



Cloudera Installation

Important Notice

© 2010-2019 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 6.2.x

Date: April 8, 2019

Table of Contents

Cloudera Installation Guide.....	9
Before You Install.....	10
Storage Space Planning for Cloudera Manager.....	10
Cloudera Manager Server.....	11
Cloudera Management Service.....	11
Cloudera Navigator.....	16
Cluster Lifecycle Management with Cloudera Manager.....	18
Configure Network Names.....	21
Disabling the Firewall.....	22
Setting SELinux mode.....	22
Enable an NTP Service.....	23
(RHEL 6 Compatible Only) Install Python 2.7 on Hue Hosts.....	24
Impala Requirements.....	25
Product Compatibility Matrix.....	25
Supported Operating Systems.....	25
Hive Metastore and Related Configuration.....	25
Java Dependencies.....	26
Networking Configuration Requirements.....	26
Hardware Requirements.....	26
User Account Requirements.....	27
Required Privileges for Package-based Installations of CDH.....	27
Ports.....	29
Ports Used by Cloudera Manager and Cloudera Navigator.....	29
Ports Used by Cloudera Navigator Encryption.....	35
Ports Used by CDH Components.....	35
Ports Used by DistCp.....	42
Ports Used by Third-Party Components.....	42
Recommended Cluster Hosts and Role Distribution.....	43
CDH Cluster Hosts and Role Assignments.....	44
Allocating Hosts for Key Trustee Server and Key Trustee KMS	47
Custom Installation Solutions.....	48
Introduction to Parcels.....	48
Understanding Package Management.....	48
Configuring a Local Parcel Repository.....	49
Configuring a Local Package Repository.....	54
Manually Install Cloudera Software Packages.....	58

<i>Creating Virtual Images of Cluster Hosts.....</i>	<i>60</i>
<i>Configuring a Custom Java Home Location.....</i>	<i>63</i>
<i>Creating a CDH Cluster Using a Cloudera Manager Template.....</i>	<i>63</i>
<i>Service Dependencies in Cloudera Manager.....</i>	<i>69</i>
<i>Version 6.2 Service Dependencies.....</i>	<i>69</i>
<i>Version 6.1 Service Dependencies.....</i>	<i>71</i>
<i>Version 6.0 Service Dependencies.....</i>	<i>72</i>
<i>Version 5.16 Service Dependencies.....</i>	<i>73</i>
<i>Version 5.15 Service Dependencies.....</i>	<i>74</i>
<i>Version 5.14 Service Dependencies.....</i>	<i>75</i>
<i>Version 5.13 Service Dependencies.....</i>	<i>76</i>
<i>Version 5.12 Service Dependencies.....</i>	<i>77</i>
<i>Version 5.11.2 Service Dependencies.....</i>	<i>78</i>
<i>Version 5.11.0 Service Dependencies.....</i>	<i>79</i>
<i>Version 5.10.3 Service Dependencies.....</i>	<i>80</i>
<i>Version 5.10 Service Dependencies.....</i>	<i>81</i>
<i>Version 5.9 Service Dependencies.....</i>	<i>82</i>
<i>Version 5.8 Service Dependencies.....</i>	<i>83</i>
<i>Version 5.7.1 Service Dependencies.....</i>	<i>84</i>
<i>Version 5.7.0 Service Dependencies.....</i>	<i>85</i>
<i>Version 5.6.0 Service Dependencies.....</i>	<i>86</i>
<i>Version 5.5.2 Service Dependencies.....</i>	<i>87</i>
<i>Version 5.5.0 Service Dependencies.....</i>	<i>87</i>
<i>Version 5.4.4 Service Dependencies.....</i>	<i>88</i>
<i>Version 5.4.1 Service Dependencies.....</i>	<i>89</i>
<i>Version 5.4.0 Service Dependencies.....</i>	<i>90</i>
<i>Version 5.3.0 Service Dependencies.....</i>	<i>91</i>
<i>Version 5.2.0 Service Dependencies.....</i>	<i>92</i>
<i>Version 5.1.0 Service Dependencies.....</i>	<i>93</i>
<i>Version 5.0.0 Service Dependencies.....</i>	<i>94</i>
<i>Cloudera Data Science Workbench 1.5.0 Service Dependencies.....</i>	<i>95</i>
<i>Spark 2 on YARN Service Dependencies.....</i>	<i>95</i>

Installing Cloudera Manager, CDH, and Managed Services.....96

<i>Step 1: Configure a Repository for Cloudera Manager.....</i>	<i>96</i>
<i>Step 2: Install Java Development Kit.....</i>	<i>97</i>
<i>Requirements.....</i>	<i>97</i>
<i>Installing Oracle JDK Using Cloudera Manager.....</i>	<i>98</i>
<i>Manually Installing Oracle JDK.....</i>	<i>98</i>
<i>Manually Installing OpenJDK.....</i>	<i>99</i>
<i>Step 3: Install Cloudera Manager Server.....</i>	<i>99</i>
<i>Install Cloudera Manager Packages.....</i>	<i>99</i>
<i>(Recommended) Enable Auto-TLS.....</i>	<i>100</i>
<i>Install and Configure Databases.....</i>	<i>101</i>

Step 4: Install and Configure Databases.....	101
<i>Required Databases.....</i>	<i>101</i>
<i>Installing and Configuring Databases.....</i>	<i>102</i>
<i>Install and Configure MariaDB for Cloudera Software.....</i>	<i>102</i>
<i>Install and Configure MySQL for Cloudera Software.....</i>	<i>107</i>
<i>Install and Configure PostgreSQL for Cloudera Software.....</i>	<i>113</i>
<i>Install and Configure Oracle Database for Cloudera Software.....</i>	<i>119</i>
<i>Configuring an External Database for Sqoop 2.....</i>	<i>129</i>
Step 5: Set up the Cloudera Manager Database.....	130
<i>Syntax for scm_prepare_database.sh.....</i>	<i>130</i>
<i>Preparing the Cloudera Manager Server Database.....</i>	<i>131</i>
<i>Installing CDH.....</i>	<i>132</i>
Step 6: Install CDH and Other Software.....	133
<i>Welcome.....</i>	<i>133</i>
<i>Accept License.....</i>	<i>133</i>
<i>Select Edition.....</i>	<i>134</i>
<i>Welcome (Add Cluster - Installation).....</i>	<i>134</i>
<i>Cluster Basics.....</i>	<i>134</i>
<i>Setup Auto-TLS.....</i>	<i>134</i>
<i>Specify Hosts.....</i>	<i>135</i>
<i>Select Repository.....</i>	<i>136</i>
<i>Accept JDK License.....</i>	<i>136</i>
<i>Enter Login Credentials.....</i>	<i>137</i>
<i>Install Agents.....</i>	<i>137</i>
<i>Install Parcels.....</i>	<i>137</i>
<i>Inspect Cluster.....</i>	<i>137</i>
Step 7: Set Up a Cluster Using the Wizard.....	138
<i>Select Services.....</i>	<i>138</i>
<i>Assign Roles.....</i>	<i>138</i>
<i>Setup Database.....</i>	<i>138</i>
<i>Review Changes.....</i>	<i>139</i>
<i>Command Details.....</i>	<i>139</i>
<i>Summary.....</i>	<i>139</i>

Installing the Cloudera Navigator Data Management Component.....140

Installing Cloudera Navigator Encryption Components.....143

Installing Cloudera Navigator Key Trustee Server.....	143
<i>Prerequisites.....</i>	<i>143</i>
<i>Setting Up an Internal Repository.....</i>	<i>143</i>
<i>Installing Key Trustee Server.....</i>	<i>143</i>
<i>Securing Key Trustee Server Host.....</i>	<i>146</i>
<i>Leveraging Native Processor Instruction Sets.....</i>	<i>147</i>

