAVA_HOME should be set up as part of the overall installation. See Java Development Kit Installation on page 58 for instructions on setting JAVA_HOME, as well as other JDK-specific instructions.

The default value for `hbase.hstore.flusher.count` has increased from 1 to 2.

The default value for `hbase.hstore.flusher.count` has been increased from one thread to two. This new configuration can improve performance when writing to HBase under some workloads. However, for high IO workloads two flusher threads can create additional contention when writing to HDFS. If after upgrading to CDH 5.2. you see an increase in flush times or performance degradation, lowering this value to 1 is recommended. Use the RegionServer's advanced configuration snippet for `hbase-site.xml` if you use Cloudera Manager, or edit the file directly otherwise.

The default value for `hbase.hregion.memstore.block.multiplier` has increased from 2 to 4.

The default value for `hbase.hregion.memstore.block.multiplier` has increased from 2 to 4 to improve both throughput and latency. If you experience performance degradation due to this change, change the value setting to 2, using the RegionServer's advanced configuration snippet for `hbase-site.xml` if you use Cloudera Manager, or by editing the file directly otherwise.

SlabCache is deprecated, and BucketCache is now the default block cache.

CDH 5.1 provided full support for the BucketCache block cache. CDH 5.2 deprecates usage of SlabCache in favor of BucketCache. To configure BucketCache, see BucketCache Block Cache on page 287

Changed Syntax of `user_permissions` Shell Command

The pattern-matching behavior for the `user_permissions` HBase Shell command has changed. Previously, either of the following two commands would return permissions of all known users in HBase:

```
hbase> user_permissions '*'
```

```
hbase> user_permissions '.*'
```

The first variant is no longer supported. The second variant is the only supported operation and also supports passing in other Java regular expressions.

New Properties for IPC Configuration

If the Hadoop configuration is read after the HBase configuration, Hadoop's settings can override HBase's settings if the names of the settings are the same. To avoid the risk of override, HBase has renamed the following settings (by prepending 'hbase.') so that you can set them independent of your setting for Hadoop. If you do not use the HBase-specific variants, the Hadoop settings will be used. If you have not experienced issues with your configuration, there is no need to change it.

Hadoop Configuration Property	New HBase Configuration Property
ipc.server.listen.queue.size	hbase.ipc.server.listen.queue.size
ipc.server.max.callqueue.size	hbase.ipc.server.max.callqueue.size
ipc.server.max.callqueue.length	hbase.ipc.server.max.callqueue.length
ipc.server.read.threadpool.size	hbase.ipc.server.read.threadpool.size
ipc.server.tcpkeepalive	hbase.ipc.server.tcpkeepalive
ipc.server.tcpnodelay	hbase.ipc.server.tcpnodelay

Hadoop Configuration Property	New HBase Configuration Property
ipc.client.call.purge.timeout	hbase.ipc.client.call.purge.timeout
ipc.client.connection.maxidletime	hbase.ipc.client.connection.maxidletime
ipc.client.idlethreshold	hbase.ipc.client.idlethreshold
ipc.client.kill.max	hbase.ipc.client.kill.max

Snapshot Manifest Configuration

Snapshot manifests were previously a feature included in HBase in CDH 5 but not in Apache HBase. They are now included in Apache HBase 0.98.6. To use snapshot manifests, you now need to set `hbase.snapshot.format.version` to 2 in `hbase-site.xml`. This is the default for HBase in CDH 5.2, but the default for Apache HBase 0.98.6 is 1. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. The new snapshot code can read both version 1 and 2. However, if you use version 2, you will not be able to read these snapshots on HBase versions prior to 0.98.

Not using manifests (setting `hbase.snapshot.format.version` to 1) can cause excess load on the NameNode and impact performance.

Tags

Tags, which allow metadata to be stored in HFiles alongside cell data, are a feature of HFile version 3, are needed for per-cell access controls and visibility labels. Tags were previously considered an experimental feature but are now fully supported.

Per-Cell Access Controls

Per-cell access controls were introduced as an experimental feature in CDH 5.1 and are fully supported in CDH 5.2. You must use HFile version 3 to use per-cell access controls. For more information about access controls, see Per-Cell Access Controls on page 290.

Experimental Features

Warning: These features are still considered experimental. Experimental features are not supported and Cloudera does not recommend using them in production environments or with important data.

Visibility Labels

You can now specify a list of visibility labels, such as CONFIDENTIAL, TOPSECRET, or PUBLIC, at the cell level. You can associate users with these labels to enforce visibility of HBase data. These labels can be grouped into complex expressions using logical operators `&`, `|`, and `!` (AND, OR, NOT). A given user is associated with a set of visibility labels, and the policy for associating the labels is pluggable. A coprocessor, `org.apache.hadoop.hbase.security.visibility.DefaultScanLabelGenerator`, checks for visibility labels on cells that would be returned by a Get or Scan and drops the cells that a user is not authorized to see, before returning the results. The same coprocessor saves visibility labels as tags, in the HFiles alongside the cell data, when a Put operation includes visibility labels. You can specify custom implementations of `ScanLabelGenerator` by setting the property `hbase.regionserver.scan.visibility.label.generator.class` to a comma-separated list of classes in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise.

No labels are configured by default. You can add a label to the system using either the `VisibilityClient#addLabels()` API or the `add_label` shell command. Similar APIs and shell commands are provided for deleting labels and assigning them to users. Only a user with superuser access (the `hbase.superuser` access level) can perform these operations.

To assign a visibility label to a cell, you can label the cell using the API method `Mutation#setCellVisibility(new CellVisibility(<labelExp>));`. An API is provided for managing visibility labels, and you can also perform many of the operations using HBase Shell.

Previously, visibility labels could not contain the symbols `&`, `|`, `!`, `(` and `)`, but this is no longer the case.

For more information about visibility labels, see the Visibility Labels section of the *Apache HBase Reference Guide*.

If you use visibility labels along with access controls, you must ensure that the Access Controller is loaded before the Visibility Controller in the list of coprocessors. This is the default configuration. See HBASE-11275.

Visibility labels are an **experimental** feature introduced in CDH 5.1, and still experimental in CDH 5.2.

Transparent Server-Side Encryption

Transparent server-side encryption can now be enabled for both HFiles and write-ahead logs (WALs), to protect their contents at rest. To configure transparent encryption, first create an encryption key, then configure the appropriate settings in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. See the Transparent Encryption section in the *Apache HBase Reference Guide* for more information.

Transparent server-side encryption is an **experimental** feature introduced in CDH 5.1, and still experimental in CDH 5.2.

Stripe Compaction

Stripe compaction is a compaction scheme that segregates the data inside a region by row key, creating "stripes" of data which are visible within the region but transparent to normal operations. This striping improves read performance in common scenarios and greatly reduces variability by avoiding large or inefficient compactions.

Configuration guidelines and more information are available at Stripe Compaction.

To configure stripe compaction for a single table from within the HBase shell, use the following syntax.

```
alter <table>, CONFIGURATION => {<setting> => <value>}
 Example: alter 'orders', CONFIGURATION => {'hbase.store.stripe.fixed.count' => 10}
```

To configure stripe compaction for a column family from within the HBase shell, use the following syntax.

```
alter <table>, {NAME => <column family>, CONFIGURATION => {<setting => <value>}}
 Example: alter 'logs', {NAME => 'blobs', CONFIGURATION =>
{'hbase.store.stripe.fixed.count' => 10}}
```

Stripe compaction is an **experimental** feature in CDH 5.1, and still experimental in CDH 5.2.

CDH 5.1 HBase Changes

CDH 5.1 introduces HBase 0.98, which represents a major upgrade to HBase. This upgrade introduces several new features, including a section of features which are considered experimental and should not be used in a production environment. This overview provides information about the most important features, how to use them, and where to find out more information. Cloudera appreciates your feedback about these features.

In addition to HBase 0.98, Cloudera has pulled in changes from HBASE-10883, HBASE-10964, HBASE-10823, HBASE-10916, and HBASE-11275. Implications of these changes are detailed below and in the Release Notes.

BucketCache Block Cache

A new offheap BlockCache implementation, BucketCache, was introduced as an experimental feature in CDH 5 Beta 1, and is now fully supported in CDH 5.1. BucketCache can be used in either of the following two configurations:

- As a CombinedBlockCache with both onheap and offheap caches.
- As an L2 cache for the default onheap LruBlockCache

BucketCache requires less garbage-collection than SlabCache, which is the other offheap cache implementation in HBase. It also has many optional configuration settings for fine-tuning. All available settings are documented in the API documentation for CombinedBlockCache. Following is a simple example configuration.

1. First, edit `hbase-env.sh` and set `-XX:MaxDirectMemorySize` to the total size of the desired onheap plus offheap, in this case, 5 GB (but expressed as `5G`). To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise.

```
-XX:MaxDirectMemorySize=5G
```

2. Next, add the following configuration to `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise. This configuration uses 80% of the `-XX:MaxDirectMemorySize` (4 GB) for offheap, and the remainder (1 GB) for onheap.

```
<property>
  <name>hbase.bucketcache.ioengine</name>
  <value>offheap</value>
</property>
<property>
  <name>hbase.bucketcache.percentage.in.combinedcache</name>
  <value>0.8</value>
</property>
<property>
  <name>hbase.bucketcache.size</name>
  <value>5120</value>
</property>
```

3. Restart or rolling restart your cluster for the configuration to take effect.

Access Control for EXEC Permissions

A new access control level has been added to check whether a given user has EXEC permission. This can be specified at the level of the cluster, table, row, or cell.

To use EXEC permissions, perform the following procedure.

- Install the AccessController coprocessor either as a system coprocessor or on a table as a table coprocessor.
- Set the `hbase.security.exec.permission.checks` configuration setting in `hbase-site.xml` to `true` To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise..

For more information on setting and revoking security permissions, see the Access Control section of the *Apache HBase Reference Guide*.

Reverse Scan API

A reverse scan API has been introduced. This allows you to scan a table in reverse. Previously, if you wanted to be able to access your data in either direction, you needed to store the data in two separate tables, each ordered differently. This feature was implemented in HBASE-4811.

To use the reverse scan feature, use the new `Scan.setReversed(boolean reversed)` API. If you specify a startRow and stopRow, to scan in reverse, the startRow needs to be lexicographically after the stopRow. See the Scan API documentation for more information.

MapReduce Over Snapshots

You can now run a MapReduce job over a snapshot from HBase, rather than being limited to live data. This provides the ability to separate your client-side work load from your live cluster if you need to run resource-intensive MapReduce jobs and can tolerate using potentially-stale data. You can either run the MapReduce job on the snapshot within HBase, or export the snapshot and run the MapReduce job against the exported file.

Running a MapReduce job on an exported file outside of the scope of HBase relies on the permissions of the underlying filesystem and server, and bypasses ACLs, visibility labels, and encryption that may otherwise be provided by your HBase cluster.

A new API, `TableSnapshotInputFormat`, is provided. For more information, see TableSnapshotInputFormat.

MapReduce over snapshots was introduced in CDH 5.0.

Stateless Streaming Scanner over REST

A new stateless streaming scanner is available over the REST API. Using this scanner, clients do not need to restart a scan if the REST server experiences a transient failure. All query parameters are specified during the REST request. Query parameters include `startrow`, `endrow`, `columns`, `starttime`, `endtime`, `maxversions`, `batchtime`, and limit. Following are a few examples of using the stateless streaming scanner.

Scan the entire table, return the results in JSON.

```
curl -H "Accept: application/json" https://localhost:8080/ExampleScanner/*
```

Scan the entire table, return the results in XML.

```
curl -H "Content-Type: text/xml" https://localhost:8080/ExampleScanner/*
```

Scan only the first row.

```
curl -H "Content-Type: text/xml" \
https://localhost:8080/ExampleScanner/*?limit=1
```

Scan only specific columns.

```
curl -H "Content-Type: text/xml" \
https://localhost:8080/ExampleScanner/*?columns=a:1,b:1
```

Scan for rows between starttime and endtime.

```
curl -H "Content-Type: text/xml" \
https://localhost:8080/ExampleScanner/*?starttime=1389900769772\
&endtime=1389900800000
```

Scan for a given row prefix.

```
curl -H "Content-Type: text/xml" https://localhost:8080/ExampleScanner/test*
```

For full details about the stateless streaming scanner, see the API documentation for this feature.

Delete Methods of Put Class Now Use Constructor Timestamps

The `Delete()` methods of the `Put` class of the HBase Client API previously ignored the constructor's timestamp, and used the value of `HConstants.LATEST_TIMESTAMP`. This behavior was different from the behavior of the `add()` methods. The `Delete()` methods now use the timestamp from the constructor, creating consistency in behavior across the `Put` class. See HBASE-10964.

Experimental Features

> **Warning:** These features are still considered experimental. Experimental features are not supported and Cloudera does not recommend using them in production environments or with important data.

Visibility Labels

You can now specify a list of visibility labels, such as CONFIDENTIAL, TOPSECRET, or PUBLIC, at the cell level. You can associate users with these labels to enforce visibility of HBase data. These labels can be grouped into complex expressions using logical operators `&`, `|`, and `!` (AND, OR, NOT). A given user is associated with a set of visibility labels, and the policy for associating the labels is pluggable. A coprocessor, `org.apache.hadoop.hbase.security.visibility.DefaultScanLabelGenerator`, checks for visibility labels on cells that would be returned by a Get or Scan and drops the cells that a user is not authorized to see, before returning the results. The same coprocessor saves visibility labels as tags, in the HFiles alongside the cell data, when a Put operation includes visibility labels. You can specify custom implementations of `ScanLabelGenerator` by setting the property `hbase.regionserver.scan.visibility.label.generator.class` to a comma-separated list of classes.

No labels are configured by default. You can add a label to the system using either the `VisibilityClient#addLabels()` API or the `add_label` shell command. Similar APIs and shell commands are provided for deleting labels and assigning them to users. Only a user with superuser access (the `hbase.superuser` access level) can perform these operations.

To assign a visibility label to a cell, you can label the cell using the API method `Mutation#setCellVisibility(new CellVisibility(<labelExp>));`.

Visibility labels and request authorizations cannot contain the symbols `&`, `|`, `!`, `(` and `)` because they are reserved for constructing visibility expressions. See HBASE-10883.

For more information about visibility labels, see the Visibility Labels section of the *Apache HBase Reference Guide*.

If you use visibility labels along with access controls, you must ensure that the Access Controller is loaded before the Visibility Controller in the list of coprocessors. This is the default configuration. See HBASE-11275.

To use per-cell access controls or visibility labels, you must use HFile version 3. To enable HFile version 3, add the following to `hbase-site.xml`, using an advanced code snippet if you use Cloudera Manager, or directly to the file if your deployment is unmanaged.. Changes will take effect after the next major compaction.

```
<property>
  <name>hfile.format.version</name>
  <value>3</value>
</property>
```

Visibility labels are an **experimental** feature introduced in CDH 5.1.

Per-Cell Access Controls

You can now specify access control levels at the per-cell level, as well as at the level of the cluster, table, or row.

A new parent class has been provided, which encompasses `Get`, `Scan`, and `Query`. This change also moves the `getFilter` and `setFilter` methods of `Get` and `Scan` to the common parent class. Client code may need to be recompiled to take advantage of per-cell ACLs. See the Access Control section of the *Apache HBase Reference Guide* for more information.

The ACLS for cells having timestamps in the future are not considered for authorizing the pending mutation operations. See HBASE-10823.

If you use visibility labels along with access controls, you must ensure that the Access Controller is loaded before the Visibility Controller in the list of coprocessors. This is the default configuration.

To use per-cell access controls or visibility labels, you must use HFile version 3. To enable HFile version 3, add the following to `hbase-site.xml`, using an advanced code snippet if you use Cloudera Manager, or directly to the file if your deployment is unmanaged.. Changes will take effect after the next major compaction.

```
<property>
  <name>hfile.format.version</name>
  <value>3</value>
</property>
```

Per-cell access controls are an **experimental** feature introduced in CDH 5.1.

Transparent Server-Side Encryption

Transparent server-side encryption can now be enabled for both HFiles and write-ahead logs (WALs), to protect their contents at rest. To configure transparent encryption, first create an encryption key, then configure the appropriate settings in `hbase-site.xml` . See the Transparent Encryption section in the *Apache HBase Reference Guide* for more information.

Transparent server-side encryption is an **experimental** feature introduced in CDH 5.1.

Stripe Compaction

Stripe compaction is a compaction scheme that segregates the data inside a region by row key, creating "stripes" of data which are visible within the region but transparent to normal operations. This striping improves read performance in common scenarios and greatly reduces variability by avoiding large or inefficient compactions.

Configuration guidelines and more information are available at Stripe Compaction.

To configure stripe compaction for a single table from within the HBase shell, use the following syntax.

```
alter <table>, CONFIGURATION => {<setting> => <value>}
 Example: alter 'orders', CONFIGURATION => {'hbase.store.stripe.fixed.count' => 10}
```

To configure stripe compaction for a column family from within the HBase shell, use the following syntax.

```
alter <table>, {NAME => <column family>, CONFIGURATION => {<setting => <value>}}
 Example: alter 'logs', {NAME => 'blobs', CONFIGURATION =>
{'hbase.store.stripe.fixed.count' => 10}}
```

Stripe compaction is an **experimental** feature in CDH 5.1.

Distributed Log Replay

After a RegionServer fails, its failed region is assigned to another RegionServer, which is marked as "recovering" in ZooKeeper. A SplitLogWorker directly replays edits from the WAL of the failed RegionServer to the region at its new location. When a region is in "recovering" state, it can accept writes but no reads (including Append and Increment), region splits or merges. Distributed Log Replay extends the distributed log splitting framework. It works by directly replaying WAL edits to another RegionServer instead of creating `recovered.edits` files.

Distributed log replay provides the following advantages over using the current distributed log splitting functionality on its own.

- It eliminates the overhead of writing and reading a large number of `recovered.edits` files. It is not unusual for thousands of recovered.edits files to be created and written concurrently during a RegionServer recovery. Many small random writes can degrade overall system performance.
- It allows writes even when a region is in recovering state. It only takes seconds for a recovering region to accept writes again.

To enable distributed log replay, set `hbase.master.distributed.log.replay` to `true`. You must also enable HFile version 3. Distributed log replay is unsafe for rolling upgrades.

Distributed log replay is an **experimental** feature in CDH 5.1.

CDH 5.0.x HBase Changes

HBase in CDH 5.0.x is based on the Apache HBase 0.96 release. When upgrading to CDH 5.0.x, keep the following in mind.

Upgrade is Not Reversible

The upgrade from CDH 4 HBase to CDH 5 HBase is irreversible and requires HBase to be shut down completely. Executing the upgrade script reorganizes existing HBase data stored on HDFS into new directory structures, converts HBase 0.90 HFile v1 files to the improved and optimized HBase 0.96 HFile v2 file format, and rewrites the `hbase.version` file. This upgrade also removes transient data stored in ZooKeeper during the conversion to the new data format.

These changes were made to reduce the impact in future major upgrades. Previously HBase used brittle custom data formats and this move shifts HBase's RPC and persistent data to a more evolvable Protocol Buffer data format.

API Changes

The HBase User API (Get, Put, Result, Scanner etc; see Apache HBase API documentation) has evolved and attempts have been made to make sure the HBase Clients are source code compatible and thus should recompile without needing any source code modifications. This cannot be guaranteed however, since with the conversion to Protocol Buffers (ProtoBufs), some relatively obscure APIs have been removed. Rudimentary efforts have also been made to preserve recompile compatibility with advanced APIs such as Filters and Coprocessors. These advanced APIs are still evolving and our guarantees for API compatibility are weaker here.

For information about changes to custom filters, see Custom Filters.

As of 0.96, the User API has been marked and all attempts at compatibility in future versions will be made. A version of the javadoc that only contains the User API can be found here.

HBase Metrics Changes

HBase provides a metrics framework based on JMX beans. Between HBase 0.94 and 0.96, the metrics framework underwent many changes. Some beans were added and removed, some metrics were moved from one bean to another, and some metrics were renamed or removed. Click here to download the CSV spreadsheet which provides a mapping.

Custom Filters

If you used custom filters written for HBase 0.94, you need to recompile those filters for HBase 0.96. The custom filter must be altered to fit with the newer interface that uses protocol buffers. Specifically two new methods, `toByteArray(…)` and `parseFrom(…)`, which are detailed in detailed in the Filter API. These should be used instead of the old methods `write(…)` and `readFields(…)`, so that protocol buffer serialization is used. To see what changes were required to port one of HBase's own custom filters, see the Git commit that represented porting the `SingleColumnValueFilter` filter.

Checksums

When you upgrade to CDH 5, HBase checksums are now turned on by default. With this configuration, HBase reads data and then verifies the checksums. Checksum verification inside HDFS will be switched off. If the HBase-checksum verification fails, then the HDFS checksums are used instead for verifying data that is being read from storage. Once you turn on HBase checksums, you will not be able to roll back to an earlier HBase version.

You should see a modest performance gain after setting `hbase.regionserver.checksum.verify` to true for data that is not already present in the RegionServer's block cache.

To enable or disable checksums, modify the following configuration properties in `hbase-site.xml`. To edit the configuration, use an Advanced Configuration Snippet if you use Cloudera Manager, or edit the file directly otherwise.

```
<property>
  <name>hbase.regionserver.checksum.verify</name>
  <value>true</value>
  <description>
      If set to  true, HBase will read data and then verify checksums  for
      hfile blocks. Checksum verification inside HDFS will be switched off.
      If the hbase-checksum verification fails, then it will  switch back to
      using HDFS checksums.
  </description>
</property>
```

The default value for the `hbase.hstore.checksum.algorithm` property has also changed to `CRC32`. Previously, Cloudera advised setting it to `NULL` due to performance issues which are no longer a problem.

```
<property>
   <name>hbase.hstore.checksum.algorithm</name>
   <value>CRC32</value>
   <description>
     Name of an algorithm that is used to compute checksums. Possible values
     are NULL, CRC32, CRC32C.
   </description>
 </property>
```

Upgrading HBase

Note: To see which version of HBase is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Important: Before you start, make sure you have read and understood the previous section, New Features and Changes for HBase in CDH 5 on page 283, and check the Known Issues in CDH 5 and Incompatible Changes and Limitations for HBase.

Coprocessors and Custom JARs

When upgrading HBase from one major version to another, you must recompile coprocessors and custom JARs *after* the upgrade.

Never rely on HBase directory layout on disk.

The HBase directory layout is an implementation detail and is subject to change. Do not rely on the directory layout for client or administration functionality. Instead, access HBase using the supported APIs.

Upgrading HBase from a Lower CDH 5 Release

Important: Rolling upgrade is not supported between a CDH 5 Beta release and a CDH 5 GA release. Cloudera recommends using Cloudera Manager if you need to do rolling upgrades.

To upgrade HBase from a lower CDH 5 release, proceed as follows.

The instructions that follow assume that you are upgrading HBase as part of an upgrade to the latest CDH 5 release, and have already performed the steps under Upgrading from an Earlier CDH 5 Release to the Latest Release.

During a rolling upgrade from CDH 5.0.x to CDH 5.4.x the HBase Master UI will display the URLs to the old HBase RegionServers using an incorrect info port number. Once the rolling upgrade completes the HBase master UI will use the correct port number.

Step 1: Perform a Graceful Cluster Shutdown

Note: Upgrading using rolling restart is not supported.

To shut HBase down gracefully:

1. Stop the Thrift server and clients, then stop the cluster.
   a. Stop the Thrift server and clients:

```
sudo service hbase-thrift stop
```

   b. Stop the cluster by shutting down the master and the RegionServers:
      - Use the following command on the master node:

```
sudo service hbase-master stop
```

      - Use the following command on each node hosting a RegionServer:

```
sudo service hbase-regionserver stop
```

2. Stop the ZooKeeper Server:

```
$ sudo service zookeeper-server stop
```

Step 2: Install the new version of HBase

> **Note:** You may want to take this opportunity to upgrade ZooKeeper, but you do not *have* to upgrade Zookeeper before upgrading HBase; the new version of HBase will run with the older version of Zookeeper. For instructions on upgrading ZooKeeper, see Upgrading ZooKeeper from an Earlier CDH 5 Release on page 416.

To install the new version of HBase, follow directions in the next section, HBase Installation on page 282.

> **Important: Configuration files**
>
> - If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
> - If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

Installing HBase

To install HBase On RHEL-compatible systems:

```
$ sudo yum install hbase
```

To install HBase on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase
```

To install HBase on SLES systems:

```
$ sudo zypper install hbase
```

> **Note:** See also Starting HBase in Standalone Mode on page 303, Configuring HBase in Pseudo-Distributed Mode on page 305, and Deploying HBase on a Cluster on page 307 for more information on configuring HBase for different modes.

To list the installed files on Ubuntu and Debian systems:

```
$ dpkg -L hbase
```

To list the installed files on RHEL and SLES systems:

```
$ rpm -ql hbase
```

You can see that the HBase package has been configured to conform to the Linux Filesystem Hierarchy Standard. (To learn more, run `man hier`).

You are now ready to enable the server daemons you want to use with Hadoop. You can also enable Java-based client access by adding the JAR files in `/usr/lib/hbase/` and `/usr/lib/hbase/lib/` to your Java class path.

Configuration Settings for HBase

This section contains information on configuring the Linux host and HDFS for HBase.

Using DNS with HBase

HBase uses the local hostname to report its IP address. Both forward and reverse DNS resolving should work. If your server has multiple interfaces, HBase uses the interface that the primary hostname resolves to. If this is insufficient, you can set `hbase.regionserver.dns.interface` in the `hbase-site.xml` file to indicate the primary interface. To work properly, this setting requires that your cluster configuration is consistent and every host has the same network interface configuration. As an alternative, you can set `hbase.regionserver.dns.nameserver` in the `hbase-site.xml` file to use a different DNS name server than the system-wide default.

Using the Network Time Protocol (NTP) with HBase

The clocks on cluster members must be synchronized for your cluster to function correctly. Some skew is tolerable, but excessive skew could generate odd behaviors. Run NTP or another clock synchronization mechanism on your cluster. If you experience problems querying data or unusual cluster operations, verify the system time. For more information about NTP, see the NTP website.

Setting User Limits for HBase

Because HBase is a database, it opens many files at the same time. The default setting of 1024 for the maximum number of open files on most Unix-like systems is insufficient. Any significant amount of loading will result in failures and cause error message such as `java.io.IOException...(Too many open files)` to be logged in the HBase or HDFS log files. For more information about this issue, see the Apache HBase Book. You may also notice errors such as:

```
2010-04-06 03:04:37,542 INFO org.apache.hadoop.hdfs.DFSClient: Exception
increateBlockOutputStream java.io.EOFException
2010-04-06 03:04:37,542 INFO org.apache.hadoop.hdfs.DFSClient: Abandoning block
blk_-6935524980745310745_1391901
```

Another setting you should configure is the number of processes a user is permitted to start. The default number of processes is typically 1024. Consider raising this value if you experience `OutOfMemoryException` errors.

Configuring ulimit for HBase Using Cloudera Manager

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

1. Go to the HBase service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Master** or **Scope** > **RegionServer**.
4. Locate the **Maximum Process File Descriptors** property or search for it by typing its name in the Search box.
5. Edit the property value.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager.

6. Click **Save Changes** to commit the changes.
7. Restart the role.
8. Restart the service.

Configuring ulimit for HBase Using the Command Line

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

Cloudera recommends increasing the maximum number of file handles to more than 10,000. Increasing the file handles for the user running the HBase process is an operating system configuration, not an HBase configuration. A common mistake is to increase the number of file handles for a particular user when HBase is running as a different user. HBase prints the ulimit it is using on the first line in the logs. Make sure that it is correct.

To change the maximum number of open files for a user, use the `ulimit -n` command while logged in as that user.

To set the maximum number of processes a user can start, use the `ulimit -u` command. You can also use the `ulimit` command to set many other limits. For more information, see the online documentation for your operating system, or the output of the `man ulimit` command.

To make the changes persistent, add the command to the user's Bash initialization file (typically `~/.bash_profile` or `~/.bashrc`). Alternatively, you can configure the settings in the `Pluggable Authentication Module (PAM)` configuration files if your operating system uses PAM and includes the `pam_limits.so` shared library.

Configuring ulimit using Pluggable Authentication Modules Using the Command Line

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

If you are using `ulimit`, you must make the following configuration changes:

1. In the `/etc/security/limits.conf` file, add the following lines, adjusting the values as appropriate. This assumes that your HDFS user is called `hdfs` and your HBase user is called `hbase`.

```
hdfs   -       nofile  32768
hdfs   -       nproc   2048
hbase  -       nofile  32768
hbase  -       nproc   2048
```

Note:

- Only the `root` user can edit this file.
- If this change does not take effect, check other configuration files in the `/etc/security/limits.d/` directory for lines containing the `hdfs` or `hbase` user and the `nofile` value. Such entries may be overriding the entries in `/etc/security/limits.conf`.

To apply the changes in `/etc/security/limits.conf` on Ubuntu and Debian systems, add the following line in the `/etc/pam.d/common-session` file:

```
session required  pam_limits.so
```

For more information on the `ulimit` command or per-user operating system limits, refer to the documentation for your operating system.

Using `dfs.datanode.max.transfer.threads` with HBase

A Hadoop HDFS DataNode has an upper bound on the number of files that it can serve at any one time. The upper bound is controlled by the `dfs.datanode.max.transfer.threads` property (the property is spelled in the code exactly as shown here). Before loading, make sure you have configured the value for `dfs.datanode.max.transfer.threads` in the `conf/hdfs-site.xml` file (by default found in `/etc/hadoop/conf/hdfs-site.xml`) to at least `4096` as shown below:

```
<property>
  <name>dfs.datanode.max.transfer.threads</name>
  <value>4096</value>
</property>
```

Restart HDFS after changing the value for `dfs.datanode.max.transfer.threads`. If the value is not set to an appropriate value, strange failures can occur and an error message about exceeding the number of transfer threads will be added to the DataNode logs. Other error messages about missing blocks are also logged, such as:

```
06/12/14 20:10:31 INFO hdfs.DFSClient: Could not obtain block
blk_XXXXXXXXXXXXXXXXXXXXXX_YYYYYYYY from any node:
java.io.IOException: No live nodes contain current block. Will get new block locations
 from namenode and retry...
```

Note: The property `dfs.datanode.max.transfer.threads` is a HDFS 2 property which replaces the deprecated property `dfs.datanode.max.xcievers`.

Configuring BucketCache in HBase

The default `BlockCache` implementation in HBase is `CombinedBlockCache`, and the default off-heap `BlockCache` is `BucketCache`. `SlabCache` is now deprecated. See Configuring the HBase BlockCache for information about configuring the `BlockCache` using Cloudera Manager or the command line.

Configuring Encryption in HBase

It is possible to encrypt the HBase root directory within HDFS, using HDFS Transparent Encryption. This provides an additional layer of protection in case the HDFS filesystem is compromised.

If you use this feature in combination with bulk-loading of HFiles, you must configure `hbase.bulkload.staging.dir` to point to a location within the same encryption zone as the HBase root directory. Otherwise, you may encounter errors such as:

```
org.apache.hadoop.ipc.RemoteException(java.io.IOException):
/tmp/output/f/5237a8430561409bb641507f0c531448 can't be moved into an encryption zone.
```

You can also choose to only encrypt specific column families, which encrypts individual HFiles while leaving others unencrypted, using HBase Transparent Encryption at Rest. This provides a balance of data security and performance.

Configuring Cell Level TTL in HBase

Cell TTLs are defined internally as Cell Tags. Cell Tags are only supported for HFile Version 3 and higher, therefore HFile Version 3 must be set to enable Cell TTL use. For more information, see Enabling HFile Version 3 Using Clouder Manager.

Using Hedged Reads

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

Note:

To enable hedged reads for HBase, edit the `hbase-site.xml` file on each server. Set `dfs.client.hedged.read.threadpool.size` to the number of threads to dedicate to running hedged threads, and set the `dfs.client.hedged.read.threshold.millis` configuration property to the number of milliseconds to wait before starting a second read against a different block replica. Set `dfs.client.hedged.read.threadpool.size` to 0 or remove it from the configuration to disable the feature. After changing these properties, restart your cluster.

The following is an example configuration for hedged reads for HBase.

```
<property>
  <name>dfs.client.hedged.read.threadpool.size</name>
  <value>20</value>  <!-- 20 threads -->
</property>
<property>
  <name>dfs.client.hedged.read.threshold.millis</name>
  <value>10</value>  <!-- 10 milliseconds -->
</property>
```

Accessing HBase by using the HBase Shell

After you have started HBase, you can access the database in an interactive way by using the HBase Shell, which is a command interpreter for HBase which is written in Ruby. Always run HBase administrative commands such as the HBase Shell, `hbck`, or bulk-load commands as the HBase user (typically `hbase`).

```
$ hbase shell
```

HBase Shell Overview

- To get help and to see all available commands, use the `help` command.
- To get help on a specific command, use `help "command"`. For example:

```
hbase> help "create"
```

- To remove an attribute from a table or column family or reset it to its default value, set its value to `nil`. For example, use the following command to remove the `KEEP_DELETED_CELLS` attribute from the `f1` column of the `users` table:

```
hbase> alter 'users', { NAME => 'f1', KEEP_DELETED_CELLS => nil }
```

- To exit the HBase Shell, type `quit`.

Setting Virtual Machine Options for HBase Shell

HBase in CDH 5.2 and higher allows you to set variables for the virtual machine running HBase Shell, by using the HBASE_SHELL_OPTS environment variable. This example sets several options in the virtual machine.

```
$ HBASE_SHELL_OPTS="-verbose:gc -XX:+PrintGCApplicationStoppedTime -XX:+PrintGCDateStamps
      -XX:+PrintGCDetails -Xloggc:$HBASE_HOME/logs/gc-hbase.log" ./bin/hbase shell
```

Scripting with HBase Shell

CDH 5.2 and higher include non-interactive mode. This mode allows you to use HBase Shell in scripts, and allow the script to access the exit status of the HBase Shell commands. To invoke non-interactive mode, use the `-n` or `--non-interactive` switch. This small example script shows how to use HBase Shell in a Bash script.

```
#!/bin/bash
echo 'list' | hbase shell -n
status=$?
```

```
if [ $status -ne 0 ]; then
  echo "The command may have failed."
fi
```

Successful HBase Shell commands return an exit status of 0. However, an exit status other than 0 does not necessarily indicate a failure, but should be interpreted as unknown. For example, a command may succeed, but while waiting for the response, the client may lose connectivity. In that case, the client has no way to know the outcome of the command. In the case of a non-zero exit status, your script should check to be sure the command actually failed before taking further action.

CDH 5.7 and higher include the `get_splits` command, which returns the split points for a given table:

```
hbase> get_splits 't2'
Total number of splits = 5

=> ["", "10", "20", "30", "40"]
```

You can also write Ruby scripts for use with HBase Shell. Example Ruby scripts are included in the `hbase-examples/src/main/ruby/` directory.

Configuring HBase Online Merge

CDH 5 supports online merging of regions. HBase splits big regions automatically but does not support merging small regions automatically. To complete an online merge of two regions of a table, use the HBase shell to issue the online merge command. By default, both regions to be merged should be neighbors; that is, one end key of a region should be the start key of the other region. Although you can "force merge" any two regions of the same table, this can create overlaps and is not recommended.

The Master and RegionServer both participate in online merges. When the request to merge is sent to the Master, the Master moves the regions to be merged to the same RegionServer, usually the one where the region with the higher load resides. The Master then requests the RegionServer to merge the two regions. The RegionServer processes this request locally. Once the two regions are merged, the new region will be online and available for server requests, and the old regions are taken offline.

For merging two consecutive regions, use the following command:

```
hbase> merge_region 'ENCODED_REGIONNAME', 'ENCODED_REGIONNAME'
```

For merging regions that are not adjacent, passing `true` as the third parameter forces the merge.