<i>Initializing Key Trustee Server.....</i>	<i>147</i>
<i>Installing Cloudera Navigator Key HSM.....</i>	<i>148</i>
<i>Prerequisites.....</i>	<i>148</i>
<i>Setting Up an Internal Repository.....</i>	<i>148</i>
<i>Installing Navigator Key HSM.....</i>	<i>148</i>
<i>Installing Key Trustee KMS.....</i>	<i>149</i>
<i>Setting Up an Internal Repository.....</i>	<i>149</i>
<i>Installing Key Trustee KMS Using Parcels.....</i>	<i>149</i>
<i>Installing Key Trustee KMS Using Packages.....</i>	<i>149</i>
<i>Post-Installation Configuration.....</i>	<i>150</i>
<i>Installing Navigator HSM KMS Backed by Thales HSM</i>	<i>150</i>
<i>Client Prerequisites.....</i>	<i>150</i>
<i>Setting Up an Internal Repository.....</i>	<i>151</i>
<i>Installing Navigator HSM KMS Backed by Thales HSM Using Parcels.....</i>	<i>151</i>
<i>Installing Navigator HSM KMS Backed by Thales HSM Using Packages.....</i>	<i>151</i>
<i>Installing Navigator HSM KMS Backed by Luna HSM</i>	<i>152</i>
<i>Client Prerequisites.....</i>	<i>152</i>
<i>Setting Up an Internal Repository.....</i>	<i>152</i>
<i>Installing Navigator HSM KMS Backed by Luna HSM Using Parcels.....</i>	<i>152</i>
<i>Installing Navigator HSM KMS Backed by Luna HSM Using Packages.....</i>	<i>153</i>
<i>Post-Installation Configuration.....</i>	<i>153</i>
<i>Installing Cloudera Navigator Encrypt.....</i>	<i>153</i>
<i>Prerequisites.....</i>	<i>153</i>
<i>Setting Up an Internal Repository.....</i>	<i>153</i>
<i>Installing Navigator Encrypt (RHEL-Compatible).....</i>	<i>153</i>
<i>Installing Navigator Encrypt (SLES).....</i>	<i>155</i>
<i>Installing Navigator Encrypt (Debian or Ubuntu).....</i>	<i>156</i>
<i>Post Installation.....</i>	<i>157</i>
<i>Setting Up TLS for Navigator Encrypt Clients.....</i>	<i>157</i>
<i>Entropy Requirements.....</i>	<i>158</i>
<i>Uninstalling and Reinstalling Navigator Encrypt.....</i>	<i>159</i>

After Installation.....160

<i>Deploying Clients.....</i>	<i>160</i>
<i>Testing the Installation.....</i>	<i>160</i>
<i>Checking Host Heartbeats.....</i>	<i>161</i>
<i>Running a MapReduce Job.....</i>	<i>161</i>
<i>Testing with Hue.....</i>	<i>161</i>
<i>Installing the GPL Extras Parcel.....</i>	<i>161</i>
<i>Migrating from Packages to Parcels.....</i>	<i>162</i>
<i>Migrating from Parcels to Packages.....</i>	<i>164</i>
<i>Install CDH and Managed Service Packages.....</i>	<i>164</i>
<i>Deactivate Parcels.....</i>	<i>166</i>

Restart the Cluster.....	167
Remove and Delete Parcels.....	167
Secure Your Cluster.....	167

Troubleshooting Installation Problems.....168

Navigator HSM KMS Backed by Thales HSM installation fails.....	168
Possible Reasons.....	168
Possible Solutions.....	168
Failed to start server reported by cloudera-manager-installer.bin.....	168
Possible Reasons.....	168
Possible Solutions.....	168
Installation interrupted and installer does not restart.....	168
Possible Reasons.....	168
Possible Solutions.....	168
Cloudera Manager Server fails to start with MySQL.....	169
Possible Reasons.....	169
Possible Solutions.....	169
Agents fail to connect to Server.....	169
Possible Reasons.....	169
Possible Solutions.....	169
Cluster hosts do not appear.....	169
Possible Reasons.....	169
Possible Solutions.....	169
"Access denied" in install or update wizard.....	169
Possible Reasons.....	169
Possible Solutions.....	169
Databases fail to start.....	170
Possible Reasons.....	170
Possible Solutions.....	170
Cloudera services fail to start.....	170
Possible Reasons.....	170
Possible Solutions.....	170
Activity Monitor displays a status of BAD	170
Possible Reasons.....	170
Possible Solutions.....	171
Activity Monitor fails to start.....	171
Possible Reasons.....	171
Possible Solutions.....	171
Attempts to reinstall lower version of Cloudera Manager fail.....	171
Possible Reasons.....	171
Possible Solutions.....	171
Create Hive Metastore Database Tables command fails.....	171
Possible Reasons.....	171

<i>Possible Solutions</i>	172
Oracle invalid identifier.....	172
<i>Possible Reasons</i>	172
<i>Possible Solutions</i>	172

Uninstalling Cloudera Manager and Managed Software.....173

Uninstalling Cloudera Manager and Managed Software.....	173
<i>Record User Data Paths</i>	173
<i>Stop all Services</i>	173
<i>Deactivate and Remove Parcels</i>	173
<i>Delete the Cluster</i>	174
<i>Uninstall the Cloudera Manager Server</i>	174
<i>Uninstall Cloudera Manager Agent and Managed Software</i>	174
<i>Remove Cloudera Manager and User Data</i>	176
Uninstalling a CDH Component From a Single Host.....	176

Appendix: Apache License, Version 2.0.....178

Cloudera Installation Guide

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

For non-production environments (such as testing and proof-of- concept use cases), see [Proof-of-Concept Installation Guide](#) for a simplified (but limited) installation procedure.

This guide includes the following sections:

Before You Install

Before you install Cloudera Manager, CDH, and other managed services:

- Review [Cloudera Enterprise 6 Requirements and Supported Versions](#).
- Review the [Cloudera Manager 6 Release Notes](#) and the [CDH 6 Release Notes](#).

For planning, best practices, and recommendations, review the [reference architecture](#) for your environment. For example, for on-premises deployments, review the [Cloudera Enterprise Reference Architecture for Bare Metal Deployments \(PDF\)](#).

The following topics describe additional considerations you should be aware of before beginning an installation:

Storage Space Planning for Cloudera Manager



Note: This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

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 can 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 helps you 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 might not be able to retain historical operational data to meet internal requirements.
- The cluster might miss critical audit information that was not gathered or retained for the required length of time.
- Administrators might be unable to research past events or health status.
- Administrators might not have historical MR1, YARN, or Impala usage data when they need to reference or report on them later.
- There might be gaps in metrics collection and charts.
- The cluster might experience data loss due to filling storage locations to 100% of capacity. The effects of such an event can impact many other components.

The main theme here is that you must architect your data storage needs well in advance. You must 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 internal build documentation and run books.

This topic describes both local disk storage and RDBMS storage. 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 to enable Cloudera Manager administrators to make appropriate storage and lifecycle planning decisions.

Cloudera Manager Server

Table 1: Cloudera Manager Server

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

Cloudera Management Service

Table 2: Cloudera Management Service - Activity Monitor Configuration

Configuration Topic	Activity Monitor
Default Storage Location	Any Supported RDBMS. For more information, see Database Requirements .
Storage Configuration Defaults / Minimum / Maximum	Default: 14 Days worth of MapReduce (MRv1) jobs/tasks
Where to Control Data Retention or Size	<p>You control Activity Monitor storage usage by configuring the number of days or hours of data to retain. Older data is purged.</p> <p>To configure data retention in the Cloudera Manager Administration Console:</p> <ol style="list-style-type: none"> 1. Go the Cloudera Management Service. 2. Click the Configuration tab.

Configuration Topic	Activity Monitor
	<p>3. Select Scope > Activity Monitor or Cloudera Management Service (Service-Wide).</p> <p>4. Select Category > Main.</p> <p>5. Locate the following properties or search for them by typing the property name in the Search box:</p> <p>Purge Activities Data at This Age</p> <p>In Activity Monitor, purge data about MapReduce jobs and aggregate activities when the data reaches this age in hours. By default, Activity Monitor keeps data about activities for 336 hours (14 days).</p> <p>Purge Attempts Data at This Age</p> <p>In the Activity Monitor, purge data about MapReduce attempts when the data reaches this age in hours. Because attempt data can consume large amounts of database space, you might want to purge it more frequently than activity data. By default, Activity Monitor keeps data about attempts for 336 hours (14 days).</p> <p>Purge MapReduce Service Data at This Age</p> <p>The number of hours of past service-level data to keep in the Activity Monitor database, such as total slots running. The default is to keep data for 336 hours (14 days).</p> <p>6. Enter a Reason for change, and then click Save Changes to commit the changes.</p>
Sizing, Planning, and Best Practices	<p>The Activity Monitor only monitors MapReduce jobs, and does not monitor YARN applications. If you no longer use MapReduce (MRv1) in your cluster, the Activity Monitor is not required for Cloudera Manager or CDH.</p> <p>The amount of storage space needed for 14 days worth of MapReduce activities can vary greatly and directly depends on the size of your cluster and the level of activity that uses MapReduce. It might be necessary to adjust and readjust the amount of storage as you determine the "stable state" and "burst state" of the MapReduce activity in your cluster.</p> <p>For example, consider the following test cluster and usage:</p> <ul style="list-style-type: none"> • A simulated 1000-host cluster, each host with 32 slots • MapReduce jobs with 200 attempts (tasks) per activity (job) <p>Sizing observations for this cluster:</p> <ul style="list-style-type: none"> • Each attempt takes 10 minutes to complete. • This usage results in roughly 20 thousand jobs a day with approximately 5 million total attempts. • For a retention period of 7 days, this Activity Monitor database required 200 GB.

Table 3: Cloudera Management Service - Service Monitor Configuration

Configuration Topic	Service Monitor Configuration
Default Storage Location	<code>/var/lib/cloudera-service-monitor/</code> on the host where the Service Monitor role is configured to run.
Storage Configuration Defaults / Minimum / Maximum	<ul style="list-style-type: none"> • 10 GiB Services Time Series Storage • 1 GiB Impala Query Storage

Configuration Topic	Service Monitor Configuration
	<ul style="list-style-type: none"> • 1 GiB YARN Application Storage <p>Total: ~12 GiB Minimum (No Maximum)</p>
Where to Control Data Retention or Size	<p>Service Monitor data growth is controlled by configuring the maximum amount of storage space it can use.</p> <p>To configure data retention in Cloudera Manager Administration Console:</p> <ol style="list-style-type: none"> 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 <i>propertyName</i> property or search for it by typing its name in the Search box. <p>Time-Series Storage</p> <p>The approximate amount of disk space dedicated to storing time series and health data. When the store has reached its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data when it reaches the limit.</p> <p>Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. The Service Monitor stores metric data not only as raw data points but also as ten-minute, hourly, six-hourly, daily, and weekly summary data points. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time while weekly summary points are unlikely to ever be deleted.</p> <p>Select Cloudera Management Service > Charts Library tab in Cloudera Manager for information about how space is consumed within the Service Monitor. These pre-built charts also show information about the amount of data retained and time window covered by each data granularity.</p> <p>Impala Storage</p> <p>The approximate amount of disk space dedicated to storing Impala query data. When the store reaches its maximum size, it deletes older data to make room for newer queries. The disk usage is approximate because the store only begins deleting data when it reaches the limit.</p> <p>YARN Storage</p> <p>The approximate amount of disk space dedicated to storing YARN application data. When the store reaches its maximum size, it deletes older data to make room for newer applications. The disk usage is approximate because Cloudera Manager only begins deleting data when it reaches the limit.</p> <ol style="list-style-type: none"> 6. Enter a Reason for change, and then click Save Changes to commit the changes.
Sizing, Planning, and Best Practices	<p>The Service Monitor gathers metrics about configured roles and services in your cluster and also runs active health tests. These health tests run regardless of idle and use periods, because they are always relevant. The Service Monitor</p>

Configuration Topic	Service Monitor Configuration
	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 4: Cloudera Management Service - Host Monitor

Configuration Topic	Host Monitor Configuration
Default Storage Location	<code>/var/lib/cloudera-host-monitor/</code> on the host where the Host Monitor role is configured to run.
Storage Configuration Defaults / Minimum/ Maximum	Default (and minimum): 10 GiB Host Time Series Storage
Where to Control Data Retention or Size	<p>Host Monitor data growth is controlled by configuring the maximum amount of storage space it can use.</p> <p>See Data Storage for Monitoring Data.</p> <p>To configure these data retention configuration properties in the Cloudera Manager Administration Console:</p> <ol style="list-style-type: none"> 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. <p>Time-Series Storage</p> <p>The approximate amount of disk space dedicated to storing time series and health data. When the store reaches its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data when it reaches the limit.</p> <p>Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. Host Monitor stores metric data not only as raw data points but also as summaries of ten minute, one hour, six hour, one day, and one week increments. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time, while weekly summary points are unlikely to ever be deleted.</p> <p>See the Cloudera Management Service > Charts Library tab in Cloudera Manager for information on how space is consumed within the Host Monitor. These pre-built charts also show information about the amount of data retained and the time window covered by each data granularity.</p> <ol style="list-style-type: none"> 6. Enter a Reason for change, and then 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 5: Cloudera Management Service - Event Server


Configuration Topic	Event Server Configuration
Default Storage Location	<code>/var/lib/cloudera-scm-eventserver/</code> on the host where the Event Server role is configured to run.
Storage Configuration Defaults	5,000,000 events retained
Where to Control Data Retention or Minimum /Maximum	<p>The amount of storage space the Event Server uses is influenced by configuring how many discrete events it can retain.</p> <p>To configure data retention in Cloudera Manager Administration Console,</p> <ol style="list-style-type: none"> 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: <p>Maximum Number of Events in the Event Server Store</p> <p>The maximum size of the Event Server store, in events. When this size is exceeded, events are deleted starting with the oldest first until the size of the store is below this threshold</p> 6. Enter a Reason for change, and then click Save Changes to commit the changes.
Sizing, Planning, and Best Practices	<p>The Event Server is a managed Lucene index that collects relevant events that happen within your cluster, such as results of health tests, log events that are created when a log entry matches a set of rules for identifying messages of interest and makes them available for searching, filtering and additional action. You can view and filter events on the Diagnostics > Events tab of the Cloudera Manager Administration Console. You can also poll this data using the Cloudera Manager API.</p> <div>  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. </div>

Table 6: Cloudera Management Service - Reports Manager

Configuration Topic	Reports Manager Configuration
Default Storage Location	<p>RDBMS:</p> <p>Any Supported RDBMS. For more information, see Database Requirements.</p> <p>Disk:</p> <p><code>/var/lib/cloudera-scm-headlamp/</code> on the host where the Reports Manager role is configured to run.</p>
Storage Configuration Defaults	<p>RDBMS:</p> <p>There are no configurable parameters to directly control the size of this data set.</p>

Configuration Topic	Reports Manager Configuration
	<p>Disk:</p> <p>There are no configurable parameters to directly control the size of this data set. The storage utilization depends not only on the size of the HDFS <i>fsimage</i>, but also on the HDFS file path complexity. Longer file paths contribute to more space utilization.</p>
Where to Control Data Retention or Minimum / Maximum	The Reports Manager uses space in two main locations: on the Reports Manager host and on its supporting database. Cloudera recommends that the database be on a separate host from the Reports Manager host for process isolation and performance.
Sizing, Planning, and Best Practices	<p>Reports Manager downloads the <i>fsimage</i> from the NameNode (every 60 minutes by default) and stores it locally to perform operations against, including indexing the HDFS filesystem structure. More files and directories results in a larger <i>fsimage</i>, which consumes more disk space.</p> <p>Reports Manager has no control over the size of the <i>fsimage</i>. If your total HDFS usage trends upward notably or you add excessively long paths in HDFS, it might be necessary to revisit and adjust the amount of local storage allocated to the Reports Manager. Periodically monitor, review, and adjust the local storage allocation.</p>

Cloudera Navigator

Table 7: Cloudera Navigator - Navigator Audit Server

Configuration Topic	Navigator Audit Server Configuration
Default Storage Location	Any Supported RDBMS. For more information, see Database Requirements .
Storage Configuration Defaults	Default: 90 Days retention
Where to Control Data Retention or Min/Max	<p>Navigator Audit Server storage usage is controlled by configuring how many days of data it can retain. Any older data is purged.</p> <p>To configure data retention in the Cloudera Manager Administration Console:</p> <ol style="list-style-type: none"> 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. <p>Navigator Audit Server Data Expiration Period</p> <p>In Navigator Audit Server, purge audit data of various auditable services when the data reaches this age in days. By default, Navigator Audit Server keeps data about audits for 90 days.</p> <ol style="list-style-type: none"> 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 exceeds the volume of other audits (all other components like MRv1, Hive and Impala read from HDFS, which generates additional audit events).