```
hbase> merge_region 'ENCODED_REGIONNAME', 'ENCODED_REGIONNAME', true
```

Note: This command is slightly different from other region operations. You must pass the encoded region name (`ENCODED_REGIONNAME`), not the full region name . The encoded region name is the hash suffix on region names. For example, if the region name is `TestTable,0094429456,1289497600452.527db22f95c8a9e0116f0cc13c680396`, the encoded region name portion is `527db22f95c8a9e0116f0cc13c680396`.

Configuring RegionServer Grouping

You can use RegionServer Grouping (rsgroup) to impose strict isolation between RegionServers by partitioning RegionServers into distinct groups. You can use HBase Shell commands to define and manage RegionServer Grouping.

You must first create an rsgroup before you can add RegionServers to it. Once you have created an rsgroup, you can move your HBase tables into this rsgroup so that only the RegionServers in the same rsgroup can host the regions of the table.

Note: RegionServers and tables can only belong to one rsgroup at a time. By default, all the tables and RegionServers belong to the `default` rsgroup.

A custom balancer implementation tracks assignments per rsgroup and moves regions to the relevant RegionServers in that rsgroup. The rsgroup information is stored in a regular HBase table, and a ZooKeeper-based read-only cache is used at cluster bootstrap time.

Enabling RegionServer Grouping using Cloudera Manager

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

You must use Cloudera Manager to enable RegionServer Grouping before you can define and manage rsgroups.

1. Go to the HBase service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Master**.
4. Locate the **HBase Coprocessor Master Classes** property or search for it by typing its name in the Search box.
5. Add the following property value: `org.apache.hadoop.hbase.rsgroup.RSGroupAdminEndpoint`.
6. Locate the **Master Advanced Configuration Snippet (Safety Valve) for hbase-site.xml** property or search for it by typing its name in the Search box.
7. Click View as XML and add the following property:

```
<property>
  <name>hbase.master.loadbalancer.class</name>
  <value>org.apache.hadoop.hbase.rsgroup.RSGroupBasedLoadBalancer</value>
</property>
```

8. Click **Save Changes** to commit the changes.
9. Restart the role.
10. Restart the service.

Configuring RegionServer Grouping

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

When you add a new rsgroup, you are creating an rsgroup other than the default group. To configure a rsgroup, in the HBase shell:

1. Type `add_rsgroup '*mygroup*'`.
2. Add RegionServers and tables to this rsgroup using the commands:

```
hbase> move_servers_rsgroup ‘mygroup',['server1:port','server2:port']
hbase> move_tables_rsgroup 'mygroup',['table1','table2']
```

3. Run the `balance_rsgroup` command if the tables are slow to migrate to the group's dedicated server.

Note: The term rsgroup refers to servers in a cluster with only the hostname and port. It does not make use of the HBase ServerName type identifying RegionServers (hostname + port + start time) to distinguish RegionServer instances.

Monitor RegionServer Grouping

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

You can monitor the status of the commands using the Tables tab on the HBase Master UI home page. If you click on a table name, you can see the RegionServers that are deployed.

You must manually align the RegionServers referenced in rsgroups with the actual state of nodes in the cluster that is active and running.

Removing a RegionServer from RegionServer Grouping

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

You can remove a RegionServer by moving it to the `default` rsgroup. Edits made using shell commands to all rsgroups, except the `default` rsgroup, are persisted to the system `hbase:rsgroup` table. If an rsgroup references a decommissioned RegionServer, then the rsgroup should be updated to undo the reference.

1. Move the RegionServer to the default rsgroup using the command:

```
hbase> move_servers_rsgroup 'default',['server1:port']
```

2. Check the list of RegionServers in your rsgourp to ensure that that the RegionServer is successfully removed using the command:

```
hbase> get_rsgroup 'mygroup'
```

The default rsgroup's RegionServer list mirrors the current state of the cluster. If you shut down a RegionServer that was part of the `default` rsgroup, and then run the `get_rsgroup 'default'` command to list its content in the shell, the server is no longer listed. If you move the offline server from the non-default rsgroup to `default`, it will not show in the `default` list; the server will just be removed from the list.

Enabling ACL for RegionServer Grouping

Minimum Required Role: Full Administrator

You need to be a Global Admin to manage rsgroups if authorization is enabled.

To enable ACL, add the following to the `hbase-site.xml` file, and then restart your HBase Master server:

```
<property>
  <name>hbase.security.authorization</name>
  <value>true</value>
<property>
```

Best Practices when using RegionServer Grouping

You must keep in mind the following best practices when using rsgroups:

Isolate System Tables

You can either have a system rsgroup where all the system tables are present or just leave the system tables in `default` rsgroup and have all user-space tables in non-`default` rsgroups.

Handling Dead Nodes

You can have a special rsgroup of dead or questionable nodes to help you keep them without running until the nodes are repaired. Be careful when replacing dead nodes in an rsgroup, and ensure there are enough live nodes before you start moving out the dead nodes. You can move the good live nodes first before moving out the dead nodes.

If you have configured a table to be in a rsgroup, but all the RegionServers in that rsgroup die, the tables become unavailable and you can no longer access those tables.

Troubleshooting RegionServer Grouping

If you encounter an issue when using rsgroup, check the Master log to see what is causing the issue. If an rsgroup operation is unresponsive, restart the Master.

For example, if you have not enabled the rsgroup coprocessor endpoint in the Master, and run any of the rsgroup shell commands, you will encounter the following error message:

```
ERROR: org.apache.hadoop.hbase.exceptions.UnknownProtocolException: No registered master
 coprocessor service found for name RSGroupAdminService
    at
org.apache.hadoop.hbase.master.MasterRpcServices.execMasterService(MasterRpcServices.java:604)

    at
```

```
org.apache.hadoop.hbase.shaded.protobuf.generated.MasterProtos$MasterService$2.callBlockingMethod(MasterProtos.java)

    at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:1140)
    at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:133)
    at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:277)
    at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:257)
```

Disabling RegionServer Grouping

When you no longer require rsgroups, you can disable it for your cluster. Removing RegionServer Grouping for a cluster on which it was enabled involves more steps in addition to removing the relevant properties from `hbase-site.xml`. You must ensure that you clean the RegionServer grouping-related metadata so that if the feature is re-enabled in the future, the old metadata will not affect the functioning of the cluster.

To disable RegionServer Grouping:

1. Move all the tables in non-default rsgroups to `default` RegionServer group.

```
#Reassigning table t1 from the non-default group - hbase shell
hbase> move_tables_rsgroup 'default',['t1']
```

2. Move all RegionServers in non-default rsgroups to `default` regionserver group.

```
#Reassigning all the servers in the non-default rsgroup to default - hbase shell
hbase> move_servers_rsgroup
'default',['regionserver1:port','regionserver2:port','regionserver3:port']
```

3. Remove all non-default rsgroups. `default` rsgroup created implicitly does not have to be removed.

```
#removing non-default rsgroup - hbase shell
hbase> remove_rsgroup 'mygroup'
```

4. Remove the changes made in `hbase-site.xml` and restart the cluster.
5. Drop the table `hbase:rsgroup` from HBase.

```
#Through hbase shell drop table hbase:rsgroup
hbase> disable 'hbase:rsgroup'
0 row(s) in 2.6270 seconds
hbase> drop 'hbase:rsgroup'
0 row(s) in 1.2730 seconds
```

6. Remove the znode `rsgroup` from the cluster ZooKeeper using zkCli.sh.

```
#From ZK remove the node /hbase/rsgroup through zkCli.sh
rmr /hbase/rsgroup
```

Configuring the BlockCache

See Configuring the HBase BlockCache.

Configuring the Scanner Heartbeat

See Configuring the HBase Scanner Heartbeat.

Troubleshooting HBase

See Troubleshooting HBase.

Starting HBase in Standalone Mode

> **Note:**
>
> You can skip this section if you are already running HBase in distributed or pseudo-distributed mode.

By default, HBase ships configured for *standalone mode*. In this mode of operation, a single JVM hosts the HBase Master, an HBase RegionServer, and a ZooKeeper quorum peer. HBase stores your data in a location on the local filesystem, rather than using HDFS. Standalone mode is only appropriate for initial testing.

> **Important:**
>
> If you have configured High Availability for the NameNode (HA), you cannot deploy HBase in standalone mode without modifying the default configuration, because both the standalone HBase process and ZooKeeper (required by HA) will try to bind to port 2181. You can configure a different port for ZooKeeper, but in most cases it makes more sense to deploy HBase in distributed mode in an HA cluster.

In order to run HBase in standalone mode, you must install the HBase Master package.

Installing the HBase Master

To install the HBase Master on RHEL-compatible systems:

```
$ sudo yum install hbase-master
```

To install the HBase Master on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-master
```

To install the HBase Master on SLES systems:

```
$ sudo zypper install hbase-master
```

Starting the HBase Master

- On RHEL and SLES systems (using `.rpm` packages) you can now start the HBase Master by using the included service script:

```
$ sudo service hbase-master start
```

- On Ubuntu systems (using Debian packages) the HBase Master starts when the HBase package is installed.

To verify that the standalone installation is operational, visit `http://localhost:60010`. The list of RegionServers at the bottom of the page should include one entry for your local machine.

> **Note:**
>
> Although you have only started the master process, in *standalone* mode this same process is also internally running a RegionServer and a ZooKeeper peer. In the next section, you will break out these components into separate JVMs.

If you see this message when you start the HBase standalone master:

```
Starting Hadoop HBase master daemon: starting master, logging to
/usr/lib/hbase/logs/hbase-hbase-master/cloudera-vm.out
Couldnt start ZK at requested address of 2181, instead got: 2182.  Aborting. Why? Because
```

```
 clients (eg shell) wont be able to find this ZK quorum
hbase-master.
```

you will need to stop the hadoop-zookeeper-server (or zookeeper-server) or uninstall the hadoop-zookeeper-server (or zookeeper) package.

See also Accessing HBase by using the HBase Shell on page 308, Using MapReduce with HBase on page 310 and Troubleshooting HBase on page 310.

Installing and Starting the HBase Thrift Server

To install Thrift on RHEL-compatible systems:

```
$ sudo yum install hbase-thrift
```

To install Thrift on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-thrift
```

To install Thrift on SLES systems:

```
$ sudo zypper install hbase-thrift
```

You can now use the `service` command to start the Thrift server:

```
$ sudo service hbase-thrift start
```

Installing and Configuring HBase REST

To install HBase REST on RHEL-compatible systems:

```
$ sudo yum install hbase-rest
```

To install HBase REST on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-rest
```

To install HBase REST on SLES systems:

```
$ sudo zypper install hbase-rest
```

You can use the `service` command to run an `init.d` script, `/etc/init.d/hbase-rest`, to start the REST server; for example:

```
$ sudo service hbase-rest start
```

The script starts the server by default on port 8080. This is a commonly used port and so may conflict with other applications running on the same host.

If you need change the port for the REST server, configure it in `hbase-site.xml`, for example:

```
<property>
  <name>hbase.rest.port</name>
  <value>60050</value>
</property>
```

Note:

You can use HBASE_REST_OPTS in `hbase-env.sh` to pass other settings (such as heap size and GC parameters) to the REST server JVM.

Configuring HBase in Pseudo-Distributed Mode

Note: You can skip this section if you are already running HBase in distributed mode, or if you intend to use only standalone mode.

Pseudo-distributed mode differs from *standalone* mode in that each of the component processes run in a separate JVM. It differs from *distributed mode* in that each of the separate processes run on the same server, rather than multiple servers in a cluster. This section also assumes you want to store your HBase data in HDFS rather than on the local filesystem.

Note: Before you start

- This section assumes you have already installed the HBase master and gone through the standalone configuration steps.
- If the HBase master is already running in standalone mode, use the following command to stop it:

```
sudo service hbase-master stop
```

Modifying the HBase Configuration

To enable pseudo-distributed mode, you must first make some configuration changes. Open `/etc/hbase/conf/hbase-site.xml` in your editor of choice, and insert the following XML properties between the `<configuration>` and `</configuration>` tags. The `hbase.cluster.distributed` property directs HBase to start each process in a separate JVM. The `hbase.rootdir` property directs HBase to store its data in an HDFS filesystem, rather than the local filesystem. Be sure to replace `myhost` with the hostname of your HDFS NameNode (as specified by `fs.default.name` or `fs.defaultFS` in your `conf/core-site.xml` file); you may also need to change the port number from the default (8020).

```
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://myhost:8020/hbase</value>
</property>
```

Creating the /hbase Directory in HDFS

Before starting the HBase Master, you need to create the `/hbase` directory in HDFS. The HBase master runs as `hbase:hbase` so it does not have the required permissions to create a top level directory.

To create the /hbase directory in HDFS:

```
$ sudo -u hdfs hadoop fs -mkdir /hbase
$ sudo -u hdfs hadoop fs -chown hbase /hbase
```

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

Enabling Servers for Pseudo-distributed Operation

After you have configured HBase, you must enable the various servers that make up a distributed HBase cluster. HBase uses three required types of servers:

- Installing and Starting ZooKeeper Server
- Starting the HBase Master
- Starting an HBase RegionServer

Installing and Starting ZooKeeper Server

HBase uses ZooKeeper Server as a highly available, central location for cluster management. For example, it allows clients to locate the servers, and ensures that only one master is active at a time. For a small cluster, running a ZooKeeper node collocated with the NameNode is recommended. For larger clusters, contact Cloudera Support for configuration help.

Install and start the ZooKeeper Server in standalone mode by running the commands shown in the Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server on page 417

Starting the HBase Master

After ZooKeeper is running, you can start the HBase master in standalone mode.

```
$ sudo service hbase-master start
```

Starting an HBase RegionServer

The RegionServer is the HBase process that actually hosts data and processes requests. The RegionServer typically runs on all HBase nodes except for the node running the HBase master node.

To enable the HBase RegionServer On RHEL-compatible systems:

```
$ sudo yum install hbase-regionserver
```

To enable the HBase RegionServer on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-regionserver
```

To enable the HBase RegionServer on SLES systems:

```
$ sudo zypper install hbase-regionserver
```

To start the RegionServer:

```
$ sudo service hbase-regionserver start
```

Verifying the Pseudo-Distributed Operation

After you have started ZooKeeper, the Master, and a RegionServer, the pseudo-distributed cluster should be up and running. You can verify that each of the daemons is running using the `jps` tool from the Oracle JDK, which you can obtain from here. If you are running a pseudo-distributed HDFS installation and a pseudo-distributed HBase installation on one machine, `jps` will show the following output:

```
$ sudo jps
32694 Jps
30674 HRegionServer
29496 HMaster
28781 DataNode
28422 NameNode
30348 QuorumPeerMain
```

You should also be able to go to `http://localhost:60010` and verify that the local RegionServer has registered with the Master.

Installing and Starting the HBase Thrift Server

The HBase Thrift Server is an alternative gateway for accessing the HBase server. Thrift mirrors most of the HBase client APIs while enabling popular programming languages to interact with HBase. The Thrift Server is multiplatform and more performant than REST in many situations. Thrift can be run collocated along with the RegionServers, but should not be collocated with the NameNode or the JobTracker. For more information about Thrift, visit http://thrift.apache.org/.

To enable the HBase Thrift Server On RHEL-compatible systems:

```
$ sudo yum install hbase-thrift
```

To enable the HBase Thrift Server on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-thrift
```

To enable the HBase Thrift Server on SLES systems:

```
$ sudo zypper install hbase-thrift
```

To start the Thrift server:

```
$ sudo service hbase-thrift start
```

See also Accessing HBase by using the HBase Shell on page 308, Using MapReduce with HBase on page 310 and Troubleshooting HBase on page 310.

Deploying HBase on a Cluster

After you have HBase running in pseudo-distributed mode, the same configuration can be extended to running on a distributed cluster.

Note: Before you start

This section assumes that you have already installed the HBase Master and the HBase RegionServer and gone through the steps for standalone and pseudo-distributed configuration. You are now about to distribute the processes across multiple hosts; see Choosing Where to Deploy the Processes on page 307.

Choosing Where to Deploy the Processes

For small clusters, Cloudera recommends designating one node in your cluster as the HBase Master node. On this node, you will typically run the HBase Master and a ZooKeeper quorum peer. These master processes may be collocated with the Hadoop NameNode and JobTracker for small clusters.

Designate the remaining nodes as RegionServer nodes. On each node, Cloudera recommends running a RegionServer, which may be collocated with a Hadoop TaskTracker (MRv1) and a DataNode. When co-locating with TaskTrackers, be sure that the resources of the machine are not oversubscribed – it's safest to start with a small number of MapReduce slots and work up slowly.

The HBase Thrift service is light-weight, and can be run on any node in the cluster.

Configuring for Distributed Operation

After you have decided which machines will run each process, you can edit the configuration so that the nodes can locate each other. In order to do so, you should make sure that the configuration files are synchronized across the cluster. Cloudera strongly recommends the use of a configuration management system to synchronize the configuration files, though you can use a simpler solution such as `rsync` to get started quickly.

The only configuration change necessary to move from pseudo-distributed operation to fully-distributed operation is the addition of the ZooKeeper Quorum address in `hbase-site.xml`. Insert the following XML property to configure the nodes with the address of the node where the ZooKeeper quorum peer is running:

```
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>mymasternode</value>
</property>
```

The `hbase.zookeeper.quorum` property is a comma-separated list of hosts on which ZooKeeper servers are running. If one of the ZooKeeper servers is down, HBase will use another from the list. By default, the ZooKeeper service is bound to port 2181. To change the port, add the `hbase.zookeeper.property.clientPort` property to `hbase-site.xml` and set the value to the port you want ZooKeeper to use. In CDH 5.7.0 and higher, you do not need to use hbase.zookeeper.property.clientPort. Instead, you can specify the port along with the hostname for each ZooKeeper host:

```
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>zk1.example.com:2181,zk2.example.com:20000,zk3.example.com:31111</value>
</property>
```

For more information, see this chapter of the Apache HBase Reference Guide.

To start the cluster, start the services in the following order:

1. The ZooKeeper Quorum Peer
2. The HBase Master
3. Each of the HBase RegionServers

After the cluster is fully started, you can view the HBase Master web interface on port 60010 and verify that each of the RegionServer nodes has registered properly with the master.

See also Accessing HBase by using the HBase Shell on page 308, Using MapReduce with HBase on page 310 and Troubleshooting HBase on page 310. For instructions on improving the performance of local reads, see Optimizing Performance in CDH.

Accessing HBase by using the HBase Shell

After you have started HBase, you can access the database in an interactive way by using the HBase Shell, which is a command interpreter for HBase which is written in Ruby. Always run HBase administrative commands such as the HBase Shell, `hbck`, or bulk-load commands as the HBase user (typically `hbase`).

```
$ hbase shell
```

HBase Shell Overview

- To get help and to see all available commands, use the `help` command.
- To get help on a specific command, use `help "command"`. For example:

```
hbase> help "create"
```

- To remove an attribute from a table or column family or reset it to its default value, set its value to `nil`. For example, use the following command to remove the `KEEP_DELETED_CELLS` attribute from the `f1` column of the `users` table:

```
hbase> alter 'users', { NAME => 'f1', KEEP_DELETED_CELLS => nil }
```

- To exit the HBase Shell, type `quit`.

Setting Virtual Machine Options for HBase Shell

HBase in CDH 5.2 and higher allows you to set variables for the virtual machine running HBase Shell, by using the HBASE_SHELL_OPTS environment variable. This example sets several options in the virtual machine.

```
$ HBASE_SHELL_OPTS="-verbose:gc -XX:+PrintGCApplicationStoppedTime -XX:+PrintGCDateStamps
      -XX:+PrintGCDetails -Xloggc:$HBASE_HOME/logs/gc-hbase.log" ./bin/hbase shell
```

Scripting with HBase Shell

CDH 5.2 and higher include non-interactive mode. This mode allows you to use HBase Shell in scripts, and allow the script to access the exit status of the HBase Shell commands. To invoke non-interactive mode, use the `-n` or `--non-interactive` switch. This small example script shows how to use HBase Shell in a Bash script.

```
#!/bin/bash
echo 'list' | hbase shell -n
status=$?
if [ $status -ne 0 ]; then
  echo "The command may have failed."
fi
```

Successful HBase Shell commands return an exit status of 0. However, an exit status other than 0 does not necessarily indicate a failure, but should be interpreted as unknown. For example, a command may succeed, but while waiting for the response, the client may lose connectivity. In that case, the client has no way to know the outcome of the command. In the case of a non-zero exit status, your script should check to be sure the command actually failed before taking further action.

CDH 5.7 and higher include the `get_splits` command, which returns the split points for a given table:

```
hbase> get_splits 't2'
Total number of splits = 5

=> ["", "10", "20", "30", "40"]
```

You can also write Ruby scripts for use with HBase Shell. Example Ruby scripts are included in the `hbase-examples/src/main/ruby/` directory.

Configuring HBase Online Merge

CDH 5 supports online merging of regions. HBase splits big regions automatically but does not support merging small regions automatically. To complete an online merge of two regions of a table, use the HBase shell to issue the online merge command. By default, both regions to be merged should be neighbors; that is, one end key of a region should be the start key of the other region. Although you can "force merge" any two regions of the same table, this can create overlaps and is not recommended.

The Master and RegionServer both participate in online merges. When the request to merge is sent to the Master, the Master moves the regions to be merged to the same RegionServer, usually the one where the region with the higher load resides. The Master then requests the RegionServer to merge the two regions. The RegionServer processes this request locally. Once the two regions are merged, the new region will be online and available for server requests, and the old regions are taken offline.

For merging two consecutive regions, use the following command:

```
hbase> merge_region 'ENCODED_REGIONNAME', 'ENCODED_REGIONNAME'
```

For merging regions that are not adjacent, passing `true` as the third parameter forces the merge.

```
hbase> merge_region 'ENCODED_REGIONNAME', 'ENCODED_REGIONNAME', true
```

Note: This command is slightly different from other region operations. You must pass the encoded region name (`ENCODED_REGIONNAME`), not the full region name . The encoded region name is the hash suffix on region names. For example, if the region name is `TestTable,0094429456,1289497600452.527db22f95c8a9e0116f0cc13c680396`, the encoded region name portion is `527db22f95c8a9e0116f0cc13c680396`.

Using MapReduce with HBase

To run MapReduce jobs that use HBase, you need to add the HBase and Zookeeper JAR files to the Hadoop Java classpath. You can do this by adding the following statement to each job:

```
TableMapReduceUtil.addDependencyJars(job);
```

This distributes the JAR files to the cluster along with your job and adds them to the job's classpath, so that you do not need to edit the MapReduce configuration.

You can find more information about `addDependencyJars` in the documentation listed under Viewing the HBase Documentation on page 312.

When getting an `Configuration` object for a HBase MapReduce job, instantiate it using the `HBaseConfiguration.create()` method.

Troubleshooting HBase

The Cloudera HBase packages have been configured to place logs in `/var/log/hbase`. Cloudera recommends tailing the `.log` files in this directory when you start HBase to check for any error messages or failures.

Table Creation Fails after Installing LZO

If you install LZO after starting the RegionServer, you will not be able to create a table with LZO compression until you re-start the RegionServer.

Why this happens

When the RegionServer starts, it runs CompressionTest and caches the results. When you try to create a table with a given form of compression, it refers to those results. You have installed LZO since starting the RegionServer, so the cached results, which pre-date LZO, cause the create to fail.

What to do

Restart the RegionServer. Now table creation with LZO will succeed.

Thrift Server Crashes after Receiving Invalid Data

The Thrift server may crash if it receives a large amount of invalid data, due to a buffer overrun.

Why this happens

The Thrift server allocates memory to check the validity of data it receives. If it receives a large amount of invalid data, it may need to allocate more memory than is available. This is due to a limitation in the Thrift library itself.

What to do

To prevent the possibility of crashes due to buffer overruns, use the framed and compact transport protocols. These protocols are disabled by default, because they may require changes to your client code. The two options to add to your hbase-site.xml are `hbase.regionserver.thrift.framed` and `hbase.regionserver.thrift.compact`. Set each of these to `true`, as in the XML below. You can also specify the maximum frame size, using the `hbase.regionserver.thrift.framed.max_frame_size_in_mb` option.

```
<property>
  <name>hbase.regionserver.thrift.framed</name>
  <value>true</value>
</property>
<property>
```

```
  <name>hbase.regionserver.thrift.framed.max_frame_size_in_mb</name>
  <value>2</value>
</property>
<property>
  <name>hbase.regionserver.thrift.compact</name>
  <value>true</value>
</property>
```

HBase is using more disk space than expected.

HBase StoreFiles (also called HFiles) store HBase row data on disk. HBase stores other information on disk, such as write-ahead logs (WALs), snapshots, data that would otherwise be deleted but would be needed to restore from a stored snapshot.

Warning: The following information is provided to help you troubleshoot high disk usage only. Do not edit or remove any of this data outside the scope of the HBase APIs or HBase Shell, or your data is very likely to become corrupted.

Table 21: HBase Disk Usage

Location	Purpose	Troubleshooting Notes
`/hbase/.snapshots`	Contains one subdirectory per snapshot.	To list snapshots, use the HBase Shell command `list_snapshots`. To remove a snapshot, use `delete_snapshot`.
`/hbase/.archive`	Contains data that would otherwise have been deleted (either because it was explicitly deleted or expired due to TTL or version limits on the table) but that is required to restore from an existing snapshot.	To free up space being taken up by excessive archives, delete the snapshots that refer to them. Snapshots never expire so data referred to by them is kept until the snapshot is removed. Do not remove anything from `/hbase/.archive` manually, or you will corrupt your snapshots.
`/hbase/.logs`	Contains HBase WAL files that are required to recover regions in the event of a RegionServer failure.	WALs are removed when their contents are verified to have been written to StoreFiles. Do not remove them manually. If the size of any subdirectory of `/hbase/.logs/` is growing, examine the HBase server logs to find the root cause for why WALs are not being processed correctly.
`/hbase/logs/.oldWALs`	Contains HBase WAL files that have already been written to disk. A HBase maintenance thread removes them periodically based on a TTL.	To tune the length of time a WAL stays in the `.oldWALs` before it is removed, configure the `hbase.master.logcleaner.ttl` property, which defaults to 60000 milliseconds, or 1 hour.
`/hbase/.logs/.corrupt`	Contains corrupted HBase WAL files.	Do not remove corrupt WALs manually. If the size of any subdirectory of `/hbase/.logs/` is growing, examine the HBase server logs to find the root cause for why

Location	Purpose	Troubleshooting Notes
		WALs are not being processed correctly.

Viewing the HBase Documentation

For additional HBase documentation, see https://archive.cloudera.com/cdh5/cdh/5/hbase/.

HCatalog Installation

As of CDH 5, HCatalog is part of Apache Hive.

HCatalog is a table and storage management layer for Hadoop that makes the same table information available to Hive, Pig, MapReduce, and Sqoop. Table definitions are maintained in the Hive metastore, which HCatalog requires. WebHCat allows you to access HCatalog using an HTTP (REST style) interface.

This page explains how to install and configure HCatalog and WebHCat. For Sqoop, see Sqoop-HCatalog Integration in the Sqoop User Guide.

Configuring HCatalog Using Cloudera Manager

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

1. Go to the Hive service by clicking **Clusters** > **Hive**.
2. Select the Hive **Instances** tab.
3. Add a WebHCat server role:
   a. Click **Add Role Instances**.
   b. Click **Select hosts** under WebHCat Server.
   c. Select the host on which you want the WebHCat server; this adds a **WHCS** icon.
   d. Click **OK**.
4. Click **Continue**.
5. Start the new role type.
   a. Select the new role type, **WebHCat Server**.
   b. Select **Actions for Selected** > **Start**.
   c. Click **Start** and **Close**.

Configuring HCatalog Using the Command Line

This section applies to unmanaged deployments *without* Cloudera Manager. Use the following sections to install, configure and use HCatalog:

- Prerequisites
- Installing and Upgrading the HCatalog RPM or Debian Packages on page 313
- Host Configuration Changes
- Starting and Stopping the WebHCat REST Server
- Accessing Table Data with the Command-line API
- Accessing Table Data with MapReduce
- Accessing Table Data with Pig
- Accessing Table Data with REST
- Apache HCatalog Documentation

You can use HCatalog to import data to HBase. See Importing Data Into HBase.

For more information, see the HCatalog documentation.

HCatalog Prerequisites

- An operating system supported by CDH 5.
- Oracle JDK.
- The Hive metastore and its database. The Hive metastore must be running in remote mode (as a service).

Installing and Upgrading the HCatalog RPM or Debian Packages

Installing the HCatalog RPM or Debian packages is more convenient than installing the HCatalog tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations

HCatalog comprises the following packages:

- `hive-hcatalog` - HCatalog wrapper for accessing the Hive metastore, libraries for MapReduce and Pig, and a command-line program
- `hive-webhcat` - A REST API server for HCatalog
- `hive-webhcat-server` - Installs `hive-webhcat` and a server `init` script

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

Upgrading HCatalog from an Earlier CDH 5 Release

> **Important:**
>
> If you have installed the `hive-hcatalog-server` package in the past, you must remove it before you proceed; otherwise the upgrade will fail.

Follow instructions under Installing the WebHCat REST Server on page 314 and Installing HCatalog for Use with Pig and MapReduce on page 314.

> **Important: Configuration files**
>
> - If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
> - If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

The upgrade is now complete.

Installing the WebHCat REST Server

> **Note:**
>
> It is not necessary to install WebHCat if you will not be using the REST API. Pig and MapReduce do not need it.

To install the WebHCat REST server on a RHEL system:

```
$ sudo yum install hive-webhcat-server
```

To install the WebHCat REST server components on an Ubuntu or other Debian system:

```
$ sudo apt-get install hive-webhcat-server
```

To install the WebHCat REST server components on a SLES system:

```
$ sudo zypper install hive-webhcat-server
```

> **Note:**
>
> - You can change the default port 50111 by creating or editing the following file and restarting WebHCat:
>
>   ```
>   /etc/webhcat/conf/webhcat-site.xml
>   ```
>
>   The property to change is:
>
>   ```
>   <configuration>
>     <property>
>       <name>templeton.port</name>
>       <value>50111</value>
>       <description>The HTTP port for the main server.</description>
>     </property>
>   </configuration>
>   ```
>
> - To uninstall WebHCat you must remove two packages: `hive-webhcat-server` and `hive-webhcat`.

Installing HCatalog for Use with Pig and MapReduce

On hosts that will be used to launch Pig scripts or MapReduce applications using table information, install HCatalog as follows:

To install the HCatalog client components on a RHEL system:

```
$ sudo yum install hive-hcatalog
```

To install the HCatalog client components on an Ubuntu or other Debian system:

```
$ sudo apt-get install hive-hcatalog
```

To install the HCatalog client components on a SLES system:

```
$ sudo zypper install hive-hcatalog
```

Configuration Change on Hosts Used with HCatalog

You must update `/etc/hive/conf/hive-site.xml` on all hosts where WebHCat will run, as well as all hosts where Pig or MapReduce will be used with HCatalog, so that Metastore clients know where to find the Metastore.

Add or edit the `hive.metastore.uris` property as follows:

```
<property>
  <name>hive.metastore.uris</name>
  <value>thrift://<hostname>:9083</value>
</property>
```

where <hostname> is the host where the HCatalog server components are running, for example `hive.examples.com`.

Starting and Stopping the WebHCat REST server

```
$ sudo service webhcat-server start
$ sudo service webhcat-server stop
```

Accessing Table Information with the HCatalog Command-line API

```
# Create a table
$ hcat -e "create table groups(name string,placeholder string,id int) row format delimited
 fields terminated by ':' stored as textfile"
OK

# Get the schema for a table
$ hcat -e "desc groups"
OK
name string
placeholder string
id int

# Create another table
$ hcat -e "create table groupids(name string,id int)"
OK
```

See the HCatalog documentation for information on using the HCatalog command-line application.

Accessing Table Data with MapReduce

You can download an example of a MapReduce program that reads from the groups table (consisting of data from `/etc/group`), extracts the first and third columns, and inserts them into the `groupids` table. Proceed as follows.

1. Download the program from https://github.com/cloudera/hcatalog-examples.git.
2. Build the example JAR file:

```
$ cd hcatalog-examples
$ mvn package
```

3. Load data from the local filesystem into the groups table:

```
$ hive -e "load data local inpath '/etc/group' overwrite into table groups"
```

4. Set up the environment that is needed for copying the required JAR files to HDFS, for example:

```
$ export HCAT_HOME=/usr/lib/hive-hcatalog
$ export HIVE_HOME=/usr/lib/hive
$ HIVE_VERSION=0.11.0-cdh5.0.0
$ HCATJAR=$HCAT_HOME/share/hcatalog/hcatalog-core-$HIVE_VERSION.jar
$ HCATPIGJAR=$HCAT_HOME/share/hcatalog/hcatalog-pig-adapter-$HIVE_VERSION.jar
$ export HADOOP_CLASSPATH=$HCATJAR:$HCATPIGJAR:$HIVE_HOME/lib/hive-exec-$HIVE_VERSION.jar\
:$HIVE_HOME/lib/hive-metastore-$HIVE_VERSION.jar:$HIVE_HOME/lib/jdo-api-*.jar:$HIVE_HOME/lib/libfb303-*.jar\
:$HIVE_HOME/lib/libthrift-*.jar:$HIVE_HOME/lib/slf4j-api-*.jar:$HIVE_HOME/conf:/etc/hadoop/conf
$ LIBJARS=`echo $HADOOP_CLASSPATH | sed -e 's/:/,/g'`
$ export LIBJARS=$LIBJARS,$HIVE_HOME/lib/antlr-runtime-*.jar
```

> **Note:** You can find current version numbers for CDH dependencies in CDH's root `pom.xml` file for the current release, for example cdh-root-5.0.0.pom.)

5. Run the job:

```
$ hadoop jar target/UseHCat-1.0.jar com.cloudera.test.UseHCat -files $HCATJAR -libjars
 $LIBJARS groups groupids
```

Accessing Table Data with Pig

When using table information from the Hive metastore with Pig, add the `-useHCatalog` option when invoking pig:

```
$ pig -useHCatalog test.pig
```

In the script, use `HCatLoader` to have table schema retrieved automatically:

```
A = LOAD 'groups' USING org.apache.hive.hcatalog.pig.HCatLoader();
DESCRIBE A;
```

Output:

```
A: {name: chararray,placeholder: chararray,id: int}
```

Accessing Table Information with REST

Table information can be retrieved from any host that has HTTP access to the host where the WebHCat server is running. A Web browser or an HTTP client such as curl or wget can be used to verify the functionality.

The base URL for REST access to table information is `http://<SERVERHOST>:50111/templeton/v1/ddl`.

Examples of specific URLs:

```
http://<SERVERHOST>:50111/templeton/v1/ddl/database/?user.name=hive
http://<SERVERHOST>:50111/templeton/v1/ddl/database/default/table/?user.name=hive
http://<SERVERHOST>:50111/templeton/v1/ddl/database/default/table/groups?user.name=hive
```

Example output:

```
{"columns":[{"name":"name","type":"string"},{"name":"placeholder","type":"string"},{"name":"id","type":"int"}],"database":"default","table":"grouptable"}
```

Supported REST Endpoints

The General and DDL endpoints are supported, for accessing Hive metadata. If you need submission capabilities for MapReduce, Hive, or Pig jobs, consider using Oozie, which is a more mature interface. See Installing Oozie on page 371.

Category	Resource Type	Description
General	:version (GET)	Return a list of supported response types.
	status (GET)	Return the WebHCat server status.
	version (GET)	Return a list of supported versions and the current version.
	version/hive (GET)	Return the Hive version being run.
	version/hadoop (GET)	Return the Hadoop version being run.

Category	Resource Type	Description
DDL	ddl (POST)	Perform an HCatalog DDL command.
	ddl/database (GET)	List HCatalog databases.
	ddl/database/:db (GET)	Describe an HCatalog database.
	ddl/database/:db (PUT)	Create an HCatalog database.
	ddl/database/:db (DELETE)	Delete (drop) an HCatalog database.
	ddl/database/:db/table (GET)	List the tables in an HCatalog database.
	ddl/database/:db/table/:table (GET)	Describe an HCatalog table.
	ddl/database/:db/table/:table (PUT)	Create a new HCatalog table.
	ddl/database/:db/table/:table (POST)	Rename an HCatalog table.
	ddl/database/:db/table/:table (DELETE)	Delete (drop) an HCatalog table.
	ddl/database/:db/table/:existingtable/like/:newtable (PUT)	Create a new HCatalog table like an existing one.
	ddl/database/:db/table/:table/partition (GET)	List all partitions in an HCatalog table.
	ddl/database/:db/table/:table/partition/:partition (GET)	Describe a single partition in an HCatalog table.
	ddl/database/:db/table/:table/partition/:partition (PUT)	Create a partition in an HCatalog table.
	ddl/database/:db/table/:table/partition/:partition (DELETE)	Delete (drop) a partition in an HCatalog table.
	ddl/database/:db/table/:table/column (GET)	List the columns in an HCatalog table.
	ddl/database/:db/table/:table/column/:column (GET)	Describe a single column in an HCatalog table.
	ddl/database/:db/table/:table/column/:column (PUT)	Create a column in an HCatalog table.
	ddl/database/:db/table/:table/property (GET)	List table properties.
	ddl/database/:db/table/:table/property/:property (GET)	Return the value of a single table property.
	ddl/database/:db/table/:table/property/:property (PUT)	Set a table property.

Viewing the HCatalog Documentation

See Apache wiki page.

Impala Installation

Impala is an open-source add-on to the Cloudera Enterprise Core that returns rapid responses to queries.

Note:

Under CDH 5, Impala is included as part of the CDH installation and no separate steps are needed.

What is Included in an Impala Installation

Impala is made up of a set of components that can be installed on multiple nodes throughout your cluster. The key installation step for performance is to install the `impalad` daemon (which does most of the query processing work) on *all* DataNodes in the cluster.

The Impala package installs these binaries:

- `impalad` - The Impala daemon. Plans and executes queries against HDFS, HBase, and Amazon S3 data. Run one impalad process on each node in the cluster that has a DataNode.
- `statestored` - Name service that tracks location and status of all `impalad` instances in the cluster. Run one instance of this daemon on a node in your cluster. Most production deployments run this daemon on the namenode.
- `catalogd` - Metadata coordination service that broadcasts changes from Impala DDL and DML statements to all affected Impala nodes, so that new tables, newly loaded data, and so on are immediately visible to queries submitted through any Impala node. (Prior to Impala 1.2, you had to run the REFRESH or `INVALIDATE METADATA` statement on each node to synchronize changed metadata. Now those statements are only required if you perform the DDL or DML through an external mechanism such as Hive or by uploading data to the Amazon S3 filesystem.) Run one instance of this daemon on a node in your cluster, preferably on the same host as the `statestored` daemon.
- `impala-shell` - Command-line interface for issuing queries to the Impala daemon. You install this on one or more hosts anywhere on your network, not necessarily DataNodes or even within the same cluster as Impala. It can connect remotely to any instance of the Impala daemon.

Before doing the installation, ensure that you have all necessary prerequisites. See Impala Requirements on page 318 for details.

Impala Requirements

To perform as expected, Impala depends on the availability of the software, hardware, and configurations described in the following sections.

Product Compatibility Matrix

The ultimate source of truth about compatibility between various versions of CDH, Cloudera Manager, and various CDH components is the Product Compatibility Matrix for CDH and Cloudera Manager.

Supported Operating Systems

The relevant supported operating systems and versions for Impala are the same as for the corresponding CDH 5 platforms. For details, see the *Supported Operating Systems* page for CDH 5.

Hive Metastore and Related Configuration

Impala can interoperate with data stored in Hive, and uses the same infrastructure as Hive for tracking metadata about schema objects such as tables and columns. The following components are prerequisites for Impala:

- MySQL or PostgreSQL, to act as a metastore database for both Impala and Hive.

Note:

Installing and configuring a Hive metastore is an Impala requirement. Impala does not work without the metastore database. For the process of installing and configuring the metastore, see Impala Installation on page 317.

Always configure a **Hive metastore service** rather than connecting directly to the metastore database. The Hive metastore service is required to interoperate between possibly different levels of metastore APIs used by CDH and Impala, and avoids known issues with connecting directly to the metastore database. The Hive metastore service is set up for you by default if you install through Cloudera Manager 4.5 or higher.

A summary of the metastore installation process is as follows:

- Install a MySQL or PostgreSQL database. Start the database if it is not started after installation.
- Download the MySQL connector or the PostgreSQL connector and place it in the `/usr/share/java/` directory.
- Use the appropriate command line tool for your database to create the metastore database.
- Use the appropriate command line tool for your database to grant privileges for the metastore database to the `hive` user.
- Modify `hive-site.xml` to include information matching your particular database: its URL, username, and password. You will copy the `hive-site.xml` file to the Impala Configuration Directory later in the Impala installation process.

- **Optional:** Hive. Although only the Hive metastore database is required for Impala to function, you might install Hive on some client machines to create and load data into tables that use certain file formats. See How Impala Works with Hadoop File Formats for details. Hive does not need to be installed on the same DataNodes as Impala; it just needs access to the same metastore database.

Java Dependencies

Although Impala is primarily written in C++, it does use Java to communicate with various Hadoop components:

- The officially supported JVM for Impala is the Oracle JVM. Other JVMs might cause issues, typically resulting in a failure at `impalad` startup. In particular, the JamVM used by default on certain levels of Ubuntu systems can cause `impalad` to fail to start.
- Internally, the `impalad` daemon relies on the JAVA_HOME environment variable to locate the system Java libraries. Make sure the `impalad` service is not run from an environment with an incorrect setting for this variable.
- All Java dependencies are packaged in the `impala-dependencies.jar` file, which is located at `/usr/lib/impala/lib/`. These map to everything that is built under `fe/target/dependency`.

Networking Configuration Requirements

As part of ensuring best performance, Impala attempts to complete tasks on local data, as opposed to using network connections to work with remote data. To support this goal, Impala matches the **hostname** provided to each Impala daemon with the **IP address** of each DataNode by resolving the hostname flag to an IP address. For Impala to work with local data, use a single IP interface for the DataNode and the Impala daemon on each machine. Ensure that the Impala daemon's hostname flag resolves to the IP address of the DataNode. For single-homed machines, this is usually automatic, but for multi-homed machines, ensure that the Impala daemon's hostname resolves to the correct interface. Impala tries to detect the correct hostname at start-up, and prints the derived hostname at the start of the log in a message of the form:

```
Using hostname: impala-daemon-1.example.com
```

In the majority of cases, this automatic detection works correctly. If you need to explicitly set the hostname, do so by setting the `--hostname` flag.

Hardware Requirements

The memory allocation should be consistent across Impala executor nodes. A single Impala executor with a lower memory limit than the rest can easily become a bottleneck and lead to suboptimal performance.

This guideline does not apply to coordinator-only nodes.

Hardware Requirements for Optimal Join Performance

During join operations, portions of data from each joined table are loaded into memory. Data sets can be very large, so ensure your hardware has sufficient memory to accommodate the joins you anticipate completing.

While requirements vary according to data set size, the following is generally recommended:

- CPU

  Impala version 2.2 and higher uses the SSSE3 instruction set, which is included in newer processors.

  > **Note:** This required level of processor is the same as in Impala version 1.x. The Impala 2.0 and 2.1 releases had a stricter requirement for the SSE4.1 instruction set, which has now been relaxed.

- Memory

  128 GB or more recommended, ideally 256 GB or more. If the intermediate results during query processing on a particular node exceed the amount of memory available to Impala on that node, the query writes temporary work data to disk, which can lead to long query times. Note that because the work is parallelized, and intermediate results for aggregate queries are typically smaller than the original data, Impala can query and join tables that are much larger than the memory available on an individual node.

- Storage

  DataNodes with 12 or more disks each. I/O speeds are often the limiting factor for disk performance with Impala. Ensure that you have sufficient disk space to store the data Impala will be querying.

User Account Requirements

Impala creates and uses a user and group named `impala`. Do not delete this account or group and do not modify the account's or group's permissions and rights. Ensure no existing systems obstruct the functioning of these accounts and groups. For example, if you have scripts that delete user accounts not in a white-list, add these accounts to the list of permitted accounts.

For correct file deletion during `DROP TABLE` operations, Impala must be able to move files to the HDFS trashcan. You might need to create an HDFS directory `/user/impala`, writeable by the `impala` user, so that the trashcan can be created. Otherwise, data files might remain behind after a `DROP TABLE` statement.

Impala should not run as root. Best Impala performance is achieved using direct reads, but root is not permitted to use direct reads. Therefore, running Impala as root negatively affects performance.

By default, any user can connect to Impala and access all the associated databases and tables. You can enable authorization and authentication based on the Linux OS user who connects to the Impala server, and the associated groups for that user. Overview of Impala Security for details. These security features do not change the underlying file permission requirements; the `impala` user still needs to be able to access the data files.

Installing Impala from the Command Line

Before installing Impala manually, make sure all applicable nodes have the appropriate hardware configuration, levels of operating system and CDH, and any other software prerequisites. See Impala Requirements on page 318 for details.

You can install Impala across many hosts or on one host:

- Installing Impala across multiple machines creates a distributed configuration. For best performance, install Impala on **all** DataNodes.
- Installing Impala on a single machine produces a pseudo-distributed cluster.

To install Impala on a host:

1. Install CDH as described in the Installation section of the CDH 5 Installation Guide.
2. Install the Hive metastore somewhere in your cluster, as described in the Hive Installation topic in the CDH 5 Installation Guide. As part of this process, you configure the Hive metastore to use an external database as a metastore. Impala uses this same database for its own table metadata. You can choose either a MySQL or PostgreSQL database as the metastore. The process for configuring each type of database is described in the CDH Installation Guide).

   Cloudera recommends setting up a Hive metastore service rather than connecting directly to the metastore database; this configuration is required when running Impala under CDH 4.1. Make sure the `/etc/impala/conf/hive-site.xml` file contains the following setting, substituting the appropriate hostname for *metastore_server_host*:

```
<property>
<name>hive.metastore.uris</name>
<value>thrift://metastore_server_host:9083</value>
</property>
<property>
<name>hive.metastore.client.socket.timeout</name>
<value>3600</value>
<description>MetaStore Client socket timeout in seconds</description>
</property>
```

3. (Optional) If you installed the full Hive component on any host, you can verify that the metastore is configured properly by starting the Hive console and querying for the list of available tables. Once you confirm that the console starts, exit the console to continue the installation:

```
$ hive
Hive history file=/tmp/root/hive_job_log_root_201207272011_678722950.txt
hive> show tables;
table1
table2
hive> quit;
$
```

4. Confirm that your package management command is aware of the Impala repository settings, as described in Impala Requirements on page 318. (For CDH 4, this is a different repository than for CDH.) You might need to download a repo or list file into a system directory underneath `/etc`.
5. Use **one** of the following sets of commands to install the Impala package:

   For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum install impala             # Binaries for daemons
$ sudo yum install impala-server      # Service start/stop script
$ sudo yum install impala-state-store # Service start/stop script
$ sudo yum install impala-catalog     # Service start/stop script
```

   For SUSE systems:

```
$ sudo zypper install impala             # Binaries for daemons
$ sudo zypper install impala-server      # Service start/stop script
$ sudo zypper install impala-state-store # Service start/stop script
$ sudo zypper install impala-catalog     # Service start/stop script
```

   For Debian or Ubuntu systems:

```
$ sudo apt-get install impala             # Binaries for daemons
$ sudo apt-get install impala-server      # Service start/stop script
$ sudo apt-get install impala-state-store # Service start/stop script
$ sudo apt-get install impala-catalog     # Service start/stop script
```

> **Note:** Cloudera recommends that you not install Impala on any HDFS NameNode. Installing Impala on NameNodes provides no additional data locality, and executing queries with such a configuration might cause memory contention and negatively impact the HDFS NameNode.

6. Copy the client `hive-site.xml`, `core-site.xml`, `hdfs-site.xml`, and `hbase-site.xml` configuration files to the Impala configuration directory, which defaults to `/etc/impala/conf`. Create this directory if it does not already exist.
7. Use **one** of the following commands to install `impala-shell` on the machines from which you want to issue queries. You can install `impala-shell` on any supported machine that can connect to DataNodes that are running `impalad`.

   For RHEL/CentOS systems:

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

   For SUSE systems:

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

   For Debian/Ubuntu systems:

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

8. Complete any required or recommended configuration, as described in Post-Installation Configuration for Impala. Some of these configuration changes are mandatory.

Once installation and configuration are complete, see Starting Impala on page 326 for how to activate the software on the appropriate nodes in your cluster.

If this is your first time setting up and using Impala in this cluster, run through some of the exercises in Impala Tutorials to verify that you can do basic operations such as creating tables and querying them.

Upgrading Impala

Upgrading Impala involves stopping Impala services, using your operating system's package management tool to upgrade Impala to the latest version, and then restarting Impala services.

> **Note:**
>
> - Each version of CDH 5 has an associated version of Impala.
> - When you upgrade Impala, also upgrade Cloudera Manager if necessary. Cloudera Manager is continually updated with configuration settings for features introduced in the latest Impala releases.
> - Make sure you are using the appropriate CDH 5 repositories shown on the CDH version and packaging page, then follow the procedures throughout the rest of this section.
> - Every time you upgrade to a new major or minor Impala release, see Apache Impala Incompatible Changes and Limitations in the *Release Notes* for any changes needed in your source code, startup scripts, and so on.
> - Also check Apache Impala Known Issues in the *Release Notes* for any issues or limitations that require workarounds.

Upgrading Impala through Cloudera Manager - Parcels

Parcels are an alternative binary distribution format available in Cloudera Manager 4.5 and higher.

Important: In CDH 5, there is not a separate Impala parcel; Impala is part of the main CDH 5 parcel. Each level of CDH 5 has a corresponding version of Impala, and you upgrade Impala by upgrading CDH. See the CDH 5 upgrade instructions and choose the instructions for parcels. The remainder of this section only covers parcel upgrades for Impala under CDH 4.

To upgrade Impala for CDH 4 in a Cloudera Managed environment, using parcels:

1. If you originally installed using packages and now are switching to parcels, remove all the Impala-related packages first. You can check which packages are installed using one of the following commands, depending on your operating system:

```
rpm -qa               # RHEL, Oracle Linux, CentOS, Debian
dpkg --get-selections # Debian
```

   and then remove the packages using one of the following commands:

```
sudo yum remove pkg_names    # RHEL, Oracle Linux, CentOS
sudo zypper remove pkg_names # SLES
sudo apt-get purge pkg_names # Ubuntu, Debian
```

2. Connect to the Cloudera Manager Admin Console.
3. Go to the **Hosts** > **Parcels** tab. You should see a parcel with a newer version of Impala that you can upgrade to.
4. Click **Download**, then **Distribute**. (The button changes as each step completes.)
5. Click **Activate**.
6. When prompted, click **Restart** to restart the Impala service.

Upgrading Impala through Cloudera Manager - Packages

To upgrade Impala in a Cloudera Managed environment, using packages:

1. Connect to the Cloudera Manager Admin Console.
2. In the **Services** tab, click the **Impala** service.
3. Click **Actions** and click **Stop**.
4. Use **one** of the following sets of commands to update Impala on each Impala node in your cluster:

   For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala
$ sudo yum update hadoop-lzo-cdh4 # Optional; if this package is already installed
```

   For SUSE systems:

```
$ sudo zypper update impala
$ sudo zypper update hadoop-lzo-cdh4 # Optional; if this package is already installed
```

   For Debian or Ubuntu systems:

```
$ sudo apt-get install impala
$ sudo apt-get install hadoop-lzo-cdh4 # Optional; if this package is already installed
```

5. Use **one** of the following sets of commands to update Impala shell on each node on which it is installed:

   For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala-shell
```

For SUSE systems:

```
$ sudo zypper update impala-shell
```

For Debian or Ubuntu systems:

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

6. Connect to the Cloudera Manager Admin Console.
7. In the **Services** tab, click the Impala service.
8. Click **Actions** and click **Start**.

Upgrading Impala from the Command Line

To upgrade Impala on a cluster by using the command-line, run these Linux commands on the appropriate hosts in your cluster:

1. Stop Impala services.
   a. Stop `impalad` on each Impala node in your cluster:

```
$ sudo service impala-server stop
```

   b. Stop any instances of the state store in your cluster:

```
$ sudo service impala-state-store stop
```

   c. Stop any instances of the catalog service in your cluster:

```
$ sudo service impala-catalog stop
```

2. Check if there are new recommended or required configuration settings to put into place in the configuration files, typically under `/etc/impala/conf`. See Post-Installation Configuration for Impala for settings related to performance and scalability.
3. Use **one** of the following sets of commands to update Impala on each Impala node in your cluster:

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala-server
$ sudo yum update hadoop-lzo-cdh4 # Optional; if this package is already installed
$ sudo yum update impala-catalog # New in Impala 1.2; do yum install when upgrading from
 1.1.
```

For SUSE systems:

```
$ sudo zypper update impala-server
$ sudo zypper update hadoop-lzo-cdh4 # Optional; if this package is already installed
$ sudo zypper update impala-catalog # New in Impala 1.2; do zypper install when upgrading
 from 1.1.
```

For Debian or Ubuntu systems:

```
$ sudo apt-get install impala-server
$ sudo apt-get install hadoop-lzo-cdh4 # Optional; if this package is already installed
$ sudo apt-get install impala-catalog # New in Impala 1.2.
```

4. Use **one** of the following sets of commands to update Impala shell on each node on which it is installed:

For RHEL, Oracle Linux, or CentOS systems:

```
$ sudo yum update impala-shell
```

For SUSE systems:

```
$ sudo zypper update impala-shell
```

For Debian or Ubuntu systems:

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

5. Depending on which release of Impala you are upgrading from, you might find that the symbolic links `/etc/impala/conf` and `/usr/lib/impala/sbin` are missing. If so, see Apache Impala Known Issues for the procedure to work around this problem.
6. Restart Impala services:
   a. Restart the Impala state store service on the desired nodes in your cluster. Expect to see a process named `statestored` if the service started successfully.

```
$ sudo service impala-state-store start
$ ps ax | grep [s]tatestored
 6819 ?        Sl     0:07 /usr/lib/impala/sbin/statestored -log_dir=/var/log/impala
-state_store_port=24000
```

   Restart the state store service *before* the Impala server service to avoid "Not connected" errors when you run `impala-shell`.

   b. Restart the Impala catalog service on whichever host it runs on in your cluster. Expect to see a process named `catalogd` if the service started successfully.

```
$ sudo service impala-catalog restart
$ ps ax | grep [c]atalogd
 6068 ?        Sl     4:06 /usr/lib/impala/sbin/catalogd
```

   c. Restart the Impala daemon service on each node in your cluster. Expect to see a process named `impalad` if the service started successfully.

```
$ sudo service impala-server start
$ ps ax | grep [i]mpalad
 7936 ?        Sl     0:12 /usr/lib/impala/sbin/impalad -log_dir=/var/log/impala
-state_store_port=24000 -use_statestore
-state_store_host=127.0.0.1 -be_port=22000
```

Note:

If the services did not start successfully (even though the `sudo service` command might display `[OK]`), check for errors in the Impala log file, typically in `/var/log/impala`.

Converting Legacy UDFs During Upgrade to CDH 5.12 or Higher

In and higher, new syntax is available for creating Java-based UDFs. UDFs created with the new syntax persist across Impala restarts, and are more compatible with Hive UDFs. Because the replication features in CDH 5.12 and higher only work with the new-style syntax, convert any older Java UDFs to use the new syntax at the same time you upgrade to CDH 5.12 or higher.

Follow these steps to convert old-style Java UDFs to the new persistent kind:

- Use `SHOW FUNCTIONS` to identify all UDFs and UDAs.
- For each function, use `SHOW CREATE FUNCTION` and save the statement in a script file.
- For Java UDFs, change the output of `SHOW CREATE FUNCTION` to use the new `CREATE FUNCTION` syntax (without argument types), which makes the UDF persistent.

- For each function, drop it and re-create it, using the new `CREATE FUNCTION` syntax for all Java UDFs.

Starting Impala

To activate Impala if it is installed but not yet started:

1. Set any necessary configuration options for the Impala services. See Modifying Impala Startup Options on page 327 for details.
2. Start one instance of the Impala statestore. The statestore helps Impala to distribute work efficiently, and to continue running in the event of availability problems for other Impala nodes. If the statestore becomes unavailable, Impala continues to function.
3. Start one instance of the Impala catalog service.
4. Start the main Impala daemon services.

Once Impala is running, you can conduct interactive experiments using the instructions in Impala Tutorials and try Using the Impala Shell (impala-shell Command).

Starting Impala through Cloudera Manager

If you installed Impala with Cloudera Manager, use Cloudera Manager to start and stop services. The Cloudera Manager GUI is a convenient way to check that all services are running, to set configuration options using form fields in a browser, and to spot potential issues such as low disk space before they become serious. Cloudera Manager automatically starts all the Impala-related services as a group, in the correct order. See the Cloudera Manager Documentation for details.

Note:

In and higher, Impala UDFs and UDAs written in C++ are persisted in the metastore database. Java UDFs are also persisted, if they were created with the new `CREATE FUNCTION` syntax for Java UDFs, where the Java function argument and return types are omitted. Java-based UDFs created with the old `CREATE FUNCTION` syntax do not persist across restarts because they are held in the memory of the `catalogd` daemon. Until you re-create such Java UDFs using the new `CREATE FUNCTION` syntax, you must reload those Java-based UDFs by running the original `CREATE FUNCTION` statements again each time you restart the `catalogd` daemon. Prior to the requirement to reload functions after a restart applied to both C++ and Java functions.

Starting Impala from the Command Line

To start the Impala state store and Impala from the command line or a script, you can either use the `service` command or you can start the daemons directly through the `impalad`, `statestored`, and `catalogd` executables.

Start the Impala statestore and then start `impalad` instances. You can modify the values the service initialization scripts use when starting the statestore and Impala by editing `/etc/default/impala`.

Start the statestore service using a command similar to the following:

```
$ sudo service impala-state-store start
```

Start the catalog service using a command similar to the following:

```
$ sudo service impala-catalog start
```

Start the Impala daemon services using a command similar to the following:

```
$ sudo service impala-server start
```

Note:

In and higher, Impala UDFs and UDAs written in C++ are persisted in the metastore database. Java UDFs are also persisted, if they were created with the new `CREATE FUNCTION` syntax for Java UDFs, where the Java function argument and return types are omitted. Java-based UDFs created with the old `CREATE FUNCTION` syntax do not persist across restarts because they are held in the memory of the `catalogd` daemon. Until you re-create such Java UDFs using the new `CREATE FUNCTION` syntax, you must reload those Java-based UDFs by running the original `CREATE FUNCTION` statements again each time you restart the `catalogd` daemon. Prior to the requirement to reload functions after a restart applied to both C++ and Java functions.

If any of the services fail to start, review:

- Reviewing Impala Logs
- Troubleshooting Impala

Modifying Impala Startup Options

The configuration options for the Impala-related daemons let you choose which hosts and ports to use for the services that run on a single host, specify directories for logging, control resource usage and security, and specify other aspects of the Impala software.

Configuring Impala Startup Options through Cloudera Manager

If you manage your cluster through Cloudera Manager, configure the settings for all the Impala-related daemons by navigating to this page: **Clusters** > **Impala** > **Configuration** > **View and Edit**. See the Cloudera Manager documentation for instructions about how to configure Impala through Cloudera Manager.

If the Cloudera Manager interface does not yet have a form field for a newly added option, or if you need to use special options for debugging and troubleshooting, the **Advanced** option page for each daemon includes one or more fields where you can enter option names directly. In Cloudera Manager 4, these fields are labelled **Safety Valve**; in Cloudera Manager 5, they are called **Advanced Configuration Snippet**. There is also a free-form field for query options, on the top-level **Impala Daemon** options page.

Configuring Impala Startup Options through the Command Line

When you run Impala in a non-Cloudera Manager environment, the Impala server, statestore, and catalog services start up using values provided in a defaults file, `/etc/default/impala`.

This file includes information about many resources used by Impala. Most of the defaults included in this file should be effective in most cases. For example, typically you would not change the definition of the `CLASSPATH` variable, but you would always set the address used by the statestore server. Some of the content you might modify includes:

```
IMPALA_STATE_STORE_HOST=127.0.0.1
IMPALA_STATE_STORE_PORT=24000
IMPALA_BACKEND_PORT=22000
IMPALA_LOG_DIR=/var/log/impala
IMPALA_CATALOG_SERVICE_HOST=...
IMPALA_STATE_STORE_HOST=...

export IMPALA_STATE_STORE_ARGS=${IMPALA_STATE_STORE_ARGS:- \
    -log_dir=${IMPALA_LOG_DIR} -state_store_port=${IMPALA_STATE_STORE_PORT}}
IMPALA_SERVER_ARGS=" \
-log_dir=${IMPALA_LOG_DIR} \
-catalog_service_host=${IMPALA_CATALOG_SERVICE_HOST} \
-state_store_port=${IMPALA_STATE_STORE_PORT} \
-use_statestore \
-state_store_host=${IMPALA_STATE_STORE_HOST} \
-be_port=${IMPALA_BACKEND_PORT}"
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-false}
```

To use alternate values, edit the defaults file, then restart all the Impala-related services so that the changes take effect. Restart the Impala server using the following commands:

```
$ sudo service impala-server restart
Stopping Impala Server:                                    [  OK  ]
Starting Impala Server:                                    [  OK  ]
```

Restart the Impala statestore using the following commands:

```
$ sudo service impala-state-store restart
Stopping Impala State Store Server:                        [  OK  ]
Starting Impala State Store Server:                        [  OK  ]
```

Restart the Impala catalog service using the following commands:

```
$ sudo service impala-catalog restart
Stopping Impala Catalog Server:                            [  OK  ]
Starting Impala Catalog Server:                            [  OK  ]
```

Some common settings to change include:

- Statestore address. Where practical, put the statestore on a separate host not running the `impalad` daemon. In that recommended configuration, the `impalad` daemon cannot refer to the statestore server using the loopback address. If the statestore is hosted on a machine with an IP address of 192.168.0.27, change:

```
IMPALA_STATE_STORE_HOST=127.0.0.1
```

  to:

```
IMPALA_STATE_STORE_HOST=192.168.0.27
```

- Catalog server address (including both the hostname and the port number). Update the value of the `IMPALA_CATALOG_SERVICE_HOST` variable. Cloudera recommends the catalog server be on the same host as the statestore. In that recommended configuration, the `impalad` daemon cannot refer to the catalog server using the loopback address. If the catalog service is hosted on a machine with an IP address of 192.168.0.27, add the following line:

```
IMPALA_CATALOG_SERVICE_HOST=192.168.0.27:26000
```

  The `/etc/default/impala` defaults file currently does not define an `IMPALA_CATALOG_ARGS` environment variable, but if you add one it will be recognized by the service startup/shutdown script. Add a definition for this variable to `/etc/default/impala` and add the option `-catalog_service_host=`*`hostname`*. If the port is different than the default 26000, also add the option `-catalog_service_port=`*`port`*.

- Memory limits. You can limit the amount of memory available to Impala. For example, to allow Impala to use no more than 70% of system memory, change:

```
export IMPALA_SERVER_ARGS=${IMPALA_SERVER_ARGS:- \
    -log_dir=${IMPALA_LOG_DIR} \
    -state_store_port=${IMPALA_STATE_STORE_PORT} \
    -use_statestore -state_store_host=${IMPALA_STATE_STORE_HOST} \
    -be_port=${IMPALA_BACKEND_PORT}}
```

  to:

```
export IMPALA_SERVER_ARGS=${IMPALA_SERVER_ARGS:- \
    -log_dir=${IMPALA_LOG_DIR} -state_store_port=${IMPALA_STATE_STORE_PORT} \
    -use_statestore -state_store_host=${IMPALA_STATE_STORE_HOST} \
    -be_port=${IMPALA_BACKEND_PORT} -mem_limit=70%}
```

You can specify the memory limit using absolute notation such as `500m` or `2G`, or as a percentage of physical memory such as `60%`.

> **Note:** Queries that exceed the specified memory limit are aborted. Percentage limits are based on the physical memory of the machine and do not consider cgroups.

- Core dump enablement. To enable core dumps on systems not managed by Cloudera Manager, change:

```
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-false}
```

to:

```
export ENABLE_CORE_DUMPS=${ENABLE_COREDUMPS:-true}
```

On systems managed by Cloudera Manager, enable the **Enable Core Dump** setting for the Impala service.

> **Note:**
>
> - The location of core dump files may vary according to your operating system configuration.
> - Other security settings may prevent Impala from writing core dumps even when this option is enabled.
> - On systems managed by Cloudera Manager, the default location for core dumps is on a temporary filesystem, which can lead to out-of-space issues if the core dumps are large, frequent, or not removed promptly. To specify an alternative location for the core dumps, filter the Impala configuration settings to find the `core_dump_dir` option, which is available in Cloudera Manager 5.4.3 and higher. This option lets you specify a different directory for core dumps for each of the Impala-related daemons.

- Authorization using the open source Sentry plugin. Specify the `-server_name` and `-authorization_policy_file` options as part of the `IMPALA_SERVER_ARGS` and `IMPALA_STATE_STORE_ARGS` settings to enable the core Impala support for authentication. See Starting the impalad Daemon with Sentry Authorization Enabled for details.
- Auditing for successful or blocked Impala queries, another aspect of security. Specify the `-audit_event_log_dir=`*`directory_path`* option and optionally the `-max_audit_event_log_file_size=`*`number_of_queries`* and `-abort_on_failed_audit_event` options as part of the `IMPALA_SERVER_ARGS` settings, for each Impala node, to enable and customize auditing. See Auditing Impala Operations for details.
- Password protection for the Impala web UI, which listens on port 25000 by default. This feature involves adding some or all of the `--webserver_password_file`, `--webserver_authentication_domain`, and `--webserver_certificate_file` options to the `IMPALA_SERVER_ARGS` and `IMPALA_STATE_STORE_ARGS` settings. See Security Guidelines for Impala for details.
- Another setting you might add to `IMPALA_SERVER_ARGS` is a comma-separated list of query options and values:

```
-default_query_options='option=value,option=value,...'
```

These options control the behavior of queries performed by this `impalad` instance. The option values you specify here override the default values for Impala query options, as shown by the `SET` statement in `impala-shell`.

- During troubleshooting, might direct you to change other values, particularly for `IMPALA_SERVER_ARGS`, to work around issues or gather debugging information.

Note:

These startup options for the `impalad` daemon are different from the command-line options for the `impala-shell` command. For the `impala-shell` options, see impala-shell Configuration Options.

Checking the Values of Impala Configuration Options

You can check the current runtime value of all these settings through the Impala web interface, available by default at `http://impala_hostname:25000/varz` for the `impalad` daemon, `http://impala_hostname:25010/varz` for the `statestored` daemon, or `http://impala_hostname:25020/varz` for the `catalogd` daemon. In the Cloudera Manager interface, you can see the link to the appropriate ***service_name* Web UI** page when you look at the status page for a specific daemon on a specific host.

Startup Options for impalad Daemon

The `impalad` daemon implements the main Impala service, which performs query processing and reads and writes the data files. Some of the noteworthy options are:

- The `fe_service_threads` option specifies the maximum number of concurrent client connections allowed. The default value is 64 with which 64 queries can run simultaneously.

  If you have more clients trying to connect to Impala than the value of this setting, the later arriving clients have to wait until previous clients disconnect. You can increase this value to allow more client connections. However, a large value means more threads to be maintained even if most of the connections are idle, and it could negatively impact query latency. Client applications should use the connection pool to avoid the need for large number of sessions.

Startup Options for statestored Daemon

The `statestored` daemon implements the Impala statestore service, which monitors the availability of Impala services across the cluster, and handles situations such as nodes becoming unavailable or becoming available again.

Startup Options for catalogd Daemon

The `catalogd` daemon implements the Impala catalog service, which broadcasts metadata changes to all the Impala nodes when Impala creates a table, inserts data, or performs other kinds of DDL and DML operations.

Use `--load_catalog_in_background` option to control when the metadata of a table is loaded.

- If set to `false`, the metadata of a table is loaded when it is referenced for the first time. This means that the first run of a particular query can be slower than subsequent runs. Starting in Impala 2.2, the default for `load_catalog_in_background` is `false`.
- If set to `true`, the catalog service attempts to load metadata for a table even if no query needed that metadata. So metadata will possibly be already loaded when the first query that would need it is run. However, for the following reasons, we recommend not to set the option to `true`.
  - Background load can interfere with query-specific metadata loading. This can happen on startup or after invalidating metadata, with a duration depending on the amount of metadata, and can lead to a seemingly random long running queries that are difficult to diagnose.
  - Impala may load metadata for tables that are possibly never used, potentially increasing catalog size and consequently memory usage for both catalog service and Impala Daemon.

Hive Installation

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

Using Hive data in HBase is a common task. See Importing Data Into HBase.

For information about Hive on Spark, see Running Apache Hive on Spark in CDH.

Use the following sections to install, update, and configure Hive.

Apache Hive is a powerful data warehousing application for Hadoop. It enables you to access your data using HiveQL, a language similar to SQL.

Install Hive on your client machine(s) from which you submit jobs; you do not need to install it on the nodes in your Hadoop cluster. As of CDH 5, Hive supports HCatalog which must be installed separately.

HiveServer2

HiveServer2 is an improved version of HiveServer that supports a Thrift API tailored for JDBC and ODBC clients, Kerberos authentication, and multi-client concurrency. The CLI for HiveServer2 is Beeline.

> **Warning:** Because of concurrency and security issues, HiveServer1 and the Hive CLI are deprecated in CDH 5 and will be removed in a future release. Cloudera recommends you migrate to Beeline and HiveServer2 as soon as possible. The Hive CLI is not needed if you are using Beeline with HiveServer2.

Installing Hive

Install the appropriate Hive packages using the appropriate command for your distribution.

OS	Command
RHEL-compatible	`$ sudo yum install <pkg1> <pkg2> ...`
SLES	`$ sudo zypper install <pkg1> <pkg2> ...`
Ubuntu or Debian	`$ sudo apt-get install <pkg1> <pkg2> ...`

The packages are:

- `hive` – base package that provides the complete language and runtime
- `hive-metastore` – provides scripts for running the metastore as a standalone service (optional)
- `hive-server2` – provides scripts for running HiveServer2
- `hive-hbase` - optional; install this package if you want to use Hive with HBase.

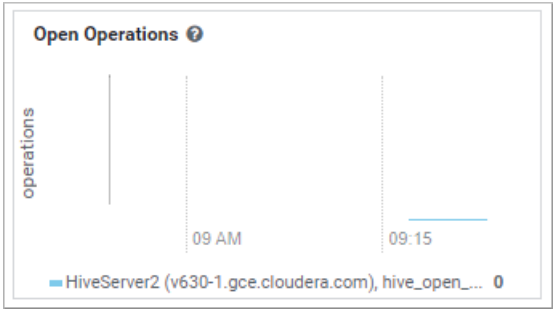
> **Important:** After installing Hive, see HiveServer2 Performance Best Practices for information about optimizing your Hive deployment and your Hive workloads for best performance results.

Heap Size and Garbage Collection for Hive Components

This section provides guidelines for setting HiveServer2 and Hive metastore memory and garbage-collection properties.