Configuration Topic	Navigator Audit Server Configuration
	<p>The average size of a discrete HDFS audit event is ~1 KB. For a busy cluster of 50 hosts with ~100K audit events generated per hour, the Navigator Audit Server database would consume ~2.5 GB per day. To retain 90 days of audits at that level, plan for a database size of around 250 GB. If other configured cluster services generate roughly the same amount of data as the HDFS audits, plan for the Navigator Audit Server database to require around 500 GB of storage for 90 days of data.</p> <p>Notes:</p> <ul style="list-style-type: none"> Individual Hive and Impala queries themselves can be very large. Since the query itself is part of an audit event, such audit events consume space in proportion to the length of the query. The amount of space required increases as activity on the cluster increases. In some cases, Navigator Audit Server databases can exceed 1 TB for 90 days of audit events. Benchmark your cluster periodically and adjust accordingly. <p>To map Cloudera Navigator versions to Cloudera Manager versions, see Product Compatibility Matrix for Cloudera Navigator.</p>

Table 8: Cloudera Navigator - Navigator Metadata Server

Configuration Topic	Navigator Metadata Server Configuration
Default Storage Location	<p>RDBMS:</p> <p>Any Supported RDBMS. For more information, see Database Requirements.</p> <p>Disk:</p> <p><code>/var/lib/cloudera-scm-navigator/</code> on the host where the Navigator Metadata Server role is configured to run.</p>
Storage Configuration Defaults	<p>RDBMS:</p> <p>There are no exposed defaults or configurations to directly cull or purge the size of this data set.</p> <p>Disk:</p> <p>There are no configuration defaults to influence the size of this location. You can change the location itself with the Navigator Metadata Server Storage Dir property. The size of the data in this location depends on the amount of metadata in the system (HDFS fsimage size, Hive Metastore size) and activity on the system (the number of MapReduce Jobs run, Hive queries executed, etc).</p>
Where to Control Data Retention or Min/Max	<p>RDBMS:</p> <p>The Navigator Metadata Server database should be carefully tuned to support large volumes of metadata.</p> <p>Disk:</p> <p>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 metadata from the system.</p>
Sizing, Planning, and Best Practices	<p>Memory:</p>

Configuration Topic	Navigator Metadata Server Configuration
	<p>See Navigator Metadata Server Tuning.</p> <p>RDBMS:</p> <p>The database is used to store policies and authorization data. The dataset is small, but this database is also used during a Solr schema upgrade, where Solr documents are extracted and inserted again in Solr. This has same space requirements as above use case, but the space is only used temporarily during product upgrades.</p> <p>Use the Product Compatibility Matrix for Cloudera Navigator product compatibility matrix to map Cloudera Navigator and Cloudera Manager versions.</p> <p>Disk:</p> <p>This filesystem location contains all the metadata that is extracted from managed clusters. The data is stored in Solr, so this is the location where Solr stores its index and documents. Depending on the size of the cluster, this data can occupy tens of gigabytes. A guideline is to look at the size of HDFS fsimage and allocate two to three times that size as the initial size. The data here is incremental and continues to grow as activity is performed on the cluster. The rate of growth can be on order of tens of megabytes per day.</p>

General Performance Notes

When possible:

- For entities that use an RDBMS, install the database on a separate host from the service, and consolidate roles that use databases on as few servers as possible.
- 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 9: Parcel Lifecycle Management

Parcel Lifecycle Path (default)	Notes
Local Parcel Repository Path (<code>/opt/cloudera/parcel-repo</code>)	<p>This path exists only on the host where Cloudera Manager Server (<code>cloudera-scm-server</code>) runs. The Cloudera Manager Server stages all new parcels in this location as it fetches them from any external repositories. Cloudera Manager Agents are then instructed to fetch the parcels from this location when the administrator distributes the parcel using the Cloudera Manager Administration Console or the Cloudera Manager API.</p> <p>Sizing and Planning</p> <p>The default location is <code>/opt/cloudera/parcel-repo</code> but you can configure another local filesystem location on the host where Cloudera Manager Server runs. See Parcel Configuration Settings.</p> <p>Provide sufficient space to hold all the parcels you download from all configured Remote Parcel Repository URLs (See Parcel Configuration Settings). Cloudera Manager deployments that manage multiple clusters store all applicable parcels for all clusters.</p>

Parcel Lifecycle Path (default)	Notes
	<p>Parcels are provided for each operating system, so be aware that heterogeneous clusters (distinct operating systems represented in the cluster) require more space than clusters with homogeneous operating systems.</p> <p>For example, a cluster with both RHEL6.x and 7.x hosts must hold -el6 and -el7 parcels in the Local Parcel Repository Path, which requires twice the amount of space.</p> <p>Lifecycle Management and Best Practices</p> <p>Delete any parcels that are no longer in use from the Cloudera Manager Administration Console, (never delete them manually from the command line) to recover disk space in the Local Parcel Repository Path and simultaneously across all managed cluster hosts which hold the parcel.</p> <p>Backup Considerations</p> <p>Perform regular backups of this path, and consider it a non-optional accessory to backing up Cloudera Manager Server. If you migrate Cloudera Manager Server to a new host or restore it from a backup (for example, after a hardware failure), recover the full content of this path to the new host, in the <code>/opt/cloudera/parcel-repo</code> directory before starting any <code>cloudera-scm-agent</code> or <code>cloudera-scm-server</code> processes.</p>
<p>Parcel Cache (<code>/opt/cloudera/parcel-cache</code>)</p>	<p>Managed Hosts running a Cloudera Manager Agent stage distributed parcels into this path (as <code>.parcel</code> files, unextracted). Do not manually manipulate this directory or its files.</p> <p>Sizing and Planning</p> <p>Provide sufficient space per-host to hold all the parcels you distribute to each host.</p> <p>You can configure Cloudera Manager to remove these cached <code>.parcel</code> files after they are extracted and placed in <code>/opt/cloudera/parcels/</code>. It is not mandatory to keep these temporary files but keeping them avoids the need to transfer the <code>.parcel</code> file from the Cloudera Manager Server repository should you need to extract the parcel again for any reason.</p> <p>To configure this behavior in the Cloudera Manager Administration Console, select Administration > Settings > Parcels > Retain Downloaded Parcel Files</p>
<p>Host Parcel Directory (<code>/opt/cloudera/parcels</code>)</p>	<p>Managed cluster hosts running a Cloudera Manager Agent extract parcels from the <code>/opt/cloudera/parcel-cache</code> directory into this path upon parcel activation. Many critical system symlinks point to files in this path and you should never manually manipulate its contents.</p> <p>Sizing and Planning</p> <p>Provide sufficient space on each host to hold all the parcels you distribute to each host. Be aware that the typical CDH parcel size is approximately 2 GB per parcel, and some third party parcels can exceed 3 GB. If you maintain various versions of parcels staged before and after upgrading, be aware of the disk space implications.</p> <p>You can configure Cloudera Manager to automatically remove older parcels when they are no longer in use. As an administrator you can always manually delete parcel versions not in use, but configuring these settings can handle the deletion automatically, in case you forget.</p>