Memory Recommendations

HiveServer2 and the Hive metastore require sufficient memory to run correctly. The default heap size of 256 MB for each component is inadequate for production workloads. The table below contains guidelines for sizing the heap for each component, based on your cluster size. The table refers to connections, the number of open connections to HiveServer (Cloudera Manager hive_open_connections metric). In Cloudera Manager, HiveServer2, Status, the visual representation of this metric appears. For example:



Number of Concurrent Connections	HiveServer2 Heap Size Recommended Range	Hive Metastore Heap Size Recommended Range
Up to 40 concurrent connections Cloudera recommends splitting HiveServer2 into multiple instances and load-balancing once you start allocating over 16 GB to HiveServer2. This reduces the impact of Java garbage collection on active processing by the service.	12 - 16 GB	12 - 16 GB
Up to 20 concurrent connections	6 - 12 GB	10 - 12 GB
Up to 10 concurrent connections	4 - 6 GB	4 - 10 GB
One connection	4 GB	4 GB

Important: These numbers are general guidance only, and can be affected by factors such as number of columns, partitions, complex joins, and client activity. Based on your anticipated deployment, refine through testing to arrive at the best values for your environment.

In addition, the Beeline CLI should use a heap size of at least 2 GB.

Set the PermGen space for Java garbage collection to 512 MB for all.

Configuring Heap Size and Garbage Collection

Using Cloudera Manager

To configure heap size and garbage collection for HiveServer2:

1. To set heap size, go to **Home > Hive > Configuration > HiveServer2 > Resource Management**.
2. Set **Java Heap Size of HiveServer2 in Bytes** to the desired value, and click **Save Changes**.
3. To set garbage collection, go to **Home > Hive > Configuration > HiveServer2 > Advanced**.
4. Set the PermGen space for Java garbage collection to `512M`, the type of garbage collector used (`ConcMarkSweepGC` or `ParNewGC`), and enable or disable the garbage collection overhead limit in **Java Configuration Options for HiveServer2**.

The following example sets the PermGen space to 512M, uses the new Parallel Collector, and disables the garbage collection overhead limit:

```
-XX:MaxPermSize=512M -XX:+UseParNewGC -XX:-UseGCOverheadLimit
```

5. From the **Actions** drop-down menu, select Restart to restart the HiveServer2 service.

To configure heap size and garbage collection for the Hive metastore:

1. To set heap size, go to **Home > Hive > Configuration > Hive Metastore > Resource Management**.
2. Set **Java Heap Size of Hive Metastore Server in Bytes** to the desired value, and click **Save Changes**.
3. To set garbage collection, go to **Home > Hive > Configuration > Hive Metastore Server > Advanced**.
4. Set the PermGen space for Java garbage collection to 512M, the type of garbage collector used (ConcMarkSweepGC or ParNewGC), and enable or disable the garbage collection overhead limit in **Java Configuration Options for Hive Metastore Server**. For an example of this setting, see step 4 above for configuring garbage collection for HiveServer2.
5. From the **Actions** drop-down menu, select Restart to restart the Hive Metastore service.

To configure heap size and garbage collection for the Beeline CLI:

1. To set heap size, go to **Home > Hive > Configuration > Gateway > Resource Management**.
2. Set **Client Java Heap Size in Bytes** to at least 2 GiB and click **Save Changes**.
3. To set garbage collection, go to **Home > Hive > Configuration > Gateway > Advanced**.
4. Set the PermGen space for Java garbage collection to 512M in **Client Java Configuration Options**.

   The following example sets the PermGen space to 512M and specifies IPv4:

```
-XX:MaxPermSize=512M -Djava.net.preferIPv4Stack=true
```

5. From the **Actions** drop-down menu, select Restart to restart the client service.

Using the Command Line

To configure the heap size for HiveServer2 and Hive metastore, set the -Xmx parameter in the HADOOP_OPTS variable to the desired maximum heap size in /etc/hive/hive-env.sh.

To configure the heap size for the Beeline CLI, set the HADOOP_HEAPSIZE environment variable in /etc/hive/hive-env.sh before starting the Beeline CLI.

The following example shows a configuration with the following settings:

- HiveServer2 uses 12 GB heap.
- Hive metastore uses 12 GB heap.
- Hive clients use 2 GB heap.

The settings to change are in bold. All of these lines are commented out (prefixed with a # character) by default.

```
if [ "$SERVICE" = "cli" ]; then
  if [ -z "$DEBUG" ]; then
    export HADOOP_OPTS="$HADOOP_OPTS -XX:NewRatio=12 -Xmx12288m -Xms12288m
-XX:MaxHeapFreeRatio=40 -XX:MinHeapFreeRatio=15 -XX:+UseParNewGC -XX:-UseGCOverheadLimit"

  else
    export HADOOP_OPTS="$HADOOP_OPTS -XX:NewRatio=12 -Xmx12288m -Xms12288m
-XX:MaxHeapFreeRatio=40 -XX:MinHeapFreeRatio=15 -XX:-UseGCOverheadLimit"
  fi
fi

export HADOOP_HEAPSIZE=2048
```

You can use either the Concurrent Collector or the new Parallel Collector for garbage collection by passing -XX:+UseConcMarkSweepGC or -XX:+UseParNewGC in the HADOOP_OPTS lines above. To enable the garbage

collection overhead limit, remove the `-XX:-UseGCOverheadLimit` setting or change it to `-XX:+UseGCOverheadLimit`.

Set the PermGen space for Java garbage collection to `512M` for all in the `JAVA_OPTS` environment variable. For example:

```
set JAVA_OPTS="-Xms256m -Xmx1024m -XX:PermSize=512m -XX:MaxPermSize=512m"
```

Configuration for WebHCat

If you want to use WebHCat, you need to set the `PYTHON_CMD` variable in `/etc/default/hive-webhcat-server` after installing Hive; for example:

```
export PYTHON_CMD=/usr/bin/python
```

Upgrading Hive

Upgrade Hive on all the hosts on which it is running including both servers and clients.

Warning: Because of concurrency and security issues, HiveServer1 and the Hive CLI are deprecated in CDH 5 and will be removed in a future release. Cloudera recommends you migrate to Beeline and HiveServer2 as soon as possible. The Hive CLI is not needed if you are using Beeline with HiveServer2.

Note: To see which version of Hive is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Checklist to Help Ensure Smooth Upgrades

The following best practices for configuring and maintaining Hive will help ensure that upgrades go smoothly.

- Configure periodic backups of the metastore database. Use `mysqldump`, or the equivalent for your vendor if you are not using MySQL.
- Make sure `datanucleus.autoCreateSchema` is set to false (in all types of database) and `datanucleus.fixedDatastore` is set to true (for MySQL and Oracle) in *all* `hive-site.xml` files. See the configuration instructions for more information about setting the properties in `hive-site.xml`.
- Insulate the metastore database from users by running the metastore service in Remote mode. If you do not follow this recommendation, make sure you remove `DROP`, `ALTER`, and `CREATE` privileges from the Hive user configured in `hive-site.xml`. See Configuring the Hive Metastore for CDH for complete instructions for each type of supported database.

Warning: Make sure you have read and understood all incompatible changes and known issues before you upgrade Hive.

Upgrading Hive from a Lower Version of CDH 5

The instructions that follow assume that you are upgrading Hive as part of a CDH 5 upgrade, and have already performed the steps under Upgrading from an Earlier CDH 5 Release to the Latest Release.

Important:

- If you are currently running Hive under MRv1, check for the following property and value in `/etc/mapred/conf/mapred-site.xml`:

```
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

  Remove this property before you proceed; otherwise Hive queries spawned from MapReduce jobs will fail with a null pointer exception (NPE).
- If you have installed the `hive-hcatalog-server` package in the past, you must remove it before you proceed; otherwise the upgrade will fail.
- If you are upgrading Hive from CDH 5.0.5 to CDH 5.4, 5.3 or 5.2 on Debian 7.0, and a Sentry version higher than 5.0.4 and lower than 5.1.0 is installed, you must upgrade Sentry before upgrading Hive; otherwise the upgrade will fail. See Apache Hive Known Issues for more details.
- CDH 5.2 and higher clients cannot communicate with CDH 5.1 and lower servers. This means that you must upgrade the server before the clients.

To upgrade Hive from a lower version of CDH 5, proceed as follows.

Step 1: Stop all Hive Processes and Daemons

Warning: You **must** make sure no Hive processes are running. If Hive processes are running during the upgrade, the new version will not work correctly.

1. Stop any HiveServer processes that are running:

```
$ sudo service hive-server stop
```

2. Stop any HiveServer2 processes that are running:

```
$ sudo service hive-server2 stop
```

3. Stop the metastore:

```
$ sudo service hive-metastore stop
```

Step 2: Install the new Hive version on all hosts (Hive servers and clients)

See Installing Hive on page 331

Step 3: Verify that the Hive Metastore is Properly Configured

See Configuring the Hive Metastore for CDH for detailed instructions.

Step 4: Upgrade the Metastore Schema

Important:

- Cloudera recommends that you make a backup copy of your metastore database before running the `schematool` or the upgrade scripts. You might need this backup copy if there are problems during the upgrade or if you need to downgrade to a previous version.
- You *must* upgrade the metastore schema to the version corresponding to the new version of Hive before starting Hive after the upgrade. Failure to do so may result in metastore corruption.

To upgrade the Hive metastore schema, you can use either the Hive `schematool` or use the schema upgrade scripts that are provided with the Hive package. Cloudera recommends that you use the `schematool`.

Using Hive schematool (Recommended):

The Hive distribution includes a command-line tool for Hive metastore schema manipulation called `schematool`. This tool can be used to initialize the metastore schema for the current Hive version. It can also upgrade the schema from an older version to the current one. You must add properties to the `hive-site.xml` before you can use it. See Using the Hive Schema Tool in CDH for information about how to set the tool up and for usage examples. To upgrade the schema, use the `upgradeSchemaFrom` option to specify the version of the schema you are currently using. For example, if you are upgrading a MySQL metastore schema from Hive 0.13.1, use the following syntax:

```
$ schematool -dbType mysql -passWord <db_user_pswd> -upgradeSchemaFrom
  0.13.1 -userName <db_user_name>
Metastore connection URL:
jdbc:mysql://<cluster_address>:3306/<user_name>?useUnicode=true&characterEncoding=UTF-8
Metastore Connection Driver : com.mysql.jdbc.Driver
Metastore connection User: <user_name>
Starting upgrade metastore schema from version 0.13.1 to <new_version>
Upgrade script upgrade-0.13.1-to-<new_version>.mysql.sql
Completed pre-0-upgrade-0.13.1-to-<new_version>.mysql.sql
Completed upgrade-0.13.1-to-<new_version>.mysql.sql
schemaTool completed
```

Note: The `upgradeSchemaFrom` option requires the Hive version and not the CDH version. See CDH 5 Packaging and Tarball Information for information about which Hive version ships with each CDH release.

Using Schema Upgrade Scripts:

Navigate to the directory where the schema upgrade scripts are located:

- If you installed CDH with parcels, the scripts are in the following location:

```
/opt/cloudera/parcels/CDH/lib/hive/scripts/metastore/upgrade/<database_name>
```

- If you installed CDH with packages, the scripts are in the following location:

```
/usr/lib/hive/scripts/metastore/upgrade/<database_name>
```

For example, if your Hive metastore is MySQL and you installed CDH with packages, navigate to `/usr/lib/hive/scripts/metastore/upgrade/mysql`.

Run the appropriate schema upgrade scripts in order. Start with the script for your database type and Hive version, and run all subsequent scripts.

For example, if you are currently running Hive 0.13.1 with MySQL and upgrading to Hive 1.1.0, start with the script for 0.13.0 to 0.14.0 for MySQL, and then run the script for Hive 0.14.0 to 1.1.0.

Important: If there are scripts with file names that start with `pre-`, like `pre-0-upgrade-1.1.0-to-1.1.0-cdh5.12.0.<database_type>.sql`, run them before running the script without `pre-` in the name. For example, if you are upgrading to CDH 5.12, run `pre-0-upgrade-1.1.0-to-1.1.0-cdh5.12.0` before you run `upgrade-1.1.0-to-1.1.0-cdh5.12.0.<database_type>.sql`.

For more information about using the scripts to upgrade the schema, see the README in the directory with the scripts.

Step 5: Start the Metastore, HiveServer2, and Beeline

See:

- Starting the Hive Metastore in CDH
- Starting, Stopping, and Using HiveServer2 in CDH

The upgrade is now complete.

Troubleshooting: If you failed to upgrade the metastore

If you failed to upgrade the metastore as instructed above, proceed as follows.

1. Identify the problem.

   The symptoms are as follows:

   - Hive stops accepting queries.
   - In a cluster managed by Cloudera Manager, the Hive Metastore canary fails.
   - An error such as the following appears in the Hive Metastore Server logs:

```
Hive Schema version 0.13.0 does not match metastore's schema version 0.12.0 Metastore
is not upgraded or corrupt.
```

2. Resolve the problem.

   If the problem you are having matches the symptoms just described, do the following:

   1. Stop all Hive services; for example:

```
$ sudo service hive-server2 stop
$ sudo service hive-metastore stop
```

   2. Run the Hive schematool, as instructed here.

      Make sure the value you use for the `-upgradeSchemaFrom` option matches the version you are *currently running* (not the new version). For example, if the error message in the log is

```
Hive Schema version 0.13.0 does not match metastore's schema version 0.12.0 Metastore
is not upgraded or corrupt.
```

      then the value of `-upgradeSchemaFrom` must be `0.12.0`.

   3. Restart the Hive services you stopped.

HttpFS Installation

> **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).

Use the following sections to install and configure HttpFS:

About HttpFS

Apache Hadoop HttpFS is a service that provides HTTP access to HDFS.

HttpFS has a REST HTTP API supporting all HDFS filesystem operations (both read and write).

Common HttpFS use cases are:

- Read and write data in HDFS using HTTP utilities (such as `curl` or `wget`) and HTTP libraries from languages other than Java (such as Perl).
- Transfer data between HDFS clusters running different versions of Hadoop (overcoming RPC versioning issues), for example using Hadoop DistCp.
- Accessing WebHDFS using the Namenode WebUI port (default port 50070). Access to all data hosts in the cluster is required, because WebHDFS redirects clients to the datanode port (default 50075). If the cluster is behind a firewall, and you use WebHDFS to read and write data to HDFS, then Cloudera recommends you use the HttpFS server. The HttpFS server acts as a gateway. It is the only system that is allowed to send and receive data through the firewall.

HttpFS supports Hadoop pseudo-authentication, HTTP SPNEGO Kerberos, and additional authentication mechanisms using a plugin API. HttpFS also supports Hadoop proxy user functionality.

The `webhdfs` client file system implementation can access HttpFS using the Hadoop filesystem command (`hadoop fs`), by using Hadoop DistCp, and from Java applications using the Hadoop file system Java API.

The HttpFS HTTP REST API is interoperable with the WebHDFS REST HTTP API.

For more information about HttpFS, see Hadoop HDFS over HTTP.

HttpFS Packaging

There are two packaging options for installing HttpFS:

- The `hadoop-httpfs` RPM package
- The `hadoop-httpfs` Debian package

You can also download a Hadoop tarball, which includes HttpFS, from here.

HttpFS Prerequisites

Prerequisites for installing HttpFS are:

- An operating system supported by CDH 5.
- Java: see Java Development Kit Installation for details.

> **Note:**
>
> To see which version of HttpFS is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes. CDH 5 Hadoop works with the CDH 5 version of HttpFS.

Installing HttpFS

HttpFS is distributed in the `hadoop-httpfs` package. To install it, use your preferred package manager application. Install the package on the system that will run the HttpFS server.

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install the HttpFS package on a RHEL-compatible system:

```
$ sudo yum install hadoop-httpfs
```

To install the HttpFS server package on a SLES system:

```
$ sudo zypper install hadoop-httpfs
```

To install the HttpFS package on an Ubuntu or Debian system:

```
$ sudo apt-get install hadoop-httpfs
```

> **Note:**
>
> Installing the `httpfs` package creates an `httpfs` service configured to start HttpFS at system startup time.

You are now ready to configure HttpFS. See the next section.

Configuring HttpFS

When you install HttpFS from an RPM or Debian package, HttpFS creates all configuration, documentation, and runtime files in the standard Unix directories, as follows.

Type of File	Where Installed
Binaries	`/usr/lib/hadoop-httpfs/`
Configuration	`/etc/hadoop-httpfs/conf/`
Documentation	*for SLES:* `/usr/share/doc/packages/hadoop-httpfs/`
	for other platforms: `/usr/share/doc/hadoop-httpfs/`

Type of File	Where Installed
Data	`/var/lib/hadoop-httpfs/`
Logs	`/var/log/hadoop-httpfs/`
temp	`/var/tmp/hadoop-httpfs/`
PID file	`/var/run/hadoop-httpfs/`

Configuring the HDFS HttpFS Will Use

HttpFS reads the HDFS configuration from the `core-site.xml` and `hdfs-site.xml` files in `/etc/hadoop/conf/`. If necessary edit those files to configure the HDFS HttpFS will use.

Configuring the HttpFS Proxy User

Edit `core-site.xml` and define the Linux user that will run the HttpFS server as a Hadoop proxy user. For example:

```
<property>
<name>hadoop.proxyuser.httpfs.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.httpfs.groups</name>
<value>*</value>
</property>
```

Then restart Hadoop to make the proxy user configuration active.

Configuring HttpFS with Kerberos Security

To configure HttpFS with Kerberos Security, see HttpFS Authentication.

Starting the HttpFS Server

After you have completed all of the required configuration steps, you can start HttpFS:

```
$ sudo service hadoop-httpfs start
```

If you see the message `Server httpfs started!, status NORMAL` in the `httpfs.log` log file, the system has started successfully.

Note:

By default, HttpFS server runs on port 14000 and its URL is `http://<HTTPFS_HOSTNAME>:14000/webhdfs/v1`.

Stopping the HttpFS Server

To stop the HttpFS server:

```
$ sudo service hadoop-httpfs stop
```

Using the HttpFS Server with curl

You can use a tool such as `curl` to access HDFS using HttpFS. For example, to obtain the home directory of the user babu, use a command such as this:

```
$ curl "http://localhost:14000/webhdfs/v1?op=gethomedirectory&user.name=babu"
```

You should see output such as this:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie:
hadoop.auth="u=babu&p=babu&t=simple&e=1332977755010&s=JVfT4T785K4jeeLNWXK68rc/0xI=";
Version=1; Path=/
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 28 Mar 2012 13:35:55 GMT

{"Path":"\/user\/babu"}
```

See the WebHDFS REST API web page for complete documentation of the API.

Hue Installation

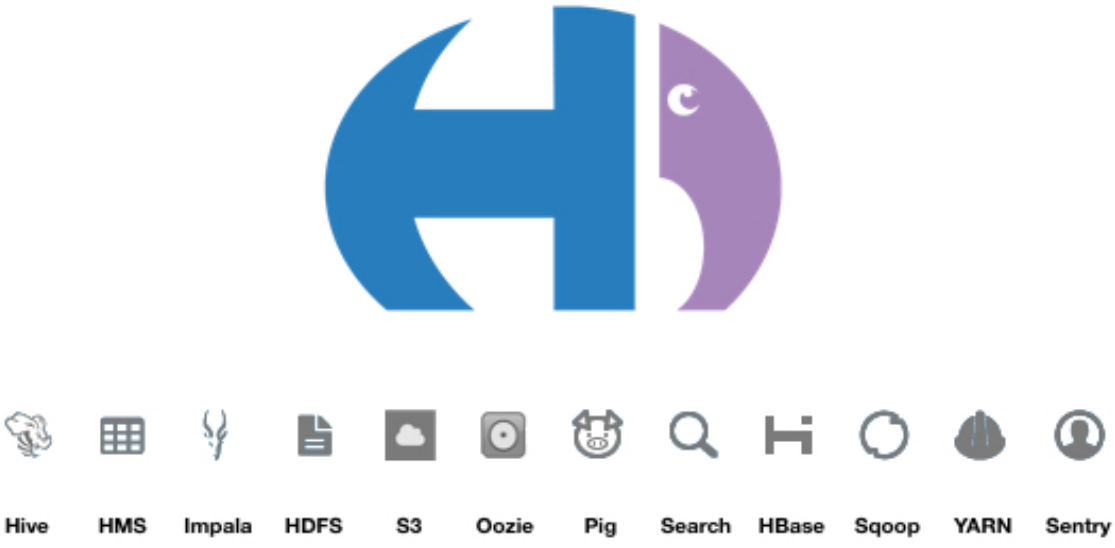
Hue Installation & Upgrade

Hue is included in Cloudera CDH, which you can install using one of the following methods:

- Path A – Installs Cloudera Manager and CDH using an automated installer and is intended only for non-production use. The installer configures an embedded PostgreSQL database for use with Hue, which is not suitable for production use.
- Path B – Installs Cloudera Manager using system packages and installs CDH using either packages or parcels.
- Path C – Installs Cloudera Manager using tarballs and CDH using parcels.

See Installing Cloudera Manager and CDH on page 54.

The Hue Server is a container web application that sits between your CDH installation and the browser. The Hue server hosts a suite of Hue applications and communicates with CDH component servers.



Supported Browsers for Hue

Hue works with the two most recent LTS (long term support) or ESR (extended support release) browsers. Cookies and JavaScript must be on.

- **Chrome**: Version history
- **Firefox**: Version history
- **Internet Explorer**: Version history
- **Safari** (Mac only): Version history

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

Important: To see all icons in the Hue Web UI, users with IE and HTTPS must add a Load Balancer.

Upgrading Hue

Note:

To see which version of Hue is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Upgrading Hue from CDH 4 to CDH 5

If you have already removed Hue as part of your upgrade to CDH 5, skip to Installing and Configuring Hue.

Step 1: Stop the Hue Server

See Starting and Stopping the Hue Server on page 361.

Step 2: Uninstall the Old Version of Hue

- On RHEL systems:

```
$ sudo yum remove hue-common
```

- On SLES systems:

```
$ sudo zypper remove hue-common
```

- On Ubuntu or Debian systems:

```
sudo apt-get remove hue-common
```

Step 3: Install Hue 3.x

Follow the instructions under Installing Hue.

If Using MySQL as Hue Backend: You may face issues after the upgrade if the default engine for MySQL does not match the engine used by the Hue tables. To confirm the match:

1. Open the `my.cnf` file for MySQL, search for `"default-storage-engine"` and note its value.
2. Connect to MySQL and run the following commands:

```
use hue;
show create table auth_user;
```

3. Search for the `"ENGINE="` line and confirm that its value matches the one for the `"default-storage-engine"` above.

If the default engines do not match, Hue will display a warning on its start-up page (`http://$HUE_HOST:$HUE_PORT/about`). Work with your database administrator to convert the current Hue MySQL tables to the engine in use by MySQL, as noted by the `"default-storage-engine"` property.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by dpkg.

Step 4: Start the Hue Server

See Starting and Stopping the Hue Server on page 361.

Upgrading Hue from an Earlier CDH 5 Release

You can upgrade Hue either as part of an overall upgrade to the latest CDH 5 release (see Upgrading from an Earlier CDH 5 Release to the Latest Release) or independently. To upgrade Hue from an earlier CDH 5 release to the latest CDH 5 release, proceed as follows.

Step 1: Stop the Hue Server

See Starting and Stopping the Hue Server on page 361.

Warning:

You **must** stop Hue. If Hue is running during the upgrade, the new version will not work correctly.

Step 2: Install the New Version of Hue

Follow the instructions under Installing Hue on page 344.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by dpkg.

Step 3: Start the Hue Server

See Starting and Stopping the Hue Server on page 361.

Installing Hue

> **Warning:** This page is deprecated and will be removed in C5.13. See the Hue Guide for new docs on installing Hue.

This section describes Hue installation and configuration on a cluster. The steps in this section apply whether you are installing on a single machine in pseudo-distributed mode, or on a cluster.

Install Python 2.6 or 2.7

CDH 5 Hue works only with the default system Python version of the operating system it is being installed on. For example, on RHEL/CentOS 6, you need Python 2.6 to start Hue.

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

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

Installing the Hue Packages

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera yum, zypper/YaST or apt repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

You must install the hue-common package on the machine where you will run the Hue Server. In addition, if you will be using Hue with MRv1, you must install the hue-plugins package on the system where you are running the JobTracker. (In pseudo-distributed mode, these will all be the same system.)

The hue meta-package installs the hue-common package and all the Hue applications; you also need to install hue-server, which contains the Hue start and stop scripts.

> **Note:** If you do not know which system your JobTracker is on, install the hue-plugins package on every node in the cluster.

On RHEL systems:

- On the Hue Server machine, install the hue package:

```
$ sudo yum install hue
```

- For MRv1: on the system that hosts the JobTracker, if different from the Hue server machine, install the hue-plugins package:

```
$ sudo yum install hue-plugins
```

On SLES systems:

- On the Hue Server machine, install the hue package:

```
$ sudo zypper install hue
```

- For MRv1: on the system that hosts the JobTracker, if different from the Hue server machine, install the `hue-plugins` package:

```
$ sudo zypper install hue-plugins
```

On Ubuntu or Debian systems:

- On the Hue Server machine, install the hue package:

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

- For MRv1: on the system that hosts the JobTracker, if different from the Hue server machine, install the `hue-plugins` package:

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

Important: For all operating systems, restart the Hue service once installation is complete. See Starting and Stopping the Hue Server on page 361.

Hue Dependencies

The following table shows the components that are dependencies for the different Hue applications:

Component	Dependent Applications
HDFS	Core, File Browser
MapReduce	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
YARN	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
Oozie	Job Designer, Oozie Editor/Dashboard
Hive	Hive Editor, Metastore Tables
Impala	Impala Editor, Metastore Tables
HBase	HBase Browser
Pig	Pig Editor, Oozie
Search	Solr Search
Spark	Spark
Sentry	Hadoop Security
Sqoop	Oozie
Sqoop 2	Sqoop Transfer
ZooKeeper	ZooKeeper

Configuring CDH Components for Hue

To enable communication between the Hue Server and CDH components, you must make minor changes to your CDH installation by adding the properties described in this section to your CDH configuration files in `/etc/hadoop-0.20/conf/` or `/etc/hadoop/conf/`. If you are installing on a cluster, make the following configuration changes to your existing CDH installation on **each node** in your cluster.

WebHDFS or HttpFS Configuration

Hue can use either of the following to access HDFS data:

- **WebHDFS** provides high-speed data transfer with good locality because clients talk directly to the DataNodes inside the Hadoop cluster.
- **HttpFS** is a proxy service appropriate for integration with external systems that are not behind the cluster's firewall.

Both WebHDFS and HttpFS use the HTTP REST API so they are fully interoperable, but Hue must be configured to use one or the other. For HDFS HA deployments, you must use HttpFS.

To configure Hue to use either WebHDFS or HttpFS, do the following steps:

1. **For WebHDFS only:**
   a. Add the following property in `hdfs-site.xml` to enable WebHDFS in the NameNode and DataNodes:

```
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
```

   b. Restart your HDFS cluster.

2. Configure Hue as a proxy user for all other users and groups, meaning it may submit a request on behalf of any other user:

   WebHDFS: Add to `core-site.xml`:

```
<!-- Hue WebHDFS proxy user setting -->
<property>
  <name>hadoop.proxyuser.hue.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.hue.groups</name>
  <value>*</value>
</property>
```

   HttpFS: Verify that `/etc/hadoop-httpfs/conf/httpfs-site.xml` has the following configuration:

```
<!-- Hue HttpFS proxy user setting -->
<property>
  <name>httpfs.proxyuser.hue.hosts</name>
  <value>*</value>
</property>
<property>
  <name>httpfs.proxyuser.hue.groups</name>
  <value>*</value>
</property>
```

   If the configuration is not present, add it to `/etc/hadoop-httpfs/conf/httpfs-site.xml` and restart the HttpFS daemon.

3. Verify that `core-site.xml` has the following configuration:

```
<property>
<name>hadoop.proxyuser.httpfs.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.httpfs.groups</name>
<value>*</value>
</property>
```

   If the configuration is not present, add it to `/etc/hadoop/conf/core-site.xml` and restart Hadoop.

4. With root privileges, update `hadoop.hdfs_clusters.default.webhdfs_url` in `hue.ini` to point to the address of either WebHDFS or HttpFS.

```
[hadoop]
[[hdfs_clusters]]
[[[default]]]
# Use WebHdfs/HttpFs as the communication mechanism.
```

WebHDFS:

```
...
webhdfs_url=http://FQDN:50070/webhdfs/v1/
```

HttpFS:

```
...
webhdfs_url=http://FQDN:14000/webhdfs/v1/
```

Note: If the `webhdfs_url` is uncommented and explicitly set to the empty value, Hue falls back to using the Thrift plugin used in Hue 1.x. This is not recommended.

MRv1 Configuration

Hue communicates with the JobTracker using the Hue plugin, which is a `.jar` file that should be placed in your MapReduce `lib` directory.

Important: The `hue-plugins` package installs the Hue plugins in your MapReduce `lib` directory, `/usr/lib/hadoop/lib`. If you are not using the package-based installation procedure, perform the following steps to install the Hue plugins.

If your JobTracker and Hue Server are located on the same host, copy the file over. If you are currently using CDH 4, your MapReduce library directory might be in `/usr/lib/hadoop/lib`.

```
$ cd /usr/lib/hue
$ cp desktop/libs/hadoop/java-lib/hue-plugins-*.jar /usr/lib/hadoop-0.20-mapreduce/lib
```

If your JobTracker runs on a different host, `scp` the Hue plugins `.jar` file to the JobTracker host.

Add the following properties to `mapred-site.xml`:

```
<property>
  <name>jobtracker.thrift.address</name>
  <value>0.0.0.0:9290</value>
</property>
<property>
  <name>mapred.jobtracker.plugins</name>
  <value>org.apache.hadoop.thriftfs.ThriftJobTrackerPlugin</value>
  <description>Comma-separated list of jobtracker plug-ins to be activated.</description>
</property>
```

You can confirm that the plugins are running correctly by tailing the daemon logs:

```
$ tail --lines=500 /var/log/hadoop-0.20-mapreduce/hadoop*jobtracker*.log | grep
ThriftPlugin
2009-09-28 16:30:44,337 INFO org.apache.hadoop.thriftfs.ThriftPluginServer: Starting
Thrift server
2009-09-28 16:30:44,419 INFO org.apache.hadoop.thriftfs.ThriftPluginServer:
Thrift server listening on 0.0.0.0:9290
```

Note: If you enable ACLs in the JobTracker, you must add users to the JobTracker `mapred.queue.default.acl-administer-jobs` property to allow Hue to display jobs in the Job Browser application. For example, to give the hue user access to the JobTracker, you would add the following property:

```
<property>
  <name>mapred.queue.default.acl-administer-jobs</name>
  <value>hue</value>
</property>
```

Repeat this for every user that requires access to the job details displayed by the JobTracker.

If you have any mapred queues besides "default", you must add a property for each queue:

```
<property>
<name>mapred.queue.default.acl-administer-jobs</name>
<value>hue</value>
</property>
<property>
<name>mapred.queue.queue1.acl-administer-jobs</name>
<value>hue</value>
</property>
<property>
<name>mapred.queue.queue2.acl-administer-jobs</name>
<value>hue</value>
</property>
```

Oozie Configuration

To run DistCp, Streaming, Pig, Sqoop, and Hive jobs in Job Designer or the Oozie Editor/Dashboard application, you must make sure the Oozie shared libraries are installed for the correct version of MapReduce (MRv1 or YARN). See Installing the Oozie ShareLib in Hadoop HDFS for instructions.

To configure Hue as a default proxy user, add the following properties to `/etc/oozie/conf/oozie-site.xml`:

```
<!-- Default proxyuser configuration for Hue -->
<property>
    <name>oozie.service.ProxyUserService.proxyuser.hue.hosts</name>
    <value>*</value>
</property>
<property>
    <name>oozie.service.ProxyUserService.proxyuser.hue.groups</name>
    <value>*</value>
</property>
```

Search Configuration

See Search Configuration on page 353 for details on how to configure the Search application for Hue.

HBase Configuration

See HBase Configuration on page 353 for details on how to configure the HBase Browser application.

Note: HBase Browser requires Thrift Server 1 to be running.

Hive Configuration

The Beeswax daemon has been replaced by HiveServer2. Hue should therefore point to a running HiveServer2. This change involved the following major updates to the `[beeswax]` section of the Hue configuration file, `hue.ini`.

```
[beeswax]
  # Host where Hive server Thrift daemon is running.
```

```
# If Kerberos security is enabled, use fully-qualified domain name (FQDN).
## hive_server_host=<FQDN of HiveServer2>

# Port where HiveServer2 Thrift server runs on.
## hive_server_port=10000
```

Existing Hive Installation

In the Hue configuration file `hue.ini`, modify `hive_conf_dir` to point to the directory containing `hive-site.xml`.

No Existing Hive Installation

Familiarize yourself with the configuration options in `hive-site.xml`. See Hive Installation. Having a `hive-site.xml` is optional but often useful, particularly on setting up a metastore. You can locate it using the `hive_conf_dir` configuration variable.

Permissions

See File System Permissions in the Hive Installation section.

Hue Configuration

This section describes the Hue configuration file, `hue.ini`. The location of the Hue configuration file varies depending on how Hue is installed. The location of the Hue configuration folder is displayed when you view the Hue configuration.

> **Note:** Only the root user can edit the Hue configuration file.

Viewing the Hue Configuration

> **Note:** You must be a Hue superuser to view the Hue configuration.

When you log in to Hue, the start-up page displays information about any misconfiguration detected.

To view the Hue configuration, do one of the following:

- Visit `http://myserver:port` and click the **Configuration** tab.
- Visit `http://myserver:port/desktop/dump_config`.

Hue Server Configuration

This section describes Hue Server settings.

Specifying the Hue Server HTTP Address

These configuration properties are under the `[desktop]` section in the Hue configuration file.

Hue uses the CherryPy web server. You can use the following options to change the IP address and port that the web server listens on. The default setting is port 8888 on all configured IP addresses.

```
# Webserver listens on this address and port
http_host=0.0.0.0
http_port=8888
```

Specifying the Secret Key

For security, you should specify the secret key that is used for secure hashing in the session store:

1. Open the Hue configuration file.

2. In the [desktop] section, set the secret_key property to a long series of random characters (30 to 60 characters is recommended). For example,

```
secret_key=qpbdxoewsqlkhztybvfidtvwekftusgdlofbcfghaswuicmqp
```

> **Note:** If you do not specify a secret key, your session cookies will not be secure. Hue will run but it will also display error messages telling you to set the secret key.

Authentication

By default, the first user who logs in to Hue can choose any username and password and automatically becomes an administrator. This user can create other user and administrator accounts. Hue users should correspond to the Linux users who will use Hue; make sure you use the same name as the Linux username.

By default, user information is stored in the Hue database. However, the authentication system is pluggable. You can configure authentication to use an LDAP directory (Active Directory or OpenLDAP) to perform the authentication, or you can import users and groups from an LDAP directory. See Configuring an LDAP Server for User Admin on page 354.

For more information, see the Hue SDK Documentation.

Configuring the Hue Server for TLS/SSL

You can optionally configure Hue to serve over HTTPS. As of CDH 5, pyOpenSSL is now part of the Hue build and does not need to be installed manually. To configure TLS/SSL, perform the following steps from the root of your Hue installation path:

1. Configure Hue to use your private key by adding the following options to the Hue configuration file:

```
ssl_certificate=/path/to/certificate
ssl_private_key=/path/to/key
```

> **Note:** Hue can only support a private key without a passphrase.

2. On a production system, you should have an appropriate key signed by a well-known Certificate Authority. If you're just testing, you can create a self-signed key using the openssl command that may be installed on your system:

```
# Create a key
$ openssl genrsa 1024 > host.key
# Create a self-signed certificate
$ openssl req -new -x509 -nodes -sha1 -key host.key > host.cert
```

> **Note:** Uploading files using the Hue File Browser over HTTPS requires using a proper TLS/SSL Certificate. Self-signed certificates do not work.

Authentication Backend Options for Hue

The table below gives a list of authentication backends Hue can be configured with including the recent SAML backend that enables single sign-on authentication. The backend configuration property is available in the [[auth]] section under [desktop].

backend	django.contrib.auth.backends.ModelBackend	This is the default authentication backend used by Django.

desktop.auth.backend.AllowAllBackend	This backend does not require a password for users to log in. All users are automatically authenticated and the username is set to what is provided.
desktop.auth.backend.AllowFirstUserDjangoBackend	This is the default Hue backend. It creates the first user that logs in as the super user. After this, it relies on Django and the user manager to authenticate users.
desktop.auth.backend.LdapBackend	Authenticates users against an LDAP service.
desktop.auth.backend.PamBackend	Authenticates users with PAM (pluggable authentication module). The authentication mode depends on the PAM module used.
desktop.auth.backend.SpnegoDjangoBackend	SPNEGO is an authentication mechanism negotiation protocol. Authentication can be delegated to an authentication server, such as a Kerberos KDC, depending on the mechanism negotiated.
desktop.auth.backend.RemoteUserDjangoBackend	Authenticating remote users with the Django backend.
desktop.auth.backend.OAuthBackend	Delegates authentication to a third-party OAuth server.
libsaml.backend.SAML2Backend	Secure Assertion Markup Language (SAML) single sign-on (SSO) backend. Delegates authentication to the configured Identity Provider. See Configuring Hue for SAML for more details.

Note: All backends that delegate authentication to a third-party authentication server eventually import users into the Hue database. While the metadata is stored in the database, user authentication will still take place outside Hue.

Beeswax Configuration

In the [beeswax] section of the configuration file, you can optionally specify the following:

hive_server_host	The fully qualified domain name or IP address of the host running HiveServer2.
hive_server_port	The port of the HiveServer2 Thrift server. Default: 10000.
hive_conf_dir	The directory containing hive-site.xml, the HiveServer2 configuration file.

Impala Query UI Configuration

In the [impala] section of the configuration file, you can optionally specify the following:

server_host	The hostname or IP address of the Impala Server. Default: localhost.
server_port	The port of the Impalad Server.

	Default: `21050`
`impersonation_enabled`	Turn on/off impersonation mechanism when talking to Impala. Default: `False`

Note: The Impala load balancer must assign each Hue user the same Impalad instance, or queries may fail with errorMessage='Invalid query handle'.

DB Query Configuration

The DB Query app can have any number of databases configured in the `[[databases]]` section under `[librdbms]`. A database is known by its section name (`sqlite`, `mysql`, `postgresql`, and `oracle` as in the list below). For details on supported databases and versions, see CDH and Cloudera Manager Supported Databases.

Database Type	Configuration Properties

SQLite: `[[[sqlite]]]`	`# Name to show in the UI.`
	`## nice_name=SQLite`
	
	`# For SQLite, name defines the path to the database.`
	
	`## name=/tmp/sqlite.db`
	
	`# Database backend to use.`
	`## engine=sqlite`

MySQL, Oracle or PostgreSQL:
`[[[mysql]]]`
Note: Replace with `oracle` or `postgresql` as required.
	`# Name to show in the UI.`
	`## nice_name="My SQL DB"`
	
	`# For MySQL and PostgreSQL, name is the name of the database.`
	`# For Oracle, Name is instance of the Oracle server. For express edition`
	`# this is 'xe' by default.`
	`## name=mysqldb`
	
	`# Database backend to use. This can be:`
	`# 1. mysql`
	`# 2. postgresql`
	`# 3. oracle`
	`## engine=mysql`
	
	`# IP or hostname of the database to connect to.`
	
	`## host=localhost`
	
	`# Port the database server is listening to. Defaults are:`
	`# 1. MySQL: 3306`
	`# 2. PostgreSQL: 5432`
	`# 3. Oracle Express Edition: 1521`
	`## port=3306`
	
	`# Username to authenticate with when connecting to the database.`
	`## user=example`
	
	`# Password matching the username to authenticate with when`
	`# connecting to the database.`
	`## password=example`

Pig Editor Configuration

In the `[pig]` section of the configuration file, you can optionally specify the following:

remote_data_dir	Location on HDFS where the Pig examples are stored.

Sqoop Configuration

In the [sqoop] section of the configuration file, you can optionally specify the following:

server_url	The URL of the sqoop2 server.

Job Browser Configuration

By default, any user can see submitted job information for all users. You can restrict viewing of submitted job information by optionally setting the following property under the [jobbrowser] section in the Hue configuration file:

share_jobs	Indicate that jobs should be shared with all users. If set to false, they will be visible only to the owner and administrators.

Job Designer

In the [jobsub] section of the configuration file, you can optionally specify the following:

remote_data_dir	Location in HDFS where the Job Designer examples and templates are stored.

Oozie Editor/Dashboard Configuration

By default, any user can see all workflows, coordinators, and bundles. You can restrict viewing of workflows, coordinators, and bundles by configuring either of the following properties under the [oozie] section of the Hue configuration file:

oozie_jobs_count	Maximum number of Oozie workflows or coordinators or bundles to retrieve in one API call.
remote_data_dir	The location in HDFS where Oozie workflows are stored.

As of CDH 5.4, Hue uses a new editor for Oozie documents. If documents were created in the old editor, they won't immediately be available to users other than the document owner. To resolve this problem, the document owner can share any documents again. Alternatively, you can revert to the old editor by setting the flag use_new_editor=false in the [oozie] section of the Hue configuration file.

Also see Liboozie Configuration on page 358.

Search Configuration

In the [search] section of the configuration file, you can optionally specify the following:

security_enabled	Indicate whether Solr requires clients to perform Kerberos authentication.
empty_query	Query sent when no term is entered. Default: *:*.
solr_url	URL of the Solr server.

HBase Configuration

In the [hbase] section of the configuration file, you can optionally specify the following:

truncate_limit	Hard limit of rows or columns per row fetched before truncating. Default: 500
hbase_clusters	Comma-separated list of HBase Thrift servers for clusters in the format of "(name|host:port)".

	Default: `(Cluster|localhost:9090)`

HBase Impersonation: - To enable the HBase app to use impersonation, perform the following steps:

1. Ensure you have a secure HBase Thrift server.
2. Enable impersonation for the Thrift server by adding the following properties to `hbase-site.xml` on each Thrift gateway:

```
<property>
  <name>hbase.regionserver.thrift.http</name>
  <value>true</value>
</property>
<property>
  <name>hbase.thrift.support.proxyuser</name>
  <value>true/value>
</property>
```

   See: Configure doAs Impersonation for the HBase Thrift Gateway.

3. Configure Hue to point to a valid HBase configuration directory. You will find this property under the `[hbase]` section of the `hue.ini` file.

`hbase_conf_dir`	HBase configuration directory, where hbase-site.xml is located. Default: `/etc/hbase/conf`

User Admin Configuration

In the `[useradmin]` section of the configuration file, you can optionally specify the following:

`default_user_group`	The name of the group to which a manually created user is automatically assigned. Default: default.

Configuring an LDAP Server for User Admin

User Admin can interact with an LDAP server, such as Active Directory, in one of two ways:

- You can import user and group information from your current Active Directory infrastructure using the LDAP Import feature in the User Admin application. User authentication is then performed by User Admin based on the imported user and password information. You can then manage the imported users, along with any users you create directly in User Admin. See Enabling Import of Users and Groups from an LDAP Directory on page 354.
- You can configure User Admin to use an LDAP server as the authentication back end, which means users logging in to Hue will authenticate to the LDAP server, rather than against a username and password kept in User Admin. In this scenario, your users must all reside in the LDAP directory. See Enabling the LDAP Server for User Authentication on page 355 for further information.

Enabling Import of Users and Groups from an LDAP Directory

User Admin can import users and groups from an Active Directory using the Lightweight Directory Authentication Protocol (LDAP). In order to use this feature, you must configure User Admin with a set of LDAP settings in the Hue configuration file.

Note: If you import users from LDAP, you must set passwords for them manually; password information is not imported.

1. In the Hue configuration file, configure the following properties in the `[[ldap]]` section:

Property	Description	Example
base_dn	The search base for finding users and groups.	base_dn="DC=mycompany,DC=com"
nt_domain	The NT domain to connect to (only for use with Active Directory).	nt_domain=mycompany.com
ldap_url	URL of the LDAP server.	ldap_url=ldap://auth.mycompany.com
ldap_cert	Path to certificate for authentication over TLS (optional).	ldap_cert=/mycertsdir/myTLScert
bind_dn	Distinguished name of the user to bind as – not necessary if the LDAP server supports anonymous searches.	bind_dn="CN=ServiceAccount,DC=mycompany,DC=com"
bind_password	Password of the bind user – not necessary if the LDAP server supports anonymous searches.	bind_password=P@ssw0rd

2. Configure the following properties in the [[[users]]] section:

Property	Description	Example
user_filter	Base filter for searching for users.	user_filter="objectclass=*"
user_name_attr	The username attribute in the LDAP schema.	user_name_attr=sAMAccountName

3. Configure the following properties in the [[[groups]]] section:

Property	Description	Example
group_filter	Base filter for searching for groups.	group_filter="objectclass=*"
group_name_attr	The username attribute in the LDAP schema.	group_name_attr=cn

Note: If you provide a TLS certificate, it must be signed by a Certificate Authority that is trusted by the LDAP server.

Enabling the LDAP Server for User Authentication

You can configure User Admin to use an LDAP server as the authentication back end, which means users logging in to Hue will authenticate to the LDAP server, rather than against usernames and passwords managed by User Admin.

Important:

Be aware that when you enable the LDAP back end for user authentication, user authentication by User Admin will be disabled. This means there will be no superuser accounts to log into Hue unless you take one of the following actions:

- Import one or more superuser accounts from Active Directory and assign them superuser permission.
- If you have already enabled the LDAP authentication back end, log into Hue using the LDAP back end, which will create a LDAP user. Then disable the LDAP authentication back end and use User Admin to give the superuser permission to the new LDAP user.

After assigning the superuser permission, enable the LDAP authentication back end.

1. In the Hue configuration file, configure the following properties in the `[[ldap]]` section:

Property	Description	Example
`ldap_url`	URL of the LDAP server, prefixed by `ldap://` or `ldaps://`	`ldap_url=ldap://auth.mycompany.com`
`search_bind_authentication`	Search bind authentication is now the default instead of direct bind. To revert to direct bind, the value of this property should be set to `false`. When using search bind semantics, Hue will ignore the following `nt_domain` and `ldap_username_pattern` properties.	`search_bind_authentication=false`
`nt_domain`	The NT domain over which the user connects (not strictly necessary if using `ldap_username_pattern`.	`nt_domain=mycompany.com`
`ldap_username_pattern`	Pattern for searching for usernames – Use <username> for the username parameter. For use when using LdapBackend for Hue authentication	`ldap_username_pattern="uid=<username>,ou=People,dc=mycompany,dc=com"`

2. If you are using TLS or secure ports, add the following property to specify the path to a TLS certificate file:

Property	Description	Example
`ldap_cert`	Path to certificate for authentication over TLS. **Note:** If you provide a TLS certificate, it must be signed by a Certificate Authority that is trusted by the LDAP server.	`ldap_cert=/mycertsdir/myTLScert`

3. In the `[[auth]]` sub-section inside `[desktop]` change the following:

`backend`	Change the setting of `backend` from
	`backend=desktop.auth.backend.AllowFirstUserDjangoBackend`
	to
	`backend=desktop.auth.backend.LdapBackend`

Hadoop Configuration

The following configuration variables are under the `[hadoop]` section in the Hue configuration file.

HDFS Cluster Configuration

Hue currently supports only one HDFS cluster, which you define under the `[[hdfs_clusters]]` sub-section. The following properties are supported:

`[[[default]]]`	The section containing the default settings.
`fs_defaultfs`	The equivalent of `fs.defaultFS` (also referred to as `fs.default.name`) in a Hadoop configuration.
`webhdfs_url`	The HttpFS URL. The default value is the HTTP port on the NameNode.

YARN (MRv2) and MapReduce (MRv1) Cluster Configuration

Job Browser can display both MRv1 and MRv2 jobs, but must be configured to display one type at a time by specifying either `[[yarn_clusters]]` or `[[mapred_clusters]]` sections in the Hue configuration file.

The following YARN cluster properties are defined under the under the `[[yarn_clusters]]` sub-section:

`[[[default]]]`	The section containing the default settings.
`resourcemanager_host`	The fully qualified domain name of the host running the ResourceManager.
`resourcemanager_port`	The port for the ResourceManager IPC service.
`submit_to`	If your Oozie is configured to use a YARN cluster, then set this to true. Indicate that Hue should submit jobs to this YARN cluster.
`proxy_api_url`	URL of the ProxyServer API. Default: `http://localhost:8088`
`history_server_api_url`	URL of the HistoryServer API Default: `http://localhost:19888`

The following MapReduce cluster properties are defined under the `[[mapred_clusters]]` sub-section:

`[[[default]]]`	The section containing the default settings.
`jobtracker_host`	The fully qualified domain name of the host running the JobTracker.
`jobtracker_port`	The port for the JobTracker IPC service.
`submit_to`	If your Oozie is configured with to use a 0.20 MapReduce service, then set this to true. Indicate that Hue should submit jobs to this MapReduce cluster.

Note: High Availability (MRv1):

Add High Availability (HA) support for your MRv1 cluster by specifying a failover JobTracker. You can do this by configuring the following property under the `[[[ha]]]` sub-section for MRv1.

```
# Enter the host on which you are running the failover JobTracker

# jobtracker_host=<localhost-ha>
```

High Availability (YARN):

Add the following `[[[ha]]]` section under the `[hadoop] > [[yarn_clusters]]` sub-section in `hue.ini` with configuration properties for a second ResourceManager. As long as you have the `logical_name` property specified as below, jobs submitted to Oozie will work. The Job Browser, however, will *not* work with HA in this case.

```
[[[ha]]]
resourcemanager_host=<second_resource_manager_host_FQDN>
resourcemanager_api_url=http://<second_resource_manager_host_URL>
proxy_api_url=<second_resource_manager_proxy_URL>
history_server_api_url=<history_server_API_URL>
resourcemanager_port=<port_for_RM_IPC>
security_enabled=false
submit_to=true
logical_name=XXXX
```

Liboozie Configuration

In the `[liboozie]` section of the configuration file, you can optionally specify the following:

`security_enabled`	Indicate whether Oozie requires clients to perform Kerberos authentication.
`remote_deployment_dir`	The location in HDFS where the workflows and coordinators are deployed when submitted by a non-owner.
`oozie_url`	The URL of the Oozie server.

Sentry Configuration

In the `[libsentry]` section of the configuration file, specify the following:

`hostname`	Hostname or IP of server. Default: `localhost`
`port`	The port where the Sentry service is running. Default: `8038`
`sentry_conf_dir`	Sentry configuration directory, where `sentry-site.xml` is located. Default: `/etc/sentry/conf`

Hue will also automatically pick up the HiveServer2 server name from Hive's `sentry-site.xml` file at `/etc/hive/conf`.

If you have enabled Kerberos for the Sentry service, allow Hue to connect to the service by adding the hue user to the following property in the `/etc/sentry/conf/sentry-store-site.xml` file.

```
<property>
  <name>sentry.service.allow.connect</name>
  <value>impala,hive,solr,hue</value>
</property>
```

ZooKeeper Configuration

> **Warning:**
> CDH does not support using Zookeeper with Hue.

In the `[zookeeper]` section of the configuration file, you can specify the following:

`host_ports`	Comma-separated list of ZooKeeper servers in the format "host:port". Example: `localhost:2181,localhost:2182,localhost:2183`
`rest_url`	The URL of the REST Contrib service (required for znode browsing). Default: `http://localhost:9998`

Setting up REST Service for ZooKeeper

ZooKeeper Browser requires the ZooKeeper REST service to be running. Follow the instructions below to set this up.

Step 1: Git and build the ZooKeeper repository

```
git clone https://github.com/apache/zookeeper
cd zookeeper
ant
Buildfile: /home/hue/Development/zookeeper/build.xml

init:
[mkdir] Created dir: /home/hue/Development/zookeeper/build/classes
[mkdir] Created dir: /home/hue/Development/zookeeper/build/lib
[mkdir] Created dir: /home/hue/Development/zookeeper/build/package/lib
[mkdir] Created dir: /home/hue/Development/zookeeper/build/test/lib
…
```

Step 2: Start the REST service

```
cd src/contrib/rest
nohup ant run&
```

Step 3: Update ZooKeeper configuration properties (if required)

If ZooKeeper and the REST service are not on the same machine as Hue, update the Hue configuration file and specify the correct hostnames and ports as shown in the sample configuration below:

```
[zookeeper]
 ...
 [[clusters]]
  ...
  [[[default]]]
          # Zookeeper ensemble. Comma separated list of Host/Port.
          # e.g. localhost:2181,localhost:2182,localhost:2183
          ## host_ports=localhost:2181

          # The URL of the REST contrib service
          ## rest_url=http://localhost:9998
```

You should now be able to successfully run the ZooKeeper Browser app.

Administering Hue

This page explains how to manage and operate a Hue installation:

Hue Superusers and Users

The Hue User Admin application provides two levels of privileges: superusers and users.

- **Superusers** — The first user who logs into Hue after its installation becomes the first superuser. Superusers can perform administrative functions such as:
  - Add and delete users
  - Add and delete groups
  - Assign permissions to groups
  - Change a user to a superuser
  - Import users and groups from an LDAP server
- **Users** — Can change their name, email address, and password. They can login to Hue and run Hue applications according to their group permissions.

Hue UI Customization

The Hue Web UI has the following customization options in `hue.ini` under **[desktop]** > **[[custom]]**.

Banner

You can add a custom banner to the Hue Web UI by applying HTML directly to the property, `banner_top_html`. For example:

```
banner_top_html=<H4>My company's custom Hue Web UI banner</H4>
```

To configure a banner in Cloudera Manager:

1. Go to the **Hue** service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Hue Server** and **Category** > **Advanced**.
4. Locate **Top Banner Custom HTML** and input your desired HTML in the text field.
5. Click **Save Changes** to commit the changes.
6. Select **Actions** > **Restart** and, when done, click **Close**.
7. Click **Web UI** to view your changes.

Splash Screen

You can customize a splash screen on the login page by applying HTML directly to the property, `login_splash_html`. For example:

```
login_splash_html=WARNING: You are required to have authorization before you proceed.
```

To configure a splash screen in Cloudera Manager:

1. Go to the **Hue** service and click the **Configuration** tab.
2. Filter by**Scope** > **HUE-*n* (Service-Wide)** and **Category** > **Advanced**.
3. Locate the property, **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini**, and add your HTML in the text field:

```
[dekstop]
[[custom]]
login_splash_html=WARNING: You are required to have authorization before you proceed.
```

4. Click **Save Changes** to commit the changes.
5. Select **Actions** > **Restart** and, when done, click **Close**.
6. Click **Web UI** to view your changes.

Cache Timeout

You enable Hue UI caching by setting a timeout in milliseconds. The default is 86400000 milliseconds or one day. Set the timeout to 0 to disable caching.

As with the splash screen, set the cache timeout in **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini**.

```
[dekstop]
[[custom]]
cacheable_ttl=86400000
```

Starting and Stopping the Hue Server

The `hue-server` package includes service scripts to start and stop the Hue Server.

To start the Hue Server:

```
$ sudo service hue start
```

To restart the Hue Server:

```
$ sudo service hue restart
```

To stop the Hue Server:

```
$ sudo service hue stop
```

Configuring a Firewall for Hue

Hue currently requires that the machines in your cluster connect to each other freely over TCP. The machines outside your cluster must be able to open TCP port 8888 on the Hue Server (or the configured Hue web HTTP port) to interact with the system.

Anonymous Usage Data Collection

Hue tracks anonymized pages and application versions to gather information about application usage levels. The data collected does not include any hostnames or IDs.

For Hue 2.5.0 and higher, you can restrict this data collection by setting the `collect_usage` property to `false` in the `[desktop]` section in the Hue configuration file, `hue.ini`.

```
[desktop]
...
# Help improve Hue with anonymous usage analytics.
# Use Google Analytics to see how many times an application or specific section of an
application is used, nothing more.
## collect_usage=false
```

If you are using an earlier version of Hue, disable this data collection by going to Step 3 of Hue's Quick Start Wizard. Under **Anonymous usage analytics**, uncheck the box for **Check to enable usage analytics**.

Managing Hue Processes

The `supervisor` script manages all Hue processes. The supervisor is a watchdog process; its only purpose is to spawn and monitor other processes. To see active supervised processes, run `ps -f -u hue`.

A standard Hue installation starts and monitors the `runcpserver` process, which provides the core web functionality for Hue. If you installed other applications into your Hue instance, you may see other daemons running under the supervisor as well.

Supervisor automatically restarts these processes if they fail for any reason. If they fail repeatedly in a short period of time, the supervisor itself shuts down.

Viewing Hue Logs

Hue logs are stored in `/var/log/hue`. In the Hue UI, select **About Hue** > **Server Logs**. You can also view these logs at `http://myserver:port/logs`.

Hue generates `.log` and `.out` files for each supervised process. The `.log` files write log information with log4j. The `.out` files write standard output (stdout) and standard error (stderr) streams.

Available Hue Logs

The following Hue logs are available.

Log Name	Description
`access.log`	Filtered list of all successful attempts to access the Hue Web UI
`audit.log`	Audit log visible in Cloudera Navigator
`collectstatic.log`	Static files that support the Hue Web UI (images, JavaScript files, .css, and so on)
`error.log`	Filtered list of all nontrivial errors
`kt_renewer.log`	Kerberos ticket renews
`metrics_hue_server.log`	Usage data for monitoring in Cloudera Manager
`migrate.log`	Database and table migrations
`runcpserver.log`	Hue (CherryPy) web server info
`syncdb.log`	Database and table creations

Enable DEBUG

The Hue Server writes `INFO` level messages and keeps a small buffer of log messages at all levels in memory. `DEBUG` level messages are also available for troubleshooting.

To enable `DEBUG` messages for all the logs in the directory, `/var/log/hue`, choose one of these methods:

- In the Hue Web UI, go to the **Home** page, select **Server Logs**, and check the box by **Force Debug Level**. Debug is enabled on-the-fly.
- In Cloudera Manager, go to **Hue** > **Configuration**, search for and set **Enable Django Debug Mode**, click **Save Changes**, and **Restart** the Hue service.
- At the command line, open `/etc/hue/conf/hue.ini`, scroll to `[desktop]`, and set `django_debug_mode=true`. Restart the Hue service:

```
sudo service hue restart
```

KMS Installation and Upgrade

Hadoop Key Management Server (KMS) is a cryptographic key management server based on the Hadoop **KeyProvider** API. It provides a KeyProvider implementation client that interacts with the KMS using the HTTP REST API. Both the KMS and its client support HTTP SPNEGO Kerberos authentication and TLS/SSL-secured communication. The KMS is a Java-based web application that runs using a preconfigured Tomcat server bundled with the Hadoop distribution.

Cloudera provides the following implementations of the Hadoop KMS:

- **Java KeyStore KMS** - The default Hadoop KMS included in CDH that uses a file-based Java KeyStore (JKS) for its backing keystore. For parcel-based installations, no additional action is required to install or upgrade the KMS. For package-based installations, you must install additional packages. For more information, see Installing and Upgrading Java KeyStore KMS on page 363. Cloudera strongly recommends not using Java Keystore KMS in production environments.
- **Key Trustee KMS** - A custom KMS that uses Cloudera Navigator Key Trustee Server for its backing keystore instead of the file-based Java KeyStore (JKS) used by the default Hadoop KMS. Cloudera strongly recommends using Key Trustee KMS in production environments to improve the security, durability, and scalability of your cryptographic key management. For more information about the architecture and components involved in encrypting data at rest for production environments, see Cloudera Navigator Data Encryption Overview and Data at Rest Encryption Reference Architecture. For instructions on installing and upgrading Key Trustee KMS, see:

  - Installing Key Trustee KMS on page 209
  - Upgrading Key Trustee KMS

  Also, integrating Key Trustee Server with Cloudera Navigator Key HSM provides an additional layer of protection.
- **Navigator KMS Services backed by Thales HSM** - A custom KMS that uses a supported Thales Hardware Security Module (HSM) as its backing keystore. This KMS service provides the highest level of key isolation to customers who require it.
- **Navigator KMS Services backed by Luna HSM** - A custom KMS that uses a supported Luna Hardware Security Module (HSM) as its backing keystore. This KMS provides the highest level of key isolation to customers who require it.

Installing and Upgrading Java KeyStore KMS

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install or upgrade Java KeyStore KMS on a RHEL-compatible system:

```
$ sudo yum install hadoop-kms hadoop-kms-server
```

To install or upgrade Java KeyStore KMS on a SLES system:

```
$ sudo zypper install hadoop-kms hadoop-kms-server
```

To install or upgrade Java KeyStore KMS on an Ubuntu or Debian system:

```
$ sudo apt-get install hadoop-kms hadoop-kms-server
```

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

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
```

Kudu Installation

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

Follow these steps on each node in your Kudu cluster.

1. Cloudera recommends installing the Kudu repositories for your operating system. Use the links in the following table to download the appropriate repository installer. Save the repository installer to `/etc/yum.repos.d/` for RHEL, `/etc/apt/sources.list.d/` for Ubuntu/Debian, or `/etc/zypp/repos.d` for SLES. Table 17: Kudu Repository and Package Links on page 163
2. Install the kudu package, using the appropriate commands for your operating system. Also install the `kudu-master` and `kudu-tserver` packages. They provide operating system start-up scripts for the Kudu master and tablet servers.

Operating System	Install Commands

RHEL/CentOS	`sudo yum install kudu # Base Kudu files`
	`sudo yum install kudu-master # Kudu master init.d service script and default configuration`
	`sudo yum install kudu-tserver # Kudu tablet server init.d service script and default configuration`
	`sudo yum install kudu-client0 # Kudu C++ client shared library`
	`sudo yum install kudu-client-devel # Kudu C++ client SDK`

Ubuntu/Debian	`sudo apt-get install kudu # Base Kudu files`
	`sudo apt-get install kudu-master # Service scripts for managing kudu-master`
	`sudo apt-get install kudu-tserver # Service scripts for managing kudu-tserver`
	`sudo apt-get install libkuduclient0 # Kudu C++ client shared library`
	`sudo apt-get install libkuduclient-dev # Kudu C++ client SDK`

SLES	`sudo zypper install kudu # Base Kudu files`
	`sudo zypper install kudu-master # Kudu master init.d service script and default configuration`
	`sudo zypper install kudu-tserver # Kudu tablet server init.d service script and default configuration`
	`sudo zypper install kudu-client0 # Kudu C++ client shared library`
	`sudo zypper install kudu-client-devel # Kudu C++ client SDK`

3. The packages create a `kudu-conf` entry in the operating system's alternatives database, and they ship the built-in `conf.dist` alternative. To adjust your configuration, you can either edit the files in `/etc/kudu/conf/` directly, or create a new alternative using the operating system utilities. If you create a new alternative, make sure the alternative is the directory pointed to by the `/etc/kudu/conf/` symbolic link, and create custom configuration files there. Some parts of the configuration are configured in `/etc/default/kudu-master` and `/etc/default/kudu-tserver` files as well. You must include or duplicate these configuration options if you create custom configuration files.

   Review the configuration, including the default WAL and data directory locations, and adjust them according to your requirements.

4. Configure the Kudu services to start automatically when the server starts, by adding them to the default runlevel.

```
sudo chkconfig kudu-master on            # RHEL / CentOS
sudo chkconfig kudu-tserver on           # RHEL / CentOS

sudo update-rc.d kudu-master defaults    # Ubuntu / Debian
sudo update-rc.d kudu-tserver defaults   # Ubuntu / Debian
```

5. Verify the Installation on page 365.

Verify the Installation

1. Verify that the Kudu master and tablet servers are running using one of the following methods:
   - Examine the output of the `ps` command on servers to verify that the `kudu-master` and `kudu-tserver` processes are running.

- Access the master or tablet server web UI by going to `http://<_host_name_>:8051/` for masters, or `http://<_host_name_>:8050/` for tablet servers.

2. If Kudu isn't running, look at the log files in `/var/log/kudu`, and if there's a file ending with `.FATAL`, that means Kudu did not start.
   - If the error is related to a failed hole punch test or the file block manager, it might be a problem with your operating system.
   - If the error is related to clock synchronization, it is most likely a problem with the Network Time Protocol.

Upgrading Kudu

Before upgrading, you should read the release notes for the version of Kudu that you are about to install. Note that rolling upgrades are not supported. Shut down all Kudu services before you begin upgrading the software.

Important: If you have previously installed the beta `IMPALA_KUDU` service, make sure you remove it from your cluster before you proceed.

Upgrade Kudu using Cloudera Manager

Warning:

- You must upgrade Kudu *before* you upgrade CDH. Failure to do so will result in Impala queries returning the following error:

```
Query Status: Unable to open scanner: Remote error: unsupported feature
 flags
```

- If you are upgrading from Kudu 1.2.0 / CDH 5.10.x, you must upgrade both Kudu and CDH parcels (or packages) at the same time. If you upgrade Kudu but do not upgrade CDH, new Kudu features such as Security will not be available. Note that even though you might be able to see the updated configuration options for Kudu security in Cloudera Manager, configuring them will have no effect.

To use Cloudera Manager to upgrade Kudu using parcels or packages, use the following instructions. If you do not use Cloudera Manager, see Upgrade Kudu Using the Command Line on page 367.

Upgrade Kudu Using Parcels

1. Log in to Cloudera Manager.
2. Go to **Hosts**. Click **Parcels**.
3. Click **Check For New Parcels**.
4. Find the new version of KUDU in the list of parcels. Download, distribute, and activate it on your cluster.

Upgrade Kudu Using Packages

1. If you use a repository, re-download the repository list file to ensure that you have the latest information. See Table 17: Kudu Repository and Package Links on page 163.
2. Stop the Kudu service in Cloudera Manager. Go to the Kudu service and select **Actions** > **Stop**.
3. Depending on your operating system, issue the following set of commands on each Kudu host:

Operating System	Upgrade Commands
RHEL/CentOS	`sudo yum -y clean all` `sudo yum -y upgrade kudu`
Ubuntu/Debian	`sudo apt-get update` `sudo apt-get install kudu`
SLES	`sudo zypper clean --all` `sudo zypper update kudu`

4. Start the Kudu service in Cloudera Manager. Go to the Kudu service and select **Actions** > **Start**.

Upgrade Kudu Using the Command Line

Warning:

- You must upgrade Kudu *before* you upgrade CDH. Failure to do so will result in Impala queries returning the following error:

```
Query Status: Unable to open scanner: Remote error: unsupported feature
 flags
```

- If you are upgrading from Kudu 1.2.0 / CDH 5.10.x, you must upgrade both Kudu and CDH parcels (or packages) at the same time. If you upgrade Kudu but do not upgrade CDH, new Kudu features such as Security will not be available. Note that even though you might be able to see the updated configuration options for Kudu security in Cloudera Manager, configuring them will have no effect.

If you use Cloudera Manager, do not use the following command-line instructions. See Upgrade Kudu using Cloudera Manager on page 366.

1. If you use a repository, re-download the repository list file to ensure that you have the latest information. See Table 17: Kudu Repository and Package Links on page 163.
2. Stop the Kudu master and tablet servers using the following commands:

```
sudo service kudu-master stop
sudo service kudu-tserver stop
```

3. Depending on your operating system, issue the following set of commands on each Kudu host:

Operating System	Upgrade Commands
RHEL/CentOS	`sudo yum -y clean all` `sudo yum -y upgrade kudu`
Ubuntu/Debian	`sudo apt-get update` `sudo apt-get install kudu`
SLES	`sudo zypper clean --all` `sudo zypper update kudu`

4. Start the Kudu master and tablet servers using the following commands:

```
$ sudo service kudu-master start
$ sudo service kudu-tserver start
```

Mahout Installation

Important: This item is deprecated and will be removed in a future release. Cloudera supports items that are deprecated until they are removed. For more information about deprecated and removed items, see Deprecated Items.

Apache Mahout is a machine-learning tool. By enabling you to build machine-learning libraries that are scalable to "reasonably large" datasets, it aims to make building intelligent applications easier and faster.

Note:

To see which version of Mahout is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

The main use cases for Mahout are:

- **Recommendation mining**, which tries to identify things users will like on the basis of their past behavior (for example shopping or online-content recommendations)
- **Clustering**, which groups similar items (for example, documents on similar topics)
- **Classification**, which learns from existing categories what members of each category have in common, and on that basis tries to categorize new items
- **Frequent item-set mining**, which takes a set of item-groups (such as terms in a query session, or shopping-cart content) and identifies items that usually appear together

Important:

If you have not already done so, install the Cloudera `yum`, `zypper`/YaST or `apt` repository before using the instructions below to install Mahout. For instructions, see Installing the Latest CDH 5 Release on page 223.

Upgrading Mahout

Note:

To see which version of Mahout is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Upgrading Mahout from an Earlier CDH 5 Release to the Latest CDH 5 Release

To upgrade Mahout to the latest release, simply install the new version; see Installing Mahout on page 369.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

Installing Mahout

You can install Mahout from an RPM or Debian package, or from a tarball.

Note:

To see which version of Mahout is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Installing from packages is more convenient than installing the tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations

These instructions assume that you will install from packages if possible.

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper`/YaST or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install Mahout on a RHEL system:

```
$ sudo yum install mahout
```

To install Mahout on a SLES system:

```
$ sudo zypper install mahout
```

To install Mahout on an Ubuntu or Debian system:

```
$ sudo apt-get install mahout
```

To access Mahout documentation:

The Mahout docs are bundled in a `mahout-doc` package that should be installed separately.