Parcel Lifecycle Path (default)	Notes
	<p>To configure this behavior in the Cloudera Manager Administration Console, select Administration > Settings > Parcels and configure the following property:</p> <p>Automatically Remove Old Parcels</p> <p>This parameter controls whether parcels for old versions of an activated product should be removed from a cluster when they are no longer in use.</p> <p>The default value is Disabled.</p> <p>Number of Old Parcel Versions to Retain</p> <p>If you enable Automatically Remove Old Parcels, this setting specifies the number of old parcels to keep. Any old parcels beyond this value are removed. If this property is set to zero, no old parcels are retained.</p> <p>The default value is 3.</p>

Table 10: Management Service Lifecycle - Space Reclamation Tasks

Task	Description
Activity Monitor (One-time)	<p>The Activity Monitor only works against a MapReduce (MR1) service, not YARN. So if your deployment has fully migrated to YARN and no longer uses a MapReduce (MR1) service, your Activity Monitor database is no longer growing. If you have waited longer than the default Activity Monitor retention period (14 days) to address this point, then the Activity Monitor has already purged it all for you and your database is mostly empty. If your deployment meets these conditions, consider cleaning up by dropping the Activity Monitor database (again, only when you are satisfied that you no longer need the data or have confirmed that it is no longer in use) and the Activity Monitor role.</p>
Service Monitor and Host Monitor (One-time)	<p>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.</p> <p>After the above migration has taken place, the RDBMS databases previously used by the Service Monitor and Host Monitor are no longer used. Space occupied by these databases is now recoverable. If appropriate in your environment (and you are satisfied that you have long-term backups or do not need the data on disk any longer), you can drop those databases.</p>
Ongoing Space Reclamation	<p>Cloudera Management Services are automatically rolling up, purging or otherwise consolidating aged data for you in the background. Configure retention and purging limits per-role to control how and when this occurs. These configurations are discussed per-entity above. Adjust the default configurations to meet your space limitations or retention needs.</p>

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.

For more information, 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, this lifecycle and backup information can help you keep critical monitoring, auditing, and metadata sources safe and properly backed up.

Configure Network Names



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.example.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.
- Unqualified hostnames (short names) must be unique in a Cloudera Manager instance. For example, you cannot have both `host01.example.com` and `host01.standby.example.com` managed by the same Cloudera Manager Server.

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:

```
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`):

```
Trying "foo-1.example.com"
...
```

```
;; ANSWER SECTION:  
foo-1.example.com. 60 IN A 172.29.82.176
```

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
```

- SLES:

```
sudo chkconfig SuSEfirewall2_setup off  
sudo chkconfig SuSEfirewall2_init off  
sudo rcSuSEfirewall2 stop
```

- Ubuntu:

```
sudo service ufw stop
```

Setting SELinux mode



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 continue on to [Disabling the Firewall](#) on page 22. 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.

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
```

(RHEL 6 Compatible Only) Install Python 2.7 on Hue Hosts

Hue in CDH 6 requires Python 2.7, which is included by default in RHEL 7 compatible operating systems (OSes).

RHEL 6 compatible OSes include Python 2.6. You must install Python 2.7 on all Hue hosts before installing or upgrading to Cloudera Enterprise 6:

RHEL 6

1. Make sure that you have access to the Software Collections Library. For more information, see the Red Hat knowledge base article, [How to use Red Hat Software Collections \(RHSC\) or Red Hat Developer Toolset \(DTS\)?](#).
2. Install Python 2.7:

```
sudo yum install python27
```

3. Verify that Python 2.7 is installed:

```
source /opt/rh/python27/enable  
python --version
```

CentOS 6

1. Enable the Software Collections Library:

```
sudo yum install centos-release-scl
```

2. Install the Software Collections utilities:

```
sudo yum install scl-utils
```

3. Install Python 2.7:

```
sudo yum install python27
```


4. Verify that Python 2.7 is installed:

```
source /opt/rh/python27/enable
python --version
```

Oracle Linux 6**1. Download the Software Collections Library repository:**

```
sudo wget -O /etc/yum.repos.d/public-yum-ol6.repo
http://yum.oracle.com/public-yum-ol6.repo
```

2. Edit /etc/yum.repos.d/public-yum-ol6.repo and make sure that enabled is set to 1, as follows:

```
[ol6_software_collections]
name=Software Collection Library release 3.0 packages for Oracle Linux 6 (x86_64)
baseurl=http://yum.oracle.com/repo/OracleLinux/OL6/SoftwareCollections/x86_64/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

For more information, see [Installing the Software Collection Library Utility From the Oracle Linux Yum Server](#) in the Oracle documentation.

3. Install the Software Collections utilities:

```
sudo yum install scl-utils
```

4. Install Python 2.7:

```
sudo yum install python27
```

5. Verify that Python 2.7 is installed:

```
source /opt/rh/python27/enable
python --version
```

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 platforms. For details, see the *Supported Operating Systems* page for [Operating System Requirements](#).

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.

Always configure a **Hive metastore service** rather than connecting directly to the metastore database. The Hive metastore service is required to interoperate between different levels of metastore APIs if this is necessary for your environment, and using it avoids known issues with connecting directly to the metastore database.

See below for a summary of the metastore installation process.

- Hive (optional). 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.

To install the metastore:

1. Install a MySQL or PostgreSQL database. Start the database if it is not started after installation.
2. Download the [MySQL connector](#) or the [PostgreSQL connector](#) and place it in the `/usr/share/java/` directory.
3. Use the appropriate command line tool for your database to create the metastore database.
4. Use the appropriate command line tool for your database to grant privileges for the metastore database to the `hive` user.
5. 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.

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 SSE3 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.

- JVM Heap Size for Catalog Server

4 GB or more recommended, ideally 8 GB or more, to accommodate the maximum numbers of tables, partitions, and data files you are planning to use with Impala.

- 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. [Impala Security Overview](#) 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.

Required Privileges for Package-based Installations of CDH

The following sections describe the user privilege requirements for package-based installation of CDH with Cloudera Manager. These requirements are standard UNIX system requirements for installing and managing packages and services.

Required Privileges



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 11: Required Privileges for Package-Based CDH Installation

Task	Permissions Required
Install Cloudera Manager Server	root or sudo access to the host on which you are installing Cloudera Manager Server.
Start, stop, or restart Cloudera Manager Server using the <code>service</code> or <code>systemctl</code> utilities	root or sudo access to the Cloudera Manager Server host. The service runs as the <code>cloudera-scm</code> user by default.
Install CDH components using Cloudera Manager	<p>One of the following, configured during initial installation of Cloudera Manager:</p> <ul style="list-style-type: none"> Access to the <code>root</code> user account using a password or SSH key file. Passwordless <code>sudo</code> access for a specific user. <p>For this task, using another system (such as PowerBroker) that provides <code>root</code> or <code>sudo</code> access is <i>not</i> supported.</p>
Install Cloudera Manager Agent using Cloudera Manager	<p>One of the following, configured during initial installation of Cloudera Manager:</p> <ul style="list-style-type: none"> Access to the <code>root</code> user account using a password or SSH key file. Passwordless <code>sudo</code> access for a specific user. <p>For this task, using another system (such as PowerBroker) that provides <code>root</code> or <code>sudo</code> access is <i>not</i> supported.</p>
Automatically start Cloudera Manager Agent process	<p>Access to the <code>root</code> user account during runtime, through one of the following scenarios:</p> <ul style="list-style-type: none"> During Cloudera Manager and CDH installation, the Agent is automatically started if installation is successful. It is then started using one of the following, as configured during the initial installation of Cloudera Manager: <ul style="list-style-type: none"> Access to the <code>root</code> user account using a password or SSH key file. Passwordless <code>sudo</code> access for a specific user. <p>For this task, using another system (such as PowerBroker) that provides <code>root</code> or <code>sudo</code> access is <i>not</i> supported.</p> <ul style="list-style-type: none"> Through automatic startup during system boot, using <code>init</code>.
Manually start, stop, or restart Cloudera Manager Agent process	<p>root or sudo access.</p> <p>This permission requirement ensures that services managed by the Cloudera Manager Agent can run as the appropriate user (such as the <code>hdfs</code> user for the HDFS service). Running commands within Cloudera Manager on a CDH service <i>does not</i> require root or sudo access, because the action is handled by the Cloudera Manager Agent, which is already running as the <code>root</code> user.</p>

sudo Commands Run by Cloudera Manager

If you want to configure specific `sudo` access for the Cloudera Manager user (`cloudera-scm` by default), you can use the following list to do so.