```
$ sudo apt-get install mahout-doc
```

The contents of this package are saved under `/usr/share/doc/mahout*`.

The Mahout Executable

The Mahout executable is installed in `/usr/bin/mahout`. Use this executable to run your analysis.

Getting Started with Mahout

To get started with Mahout, you can follow the instructions in this Apache Mahout Quickstart.

Viewing the Mahout Documentation

For more information about Mahout, see mahout.apache.org.

Oozie Installation

About Oozie

Apache Oozie Workflow Scheduler for Hadoop is a workflow and coordination service for managing Apache Hadoop jobs:

- Oozie Workflow jobs are Directed Acyclical Graphs (DAGs) of *actions*; *actions* are typically Hadoop jobs (MapReduce, Streaming, Pipes, Pig, Hive, Sqoop, etc).
- Oozie Coordinator jobs trigger recurrent Workflow jobs based on time (frequency) and data availability.
- Oozie Bundle jobs are sets of Coordinator jobs managed as a single job.

Oozie is an extensible, scalable and data-aware service that you can use to orchestrate dependencies among jobs running on Hadoop.

- To find out more about Oozie, see https://archive.cloudera.com/cdh5/cdh/5/oozie/.
- To install or upgrade Oozie, follow the directions on this page.

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).

Oozie Packaging

There are two packaging options for installing Oozie:

- Separate RPM packages for the Oozie server (`oozie`) and client (`oozie-client`)
- Separate Debian packages for the Oozie server (`oozie`) and client (`oozie-client`)

You can also download an Oozie tarball.

Oozie Prerequisites

- Prerequisites for installing Oozie server:
  - An operating system supported by CDH 5.
  - Oracle JDK
  - A supported database.

- Prerequisites for installing Oozie client:
  - Oracle JDK

Note:

- To see which version of Oozie is shipping in CDH 5,check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Upgrading Oozie

Follow these instructions to upgrade Oozie to CDH 5 from RPM or Debian Packages.

Upgrading Oozie from an Earlier CDH 5 Release

The steps that follow assume you are upgrading Oozie as part of an overall upgrade to the latest CDH 5 release and have already performed the steps under Upgrading from an Earlier CDH 5 Release to the Latest Release.

To upgrade Oozie to the latest CDH 5 release, proceed as follows.

Step 1: Back Up the Configuration

Back up the Oozie configuration files in `/etc/oozie` and the Oozie database.

For convenience you may want to save Oozie configuration files in your home directory; you will need them after installing the new version of Oozie.

Step 2: Stop the Oozie Server.

To stop the Oozie Server:

```
sudo service oozie stop
```

Step 3: Install Oozie

Follow the procedure under Installing Oozie on page 371 and then proceed to Configuring Oozie after Upgrading from an Earlier CDH 5 Release on page 373.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by dpkg.

Installing Oozie

Oozie is distributed as two separate packages—a server package (`oozie`) and a client package (`oozie-client`). Choose the appropriate packages and install them with your preferred package manager application.

Note: The Oozie server package, `oozie`, is preconfigured to work with MRv2 (YARN). To configure the Oozie server to work with MRv1, see Configuring which Hadoop Version to Use on page 372.

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install the Oozie server package on an Ubuntu and other Debian system:

```
$ sudo apt-get install oozie
```

To install the Oozie client package on an Ubuntu and other Debian system:

```
$ sudo apt-get install oozie-client
```

To install the Oozie server package on a RHEL-compatible system:

```
$ sudo yum install oozie
```

To install the Oozie client package on a RHEL-compatible system:

```
$ sudo yum install oozie-client
```

To install the Oozie server package on a SLES system:

```
$ sudo zypper install oozie
```

To install the Oozie client package on a SLES system:

```
$ sudo zypper install oozie-client
```

Note:

Installing the `oozie` package creates an `oozie` service configured to start Oozie at system startup time.

You are now ready to configure Oozie. See Configuring Oozie on page 372.

Configuring Oozie

This page explains how to configure Oozie, for new installs and upgrades, in an unmanaged deployment, *without* Cloudera Manager.

Important:

- Follow these command-line instructions on systems that do not use Cloudera Manager.
- This information applies specifically to CDH 5.12.x. See Cloudera Documentation for information specific to other releases.

Configuring which Hadoop Version to Use

The Oozie *server* works with either MRv1 or YARN, but not both simultaneously. The Oozie *client* does not interact directly with Hadoop MapReduce and does not require any MapReduce configuration.

To configure the Oozie server to work with YARN or MRv1, and with or without TLS/SSL, use the `alternatives` command (or `update-alternatives`, depending on your operating system).

Important: Stop the Oozie server before upgrading from MRv1 to YARN or workflows that depend on MRv1 may cause the MRv1 jobs to fail.

- To use YARN (without TLS/SSL):

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.http
```

- To use YARN (with TLS/SSL):

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.https
```

- To use MRv1 (without TLS/SSL) :

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.http.mr1
```

- To use MRv1 (with TLS/SSL) :

```
alternatives --set oozie-tomcat-deployment /etc/oozie/tomcat-conf.https.mr1
```

Important:

In CDH 5 Beta 2 and higher, ensure that CATALINA_BASE in `/etc/oozie/conf/oozie-env.sh` is set to:

```
export
          CATALINA_BASE=/var/lib/oozie/tomcat-deployment
```

Configuring Oozie after Upgrading from an Earlier CDH 5 Release

Note: If you are installing Oozie for the first time, skip this section and proceed with Configuring Oozie after a New Installation on page 376.

Step 1: Update Configuration Files

1. Edit the new Oozie CDH 5 `oozie-site.xml`, and set all customizable properties to the values you set in the previous `oozie-site.xml`.
2. If necessary do the same for the `oozie-log4j.properties`, `oozie-env.sh` and the `adminusers.txt` files.

Step 2: Upgrade the Database

Important:

- Do not proceed before you have edited the configuration files as instructed in Step 1.
- Before running the database upgrade tool, copy or symbolically link the JDBC driver JAR for the database you are using into the `/var/lib/oozie/` directory.

Oozie CDH 5 provides a command-line tool to perform the database schema and data upgrade. The tool uses Oozie configuration files to connect to the database and perform the upgrade.

The database upgrade tool works in two modes: it can do the upgrade in the database or it can produce an SQL script that a database administrator can run manually. If you use the tool to perform the upgrade, you must do it as a database user who has permissions to run DDL operations in the Oozie database.

- **To run the Oozie database upgrade tool against the database:**

  Important:

  This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

  ```
  $ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh upgrade -run
  ```

  You will see output such as this (the output of the script may differ slightly depending on the database vendor):

```
Validate DB Connection
DONE
Check DB schema exists
DONE
Verify there are not active Workflow Jobs
DONE
Check OOZIE_SYS table does not exist
DONE
Get Oozie DB version
DONE
Upgrade SQL schema
DONE
Upgrading to db schema for Oozie 4.0.0-cdh5.0.0
Update db.version in OOZIE_SYS table to 3
DONE
Converting text columns to bytea for all tables
DONE
Get Oozie DB version
DONE

Oozie DB has been upgraded to Oozie version '4.0.0-cdh5.0.0'

The SQL commands have been written to: /tmp/ooziedb-8676029205446760413.sql
```

- **To create the upgrade script:**

  Important: This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh upgrade -sqlfile SCRIPT
```

  For example:

```
$ sudo -u oozie  /usr/lib/oozie/bin/ooziedb.sh upgrade -sqlfile oozie-upgrade.sql
```

  You should see output such as the following (the output of the script may differ slightly depending on the database vendor):

```
Validate DB Connection
DONE
Check DB schema exists
DONE
Verify there are not active Workflow Jobs
DONE
Check OOZIE_SYS table does not exist
DONE
Get Oozie DB version
DONE
```

```
Upgrade SQL schema
DONE
Upgrading to db schema for Oozie 4.0.0-cdh5.0.0
Update db.version in OOZIE_SYS table to 3
DONE
Converting text columns to bytea for all tables
DONE
Get Oozie DB version
DONE

The SQL commands have been written to: oozie-upgrade.sql

WARN: The SQL commands have NOT been executed, you must use the '-run' option
```

Important: If you used the `-sqlfile` option instead of `-run`, Oozie database schema has not been upgraded. You need to run the `oozie-upgrade` script against your database.

Step 3: Upgrade the Oozie Shared Library

Important: This step is required; the current version of Oozie does not work with shared libraries from an earlier version.

The Oozie installation bundles two shared libraries, one for MRv1 and one for YARN. Make sure you install the right one for the MapReduce version you are using:

- The shared library file for YARN is `oozie-sharelib-yarn`.
- The shared library file for MRv1 is `oozie-sharelib-mr1`.

To upgrade the shared library, proceed as follows.

1. Delete the Oozie shared libraries from HDFS. For example:

```
$ sudo -u oozie hadoop fs -rmr /user/oozie/share
```

Note:

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

2. install the Oozie CDH 5 shared libraries. For example:

```
$ sudo oozie-setup sharelib create -fs <FS_URI> -locallib
/usr/lib/oozie/oozie-sharelib-yarn
```

where *FS_URI* is the HDFS URI of the filesystem that the shared library should be installed on (for example, `hdfs://<HOST>:<PORT>`).

Important: If you are installing Oozie to work with MRv1, make sure you use `oozie-sharelib-mr1` instead.

Step 4: Start the Oozie Server

Now you can start Oozie:

```
$ sudo service oozie start
```

Check Oozie's `oozie.log` to verify that Oozie has started successfully.

Step 5: Upgrade the Oozie Client

Although older Oozie clients work with the new Oozie server, you need to install the new version of the Oozie client to use all the functionality of the Oozie server.

To upgrade the Oozie client, if you have not already done so, follow the steps under Installing Oozie on page 371.

Configuring Oozie after a New Installation

Note: If you are upgrading Oozie from an earlier CDH 5 release, see Configuring Oozie after Upgrading from an Earlier CDH 5 Release on page 373.

When you install Oozie from an RPM or Debian package, Oozie server creates all configuration, documentation, and runtime files in the standard Linux directories, as follows.

Type of File	Where Installed
binaries	`/usr/lib/oozie/`
configuration	`/etc/oozie/conf/`
documentation	for SLES: `/usr/share/doc/packages/oozie/` for other platforms: `/usr/share/doc/oozie/`
examples TAR.GZ	for SLES: `/usr/share/doc/packages/oozie/` for other platforms: `/usr/share/doc/oozie/`
sharelib TAR.GZ	`/usr/lib/oozie/`
data	`/var/lib/oozie/`
logs	`/var/log/oozie/`
temp	`/var/tmp/oozie/`
PID file	`/var/run/oozie/`

Deciding Which Database to Use

Oozie has a built-in Derby database, but Cloudera recommends that you use a PostgreSQL, MariaDB, MySQL, or Oracle database instead, for the following reasons:

- Derby runs in embedded mode and it is not possible to monitor its health.
- Though it might be possible, Cloudera currently has no live backup strategy for the embedded Derby database.
- Under load, Cloudera has observed locks and rollbacks with the embedded Derby database that do not happen with server-based databases.

See CDH and Cloudera Manager Supported Databases for tested database versions.

Configuring Oozie to Use PostgreSQL

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

Install PostgreSQL

Create the Oozie User and Oozie Database

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

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

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

postgres=# CREATE DATABASE "oozie" WITH OWNER = oozie
 ENCODING = 'UTF8'
 TABLESPACE = pg_default
 LC_COLLATE = 'en_US.UTF-8'
 LC_CTYPE = 'en_US.UTF-8'
 CONNECTION LIMIT = -1;
CREATE DATABASE

postgres=# \q
```

Configure PostgreSQL to Accept Network Connections for the Oozie User

1. Edit the `postgresql.conf` file and set the `listen_addresses` property to *, to make sure that the PostgreSQL server starts listening on all your network interfaces. Also make sure that the `standard_conforming_strings` property is set to `off`.
2. Edit the PostgreSQL `data/pg_hba.conf` file as follows:

```
host    oozie         oozie         0.0.0.0/0             md5
```

Reload the PostgreSQL Configuration

```
$ sudo -u postgres pg_ctl reload -s -D /opt/PostgreSQL/8.4/data
```

Configure Oozie to Use PostgreSQL

Edit the `oozie-site.xml` file as follows:

```
...
    <property>
        <name>oozie.service.JPAService.jdbc.driver</name>
        <value>org.postgresql.Driver</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.url</name>
        <value>jdbc:postgresql://localhost:5432/oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.username</name>
        <value>oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.password</name>
        <value>oozie</value>
    </property>
    ...
```

Note: In the JDBC URL property, replace `localhost` with the hostname where PostgreSQL is running. In the case of PostgreSQL, unlike MySQL or Oracle, you do not need to download and install the JDBC driver separately, as it is license-compatible with Oozie and bundled with it.

Configuring Oozie to Use MariaDB

Use the procedure that follows to configure Oozie to use MariaDB instead of Apache Derby.

Install and Start MariaDB

For more information, see Installing the MariaDB Server on page 82.

Create the Oozie Database and Oozie MariaDB User

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

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

MariaDB [(none)]> create database oozie default character set utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]>  grant all privileges on oozie.* to 'oozie'@'localhost' identified by
 'oozie';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>  grant all privileges on oozie.* to 'oozie'@'%' identified by 'oozie';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
```

Configure Oozie to Use MariaDB

Edit properties in the `oozie-site.xml` file as follows:

```
...
    <property>
        <name>oozie.service.JPAService.jdbc.driver</name>
        <value>org.mysql.jdbc.Driver</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.url</name>
        <value>jdbc:mysql://localhost:3306/oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.username</name>
        <value>oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.password</name>
        <value>oozie</value>
    </property>
    ...
```

Note: In the JDBC URL property, replace `localhost` with the hostname where MariaDB is running.

Add the MariaDB JDBC Driver JAR to Oozie

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

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

Configuring Oozie to Use MySQL

Use the procedure that follows to configure Oozie to use MySQL instead of Apache Derby.

Install and Start MySQL 5.x

Create the Oozie Database and Oozie MySQL User

For example, using the MySQL `mysql` command-line tool:

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

mysql> create database oozie default character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql>  grant all privileges on oozie.* to 'oozie'@'localhost' identified by 'oozie';
Query OK, 0 rows affected (0.00 sec)

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

mysql> exit
Bye
```

Configure Oozie to Use MySQL

Edit properties in the `oozie-site.xml` file as follows:

```
...
    <property>
        <name>oozie.service.JPAService.jdbc.driver</name>
        <value>com.mysql.jdbc.Driver</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.url</name>
        <value>jdbc:mysql://localhost:3306/oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.username</name>
        <value>oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.password</name>
        <value>oozie</value>
    </property>
    ...
```

Note: In the JDBC URL property, replace `localhost` with the hostname where MySQL is running.

Add the MySQL JDBC Driver JAR to Oozie

Copy or symbolically link the MySQL JDBC driver JAR into one of the following directories:

- For installations that use *packages*: `/var/lib/oozie/`
- For installations that use *parcels*: `/opt/cloudera/parcels/CDH/lib/oozie/lib/`

directory.

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

Configuring Oozie to use Oracle

Use the procedure that follows to configure Oozie to use Oracle 11g instead of Apache Derby.

Install and Start Oracle 11g

Use Oracle's instructions.

Create the Oozie Oracle User and Grant Privileges

The following example uses the Oracle `sqlplus` command-line tool, and shows the privileges Cloudera recommends. Oozie needs CREATE SESSION to start and manage workflows. The additional roles are needed for creating and upgrading the Oozie database.

```
$ sqlplus system@localhost

Enter password: ******

SQL> create user oozie identified by oozie default tablespace users temporary tablespace
 temp;

User created.

SQL> grant alter index to oozie;
grant alter table to oozie;
grant create index to oozie;
grant create sequence to oozie;
grant create session to oozie;
grant create table to oozie;
grant drop sequence to oozie;
grant select dictionary to oozie;
grant drop table to oozie;
alter user oozie quota unlimited on users;
alter user oozie quota unlimited on system;

SQL> exit

$
```

Important:

Do *not* make the following grant:

```
grant select any table;
```

Configure Oozie to Use Oracle

Edit the `oozie-site.xml` file as follows.

```
...
    <property>
        <name>oozie.service.JPAService.jdbc.driver</name>
        <value>oracle.jdbc.OracleDriver</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.url</name>
        <value>jdbc:oracle:thin:@//myhost:1521/oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.username</name>
```

```
        <value>oozie</value>
    </property>
    <property>
        <name>oozie.service.JPAService.jdbc.password</name>
        <value>oozie</value>
    </property>
    ...
```

Note: In the JDBC URL property, replace `myhost` with the hostname where Oracle is running and replace `oozie` with the TNS name of the Oracle database.

Add the Oracle JDBC Driver JAR to Oozie

Copy or symbolically link the Oracle JDBC driver JAR into the `/var/lib/oozie/` directory.

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

Creating the Oozie Database Schema

After configuring Oozie database information and creating the corresponding database, create the Oozie database schema. Oozie provides a database tool for this purpose.

Note: The Oozie database tool uses Oozie configuration files to connect to the database to perform the schema creation; before you use the tool, make you have created a database and configured Oozie to work with it as described above.

The Oozie database tool works in 2 modes: it can create the database, or it can produce an SQL script that a database administrator can run to create the database manually. If you use the tool to create the database schema, you must have the permissions needed to execute DDL operations.

To run the Oozie database tool against the database

Important: This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh create -run
```

You should see output such as the following (the output of the script may differ slightly depending on the database vendor) :

```
Validate DB Connection.
DONE
Check DB schema does not exist
DONE
Check OOZIE_SYS table does not exist
DONE
Create SQL schema
DONE
DONE
Create OOZIE_SYS table
DONE

Oozie DB has been created for Oozie version '4.0.0-cdh5.0.0'

The SQL commands have been written to: /tmp/ooziedb-5737263881793872034.sql
```

To create the upgrade script

Important: This step must be done as the `oozie` Unix user, otherwise Oozie may fail to start or work properly because of incorrect file permissions.

Run `/usr/lib/oozie/bin/ooziedb.sh create -sqlfile` *`SCRIPT`*. For example:

```
$ sudo -u oozie /usr/lib/oozie/bin/ooziedb.sh create -sqlfile oozie-create.sql
```

You should see output such as the following (the output of the script may differ slightly depending on the database vendor) :

```
Validate DB Connection.
DONE
Check DB schema does not exist
DONE
Check OOZIE_SYS table does not exist
DONE
Create SQL schema
DONE
DONE
Create OOZIE_SYS table
DONE

Oozie DB has been created for Oozie version '4.0.0-cdh5.0.0'

The SQL commands have been written to: oozie-create.sql

WARN: The SQL commands have NOT been executed, you must use the '-run' option
```

Important: If you used the `-sqlfile` option instead of `-run`, Oozie database schema has not been created. You must run the `oozie-create.sql` script against your database.

Enabling the Oozie Web Console

To enable the Oozie web console, download and add the ExtJS library to the Oozie server.

Step 1: Download the Library

Download the ExtJS version 2.2 library from https://archive.cloudera.com/gplextras/misc/ext-2.2.zip and place it a convenient location.

Step 2: Install the Library

Extract the `ext-2.2.zip` file into `/var/lib/oozie`.

Step 3: Configure SPNEGO authentication (in Kerberos clusters only)

The web console shares a port with the Oozie REST API, and the API allows modifications of Oozie jobs (kill, submission, and inspection). SPNEGO authentication ensures that the Kerberos realm trusts the client browser credentials and that configuration of the client web browser passes these credentials. If this configuration is not possible, use the Hue Oozie Dashboard instead of the Oozie Web Console.

See How to Configure Browsers for Kerberos Authentication and Configuring a Cluster-dedicated MIT KDC with Cross-Realm Trust.

Configuring Oozie with Kerberos Security

To configure Oozie with Kerberos security, see Oozie Authentication.

Installing the Oozie Shared Library in Hadoop HDFS

The Oozie installation bundles two Oozie shared libraries, one for MRv1 and one for YARN, which contain all of the JARs required to enable workflow jobs to run streaming, DistCp, Pig, Hive, and Sqoop actions. Make sure you install the right one for the MapReduce version you are using:

- The shared library file for MRv1 is `oozie-sharelib-mr1`.
- The shared library file for YARN is `oozie-sharelib-yarn`.

Important: If Hadoop is configured with Kerberos security enabled, you must first configure Oozie with Kerberos Authentication. For instructions, see Oozie Security Configuration. Before running the commands in the following instructions, you must run the `sudo -u oozie kinit -k -t /etc/oozie/oozie.keytab` and `kinit -k hdfs` commands. Then, instead of using commands in the form `sudo -u user command`, use just *command*; for example, `$ hadoop fs -mkdir /user/oozie`

To install the Oozie shared library in Hadoop HDFS in the oozie user home directory

```
$ sudo -u hdfs hadoop fs -mkdir /user/oozie
$ sudo -u hdfs hadoop fs -chown oozie:oozie /user/oozie
$ sudo oozie-setup sharelib create -fs <FS_URI> -locallib
/usr/lib/oozie/oozie-sharelib-yarn
```

where *FS_URI* is the HDFS URI of the filesystem that the shared library should be installed on (for example, `hdfs://<HOST>:<PORT>`).

Important: If you are installing Oozie to work with MRv1 use `oozie-sharelib-mr1` instead.

Configuring Support for Oozie Uber JARs

An ***uber JAR*** is a JAR that contains other JARs with dependencies in a `lib/` folder inside the JAR

Important: When you build an application JAR, *do not* include CDH JARs, because they are already provided. If you do, upgrading CDH can break your application. To avoid this situation, set the Maven dependency `scope` to `provided`. For more information, see Using the CDH 5 Maven Repository.

You can configure the cluster to handle uber JARs properly for the MapReduce action (as long as it does not include any streaming or pipes) by setting the following property in the `oozie-site.xml` file:

```
...
    <property>
        <name>oozie.action.mapreduce.uber.jar.enable</name>
        <value>true</value>
...
```

When this property is set, users can use the `oozie.mapreduce.uber.jar` configuration property in their MapReduce workflows to notify Oozie that the specified JAR file is an uber JAR.

Configuring Oozie to Run against a Federated Cluster

To run Oozie against a federated HDFS cluster using ViewFS, configure the `oozie.service.HadoopAccessorService.supported.filesystems` property in `oozie-site.xml` as follows:

```
<property>
     <name>oozie.service.HadoopAccessorService.supported.filesystems</name>
     <value>hdfs,viewfs</value>
</property>
```

Starting, Stopping, and Accessing the Oozie Server

Starting the Oozie Server

After you have completed *all* of the required configuration steps, you can start Oozie:

```
$ sudo service oozie start
```

If you see the message `Oozie System ID [oozie-oozie] started` in the `oozie.log` log file, the system has started successfully.

> **Note:**
>
> By default, Oozie server runs on port `11000` and its URL is `http://<OOZIE_HOSTNAME>:11000/oozie`.

Stopping the Oozie Server

```
$ sudo service oozie stop
```

Accessing the Oozie Server with the Oozie Client

The Oozie client is a command-line utility that interacts with the Oozie server using the Oozie web-services API.

Use the `/usr/bin/oozie` script to run the Oozie client.

For example, if you want to invoke the client on the same machine where the Oozie server is running:

```
$ oozie admin -oozie http://localhost:11000/oozie -status
System mode: NORMAL
```

To make it convenient to use this utility, set the environment variable `OOZIE_URL` to point to the URL of the Oozie server. Then you can skip the `-oozie` option.

For example, if you want to invoke the client on the same machine where the Oozie server is running, set the `OOZIE_URL` to `http://localhost:11000/oozie`.

```
$ export OOZIE_URL=http://localhost:11000/oozie
$ oozie admin -version
Oozie server build version: 4.0.0-cdh5.0.0
```

> **Important:**
>
> If Oozie is configured with Kerberos Security enabled:
>
> - You must have a Kerberos session running. For example, you can start a session by running the `kinit` command.
> - **Do not** use `localhost` as in the above examples.
>
> As with every service that uses Kerberos, Oozie has a Kerberos *principal* in the form `<SERVICE>/<HOSTNAME>@<REALM>`. In a Kerberos configuration, you **must** use the `<HOSTNAME>` value in the Kerberos principal to specify the Oozie server; for example, if the `<HOSTNAME>` in the principal is `myoozieserver.mydomain.com`, set `OOZIE_URL` as follows:
>
> ```
> $ export OOZIE_URL=http://myoozieserver.mydomain.com:11000/oozie
> ```
>
> If you use an alternate hostname or the IP address of the service, Oozie will not work properly.

Accessing the Oozie Server with a Browser

If you have enabled the Oozie web console by adding the ExtJS library, you can connect to the console at `http://<OOZIE_HOSTNAME>:11000/oozie`.

> **Note:**
>
> If the Oozie server is configured to use Kerberos HTTP SPNEGO Authentication, you must use a web browser that supports Kerberos HTTP SPNEGO (for example, Firefox or Internet Explorer).

Using Sqoop Actions with Oozie

Sqoop 1 does not ship with third party JDBC drivers. You must download them separately and save them to the `/var/lib/sqoop/` directory on the Oozie server. For more information, see Sqoop 1 Installation on page 398.

Recommendations

- Cloudera recommends that you not use Sqoop CLI commands with an Oozie Shell Action. Such deployments are not reliable and prone to breaking during upgrades and configuration changes.
- To import data into Hive, use a combination of a Sqoop Action with a Hive2 Action.
  - A Sqoop Action to simply ingest data into HDFS.
  - A Hive2 Action that loads the data from HDFS into Hive.

Deploying and Configuring Oozie Sqoop1 Action JDBC Drivers

Before you begin this process, confirm that your Sqoop1 JDBC drivers are present in `/var/lib/sqoop`.

SSH to the Oozie server host and execute the following commands to deploy and configure the drivers on HDFS:

```
cd /var/lib/sqoop
sudo -u hdfs hdfs dfs -mkdir /user/oozie/libext
sudo -u hdfs hdfs dfs -chown oozie:oozie /user/oozie/libext
sudo -u hdfs hdfs dfs -put
/opt/cloudera/parcels/SQOOP_NETEZZA_CONNECTOR/sqoop-nz-connector*.jar /user/oozie/libext/
sudo -u hdfs hdfs dfs -put /opt/cloudera/parcels/SQOOP_TERADATA_CONNECTOR/lib/*.jar
/user/oozie/libext/
sudo -u hdfs hdfs dfs -put
/opt/cloudera/parcels/SQOOP_TERADATA_CONNECTOR/sqoop-connector-teradata*.jar
/user/oozie/libext/
sudo -u hdfs hdfs dfs -put /var/lib/sqoop/*.jar /user/oozie/libext/
sudo -u hdfs hdfs dfs -chown oozie:oozie /user/oozie/libext/*.jar
sudo -u hdfs hdfs dfs -chmod 755 /user/oozie/libext/*.jar
sudo -u hdfs hdfs dfs -ls /user/oozie/libext

# [sample contents of /user/oozie/libext]
-rwxr-xr-x   3 oozie oozie     959987 2016-05-29 09:58
/user/oozie/libext/mysql-connector-java.jar
-rwxr-xr-x   3 oozie oozie     358437 2016-05-29 09:58 /user/oozie/libext/nzjdbc3.jar
-rwxr-xr-x   3 oozie oozie    2739670 2016-05-29 09:58 /user/oozie/libext/ojdbc6.jar
-rwxr-xr-x   3 oozie oozie    3973162 2016-05-29 09:58
/user/oozie/libext/sqoop-connector-teradata-1.5c5.jar
-rwxr-xr-x   3 oozie oozie      41691 2016-05-29 09:58
/user/oozie/libext/sqoop-nz-connector-1.3c5.jar
-rwxr-xr-x   3 oozie oozie       2405 2016-05-29 09:58 /user/oozie/libext/tdgssconfig.jar
-rwxr-xr-x   3 oozie oozie     873860 2016-05-29 09:58 /user/oozie/libext/terajdbc4.jar
```

Configuring Oozie Sqoop1 Action Workflow JDBC Drivers

Use the following steps to configure Oozie Sqoop1 Action Workflows:

1. Confirm that the Sqoop1 JDBC drivers are present in HDFS. To do this, SSH to the Oozie Server host and run the following command:

```
sudo -u hdfs hdfs dfs -ls /user/oozie/libext
```

2. Configure the following Oozie Sqoop1 Action workflow variables in Oozie's `job.properties` file as follows:

```
oozie.use.system.libpath = true
oozie.libpath = /user/oozie/libext
```

Viewing the Oozie Documentation

For additional Oozie documentation, see https://archive.cloudera.com/cdh5/cdh/5/oozie/.

Pig Installation

Apache Pig enables you to analyze large amounts of data using Pig's query language called Pig Latin. Pig Latin queries run in a distributed way on a Hadoop cluster.

Use the following sections to install or upgrade Pig:

- Upgrading Pig
- Installing Pig
- Using Pig with HBase
- Installing DataFu
- Apache Pig Documentation

Upgrading Pig

Note:

To see which version of Pig is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the Release Notes.

Upgrading Pig from an Earlier CDH 5 release

The instructions that follow assume that you are upgrading Pig as part of a CDH 5 upgrade, and have already performed the steps under Upgrading from an Earlier CDH 5 Release to the Latest Release.

To upgrade Pig from an earlier CDH 5 release:

1. Exit the Grunt shell and make sure no Pig scripts are running.
2. Install the new version, following the instructions in the next section, Installing Pig on page 387.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

Installing Pig

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install Pig On RHEL-compatible systems:

```
$ sudo yum install pig
```

To install Pig on SLES systems:

```
$ sudo zypper install pig
```

To install Pig on Ubuntu and other Debian systems:

```
$ sudo apt-get install pig
```

Note:

Pig automatically uses the active Hadoop configuration (whether standalone, pseudo-distributed mode, or distributed). After installing the Pig package, you can start Pig.

To start Pig in interactive mode (YARN)

Important:

- For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop in a YARN installation, make sure that the `HADOOP_MAPRED_HOME` environment variable is set correctly, as follows:

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

- For each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop in an MRv1 installation, set the `HADOOP_MAPRED_HOME` environment variable as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

To start Pig, use the following command.

```
$ pig
```

To start Pig in interactive mode (MRv1)

Use the following command:

```
$ pig
```

You should see output similar to the following:

```
2012-02-08 23:39:41,819 [main] INFO  org.apache.pig.Main - Logging error messages to:
/home/arvind/pig-0.11.0-cdh5b1/bin/pig_1328773181817.log
2012-02-08 23:39:41,994 [main] INFO
org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop
file system at: hdfs://localhost/
...
grunt>
```

Examples

To verify that the input and output directories from the YARN or MRv1 example `grep` job exist, list an HDFS directory from the Grunt Shell:

```
grunt> ls
hdfs://localhost/user/joe/input <dir>
hdfs://localhost/user/joe/output <dir>
```

To run a `grep` example job using Pig for `grep` inputs:

```
grunt> A = LOAD 'input';
grunt> B = FILTER A BY $0 MATCHES '.*dfs[a-z.]+.*';
grunt> DUMP B;
```

Note:

To check the status of your job while it is running, look at the ResourceManager web console (YARN) or JobTracker web console (MRv1).

Using Pig with HBase

To allow Pig scripts to use HBase, add the following statement to the top of each script. Replace the `<component_version>` strings with the current HBase, ZooKeeper and CDH version numbers.

```
register /usr/lib/zookeeper/zookeeper-<ZooKeeper_version>-cdh<CDH_version>.jar
register /usr/lib/hbase/hbase-<HBase_version>-cdh<CDH_version>-security.jar
```

For example,

```
register /usr/lib/zookeeper/zookeeper-3.4.5-cdh5.0.0.jar
register /usr/lib/hbase/hbase-0.95.2-cdh5.0.0-security.jar
```

In addition, Pig needs to be able to access the `hbase-site.xml` file on the Hadoop client. Pig searches for the file within the `/etc/hbase/conf` directory on the client, or in Pig's `CLASSPATH` variable.

For more information about using Pig with HBase, see Importing Data Into HBase.

Installing DataFu

Warning: DataFu has been in decline for a significant period of time and is now officially deprecated. Cloudera recommends that you replace the DataFu UDFs with Hive UDFs. Hive UDFs provide most of DataFu's functions and many additional functions. Moreover, Hive UDFs are more stable and well-supported. In an upcoming release, Apache Pig will support Hive UDFs. For more information about using Hive UDFs in CDH, see Managing UDFs.

DataFu is a collection of Apache Pig UDFs (User-Defined Functions) for statistical evaluation. They were developed by LinkedIn and are now open source under an Apache 2.0 license.

A number of usage examples and other information are available at https://github.com/linkedin/datafu.

To Use DataFu in a Parcel-deployed Cluster

If your cluster uses parcels, DataFu is installed for you. You need to register the JAR file prior to use with the following command.

```
REGISTER /opt/cloudera/parcels/CDH/lib/pig/datafu.jar
```

To Use DataFu in a Package-deployed Cluster:

1. Install the DataFu package:

Operating system	Install command
Red-Hat-compatible	`sudo yum install pig-udf-datafu`
SLES	`sudo zypper install pig-udf-datafu`
Debian or Ubuntu	`sudo apt-get install pig-udf-datafu`

   This puts the DataFu JAR file (for example, `datafu-0.0.4-cdh5.0.0.jar`) in `/usr/lib/pig`.

2. Register the JAR. Replace the `<component_version>` string with the current DataFu and CDH version numbers.