The `sudo` commands run by Cloudera Manager are:

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

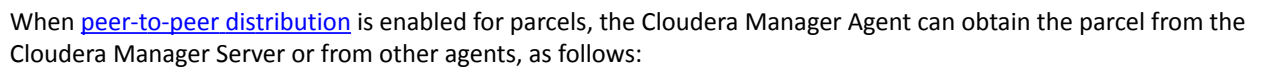
Ports

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` or `firewalld`, and cannot open all the listed ports, you must disable the firewall completely to ensure full functionality.

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

Ports Used by Cloudera Manager and Cloudera Navigator

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



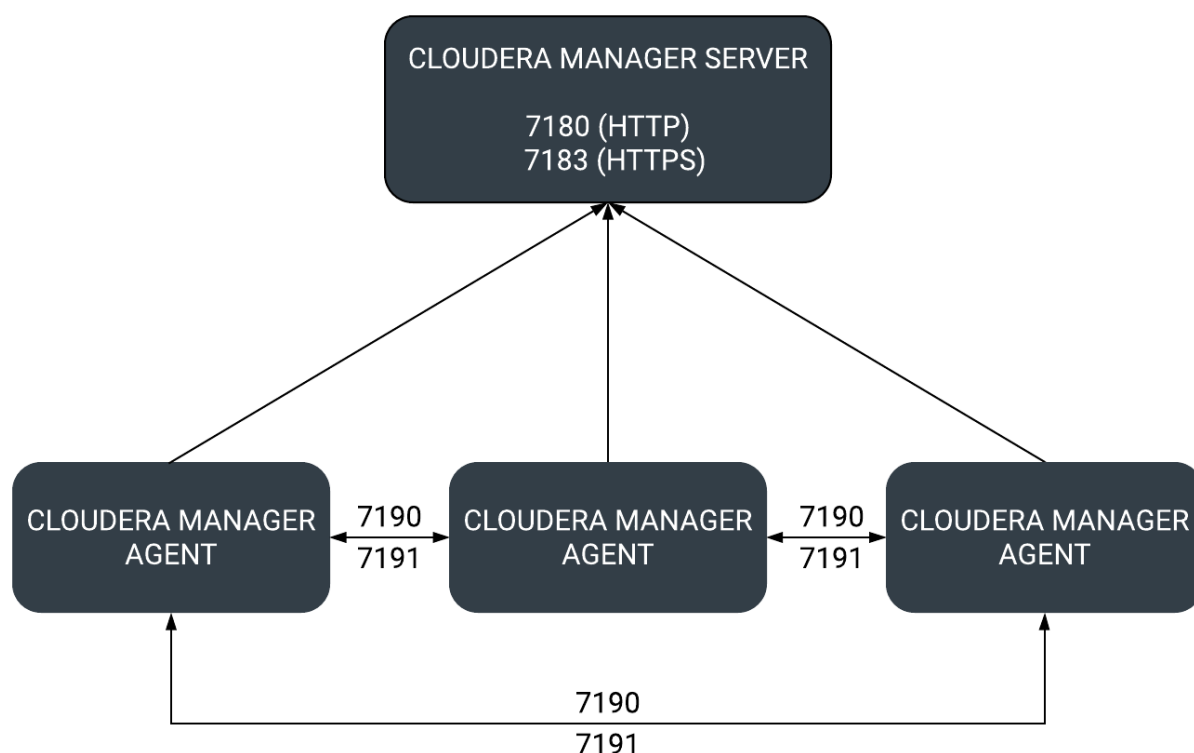


Figure 2: Ports Used in Peer-to-Peer Parcel Distribution

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

In the following tables, *Internal* means that the port is used only for communication among the components; *External* means that the port can be used for either internal or external communication.

Table 12: External Ports

Component	Service	Port	Configuration	Description
Cloudera Manager Server	HTTP (Web UI)	7180	Administration > Settings > Category > Ports and Addresses > HTTP Port for Admin Console	HTTP port used by the web console.
	HTTPS (Web UI)	7183	Administration > Settings > Category > Ports and Addresses > HTTPS Port for Admin Console	Port used by the web console if HTTPS is enabled. If enabled, port 7180 remains open, but redirects all requests to HTTPS on port 7183.
Cloudera Navigator Metadata Server	HTTP (Web UI)	7187	Cloudera Management Service > Configuration > Category > Ports and Addresses > Navigator Metadata Server Port	The port where Navigator Metadata Server listens for requests.

Component	Service	Port	Configuration	Description
Backup and Disaster Recovery	HTTP (Web UI)	7180	Administration > Settings > Category > Ports and Addresses > HTTP Port for Admin Console	Used for communication to peer (source) Cloudera Manager.
	HTTPS (Web UI)	7183	Administration > Settings > Category > Ports and Addresses > HTTPS Port for Admin Console	Used for communication to peer (source) Cloudera Manager when HTTPS is enabled.
	HDFS NameNode	8020	HDFS service > Configuration > Category > 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	HDFS service > Configuration > Category > 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 Hive hosts to destination HDFS DataNode(s).
Telemetry Publisher	HTTP	10110	Clusters > Cloudera Management Service > Category > Ports and Addresses > Telemetry Publisher Server Port	The port where the Telemetry Publisher Server listens for requests
Telemetry Publisher	HTTP (Debug)	10111	Clusters > Cloudera Management Service > Category > Ports and Addresses > Telemetry Publisher Web UI Port	The port where Telemetry Publisher starts a debug web server. Set to -1 to disable debug server.

Table 13: Internal Ports

Component	Service	Port	Configuration	Description
Cloudera Manager Server	Avro (RPC)	7182	Administration > Settings > Category > Ports and Addresses > Agent Port to connect to Server	Used for Agent to Server heartbeats
	Embedded PostgreSQL database	7432		The optional embedded PostgreSQL database used for storing configuration information for Cloudera Manager Server.
	Peer-to-peer parcel distribution	7190, 7191	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	<code>/etc/cloudera-scm-agent/config.ini</code>	
Event Server	Custom protocol	7184	Cloudera Management Service > Configuration > Category > Ports and Addresses > Event Publish Port	Port on which the Event Server listens for the publication of events.
	Custom protocol	7185	Cloudera Management Service > Configuration > Category > Ports and Addresses > Event Query Port	Port on which the Event Server listens for queries for events.
	HTTP (Debug)	8084	Cloudera Management Service > Configuration > Category > Ports and Addresses > Event Server Web UI Port	Port for the Event Server's Debug page. Set to -1 to disable debug server.
Alert Publisher	Custom protocol	10101	Cloudera Management Service > Configuration > Category > Ports and Addresses > Alerts: Listen Port	Port where the Alert Publisher listens for internal API requests.
Service Monitor	HTTP (Debug)	8086	Cloudera Management Service > Configuration > Category > Ports and Addresses > Service Monitor Web UI Port	Port for Service Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)		Cloudera Management Service > Configuration > Category > Ports and Addresses > Service Monitor Web UI HTTPS Port	Port for Service Monitor's HTTPS Debug page.
	Custom protocol	9997	Cloudera Management Service > Configuration > Category > Ports and Addresses > Service Monitor Listen Port	Port where Service Monitor is listening for agent messages.

Component	Service	Port	Configuration	Description
	Internal query API (Avro)	9996	Cloudera Management Service > Configuration > Category > Ports and Addresses > Service Monitor Nozzle Port	Port where Service Monitor's query API is exposed.
Activity Monitor	HTTP (Debug)	8087	Cloudera Management Service > Configuration > Category > Ports and Addresses > Activity Monitor Web UI Port	Port for Activity Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)		Cloudera Management Service > Configuration > Category > Ports and Addresses > Activity Monitor Web UI HTTPS Port	Port for Activity Monitor's HTTPS Debug page.
	Custom protocol	9999	Cloudera Management Service > Configuration > Category > Ports and Addresses > Activity Monitor Listen Port	Port where Activity Monitor is listening for agent messages.
	Internal query API (Avro)	9998	Cloudera Management Service > Configuration > Category > Ports and Addresses > Activity Monitor Nozzle Port	Port where Activity Monitor's query API is exposed.
Host Monitor	HTTP (Debug)	8091	Cloudera Management Service > Configuration > Category > Ports and Addresses > Host Monitor Web UI Port	Port for Host Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)	9091	Cloudera Management Service > Configuration > Category > Ports and Addresses > Host Monitor Web UI HTTPS Port	Port for Host Monitor's HTTPS Debug page.
	Custom protocol	9995	Cloudera Management Service > Configuration > Category > Ports and Addresses > Host Monitor Listen Port	Port where Host Monitor is listening for agent messages.
	Internal query API (Avro)	9994	Cloudera Management Service > Configuration > Category > Ports and Addresses > Host Monitor Nozzle Port	Port where Host Monitor's query API is exposed.
Reports Manager	Queries (Thrift)	5678	Cloudera Management Service > Configuration > Category > Ports and	The port where Reports Manager listens for requests.

Component	Service	Port	Configuration	Description
			Addresses > Reports Manager Server Port	
	HTTP (Debug)	8083	Cloudera Management Service > Configuration > Category > Ports and Addresses > Reports Manager Web UI Port	The port where Reports Manager starts a debug web server. Set to -1 to disable debug server.
Cloudera Navigator Audit Server	HTTP	7186	Cloudera Management Service > Configuration > Category > Ports and Addresses > Navigator Audit Server Port	The port where Navigator Audit Server listens for requests.
	HTTP (Debug)	8089	Cloudera Management Service > Configuration > Category > Ports and Addresses > Navigator Audit Server Web UI Port	The port where Navigator Audit Server runs a debug web server. Set to -1 to disable debug server.

Ports Used by Cloudera Navigator Encryption

All ports listed are TCP.

In the following table, 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; "External" means that the port can be used for either internal or external communication.

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 CDH Components

All ports listed are TCP.

In the following tables, *Internal* means that the port is used only for communication among the components; *External* means that the port can be used for either internal or external communication.

Table 14: External Ports

Component	Service	Port	Configuration	Comment
Apache Hadoop HDFS	DataNode	9866	dfs.datanode.address	DataNode HTTP server port
		1004	dfs.datanode.address	
		9864	dfs.datanode.http.address	
		9865	dfs.datanode.https.address	
		1006	dfs.datanode.http.address	
		9867	dfs.datanode.ipc.address	
	NameNode	8020	fs.default.name or fs.defaultFS	fs.default.name is deprecated (but still works)
		8022	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.
		9870	dfs.http.address or dfs.namenode.http-address	dfs.http.address is deprecated (but still works)
		9871	dfs.https.address or dfs.namenode.https-address	dfs.https.address is deprecated (but still works)
	NFS gateway	2049		nfs port (nfs3.server.port)
		4242		mountd port (nfs3.mountd.port)
		111		portmapper or rpcbind port
		50079	nfs.http.port	The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
		50579	nfs.https.port	The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
	HttpFS	14000		
		14001		
Apache Hadoop YARN (MRv2)	ResourceManager	8032	yarn.resourcemanager.address	
		8033	yarn.resourcemanager.admin.address	
		8088	yarn.resourcemanager.webapp.address	

Component	Service	Port	Configuration	Comment
		8090	yarn.resourcemanager.webapp.https.address	
	NodeManager	8042	yarn.nodemanager.webapp.address	
		8044	yarn.nodemanager.webapp.https.address	
	JobHistory Server	19888	mapreduce.jobhistory.webapp.address	
		19890	mapreduce.jobhistory.webapp.https.address	
	ApplicationMaster			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.
Apache Flume	Flume Agent	41414		
Apache Hadoop KMS	Key Management Server	16000	kms_http_port	Applies to both Java KeyStore KMS and Key Trustee KMS.
Apache HBase	Master	16000	hbase.master.port	IPC
		16010	hbase.master.info.port	HTTP
	RegionServer	16020	hbase.regionserver.port	IPC
		16030	hbase.regionserver.info.port	HTTP
	REST	20550	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		
	Thrift Server	9090	Pass -p <port> on CLI	
	Thrift Server	9095		
		9090	Pass --port <port> on CLI	

Component	Service	Port	Configuration	Comment
	Lily HBase Indexer	11060		
Apache Hive	Metastore	9083		
	HiveServer2	10000	hive. server2. thrift.port	The Beeline command interpreter requires that you specify this port on the command line. If you use Oracle database, you must manually reserve this port. For more information, see Reserving Ports for HiveServer 2 on page 119.
	HiveServer2 Web User Interface (UI)	10002	hive. server2. webui.port in hive-site.xml	
	WebHCat Server	50111	templeton.port	
Apache Hue	Server	8888		
	Load Balancer	8889		
Apache Impala	Impala Daemon	21000		Used to transmit commands and receive results by <code>impala-shell</code> and version 1.2 of the Cloudera ODBC driver.
		21050		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.
		25000		Impala web interface for administrators to monitor and troubleshoot.
	StateStore Daemon	25010		StateStore web interface for administrators to monitor and troubleshoot.
	Catalog Daemon	25020		Catalog service web interface for administrators to monitor and troubleshoot.
Apache Kafka	Broker	9092	port	The primary communication port used by producers and consumers; also used for inter-broker communication.
		9093	ssl_port	A secured communication port used by producers and consumers; also used for inter-broker communication.
Apache Kudu	Master	7051		Kudu Master RPC port

Component	Service	Port	Configuration	Comment
	TabletServer	8051		Kudu Master HTTP server port
		7050		Kudu TabletServer RPC port
		8050		Kudu TabletServer HTTP server port
Apache Oozie	Oozie Server	11000	OOZIE_HTTP_PORT in oozie-env.sh	HTTP
		11443		HTTPS
Apache Sentry	Sentry Server	8038	sentry.service.server.rpc-port	
		51000	sentry.service.web.port	
Apache Solr	Solr Server	8983		All Solr-specific actions, update/query.
Apache Spark	Default Master RPC port	7077		
	Default Worker RPC port	7078		
	Default Master web UI port	18080		
	Default Worker web UI port	18081		
	History Server	18088	history.port	
Apache Sqoop	Metastore	16000	sqoop.metastore.server.port	
Apache ZooKeeper	Server (with CDH or Cloudera Manager)	2181	clientPort	Client port

Table 15: Internal Ports

Component	Service	Port	Configuration	Comment
Apache Hadoop HDFS	Secondary NameNode	9868	dfs.secondary.http.address or dfs.namenode.secondary.http-address	dfs.secondary.http.address is deprecated (but still works)
		9869	dfs.secondary.https.address	
	JournalNode	8485	dfs.namenode.shared.edits.dir	
		8480	dfs.journalnode.http-address	
		8481	dfs.journalnode.https-address	

Component	Service	Port	Configuration	Comment
	Failover Controller	8019		Used for NameNode HA
Apache Hadoop YARN (MRv2)	ResourceManager	8030	yarn.resourcemanager.scheduler.address	
		8031	yarn.resourcemanager.resource-tracker.address	
	NodeManager	8040	yarn.nodemanager.localizer.address	
		8041	yarn.nodemanager.address	
	JobHistory Server	10020	mapreduce.jobhistory.address	
		10033	mapreduce.jobhistory.admin.address	
	Shuffle HTTP	13562	mapreduce.shuffle.port	
Apache Hadoop KMS	Key Management Server	16001	kms_admin_port	Applies to both Java KeyStore KMS and Key Trustee KMS.
Apache HBase	HQuorumPeer	2181	hbase.zookeeper.property.clientPort	HBase-managed ZooKeeper mode
		2888	hbase.zookeeper.peerport	HBase-managed ZooKeeper mode
		3888	hbase.zookeeper.leaderport	HBase-managed ZooKeeper mode
Apache Impala	Impala Daemon	22000		Internal use only. Impala daemons use this port to communicate with each other.
		23000		Internal use only. Impala daemons listen on this port for updates from the statestore daemon.
	StateStore Daemon	24000		Internal use only. The statestore daemon listens on this port for registration/unregistration requests.
	Catalog Daemon	23020		Internal use only. The catalog daemon listens on this port for updates from the statestore daemon.
		26000		Internal use only. The catalog service uses this port to communicate with the Impala daemons.
Apache Kafka	Broker	9092	port	The primary communication port used by producers and