```
REGISTER /usr/lib/pig/datafu-<DataFu_version>-cdh<CDH_version>.jar
```

   For example:

```
REGISTER /usr/lib/pig/datafu-0.0.4-cdh5.0.0.jar
```

Viewing the Pig Documentation

For additional Pig documentation, see https://archive.cloudera.com/cdh5/cdh/5/pig.

Search Installation

This documentation describes how to install Cloudera Search powered by Solr. It also explains how to install and start supporting tools and services such as the ZooKeeper Server, MapReduce tools for use with Cloudera Search, and Flume Solr Sink.

After installing Cloudera Search as described in this document, you can configure and use Cloudera Search as described in the Cloudera Search Guide. The user guide includes the Cloudera Search Tutorial, as well as topics that describe extracting, transforming, and loading data, establishing high availability, and troubleshooting.

Cloudera Search documentation includes:

- CDH 5 Release Notes
- CDH Version and Packaging Information
- Cloudera Search Guide
- Cloudera Search Frequently Asked Questions

Installing Cloudera Search without Cloudera Manager

Cloudera Search for CDH 5 is included with CDH 5.

To install Cloudera Search for CDH 5 using packages, see Installing the Latest CDH 5 Release on page 223.

> **Note:** This page describes how to install CDH using packages as well as how to install CDH using Cloudera Manager.

You can also install Cloudera Search manually in some situations; for example, if you have an existing installation to which you want to add Search.

- For general information about using repositories to install or upgrade Cloudera software, see Understanding Custom Installation Solutions on page 166.
- For instructions on installing or upgrading CDH, see Installing the Latest CDH 5 Release on page 223.
- For CDH 5 repository locations and client `.repo` files, which include Cloudera Search, see Version and Download Information.

Cloudera Search includes the following packages:

Package Name	Description
solr	Solr
solr-server	Platform specific service script for starting, stopping, or restart Solr.
solr-doc	Cloudera Search documentation.
solr-mapreduce	Tools to index documents using MapReduce.
solr-crunch	Tools to index documents using Crunch.
search	Examples, Contrib, and Utility code and data.

Before You Begin

The installation instructions assume that the `sudo` command is configured on the hosts where you are installing Cloudera Search. If `sudo` is not configured, use the root user (`superuser`) to configure Cloudera Search.

> **Important:**
>
> - **Running services:** When starting, stopping, and restarting CDH components, always use the `service (8)` command instead of running `/etc/init.d` scripts directly. This is important because `service` sets the current working directory to the root directory (/) and removes environment variables except `LANG` and `TERM`. This creates a predictable environment in which to administer the service. If you use `/etc/init.d` scripts directly, any environment variables continue to be applied, potentially causing unexpected results. If you install CDH from packages, `service` is installed as part of the Linux Standard Base (LSB).
> - **Install the Cloudera repository:** Before using the instructions in this guide to install or upgrade Cloudera Search from packages, install the Cloudera `yum`, `zypper/YaST` or `apt` repository, and install or upgrade CDH and make sure it is functioning correctly.

Installing Solr Packages

This topic describes how to complete a new installation of Solr packages. To upgrade an existing installation, see Upgrading Cloudera Search on page 393.

To install Cloudera Search on RHEL systems:

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

To install Cloudera Search on Ubuntu and Debian systems:

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

To install Cloudera Search on SLES systems:

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

Note: See also Deploying Cloudera Search.

To list the installed files on RHEL and SLES systems:

```
$ rpm -ql solr-server solr
```

To list the installed files on Ubuntu and Debian systems:

```
$ dpkg -L solr-server solr
```

Cloudera Search packages are configured according to the Linux Filesystem Hierarchy Standard.

Next, enable the server daemons you want to use with Hadoop. You can also enable Java-based client access by adding the JAR files in `/usr/lib/solr/` and `/usr/lib/solr/lib/` to your Java class path.

Installing the Spark Indexer

The Spark indexer uses a Spark or MapReduce ETL batch job to move data from HDFS files into Apache Solr. As part of this process, the indexer uses Morphlines to extract and transform data.

To use the Spark indexer, `solr-crunch` must be installed on hosts where you want to submit a batch indexing job.

By default, this tool is included with Cloudera Search when you have installed CDH using parcels in a Cloudera Manager deployment. If you are using a package-based installation and this tool does not exist on your system, you can install it using the commands described in this topic.

To install solr-crunch On RHEL systems:

```
$ sudo yum install solr-crunch
```

To install solr-crunch on Ubuntu and Debian systems:

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

To install solr-crunch on SLES systems:

```
$ sudo zypper install solr-crunch
```

For information on using Spark to batch index documents, see the Spark Indexing.

Installing MapReduce Tools for use with Cloudera Search

Cloudera Search provides the ability to batch index documents using MapReduce jobs. To use the MapReduce tools, `solr-mapreduce` must be installed on hosts where you want to submit a batch indexing job.

By default, this tool is included with Cloudera Search when you have installed CDH using parcels in a Cloudera Manager deployment. If you are using a package-based installation and this tool does not exist on your system, you can install it using the commands described in this topic.

To install solr-mapreduce On RHEL systems:

```
$ sudo yum install solr-mapreduce
```

To install solr-mapreduce on Ubuntu and Debian systems:

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

To install solr-mapreduce on SLES systems:

```
$ sudo zypper install solr-mapreduce
```

For information on using MapReduce to batch index documents, see the MapReduce Batch Indexing Reference.

Installing the Lily HBase Indexer Service

To query data stored in HBase, you must install the Lily HBase Indexer service. This service indexes the stream of records being added to HBase tables. This process is scalable, fault tolerant, transactional, and operates at near real-time (NRT). The typical delay is a few seconds between the time data arrives and the time the same data appears in search results.

Choosing where to Deploy the Lily HBase Indexer Service Processes

To accommodate the HBase ingest load, you can run as many Lily HBase Indexer services on different hosts as required. See the HBase replication documentation for details on how to plan the capacity. You can co-locate Lily HBase Indexer service processes with SolrCloud on the same set of hosts.

By default, this tool is included with Cloudera Search when you have installed CDH using parcels in a Cloudera Manager deployment. If you are using a package-based installation and this tool does not exist on your system, you can install it using the commands described in this topic.

To install the Lily HBase Indexer service on RHEL systems:

```
$ sudo yum install hbase-solr-indexer hbase-solr-doc
```

To install the Lily HBase Indexer service on Ubuntu and Debian systems:

```
$ sudo apt-get install hbase-solr-indexer hbase-solr-doc
```

To install the Lily HBase Indexer service on SUSE-based systems:

```
$ sudo zypper install hbase-solr-indexer hbase-solr-doc
```

Important: For the Lily HBase Indexer to work with CDH 5, you may need to run the following command before issuing Lily HBase MapReduce jobs:

```
export HADOOP_CLASSPATH=<Path to hbase-protocol-**.jar>
```

Upgrading Cloudera Search

You can upgrade an existing Cloudera Search installation in several ways. Generally, you stop Cloudera Search services, update Search to the latest version, and then restart Cloudera Search services. You can update Search to the latest version by using the package management tool for your operating system and then restarting Cloudera Search services.

Upgrading with Cloudera Manager

If you are running Cloudera Manager, you can upgrade from within the Cloudera Manager Admin Console using parcels. For Search for CDH 5, search in included in the CDH 5 parcel. To upgrade from previous versions of CDH 5, follow the instructions at Upgrading to CDH 5.x Using a Rolling Upgrade.

Upgrading Manually without Cloudera Manager

Important: Before upgrading, make backup copies of the following configuration files:

- `/etc/default/solr` or `/opt/cloudera/parcels/CDH-*/etc/default/solr`
- All collection configurations

Make sure you copy every host that is part of the SolrCloud.

- Cloudera Search for CDH 5 is included as part of CDH 5. Therefore, to upgrade from previous versions of Cloudera Search for CDH 5 to the latest version of Cloudera Search, simply upgrade CDH. For more information, see Upgrading from an Earlier CDH 5 Release to the Latest Release.

Installing Hue Search

You must install and configure Hue before you can use Search with Hue.

1. Follow the instructions for Installing Hue.
2. Use **one** of the following commands to install Search applications on the Hue machine:

   For package installation on RHEL systems:

```
sudo yum install hue-search
```

   For package installation on SLES systems:

```
sudo zypper install hue-search
```

   For package installation on Ubuntu or Debian systems:

```
sudo apt-get install hue-search
```

   For installation using tarballs:

```
$ cd /usr/share/hue
$ sudo tar -xzvf hue-search-####.tar.gz
$ sudo /usr/share/hue/tools/app_reg/app_reg.py \
--install /usr/share/hue/apps/search
```

3. Update the configuration information for the Solr Server:

Cloudera Manager Environment	Environment without Cloudera Manager

1. Connect to Cloudera Manager.
2. Select the **Hue** service.
3. Click the **Configuration** tab.
4. Search for the word "safety".
5. Add information about your Solr host to **Hue Server Advanced Configuration Snippet (Safety Valve) for hue_safety_valve_server.ini**. For example, if your hostname is SOLR_HOST, you might add the following:
`[search]`
`# URL of the Solr Server`
`solr_url=http://SOLR_HOST:8983/solr`
6. (Optional) To enable Hue in environments where Kerberos authentication is required, update the `security_enabled` property as follows:
`# Requires FQDN in solr_url if enabled`
`security_enabled=true`
	Update configuration information in `/etc/hue/hue.ini`.
	1. Specify the Solr URL. For example, to use `localhost` as your Solr host, you would add the following:
	`[search]`
	`# URL of the Solr Server, replace 'localhost' if Solr is running on another host`
	`solr_url=http://localhost:8983/solr/`
	2. (Optional) To enable Hue in environments where Kerberos authentication is required, update the `security_enabled` property as follows:
	`# Requires FQDN in solr_url if enabled`
	`security_enabled=true`

4. Configure secure impersonation for Hue.
   - If you are using Search in an environment that uses Cloudera Manager 4.8 and higher, secure impersonation for Hue is automatically configured. To review secure impersonation settings in the Cloudera Manager home page:
     1. Go to the HDFS service.
     2. Click the **Configuration** tab.
     3. Select **Scope** > **All**.
     4. Select **Category** > **All**.
     5. Type hue `proxy` in the Search box.
     6. Note the Service-Wide wild card setting for **Hue Proxy Hosts** and **Hue Proxy User Groups**.
   - If you are not using Cloudera Manager or are using a version earlier than Cloudera Manager 4.8, configure Hue to impersonate any user that makes requests by modifying `/etc/default/solr` or `/opt/cloudera/parcels/CDH-*/etc/default/solr`. The changes you make may vary according to the users for which you want to configure secure impersonation. For example, you might make the following changes:

```
SOLR_SECURITY_ALLOWED_PROXYUSERS=hue
SOLR_SECURITY_PROXYUSER_hue_HOSTS=*
SOLR_SECURITY_PROXYUSER_hue_GROUPS=*
```

   For more information about Secure Impersonation or to set up additional users for Secure Impersonation, see Enabling Secure Impersonation.

5. (Optional) To view files in HDFS, ensure that the correct `webhdfs_url` is included in `hue.ini` and WebHDFS is properly configured as described in Configuring CDH Components for Hue on page 345.
6. Restart Hue:

```
$ sudo /etc/init.d/hue restart
```

7. Open `http://hue-host.com:8888/search/` in your browser.

Updating Hue Search

To update Hue search, install updates and restart the Hue service.

1. On the Hue machine, update Hue search:

```
$ cd /usr/share/hue
$ sudo tar -xzvf hue-search-####.tar.gz
$ sudo /usr/share/hue/tools/app_reg/app_reg.py \
--install /usr/share/hue/apps/search
```

2. Restart Hue:

```
$ sudo /etc/init.d/hue restart
```

Sentry Installation

Sentry enables role-based, fine-grained authorization for HiveServer2 and Impala. It provides classic database-style authorization for Hive and Impala. For more information, and instructions on configuring Sentry for Hive and Impala, See Managing the Sentry Service.

Installing Sentry

Use the following the instructions, depending on your operating system, to install the latest version of Sentry.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

OS	Command
RHEL	`$ sudo yum install sentry`
SLES	`$ sudo zypper install sentry`
Ubuntu or Debian	`$ sudo apt-get update;` `$ sudo apt-get install sentry`

Upgrading Sentry

Upgrading from CDH 5.x to the Latest CDH 5

1. **Stop the Sentry Service**

   To stop the Sentry service, identify the PID of the Sentry Service and use the `kill` command to end the process:

```
ps -ef | grep sentry
kill -9 <PID>
```

   Replace `<PID>` with the PID of the Sentry Service.

2. Remove the previous version of Sentry.

OS	Command
RHEL	`$ sudo yum remove sentry`
SLES	`$ sudo zypper remove sentry`
Ubuntu or Debian	`$ sudo apt-get remove sentry`

3. Install the new version of Sentry.
4. **Upgrade Sentry Database Schema Using `schematool`**
   - **From a release earlier than CDH 5.2 to CDH 5.4:**

     Use the Sentry `schematool` to upgrade the database schema as follows:

```
bin/sentry --command schema-tool --conffile <sentry-site.xml> --dbType <db-type>
--upgradeSchema
```

     Where `<db-type>` should be either `mysql`, `postgres` or `oracle`.

   - **For CDH 5.5 and higher:** The newer releases include password encryption which means you can no longer run `schematool` as it requires a plaintext password. Password encryption is an important part of security and Sentry defaults to using the CredentialProvider API to accomplish this. Cloudera recommends you use Cloudera Manager to upgrade the Sentry database instead.
   - However, if using Cloudera Manager is not an option, and `scehmatool` is required, to work around the default encryption, obtain the password in plaintext from the API, open `sentry-site.xml` and manually set the `sentry.store.jdbc.password` property to use the plaintext password, and remove the `hadoop.security.credential.provider.path` property and its value. You should now be able to run `schematool`.
5. **Start the Sentry Service**
   a. Set the `SENTRY_HOME` and `HADOOP_HOME` parameters.
   b. Run the following command:

```
bin/sentry --command service --conffile <sentry-site.xml>
```

Snappy Installation

Snappy is a compression/decompression library. It optimizes for very high-speed compression and decompression, and moderate compression instead of maximum compression or compatibility with other compression libraries.

Installing Snappy

Snappy is provided in the `hadoop` package along with the other native libraries (such as native gzip compression).

> **Warning:** If you install Hadoop from a tarball, Snappy may not work, because the Snappy native library may not be compatible with the version of Linux on your system. If you want to use Snappy, install CDH 5 from the RHEL or Debian packages.

To take advantage of Snappy compression you need to set certain configuration properties, which are explained in the following sections.

Upgrading Snappy

To upgrade Snappy, simply install the `hadoop` package if you haven't already done so.

Note: To see which version of Hadoop is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Spark Installation

Spark is a fast, general engine for large-scale data processing.

See also the Apache Spark Documentation.

Spark Packages

The packaging options for installing Spark are:

- RPM packages
- Debian packages

There are five Spark packages:

- `spark-core`: delivers core functionality of Spark
- `spark-worker`: `init` scripts for `spark-worker`
- `spark-master`: `init` scripts for `spark-master`
- `spark-python`: Python client for Spark
- `spark-history-server`

Spark Prerequisites

- An operating system supported by CDH 5.
- Oracle JDK
- The `hadoop-client` package (see Installing the Latest CDH 5 Release on page 223)

Installing and Upgrading Spark

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper`/YaST or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To see which version of Spark is shipping in the current release, check the CDH Version and Packaging Information. For important information, see the CDH 5 Release Notes, in particular:

- New Features and Changes in CDH 5
- Apache Spark Incompatible Changes and Limitations
- Apache Spark Known Issues

Note: If you are using Cloudera Manager 5.9.0, 5.9.1, or 5.10.0 and have hosts with the NodeManager role but without the Spark Gateway role, you must add the Spark Gateway role to all NodeManager hosts and redeploy the client configurations.

- **RHEL-compatible system:**

- To install all Spark packages:

```
$ sudo yum install spark-core spark-master spark-worker spark-history-server spark-python
```

- To install only the packages needed to run Spark on YARN:

```
$ sudo yum install spark-core spark-history-server spark-python
```

- **SLES system:**
  - To install all Spark packages:

```
$ sudo zypper install spark-core spark-master spark-worker spark-history-server
spark-python
```

  - To install only the packages needed to run Spark on YARN:

```
$ sudo zypper install spark-core spark-history-server spark-python
```

- **Ubuntu or Debian system:**
  - To install all Spark packages:

```
$ sudo apt-get install spark-core spark-master spark-worker spark-history-server
spark-python
```

  - To install only the packages needed to run Spark on YARN:

```
$ sudo apt-get install spark-core spark-history-server spark-python
```

You are now ready to configure and start Spark. See Managing Spark Standalone Using the Command Line.

Note:

If you uploaded the Spark JAR file as described under Optimizing YARN Mode in Unmanaged CDH Deployments, use the same instructions to upload the new version of the file each time you upgrade to a new minor release of CDH (for example, any CDH 5.4.*x* release, including 5.4.0).

Sqoop 1 Installation

Apache Sqoop 1 is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. You can use Sqoop 1 to import data from external structured datastores into the Hadoop Distributed File System (HDFS) or related systems such as Hive and HBase. Conversely, you can use Sqoop 1 to extract data from Hadoop and export it to external structured datastores such as relational databases and enterprise data warehouses.

Note:

To see which version of Sqoop 1 is shipping in CDH 5, check the CDH Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Upgrading Sqoop 1 from an Earlier CDH 5 release

These instructions assume that you are upgrading Sqoop 1 as part of an upgrade to the latest CDH 5 release, and have already performed the steps under Upgrading from an Earlier CDH 5 Release to the Latest Release.

To upgrade Sqoop 1 from an earlier CDH 5 release, install the new version of Sqoop 1 using one of the methods described below: Sqoop 1 Prerequisites on page 399 or .Installing the Sqoop 1 Tarball on page 400.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

Sqoop 1 Packaging

The packaging options for installing Sqoop 1 are:

- RPM packages
- Tarball
- Debian packages

Sqoop 1 Prerequisites

Sqoop 1 requires the following:

- An operating system supported by CDH 5.
- Oracle JDK.
- Services that you want to use with Sqoop, such as HBase, Hive HCatalog, and Accumulo. When you run Sqoop, it checks to see if these services are installed and configured. It logs warnings for services it does not find. These warnings, shown below, are harmless. You can suppress these error messages by setting the variables `$HBASE_HOME`, `$HCAT_HOME` and `$ACCUMULO_HOME` to any existing directory.

```
> Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
> Please set $HBASE_HOME to the root of your HBase installation.
> Warning: /usr/lib/sqoop/../hive-hcatalog does not exist! HCatalog jobs will fail.
> Please set $HCAT_HOME to the root of your HCatalog installation.
> Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
> Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

Installing the Sqoop 1 RPM or Debian Packages

Installing the Sqoop 1 RPM or Debian packages is more convenient than installing the Sqoop 1 tarball because the packages:

- Handle dependencies
- Provide for easy upgrades
- Automatically install resources to conventional locations

The Sqoop 1 packages consist of:

- `sqoop` — Complete Sqoop 1 distribution
- `sqoop-metastore` — For installation of the Sqoop 1 metastore only

Note: Install Cloudera Repository

Before using the instructions on this page to install or upgrade:

- Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
- Install or upgrade CDH 5 and make sure it is functioning correctly.

For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install Sqoop 1 on a RHEL-compatible system:

```
$ sudo yum install sqoop
```

To install Sqoop 1 on an Ubuntu or other Debian system:

```
$ sudo apt-get install sqoop
```

To install Sqoop 1 on a SLES system:

```
$ sudo zypper install sqoop
```

If you have already configured CDH on your system, there is no further configuration necessary for Sqoop 1. You can start using Sqoop 1 by using commands such as:

```
$ sqoop help
$ sqoop version
$ sqoop import
```

Installing the Sqoop 1 Tarball

The Sqoop 1 tarball is a self-contained package containing everything necessary to use Sqoop 1 with YARN on a Unix-like system.

Important:

Make sure you have read and understood the section on tarballs before you proceed with a tarball installation.

To install Sqoop 1 from the tarball, unpack the tarball in a convenient location. Once it is unpacked, add the `bin` directory to the shell path for easy access to Sqoop 1 commands. Documentation for users and developers can be found in the `docs` directory.

To install the Sqoop 1 tarball on Linux-based systems:

Run the following command:

```
$ (cd /usr/local/ && sudo tar -zxvf _<path_to_sqoop.tar.gz>_)
```

Note:

When installing Sqoop 1 from the tarball package, you must make sure that the environment variables `JAVA_HOME` and `HADOOP_MAPRED_HOME` are configured correctly. The variable `HADOOP_MAPRED_HOME` should point to the root directory of Hadoop installation. Optionally, if you intend to use any Hive or HBase related functionality, you must also make sure that they are installed. Configure the variables `HIVE_HOME` and `HBASE_HOME` to point to the root directory of their respective installation.

Installing the JDBC Drivers for Sqoop 1

Sqoop 1 does not ship with third party JDBC drivers. You must download them separately and save them to the `/var/lib/sqoop/` directory on the server. The following sections show how to install the most common JDBC Drivers.

Note:

- The JDBC drivers need to be installed only on the machine where Sqoop runs; you do not need to install them on all hosts in your Hadoop cluster.
- Kerberos authentication is not supported by the Sqoop Connector for Teradata.

Before you begin:

Make sure the `/var/lib/sqoop` directory exists and has the correct ownership and permissions:

```
mkdir -p /var/lib/sqoop
chown sqoop:sqoop /var/lib/sqoop
chmod 755 /var/lib/sqoop
```

This sets permissions to `drwxr-xr-x`.

For JDBC drivers for Hive, Impala, Teradata, or Netezza, see the Connectors documentation.

Installing the MySQL JDBC Driver

Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html. You will need to sign up for an account if you do not already have one, and log in, before you can download it. Then copy it to the `/var/lib/sqoop/` directory. For example:

```
$ sudo cp mysql-connector-java-version/mysql-connector-java-version-bin.jar
/var/lib/sqoop/
```

Note:

At the time of publication, *version* was `5.1.31`, but the version may have changed by the time you read this.

Important:

Make sure you have at least version 5.1.31. Some systems ship with an earlier version that may not work correctly with Sqoop.

Installing the Oracle JDBC Driver

You can download the JDBC Driver from the Oracle website, for example http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html. You must accept the license agreement before you can download the driver. Download the `ojdbc6.jar` file and copy it to the `/var/lib/sqoop/` directory:

```
$ sudo cp ojdbc6.jar /var/lib/sqoop/
```

Installing the Microsoft SQL Server JDBC Driver

Download the Microsoft SQL Server JDBC driver from http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774 and copy it to the `/var/lib/sqoop/` directory. For example:

```
$ curl -L
'http://download.microsoft.com/download/0/2/A/02AAE597-3865-456C-AE7F-613F99F850A8/sqljdbc_4.0.2206.100_enu.tar.gz'
 | tar xz
$ sudo cp sqljdbc_4.0/enu/sqljdbc4.jar /var/lib/sqoop/
```

Installing the PostgreSQL JDBC Driver

Download the PostgreSQL JDBC driver from http://jdbc.postgresql.org/download.html and copy it to the `/var/lib/sqoop/` directory. For example:

```
$ curl -L 'http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar' -o
postgresql-9.2-1002.jdbc4.jar
$ sudo cp postgresql-9.2-1002.jdbc4.jar /var/lib/sqoop/
```

Syntax for Configuring JDBC Connection Strings

These are the JDBC connection strings for supported databases.

MySql Connection String

Syntax:

```
jdbc:mysql://<HOST>:<PORT>/<DATABASE_NAME>
```

Example:

```
jdbc:mysql://my_mysql_server_hostname:3306/my_database_name
```

Oracle Connection String

Syntax:

```
jdbc:oracle:thin:@<HOST>:<PORT>:<DATABASE_NAME>
```

Example:

```
jdbc:oracle:thin:@my_oracle_server_hostname:1521:my_database_name
```

PostgreSQL Connection String

Syntax:

`jdbc:postgresql://<HOST>:<PORT>/<DATABASE_NAME>`

Example:

```
jdbc:postgresql://my_postgres_server_hostname:5432/my_database_name
```

Netezza Connection String

Syntax:

`jdbc:netezza://<HOST>:<PORT>/<DATABASE_NAME>`

Example:

```
jdbc:netezza://my_netezza_server_hostname:5480/my_database_name
```

Teradata Connection String

> **Note:** Kerberos authentication is not supported by the Sqoop Connector for Teradata.

Syntax:

```
jdbc:teradata://<HOST>/DBS_PORT=1025/DATABASE=<DATABASE_NAME>
```

Example:

```
jdbc:teradata://my_teradata_server_hostname/DBS_PORT=1025/DATABASE=my_database_name
```

Setting HADOOP_MAPRED_HOME for Sqoop 1

- For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop 1 in a YARN installation, make sure that the `HADOOP_MAPRED_HOME` environment variable is set correctly, as follows:

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

- For each user who will be submitting MapReduce jobs using MapReduce v1 (MRv1), or running Pig, Hive, or Sqoop 1 in an MRv1 installation, set the `HADOOP_MAPRED_HOME` environment variable as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
```

Viewing the Sqoop 1 Documentation

For additional documentation see the Sqoop user guides.

Sqoop 2 Installation

Sqoop 2 is a server-based tool designed to transfer data between Hadoop and relational databases. You can use Sqoop 2 to import data from a relational database management system (RDBMS), such as MySQL or Oracle, into the Hadoop Distributed File System (HDFS), transform the data with Hadoop MapReduce, and then export it back into an RDBMS.

> **Note:** Sqoop 2 is deprecated. Cloudera recommends you use Sqoop 1.

Sqoop 2 has three packaging options for installation:

- Tarball (`.tgz`) that contains both the Sqoop 2 server and the client.
- Separate RPM packages for Sqoop 2 server (`sqoop2-server`) and client (`sqoop2-client`)
- Separate Debian packages for Sqoop 2 server (`sqoop2-server`) and client (`sqoop2-client`)

These topics describe the steps to install Sqoop 2.

Upgrading Sqoop 2 from an Earlier CDH 5 Release

> **Note:** Sqoop 2 is being deprecated. Cloudera recommends using Sqoop 1.

These instructions assume that you are upgrading Sqoop 2 as part of an upgrade to the latest CDH 5 release, and have already performed the steps under Upgrading from an Earlier CDH 5 Release to the Latest Release.

For more detailed instructions for upgrading Sqoop 2, see the Apache Sqoop Upgrade page.

To upgrade Sqoop 2 from an earlier CDH 5 release, proceed as follows:

1. Install the new version of Sqoop 2 following directions under Installing Sqoop 2 on page 404.
2. *If you are running MRv1 on CDH 5 Beta 1 and will continue to run it after upgrading:*
   a. Update `/etc/defaults/sqoop2-server` to point to MR1:

```
mv /etc/defaults/sqoop2-server.rpmnew /etc/defaults/sqoop2-server
```

   b. Update alternatives:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.mr1
```

3. Run the upgrade tool:

```
sqoop2-tool upgrade
```

This upgrades the repository database to the latest version.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by dpkg.

Installing Sqoop 2

Sqoop 2 Prerequisites

Note: Sqoop 2 is deprecated. Cloudera recommends you use Sqoop 1.

- An operating system supported by CDH 5.
- Oracle JDK.
- Hadoop must be installed on the host that runs the Sqoop 2 server component.
- Services that you want to use with Sqoop, such as HBase, Hive HCatalog, and Accumulo. Sqoop checks for these services when you run it, and finds services that are installed and configured. It logs warnings for services it does not find. These warnings, shown below, are harmless.

```
> Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
> Please set $HBASE_HOME to the root of your HBase installation.
> Warning: /usr/lib/sqoop/../hive-hcatalog does not exist! HCatalog jobs will fail.
> Please set $HCAT_HOME to the root of your HCatalog installation.
> Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
> Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

Installing Sqoop 2

Sqoop 2 is distributed as two separate packages: a client package (`sqoop2-client`) and a server package (`sqoop2-server`). Install the server package on one host in the cluster; because the Sqoop 2 server acts as a MapReduce client, this host must have Hadoop installed and configured.

Install the client package on each host that acts as a client. A Sqoop 2 client always connects to the Sqoop 2 server to perform any actions, so Hadoop does not need to be installed on the client hosts.

Depending on what you are planning to install, choose the appropriate package and install it using your preferred package manager application.

> **Note:** The Sqoop 2 packages cannot be installed on the same machines as Sqoop1 packages. However you can use both versions in the same Hadoop cluster by installing Sqoop1 and Sqoop 2 on different hosts.

To install the Sqoop 2 server package on a RHEL-compatible system:

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

```
$ sudo yum install sqoop2-server
```

To install the Sqoop 2 client package on a RHEL-compatible system:

```
$ sudo yum install sqoop2-client
```

To install the Sqoop 2 server package on a SLES system:

```
$ sudo zypper install sqoop2-server
```

To install the Sqoop 2 client package on a SLES system:

```
$ sudo zypper install sqoop2-client
```

To install the Sqoop 2 server package on an Ubuntu or Debian system:

```
$ sudo apt-get install sqoop2-server
```

To install the Sqoop 2 client package on an Ubuntu or Debian system:

```
$ sudo apt-get install sqoop2-client
```

> **Note:**
>
> Installing the `sqoop2-server` package creates a `sqoop-server` service configured to start Sqoop 2 at system startup time.

You are now ready to configure Sqoop 2. See the next section.

Configuring Sqoop 2

This section explains how to configure the Sqoop 2 server.

> **Note:** Sqoop 2 is deprecated. Cloudera recommends you use Sqoop 1.

Configuring which Hadoop Version to Use

The Sqoop 2 client does not interact directly with Hadoop MapReduce, and so it does not require any MapReduce configuration.

The Sqoop 2 server can work with either MRv1 or YARN. **It cannot work with both simultaneously.**You set the MapReduce version the Sqoop 2 server works with by means of the `alternatives` command (or `update-alternatives`, depending on your operating system):

- To use YARN:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.dist
```

- To use MRv1:

```
alternatives --set sqoop2-tomcat-conf /etc/sqoop2/tomcat-conf.mr1
```

> **Important: If you are upgrading from a release earlier than CDH 5 Beta 2**
>
> In earlier releases, the mechanism for setting the MapReduce version was the CATALINA_BASEvariable in the `/etc/defaults/sqoop2-server` file. This does not work as of CDH 5 Beta 2, and in fact could cause problems. **Check your `/etc/defaults/sqoop2-server` file and make sure `CATALINA_BASE` is not set.**

Configuring Sqoop 2 to Use PostgreSQL instead of Apache Derby

Deciding which Database to Use

Sqoop 2 has a built-in Derby database, but Cloudera recommends that you use a PostgreSQL database instead, for the following reasons:

- Derby runs in embedded mode and it is not possible to monitor its health.
- Though it might be possible, Cloudera currently has no live backup strategy for the embedded Derby database.
- Under load, Cloudera has observed locks and rollbacks with the embedded Derby database that do not happen with server-based databases.

See CDH and Cloudera Manager Supported Databases for tested database versions.

> **Note:**
>
> Cloudera currently has no recommended way to migrate data from an existing Derby database into the new PostgreSQL database.

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

Install PostgreSQL 8.4.x or 9.0.x

See External PostgreSQL Database on page 76.

Create the Sqoop User and Sqoop Database

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

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

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

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

postgres=# \q
```

Stop the Sqoop 2 Server

```
$ sudo /sbin/service sqoop2-server stop
```

Configure Sqoop 2 to use PostgreSQL

Edit the `sqoop.properties` file (normally `/etc/sqoop2/conf`) as follows:

```
org.apache.sqoop.repository.jdbc.handler=org.apache.sqoop.repository.postgresql.PostgresqlRepositoryHandler
org.apache.sqoop.repository.jdbc.transaction.isolation=isolation level
org.apache.sqoop.repository.jdbc.maximum.connections=max connections
org.apache.sqoop.repository.jdbc.url=jdbc URL
org.apache.sqoop.repository.jdbc.driver=org.postgresql.Driver
org.apache.sqoop.repository.jdbc.user=username
org.apache.sqoop.repository.jdbc.password=password
org.apache.sqoop.repository.jdbc.properties.property=value
```

Note:

- Replace *isolation level* with a value such as `READ_COMMITTED`.
- Replace *max connections* with a value such as `10`.
- Replace *jdbc URL* with the hostname on which you installed PostgreSQL.
- Replace *username* with (in this example) `sqoop`
- Replace *password* with (in this example) `sqoop`
- Use `org.apache.sqoop.repository.jdbc.properties.property` to set each additional property you want to configure; see https://jdbc.postgresql.org/documentation/head/connect.html for details. For example, replace *property* with `loglevel` and *value* with `3`

Restart the Sqoop 2 Server

```
$ sudo /sbin/service sqoop2-server start
```

Installing the JDBC Drivers

Sqoop 2 does not ship with third party JDBC drivers. You must download them separately and save them to the `/var/lib/sqoop2/` directory on the server. The following sections show how to install the most common JDBC drivers. Once you have installed the JDBC drivers, restart the Sqoop 2 server so that the drivers are loaded.

Note:

The JDBC drivers need to be installed only on the machine running Sqoop. You do not need to install them on all hosts in your Hadoop cluster.

Installing the MySQL JDBC Driver

Download the MySQL JDBC driver here. You must sign up for an account if you do not already have one, then log in before you can download the driver. Copy it to the `/var/lib/sqoop2/` directory. For example:

```
$ sudo cp mysql-connector-java-version/mysql-connector-java-version-bin.jar
/var/lib/sqoop2/
```

At the time of publication, *version* was `5.1.31`, but the version might change by the time you read this.

Important:

Make sure you have at least version 5.1.31. Some systems ship with an earlier version that might not work correctly with Sqoop.

Installing the Oracle JDBC Driver

You can download the JDBC Driver from the Oracle website, for example here. You must accept the license agreement before you can download the driver. Download the `ojdbc6.jar` file and copy it to `/var/lib/sqoop2/` directory:

```
$ sudo cp ojdbc6.jar /var/lib/sqoop2/
```

Installing the Microsoft SQL Server JDBC Driver

Download the Microsoft SQL Server JDBC driver here and copy it to the `/var/lib/sqoop2/` directory. For example:

```
$ curl -L
'http://download.microsoft.com/download/0/2/A/02AAE597-3865-456C-AE7F-613F99F850A8/sqljdbc_4.0.2206.100_enu.tar.gz'
 | tar xz
$ sudo cp sqljdbc_4.0/enu/sqljdbc4.jar /var/lib/sqoop2/
```

Installing the PostgreSQL JDBC Driver

Download the PostgreSQL JDBC driver here and copy it to the `/var/lib/sqoop2/` directory. For example:

```
$ curl -L 'http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar' -o
postgresql-9.2-1002.jdbc4.jar
$ sudo cp postgresql-9.2-1002.jdbc4.jar /var/lib/sqoop2/
```

Syntax for Configuring JDBC Connection Strings

These are the JDBC connection strings for supported databases.

MySql Connection String

Syntax:

```
jdbc:mysql://<HOST>:<PORT>/<DATABASE_NAME>
```

Example:

```
jdbc:mysql://my_mysql_server_hostname:3306/my_database_name
```

Oracle Connection String

Syntax:

```
jdbc:oracle:thin:@<HOST>:<PORT>:<DATABASE_NAME>
```

Example:

```
jdbc:oracle:thin:@my_oracle_server_hostname:1521:my_database_name
```

PostgreSQL Connection String

Syntax:

```
jdbc:postgresql://<HOST>:<PORT>/<DATABASE_NAME>
```

Example:

```
jdbc:postgresql://my_postgres_server_hostname:5432/my_database_name
```

Netezza Connection String

Syntax:

```
jdbc:netezza://<HOST>:<PORT>/<DATABASE_NAME>
```

Example:

```
jdbc:netezza://my_netezza_server_hostname:5480/my_database_name
```

Teradata Connection String

Syntax:

```
jdbc:teradata://<HOST>/DBS_PORT=1025/DATABASE=<DATABASE_NAME>
```

Example:

```
jdbc:teradata://my_teradata_server_hostname/DBS_PORT=1025/DATABASE=my_database_name
```

Starting, Stopping, and Accessing the Sqoop 2 Server

Starting the Sqoop 2 Server

After you have completed all of the required configuration steps, you can start Sqoop 2 server:

```
$ sudo /sbin/service sqoop2-server start
```

Stopping the Sqoop 2 Server

```
$ sudo /sbin/service sqoop2-server stop
```

Checking that the Sqoop 2 Server has Started

You can verify whether the server has started correctly by connecting to its HTTP interface. The simplest way is to get the server version using following command:

```
$ wget -qO - localhost:12000/sqoop/version
```

You should get a text fragment in JSON format similar to the following:

```
{"version":"1.99.2-cdh5.0.0",...}
```

Accessing the Sqoop 2 Server with the Sqoop 2 Client

Start the Sqoop 2 client:

```
sqoop2
```

Identify the host where your server is running (we will use `localhost` in this example):

```
sqoop:000> set server --host localhost
```

Test the connection by running the command `show version --all` to obtain the version number from server. You should see output similar to the following:

```
sqoop:000> show version --all
server version:
  Sqoop 1.99.2-cdh5.0.0 revision ...
  Compiled by jenkins on ...
client version:
  Sqoop 1.99.2-cdh5.0.0 revision ...
  Compiled by jenkins on ...
Protocol version:
  [1]
```

Viewing the Sqoop 2 Documentation

For more information about Sqoop 2, see Highlights of Sqoop 2 and https://archive.cloudera.com/cdh5/cdh/5/sqoop2.

Note: Sqoop 2 is deprecated. Cloudera recommends you use Sqoop 1.

Feature Differences - Sqoop 1 and Sqoop 2

Note: Sqoop 2 is being deprecated. Cloudera recommends using Sqoop 1.

Feature	Sqoop 1	Sqoop 2
Connectors for all major RDBMS	Supported.	Not supported. **Workaround**: Use the generic JDBC Connector which has been tested on the following databases: Microsoft SQL Server, PostgreSQL, MySQL and Oracle. This connector should work on any other JDBC compliant database. However, performance might not be comparable to that of specialized connectors in Sqoop.
Kerberos Security Integration	Supported.	Supported.
Data transfer from RDBMS to Hive or HBase	Supported.	Not supported. **Workaround:** Follow this two-step approach. 1. Import data from RDBMS into HDFS 2. Load data into Hive or HBase manually using appropriate tools and commands such as the LOAD DATA statement in Hive

Feature	Sqoop 1	Sqoop 2
Data transfer from Hive or HBase to RDBMS	Not supported. **Workaround:** Follow this two-step approach. 1. Extract data from Hive or HBase into HDFS (either as a text or Avro file) 2. Use Sqoop to export output of previous step to RDBMS	Not supported. Follow the same workaround as for Sqoop 1.

Whirr Installation

Important: This item is deprecated and will be removed in a future release. Cloudera supports items that are deprecated until they are removed. For more information about deprecated and removed items, see Deprecated Items.

Apache Whirr is a set of libraries for running cloud services. You can use Whirr to run CDH 5 clusters on cloud providers' clusters, such as Amazon Elastic Compute Cloud (Amazon EC2). There's no need to install the RPMs for CDH 5 or do any configuration; a working cluster will start immediately with one command. It's ideal for running temporary Hadoop clusters to carry out a proof of concept, or to run a few one-time jobs. When you are finished, you can destroy the cluster and all of its data with one command.

Use the following sections to install, upgrade, and deploy Whirr:

- Upgrading Whirr
- Installing Whirr
- Generating an SSH Key Pair
- Defining a Cluster
- Launching a Cluster
- Apache Whirr Documentation

Upgrading Whirr

Note:

To see which version of Whirr is shipping in CDH 5, check the Version and Packaging Information. For important information on new and changed components, see the CDH 5 Release Notes.

Upgrading Whirr from an Earlier CDH 5 Release to the Latest CDH 5 Release

Step 1: Stop the Whirr proxy.

Kill the `hadoop-proxy.sh` process by pressing Control-C.

Step 2: Destroy the Cluster.

Whirr clusters are normally short-lived. If you have a running cluster, destroy it: see Destroying a cluster on page 414.

Step 3: Install the New Version of Whirr

See Installing Whirr on page 412.

The upgrade is now complete. For more information, see Managing a Cluster with Whirr on page 413,and Viewing the Whirr Documentation on page 415.

Installing Whirr

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper`/YaST or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

To install Whirr on an Ubuntu or other Debian system:

```
$ sudo apt-get install whirr
```

To install Whirr on a RHEL-compatible system:

```
$ sudo yum install whirr
```

To install Whirr on a SLES system:

```
$ sudo zypper install whirr
```

To install Whirr on another system: Download a Whirr tarball from here.

To verify Whirr is properly installed:

```
$ whirr version
```

Generating an SSH Key Pair for Whirr

After installing Whirr, generate a password-less SSH key pair to enable secure communication with the Whirr cluster.

```
ssh-keygen -t rsa -P ''
```

> **Note:**
>
> If you specify a non-standard location for the key files in the `ssh-keygen` command (that is, not `~/.ssh/id_rsa`), then you must specify the location of the private key file in the `whirr.private-key-file` property and the public key file in the `whirr.public-key-file` property. For more information, see the next section.

Defining a Whirr Cluster

> **Note:**
>
> For information on finding your cloud credentials, see the Whirr FAQ.

After generating an SSH key pair, the only task left to do before using Whirr is to define a cluster by creating a properties file. You can name the properties file whatever you like. The example properties file used in these instructions is named `hadoop.properties`. Save the properties file in your home directory. After defining a cluster in the properties file, you will be ready to launch a cluster and run MapReduce jobs.

Important:

The properties shown below are sufficient to get a bare-bones cluster up and running, but you will probably need to do more configuration to do real-life tasks, especially if you are using HBase and ZooKeeper. You can find more comprehensive template files in the `recipes` directory, for example `recipes/hbase-cdh.properties`.

MRv1 Cluster

The following file defines a cluster with a single machine for the NameNode and JobTracker, and another machine for a DataNode and TaskTracker.

```
whirr.cluster-name=myhadoopcluster
whirr.instance-templates=1 hadoop-jobtracker+hadoop-namenode,1
hadoop-datanode+hadoop-tasktracker
whirr.provider=aws-ec2
whirr.identity=<cloud-provider-identity>
whirr.credential=<cloud-provider-credential>
whirr.private-key-file=${sys:user.home}/.ssh/id_rsa
whirr.public-key-file=${sys:user.home}/.ssh/id_rsa.pub
whirr.env.repo=cdh5
whirr.hadoop-install-function=install_cdh_hadoop
whirr.hadoop-configure-function=configure_cdh_hadoop
whirr.hardware-id=m1.large
whirr.image-id=us-east-1/ami-ccb35ea5
whirr.location-id=us-east-1
```

YARN Cluster

The following configuration provides the essentials for a YARN cluster. Change the number of instances for `hadoop-datanode+yarn-nodemanager` from 2 to a larger number if you need to.

```
whirr.cluster-name=myhadoopcluster
whirr.instance-templates=1 hadoop-namenode+yarn-resourcemanager+mapreduce-historyserver,2
 hadoop-datanode+yarn-nodemanager
whirr.provider=aws-ec2
whirr.identity=<cloud-provider-identity>
whirr.credential=<cloud-provider-credential>
whirr.private-key-file=${sys:user.home}/.ssh/id_rsa
whirr.public-key-file=${sys:user.home}/.ssh/id_rsa.pub
whirr.env.mapreduce_version=2
whirr.env.repo=cdh5
whirr.hadoop.install-function=install_cdh_hadoop
whirr.hadoop.configure-function=configure_cdh_hadoop
whirr.mr_jobhistory.start-function=start_cdh_mr_jobhistory
whirr.yarn.configure-function=configure_cdh_yarn
whirr.yarn.start-function=start_cdh_yarn
whirr.hardware-id=m1.large
whirr.image-id=us-east-1/ami-ccb35ea5
whirr.location-id=us-east-1
```

Managing a Cluster with Whirr

To launch a cluster:

```
$ whirr launch-cluster --config hadoop.properties
```

As the cluster starts up, messages are displayed in the console. You can see debug-level log messages in a file named `whirr.log` in the directory where you ran the `whirr` command. After the cluster has started, a message appears in the console showing the URL you can use to access the web UI for Whirr.

Running a Whirr Proxy

For security reasons, traffic from the network where your client is running is proxied through the master node of the cluster using an SSH tunnel (a SOCKS proxy on port 6666). A script to launch the proxy is created when you launch the cluster, and may be found in `~/.whirr/<cluster-name>`.

To launch the Whirr proxy:

1. Run the following command in a new terminal window:

```
$ . ~/.whirr/myhadoopcluster/hadoop-proxy.sh
```

2. To stop the proxy, kill the process by pressing Ctrl-C.

Running a MapReduce job

After you launch a cluster, a `hadoop-site.xml` file is automatically created in the directory `~/.whirr/<cluster-name>`. You need to update the local Hadoop configuration to use this file.

To update the local Hadoop configuration to use hadoop-site.xml:

1. On all systems, type the following commands:

```
$ cp -r /etc/hadoop/conf.empty /etc/hadoop/conf.whirr
$ rm -f /etc/hadoop/conf.whirr/*-site.xml
$ cp ~/.whirr/myhadoopcluster/hadoop-site.xml /etc/hadoop/conf.whirr
```

2. If you are using an Ubuntu, Debian, or SLES system, type these commands:

```
$ sudo update-alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.whirr
 50
$ update-alternatives --display hadoop-conf
```

3. If you are using a Red Hat system, type these commands:

```
$ sudo alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.whirr 50
$ alternatives --display hadoop-conf
```

4. You can now browse HDFS:

```
$ hadoop fs -ls /
```

To run a MapReduce job, run these commands:

- For MRv1:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-0.20-mapreduce
$ hadoop fs -mkdir input
$ hadoop fs -put $HADOOP_MAPRED_HOME/CHANGES.txt input
$ hadoop jar $HADOOP_MAPRED_HOME/hadoop-examples.jar wordcount input output
$ hadoop fs -cat output/part-* | head
```

- For YARN:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
$ hadoop fs -mkdir input
$ hadoop fs -put $HADOOP_MAPRED_HOME/CHANGES.txt input
$ hadoop jar $HADOOP_MAPRED_HOME/hadoop-mapreduce-examples.jar wordcount input output
$ hadoop fs -cat output/part-* | head
```

Destroying a cluster

When you are finished using a cluster, you can terminate the instances and clean up the resources using the commands shown in this section.

WARNING

All data will be deleted when you destroy the cluster.

To destroy a cluster:

1. Run the following command to destroy a cluster:

```
$ whirr destroy-cluster --config hadoop.properties
```

2. Shut down the SSH proxy to the cluster if you started one earlier.

Viewing the Whirr Documentation

For additional documentation see the Whirr Documentation.

ZooKeeper Installation

Note: Running Services

Use the `service` command to start, stop, and restart CDH components, 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).

Apache ZooKeeper is a highly reliable and available service that provides coordination between distributed processes.

Note: For More Information

From the Apache ZooKeeper site:

> ZooKeeper is a high-performance coordination service for distributed applications. It exposes common services — such as naming, configuration management, synchronization, and group services - in a simple interface so you do not have to write them from scratch. You can use it off-the-shelf to implement consensus, group management, leader election, and presence protocols. And you can build on it for your own, specific needs.

To learn more about Apache ZooKeeper, visit http://zookeeper.apache.org/.

Note:

To see which version of ZooKeeper is shipping in CDH 5, check the CDH Version and Packaging Information. For important information on new and changed components, see the Cloudera Release Guide.

Use the following sections to install, upgrade and administer ZooKeeper:

- Upgrading ZooKeeper from an Earlier CDH 5 Release on page 416
- Installing the ZooKeeper Packages on page 416
- Maintaining a ZooKeeper Server on page 419
- Viewing the ZooKeeper Documentation on page 419

Upgrading ZooKeeper from an Earlier CDH 5 Release

Cloudera recommends that you use a **rolling upgrade** process to upgrade ZooKeeper: that is, upgrade one server in the ZooKeeper ensemble at a time. This means bringing down each server in turn, upgrading the software, then restarting the server. The server will automatically rejoin the quorum, update its internal state with the current ZooKeeper leader, and begin serving client sessions.

This method allows you to upgrade ZooKeeper without any interruption in the service, and also lets you monitor the ensemble as the upgrade progresses, and roll back if necessary if you run into problems.

The instructions that follow assume that you are upgrading ZooKeeper as part of a CDH 5 upgrade, and have already performed the steps underUpgrading from an Earlier CDH 5 Release to the Latest Release.

Performing a ZooKeeper Rolling Upgrade

Follow these steps to perform a rolling upgrade.

Step 1: Stop the ZooKeeper Server on the First Node

To stop the ZooKeeper server:

```
$ sudo service zookeeper-server stop
```

Step 2: Install the ZooKeeper Base Package on the First Node

See Installing the ZooKeeper Base Package.

Step 3: Install the ZooKeeper Server Package on the First Node

See Installing the ZooKeeper Server Package.

Important: Configuration files

- If you install a newer version of a package that is already on the system, configuration files that you have modified will remain intact.
- If you uninstall a package, the package manager renames any configuration files you have modified from `<file>` to `<file>.rpmsave`. If you then re-install the package (probably to install a new version) the package manager creates a new `<file>` with applicable defaults. You are responsible for applying any changes captured in the original configuration file to the new configuration file. In the case of Ubuntu and Debian upgrades, you will be prompted if you have made changes to a file for which there is a new version. For details, see Automatic handling of configuration files by `dpkg`.

Step 4: Restart the Server

See Installing the ZooKeeper Server Package for instructions on starting the server.

The upgrade is now complete on this server and you can proceed to the next.

Step 5: Upgrade the Remaining Nodes

Repeat Steps 1-4 above on each of the remaining nodes.

The ZooKeeper upgrade is now complete.

Installing the ZooKeeper Packages

There are two ZooKeeper server packages:

- The `zookeeper` base package provides the basic libraries and scripts that are necessary to run ZooKeeper servers and clients. The documentation is also included in this package.

- The `zookeeper-server` package contains the `init.d` scripts necessary to run ZooKeeper as a daemon process. Because `zookeeper-server` depends on `zookeeper`, installing the server package automatically installs the base package.

> **Note: Install Cloudera Repository**
>
> Before using the instructions on this page to install or upgrade:
>
> - Install the Cloudera `yum`, `zypper/YaST` or `apt` repository.
> - Install or upgrade CDH 5 and make sure it is functioning correctly.
>
> For instructions, see Installing the Latest CDH 5 Release on page 223 and Upgrading Unmanaged CDH Using the Command Line.

Installing the ZooKeeper Base Package

To install ZooKeeper On RHEL-compatible systems:

```
$ sudo yum install zookeeper
```

To install ZooKeeper on Ubuntu and other Debian systems:

```
$ sudo apt-get install zookeeper
```

To install ZooKeeper on SLES systems:

```
$ sudo zypper install zookeeper
```

Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server

The instructions provided here deploy a single ZooKeeper server in "standalone" mode. This is appropriate for evaluation, testing and development purposes, but may not provide sufficient reliability for a production application. See Installing ZooKeeper in a Production Environment on page 418 for more information.

To install the ZooKeeper Server On RHEL-compatible systems:

```
$ sudo yum install zookeeper-server
```

To install a ZooKeeper server on Ubuntu and other Debian systems:

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

To install ZooKeeper on SLES systems:

```
$ sudo zypper install zookeeper-server
```

To create `/var/lib/zookeeper` and set permissions:

```
mkdir -p /var/lib/zookeeper
chown -R zookeeper /var/lib/zookeeper/
```

To start ZooKeeper

> **Note:**
>
> ZooKeeper may start automatically on installation on Ubuntu and other Debian systems. This automatic start will happen only if the data directory exists; otherwise you will be prompted to initialize as shown below.

- **To start ZooKeeper after an upgrade:**

```
$ sudo service zookeeper-server start
```

- **To start ZooKeeper after a first-time install:**

```
$ sudo service zookeeper-server init
$ sudo service zookeeper-server start
```

Note:

If you are deploying multiple ZooKeeper servers after a fresh install, you need to create a `myid` file in the data directory. You can do this by means of an `init` command option: `$ sudo service zookeeper-server init --myid=1`

Installing ZooKeeper in a Production Environment

In a production environment, you should deploy ZooKeeper as an ensemble with an odd number of servers. As long as a majority of the servers in the ensemble are available, the ZooKeeper service will be available. The minimum recommended ensemble size is three ZooKeeper servers, and Cloudera recommends that each server run on a separate machine. In addition, the ZooKeeper server process should have its own dedicated disk storage if possible.

Deploying a ZooKeeper ensemble requires some additional configuration. The configuration file (`zoo.cfg`) on each server must include a list of all servers in the ensemble, and each server must also have a `myid` file in its data directory (by default `/var/lib/zookeeper`) that identifies it as one of the servers in the ensemble. Proceed as follows *on each server*.

1. Use the commands under Installing the ZooKeeper Server Package and Starting ZooKeeper on a Single Server on page 417 to install `zookeeper-server` on each host.
2. Test the expected loads to set the Java heap size so as to avoid swapping. Make sure you are well below the threshold at which the system would start swapping; for example 12GB for a machine with 16GB of RAM.
3. Create a configuration file. This file can be called anything you like, and must specify settings for at least the parameters shown under "Minimum Configuration" in the ZooKeeper Administrator's Guide. You should also configure values for `initLimit`, `syncLimit`, and `server.n`; see the explanations in the administrator's guide. For example:

```
tickTime=2000
dataDir=/var/lib/zookeeper/
clientPort=2181
initLimit=5
syncLimit=2
server.1=zoo1:2888:3888
server.2=zoo2:2888:3888
server.3=zoo3:2888:3888
```

   In this example, the final three lines are in the form `server.id=hostname:port:port`. The first port is for a follower in the ensemble to listen on for the leader; the second is for leader election. You set *id* for each server in the next step.

4. Create a file named `myid` in the server's `DataDir`; in this example, `/var/lib/zookeeper/myid` . The file must contain only a single line, and that line must consist of a single unique number between 1 and 255; this is the *id* component mentioned in the previous step. In this example, the server whose hostname is `zoo1` must have a `myid` file that contains only `1`.
5. Start each server as described in the previous section.
6. Test the deployment by running a ZooKeeper client:

```
zookeeper-client -server hostname:port
```

For example:

```
zookeeper-client -server zoo1:2181
```

For more information on configuring a multi-server deployment, see Clustered (Multi-Server) Setup in the ZooKeeper Administrator's Guide.

Setting up Supervisory Process for the ZooKeeper Server

The ZooKeeper server is designed to be both highly reliable and highly available. This means that:

- If a ZooKeeper server encounters an error it cannot recover from, it will "fail fast" (the process will exit immediately)
- When the server shuts down, the ensemble remains active, and continues serving requests
- Once restarted, the server rejoins the ensemble without any further manual intervention.

Cloudera recommends that you fully automate this process by configuring a supervisory service to manage each server, and restart the ZooKeeper server process automatically if it fails. See the ZooKeeper Administrator's Guide for more information.

Maintaining a ZooKeeper Server

The ZooKeeper server continually saves `znode` snapshot files and, optionally, transactional logs in a Data Directory to enable you to recover data. It's a good idea to back up the ZooKeeper Data Directory periodically. Although ZooKeeper is highly reliable because a persistent copy is replicated on each server, recovering from backups may be necessary if a catastrophic failure or user error occurs.

When you use the default configuration, the ZooKeeper server does not remove the snapshots and log files, so they will accumulate over time. You will need to clean up this directory occasionally, taking into account on your backup schedules and processes. To automate the cleanup, a `zkCleanup.sh` script is provided in the `bin` directory of the `zookeeper` base package. Modify this script as necessary for your situation. In general, you want to run this as a `cron` task based on your backup schedule.

The data directory is specified by the `dataDir` parameter in the ZooKeeper configuration file, and the data log directory is specified by the `dataLogDir` parameter.

For more information, see Ongoing Data Directory Cleanup.

Viewing the ZooKeeper Documentation

For additional ZooKeeper documentation, see https://archive.cloudera.com/cdh5/cdh/5/zookeeper/.

Building RPMs from CDH Source RPMs

This section describes how to build binary packages (RPMs) from published CDH source packages (SRPMs):

- Prerequisites
- Setting up an Environment for Building RPMs
- Building an RPM

Prerequisites

- Oracle Java Development Kit (JDK) version 6.
- Apache Ant version 1.7 or higher.
- Apache Maven 3.0 or higher.
- The following environment variables must be set: `JAVA_HOME`, `JAVA5_HOME`, `FORREST_HOME`, and `ANT_HOME`.
- Your `PATH` must include the `JAVA_HOME`, `ANT_HOME`, `FORREST_HOME` and maven bin directories.
- If you are using RHEL or CentOS systems, the `rpmdevtools` package is required for the `rpmdev-setuptree` command used below.

Setting Up an Environment for Building RPMs

RHEL or CentOS Systems

Users of these systems can run the following command to set up their environment:

```
$ rpmdev-setuptree                                # Creates ~/rpmbuild and ~/.rpmmacros
```

SLES Systems

Users of these systems can run the following command to set up their environment:

```
$ mkdir -p ~/rpmbuild/{BUILD,RPMS,S{OURCE,PEC,RPM}S}
$ echo "%_topdir $HOME/rpmbuild">  ~/.rpmmacros
```

Building an RPM

Download SRPMs from archive.cloudera.com. The source RPMs for CDH 5 reside at https://archive.cloudera.com/cdh5/redhat/5/x86_64/cdh/5/SRPMS/, https://archive.cloudera.com/cdh5/sles/11/x86_64/cdh/5/SRPMS/ or https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5/SRPMS/. Run the following commands as a non-root user, substituting the particular SRPM that you intend to build:

```
$ export SRPM=hadoop-0.20-0.20.2+320-1.src.rpm
$ rpmbuild --nodeps --rebuild $SRPM                   # Builds the native RPMs
$ rpmbuild --nodeps --rebuild --target noarch $SRPM  # Builds the java RPMs
```

The built packages can be found in `$HOME/rpmbuild/RPMS`.

Apache and Third-Party Licenses

This section describes the licenses that apply to CDH 5.

Apache License

All software developed by Cloudera for CDH is released with an Apache 2.0 license. Please let us know if you find any file that does not explicitly state the Apache license at the top and we'll immediately fix it.

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

Copyright 2010-2013 Cloudera

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Third-Party Licenses

For a list of third-party licenses associated with CDH, see http://www.cloudera.com/content/cloudera-content/cloudera-docs/Licenses/Third-Party-Licenses/Third-Party-Licenses.html.

Uninstalling CDH Components

Before uninstalling CDH, stop all Hadoop processes, following the instructions in Stopping Services.

Here are the commands to use to uninstall the Hadoop components on different Linux systems.

Operating System	Commands	Comments
Red-Hat-compatible	`yum remove`	
Debian and Ubuntu	`apt-get remove` or `apt-get purge`	`apt-get` can be run with the `remove` option to remove only the installed packages or with the `purge` option to remove packages and configuration
SLES	`zypper remove`	

Uninstalling from Red Hat, CentOS, and Similar Systems

Component to Remove	Command
Flume	`$ sudo yum remove flume`
Hadoop core packages	`$ sudo yum remove hadoop`
Hadoop repository packages	`$ sudo yum remove cloudera-cdh`*n*
HBase	`$ sudo yum remove hadoop-hbase`
HDFS HA Journal Node	`$ sudo yum remove hadoop-hdfs-hadoop-hdfs-journalnode`
Hive	`$ sudo yum remove hive hive-metastore hive-server hive-server2`
HttpFS	`$ sudo yum remove hadoop-httpfs`
Hue	`$ sudo yum remove hue`
Mahout	`$ sudo yum remove mahout`
Pig	`$ sudo yum remove pig`
Search	`$ sudo yum remove solr hbase-solr search solr-mapreduce solr-doc search-crunch`
Sentry	`$ sudo yum remove sentry`
Spark	`$ sudo yum remove spark-core spark-master spark-worker spark-history-server spark-python`
Sqoop 1	`$ sudo yum remove sqoop`
Sqoop 2	`$ sudo yum remove sqoop2-server sqoop2-client`
Oozie client	`$ sudo yum remove oozie-client`
Oozie server	`$ sudo yum remove oozie`
Whirr	`$ sudo yum remove whirr`
ZooKeeper server	`$ sudo yum remove hadoop-zookeeper-server`
ZooKeeper client	`$ sudo yum remove hadoop-zookeeper`
ZooKeeper Failover Controller (ZKFC)	`$ sudo yum remove hadoop-hdfs-zkfc`

Uninstalling from Debian and Ubuntu

Use the `apt-get` command to uninstall software on Debian and Ubuntu systems. You can use `apt-get remove` or `apt-get purge`; the difference is that `apt-get remove` removes all your configuration data as well as the package files.

> **Warning:** For this reason, you should `apt-get remove` only with great care, and after making sure you have backed up all your configuration data.

The `apt-get remove` commands to uninstall the Hadoop components from a Debian or Ubuntu system are:

Component to Remove	Command
Flume	`$ sudo apt-get remove flume`
Hadoop core packages	`$ sudo apt-get remove hadoop`
Hadoop repository packages	`$ sudo apt-get remove cdhn-repository`
HBase	`$ sudo apt-get remove hadoop-hbase`
HDFS HA Journal Node	`$ apt-get remove hadoop-hdfs-hadoop-hdfs-journalnode`
Hive	$ sudo apt-get remove hive hive-metastore hive-server hive-server2
HttpFS	`$ sudo apt-get remove hadoop-httpfs`
Hue	`$ sudo apt-get remove hue`
Oozie client	`$ sudo apt-get remove oozie-client`
Oozie server	`$ sudo apt-get remove oozie`
Pig	`$ sudo apt-get remove pig`
Search	`$ sudo apt-get remove solr hbase-solr search solr-mapreduce solr-doc search-crunch`
Sentry	`$ sudo apt-get remove sentry`
Spark	`$ sudo apt-get remove spark-core spark-master spark-worker spark-history-server spark-python`
Sqoop 1	`$ sudo apt-get remove sqoop`
Sqoop 2	`$ sudo apt-get remove sqoop2-server sqoop2-client`
Whirr	`$ sudo apt-get remove whirr`
ZooKeeper client	`$ sudo apt-get remove hadoop-zookeeper`
ZooKeeper Failover Controller (ZKFC)	`$ sudo apt-get remove hadoop-hdfs-zkfc`
ZooKeeper server	`$ sudo apt-get remove hadoop-zookeeper-server`

Uninstalling from SLES

Component to Remove	Command
Flume	`$ sudo zypper remove flume`

Component to Remove	Command
Hadoop core packages	`$ sudo zypper remove hadoop`
Hadoop repository packages	`$ sudo zypper remove cloudera-cdh`
HBase	`$ sudo zypper remove hadoop-hbase`
HDFS HA Journal Node	`$ sudo zypper remove hadoop-hdfs-hadoop-hdfs-journalnode`
Hive	`$ sudo zypper remove hive hive-metastore hive-server hive-server2`
HttpFS	`$ sudo zypper remove hadoop-httpfs`
Hue	`$ sudo zypper remove hue`
Oozie client	`$ sudo zypper remove oozie-client`
Oozie server	`$ sudo zypper remove oozie`
Pig	`$ sudo zypper remove pig`
Search	`$ sudo zypper remove solr hbase-solr search solr-mapreduce solr-doc search-crunch`
Sentry	`$ sudo zypper remove sentry`
Spark	`$ sudo zypper remove spark-core spark-master spark-worker spark-history-server spark-python`
Sqoop 1	`$ sudo zypper remove sqoop`
Sqoop 2	`$ sudo zypper remove sqoop2-server sqoop2-client`
Whirr	`$ sudo zypper remove whirr`
ZooKeeper client	`$ sudo zypper remove hadoop-zookeeper`
ZooKeeper Failover Controller (ZKFC)	`$ sudo zypper remove hadoop-hdfs-zkfc`
ZooKeeper server	`$ sudo zypper remove hadoop-zookeeper-server`

Additional clean-up

The uninstall commands may not remove all traces of Hadoop from your system. The `apt-get purge` commands available for Debian and Ubuntu systems delete more files than the commands that use the `remove` option but are still not comprehensive. If you want to remove all vestiges of Hadoop from your system, look for the following and remove them manually:

- Log files
- Modified system configuration files
- Hadoop configuration files in directories under `/etc` such as `hadoop`, `hbase`, `hue`, `hive`, `oozie`, `sqoop`, `zookeeper`, and `zookeeper.dist`
- User/group identifiers
- Hue, Oozie, and Sqoop databases
- Documentation packages

Viewing the Apache Hadoop Documentation

- For additional Apache Hadoop documentation, see https://archive.cloudera.com/cdh5/cdh/5/hadoop.
- For more information about YARN, see the Apache Hadoop NextGen MapReduce (YARN) page at https://archive.cloudera.com/cdh5/cdh/5/hadoop/hadoop-yarn/hadoop-yarn-site/YARN.html.

Troubleshooting Installation and Upgrade Problems

This topic describes common installation issues and suggested solutions.

The Cloudera Manager Server fails to start after upgrade.

The Cloudera Manager Server fails to start after upgrade.

Possible Reasons

There were active commands running before upgrade. This includes commands a user might have run and also for commands Cloudera Manager automatically triggers, either in response to a state change, or something that's on a schedule.

Possible Solutions

For information on known issues, see Known Issues and Workarounds in Cloudera Manager 5.

Navigator HSM KMS Backed by Thales HSM installation fails

The installation of the Navigator HSM KMS backed by Thales HSM fails with the following error message in the role log:

```
ERROR: Hadoop KMS could not be started

REASON: com.ncipher.provider.nCRuntimeException:
com.ncipher.km.nfkm.nfkmCommunicationException The nfkm command program has terminated
 unexpectedly.
```

Possible Reasons

The KMS user is not part of the `nfast` group on the host(s) running the Navigator HSM KMS backed by Thales HSM role.

Possible Solutions

Add the KMS user to the `nfast` group on the host(s) running the Navigator HSM KMS backed by Thales HSM role:

```
$ sudo usermod -G nfast kms
```

Failed to start server reported by cloudera-manager-installer.bin

"Failed to start server" reported by `cloudera-manager-installer.bin`. `/var/log/cloudera-scm-server/cloudera-scm-server.log`contains a message beginning `Caused by: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver...`

Possible Reasons

You might have SELinux enabled.

Possible Solutions

Disable SELinux by running `sudo setenforce 0` on the Cloudera Manager Server host. To disable it permanently, edit `/etc/selinux/config`. For more information, see Disabling SELinux on page 251.

Installation interrupted and installer does not restart

Installation interrupted and installer does not restart.

Possible Reasons

You need to do some manual cleanup.

Possible Solutions

See Uninstalling Cloudera Manager and Managed Software on page 196.

Cloudera Manager Server fails to start with MySQL

Cloudera Manager Server fails to start and the Server is configured to use a MySQL database to store information about service configuration.

Possible Reasons

Tables might be configured with the ISAM engine. The Server does not start if its tables are configured with the MyISAM engine, and an error such as the following appears in the log file:

```
Tables ... have unsupported engine type ... .  InnoDB is required.
```

Possible Solutions

Make sure that the InnoDB engine is configured, not the MyISAM engine. To check what engine your tables are using, run the following command from the MySQL shell: `mysql> show table status;`

For more information, see MySQL Database on page 89.

Agents fail to connect to Server

Agents fail to connect to Server. You get an Error 113 ('No route to host') in `/var/log/cloudera-scm-agent/cloudera-scm-agent.log`.

Possible Reasons

You might have SELinux or `iptables` enabled.

Possible Solutions

Check `/var/log/cloudera-scm-server/cloudera-scm-server.log` on the Server host and `/var/log/cloudera-scm-agent/cloudera-scm-agent.log` on the Agent hosts. Disable SELinux and `iptables`. For more information, see Disabling SELinux on page 251 and Disabling the Firewall on page 251.

Cluster hosts do not appear

Some cluster hosts do not appear when you click **Find Hosts** in install or update wizard.

Possible Reasons

You may have network connectivity problems.

Possible Solutions

- Make sure all cluster hosts have SSH port 22 open.

- Check other common causes of loss of connectivity such as firewalls and interference from SELinux. For more information, see Disabling SELinux on page 251 and Disabling the Firewall on page 251.

"Access denied" in install or update wizard

"Access denied" in install or update wizard during database configuration for Activity Monitor or Reports Manager.

Possible Reasons

Hostname mapping or permissions are not set up correctly.

Possible Solutions

- For hostname configuration, see Configuring Network Names on page 249.
- For permissions, make sure the values you enter into the wizard match those you used when you configured the databases. The value you enter into the wizard as the database hostname *must* match the value you entered for the hostname (if any) when you configured the database.

  For example, if you had entered the following when you created the database

```
grant all on activity_monitor.* TO 'amon_user'@'myhost1.myco.com' IDENTIFIED BY
'amon_password';
```

  the value you enter here for the database hostname must be `myhost1.myco.com`. If you did not specify a host, or used a wildcard to allow access from any host, you can enter either the fully qualified domain name (FQDN), or `localhost`. For example, if you entered

```
grant all on activity_monitor.* TO 'amon_user'@'%' IDENTIFIED BY 'amon_password';
```

  the value you enter for the database hostname can be either the FQDN or `localhost`.

Databases fail to start.

Activity Monitor, Reports Manager, or Service Monitor databases fail to start.

Possible Reasons

MySQL `binlog` format problem.

Possible Solutions

Set `binlog_format=mixed` in `/etc/my.cnf`. For more information, see this MySQL bug report. See also Cloudera Manager and Managed Service Datastores on page 68.

Cannot start services after upgrade

You have upgraded the Cloudera Manager Server, but now cannot start services.

Possible Reasons

You may have mismatched versions of the Cloudera Manager Server and Agents.

Possible Solutions

Make sure you have upgraded the Cloudera Manager Agents on all hosts. (The previous version of the Agents will heartbeat with the new version of the Server, but you cannot start HDFS and MapReduce with this combination.)

Cloudera services fail to start

Cloudera services fail to start.

Possible Reasons

Java might not be installed or might be installed at a custom location.

Possible Solutions

See Configuring a Custom Java Home Location on page 173 for more information on resolving this issue.

Activity Monitor displays a status of **BAD**

The Activity Monitor displays a status of **BAD** in the Cloudera Manager Admin Console. The log file contains the following message:

```
ERROR 1436 (HY000): Thread stack overrun: 7808 bytes used of a 131072 byte stack, and
128000 bytes needed.
Use 'mysqld -O thread_stack=#' to specify a bigger stack.
```

Possible Reasons

The MySQL thread stack is too small.

Possible Solutions

1. Update the `thread_stack` value in `my.cnf` to 256KB. The `my.cnf` file is normally located in `/etc` or `/etc/mysql`.
2. Restart the `mysql` service: `$ sudo service mysql restart`
3. Restart Activity Monitor.

Activity Monitor fails to start

The Activity Monitor fails to start. Logs contain the error `read-committed isolation not safe for the statement binlog format`.

Possible Reasons

The `binlog_format` is not set to mixed.

Possible Solutions

Modify the `mysql.cnf` file to include the entry for `binlog format` as specified in MySQL Database on page 89.

Attempts to reinstall lower version of Cloudera Manager fail

Attempts to reinstall lower versions of CDH or Cloudera Manager using `yum` fails.

Possible Reasons

It is possible to install, uninstall, and reinstall CDH and Cloudera Manager. In certain cases, this does not complete as expected. If you install Cloudera Manager 5 and CDH 5, then uninstall Cloudera Manager and CDH, and then attempt to install CDH 4 and Cloudera Manager 4, incorrect cached information may result in the installation of an incompatible version of the Oracle JDK.

Possible Solutions

Clear information in the yum cache:

1. Connect to the CDH host.
2. Execute either of the following commands:

```
$ yum --enablerepo='*'clean
          all
```

or

```
$ rm -rf
            /var/cache/yum/cloudera*
```

3. After clearing the cache, proceed with installation.

Create Hive Metastore Database Tables command fails

The **Create Hive Metastore Database Tables** command fails due to a problem with an escape string.

Possible Reasons

PostgreSQL versions 9 and higher require special configuration for Hive because of a backward-incompatible change in the default value of the `standard_conforming_strings` property. Versions up to PostgreSQL 9.0 defaulted to `off`, but starting with version 9.0 the default is on.

Possible Solutions

As the administrator user, use the following command to turn `standard_conforming_strings` off:

```
ALTER DATABASE <hive_db_name> SET standard_conforming_strings = off;
```

HDFS DataNodes fail to start

After upgrading to CDH 5, HDFS DataNodes fail to start with exception:

```
Exception in secureMainjava.lang.RuntimeException: Cannot start datanode because the
configured max locked memory size (dfs.datanode.max.locked.memory) of 4294967296 bytes
 is more than the datanode's available RLIMIT_MEMLOCK ulimit of 65536 bytes.
```

Possible Reasons

HDFS caching, which is enabled by default in CDH 5, requires new memlock functionality from Cloudera Manager Agents.

Possible Solutions

Do the following:

1. Stop all CDH and managed services.
2. On all hosts with Cloudera Manager Agents, hard-restart the Agents. Before performing this step, ensure you understand the semantics of the `hard_restart` command by reading Hard Stopping and Restarting Agents.
    - Packages

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

```
sudo /etc/init.d/cloudera-scm-agent next_stop_hard
sudo systemctl restart cloudera-scm-agent
```

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

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

- Tarballs
  - To stop the Cloudera Manager Agent, run this command on each Agent host:
    - RHEL-compatible 7 and higher:

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

    - All other Linux distributions:

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

  - If you are running single user mode, start Cloudera Manager Agent using the user account you chose. For example, to run the Cloudera Manager Agent as `cloudera-scm`, you have the following options:
    - Run the following command:
      - RHEL-compatible 7 and higher:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent restart
```

      - All other Linux distributions:

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

    - Edit the configuration files so the script internally changes the user, and then run the script as root:
      1. Remove the following line from *tarball_root*`/etc/default/cloudera-scm-agent`:

```
export CMF_SUDO_CMD=" "
```

      2. Change the user and group in *tarball_root*`/etc/init.d/cloudera-scm-agent` to the user you want the Agent to run as. For example, to run as `cloudera-scm`, change the user and group as follows:

```
USER=cloudera-scm
GROUP=cloudera-scm
```

      3. Run the Agent script as root:
         - RHEL-compatible 7 and higher:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent next_stop_hard
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent restart
```

         - All other Linux distributions:

```
$ sudo -u cloudera-scm tarball_root/etc/init.d/cloudera-scm-agent hard_restart
```

3. Start all services.

Create Hive Metastore Database Tables command fails

You see the following error in NameNode log:

```
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:
```

Possible Reasons

You upgraded CDH to 5.2 using Cloudera Manager and did not run the HDFS Metadata Upgrade command.

Possible Solutions

Stop the HDFS service in Cloudera Manager and follow the steps for upgrade (depending on whether you are using packages or parcels) described in Upgrading CDH and Managed Services Using Cloudera Manager.

Oracle invalid identifier

If you are using an Oracle database and the Cloudera **Navigator Analytics** > **Audit** > **Activity** tab displays "No data available" and there is an Oracle error about "invalid identifier" with the query containing the reference to dbms_crypto in the log.

Possible Reasons

You have not granted execute permission to `sys.dbms_crypto`.

Possible Solutions

Run `GRANT EXECUTE ON sys.dbms_crypto TO` *nav*`;`, where *nav* is the user of the Navigator Audit Server database.

Appendix: Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```