Component	Service	Port	Configuration	Comment
				consumers; also used for inter-broker communication.
		9093	ssl_port	A secured communication port used by producers and consumers; also used for inter-broker communication.
		9393	jmx_port	Internal use only. Used for administration via JMX.
		9394	kafka.http.metrics.port	Internal use only. This is the port via which the HTTP metric reporter listens. It is used to retrieve metrics through HTTP instead of JMX.
	MirrorMaker	24042	jmx_port	Internal use only. Used to administer the producer and consumer of the MirrorMaker.
Apache Solr	Solr Server	8984		Solr administrative use.
Apache Spark	Shuffle service	7337		
Apache ZooKeeper	Server (with CDH only)	2888	X in server.N =host:X:Y	Peer
	Server (with CDH only)	3888	X in server.N =host:X:Y	Peer
	Server (with CDH and Cloudera Manager)	3181	X in server.N =host:X:Y	Peer
	Server (with CDH and Cloudera Manager)	4181	X in server.N =host:X:Y	Peer
	ZooKeeper JMX port	9010		<p>ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must do <i>one</i> of the following:</p> <ul style="list-style-type: none"> • Open up all ports when the connection originates from the Cloudera Manager Server • Do the following: <ol style="list-style-type: none"> 1. Open a non-ephemeral port (such as 9011) in the firewall. 2. Install Oracle Java 7u4 JDK or higher. 3. Add the port configuration to the advanced

Component	Service	Port	Configuration	Comment
				configuration snippet, for example: <pre>-Dcom.sun.management. jmxremote.rmi.port=9011</pre> 4. Restart ZooKeeper.

Ports Used by DistCp

All ports listed are TCP.

In the following table, 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; "External" means that the port can be used for either internal or external communication.

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

In the following table, 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; "External" means that the port can be used for either internal or external communication.

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	By default only UDP

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
						[kdcdefaults] or [realms] sections of kdc.conf	
	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		
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		

Recommended Cluster Hosts and Role Distribution



Important: 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 gateway 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 47.

For information about where to locate various databases that are required for Cloudera Manager and other services, see [Step 4: Install and Configure Databases](#) on page 101.

CDH Cluster Hosts and Role Assignments

Cluster hosts can be broadly described as the following types:

- **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.
- **Gateway hosts** are client access points for launching jobs in the cluster. The number of gateway 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.

The following tables describe the recommended role allocations for different cluster sizes:

3 - 10 Worker Hosts without High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: <ul style="list-style-type: none"> • NameNode • YARN ResourceManager • JobHistory Server • ZooKeeper • Kudu master • Spark History Server 	One host for all Utility and Gateway roles: <ul style="list-style-type: none"> • Secondary NameNode • Cloudera Manager • Cloudera Manager Management Service • Hive Metastore • HiveServer2 • Impala Catalog Server • Impala StateStore • Hue • Oozie • Flume • Gateway configuration 		3 - 10 Worker Hosts: <ul style="list-style-type: none"> • DataNode • NodeManager • Impalad • Kudu tablet server

3 - 20 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: <ul style="list-style-type: none"> • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • JobHistory Server • Spark History Server • Kudu master Master Host 2: <ul style="list-style-type: none"> • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper 	Utility Host 1: <ul style="list-style-type: none"> • Cloudera Manager • Cloudera Manager Management Service • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie • ZooKeeper (requires dedicated disk) • JournalNode (requires dedicated disk) 	One or more Gateway Hosts: <ul style="list-style-type: none"> • Hue • HiveServer2 • Flume • Gateway configuration 	3 - 20 Worker Hosts: <ul style="list-style-type: none"> • DataNode • NodeManager • Impalad • Kudu tablet server

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
<ul style="list-style-type: none"> Kudu master Master Host 3: <ul style="list-style-type: none"> Kudu master (Kudu requires an odd number of masters for HA.) 			

20 - 80 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: <ul style="list-style-type: none"> NameNode JournalNode FailoverController YARN ResourceManager ZooKeeper Kudu master Master Host 2: <ul style="list-style-type: none"> NameNode JournalNode FailoverController YARN ResourceManager ZooKeeper Kudu master Master Host 3: <ul style="list-style-type: none"> ZooKeeper JournalNode JobHistory Server Spark History Server Kudu master 	Utility Host 1: <ul style="list-style-type: none"> Cloudera Manager Utility Host 2: <ul style="list-style-type: none"> Cloudera Manager Management Service Hive Metastore Impala Catalog Server Oozie 	One or more Gateway Hosts: <ul style="list-style-type: none"> Hue HiveServer2 Flume Gateway configuration 	20 - 80 Worker Hosts: <ul style="list-style-type: none"> DataNode NodeManager Impalad Kudu tablet server

80 - 200 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: <ul style="list-style-type: none"> NameNode JournalNode FailoverController YARN ResourceManager ZooKeeper Kudu master Master Host 2: <ul style="list-style-type: none"> NameNode JournalNode FailoverController 	Utility Host 1: <ul style="list-style-type: none"> Cloudera Manager Utility Host 2: <ul style="list-style-type: none"> Hive Metastore Impala Catalog Server Impala StateStore Oozie Utility Host 3: <ul style="list-style-type: none"> Activity Monitor Utility Host 4:	One or more Gateway Hosts: <ul style="list-style-type: none"> Hue HiveServer2 Flume Gateway configuration 	80 - 200 Worker Hosts: <ul style="list-style-type: none"> DataNode NodeManager Impalad Kudu tablet server (Recommended maximum number of tablet servers is 100.)

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
<ul style="list-style-type: none"> YARN ResourceManager ZooKeeper Kudu master <p>Master Host 3:</p> <ul style="list-style-type: none"> ZooKeeper JournalNode JobHistory Server Spark History Server Kudu master 	<ul style="list-style-type: none"> Host Monitor <p>Utility Host 5:</p> <ul style="list-style-type: none"> Navigator Audit Server <p>Utility Host 6:</p> <ul style="list-style-type: none"> Navigator Metadata Server <p>Utility Host 7:</p> <ul style="list-style-type: none"> Reports Manager <p>Utility Host 8:</p> <ul style="list-style-type: none"> Service Monitor 		

200 - 500 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
<p>Master Host 1:</p> <ul style="list-style-type: none"> NameNode JournalNode FailoverController ZooKeeper Kudu master <p>Master Host 2:</p> <ul style="list-style-type: none"> NameNode JournalNode FailoverController ZooKeeper Kudu master <p>Master Host 3:</p> <ul style="list-style-type: none"> YARN ResourceManager ZooKeeper JournalNode Kudu master <p>Master Host 4:</p> <ul style="list-style-type: none"> YARN ResourceManager ZooKeeper JournalNode <p>Master Host 5:</p> <ul style="list-style-type: none"> JobHistory Server Spark History Server ZooKeeper JournalNode 	<p>Utility Host 1:</p> <ul style="list-style-type: none"> Cloudera Manager <p>Utility Host 2:</p> <ul style="list-style-type: none"> Hive Metastore Impala Catalog Server Impala StateStore Oozie <p>Utility Host 3:</p> <ul style="list-style-type: none"> Activity Monitor <p>Utility Host 4:</p> <ul style="list-style-type: none"> Host Monitor <p>Utility Host 5:</p> <ul style="list-style-type: none"> Navigator Audit Server <p>Utility Host 6:</p> <ul style="list-style-type: none"> Navigator Metadata Server <p>Utility Host 7:</p> <ul style="list-style-type: none"> Reports Manager <p>Utility Host 8:</p> <ul style="list-style-type: none"> Service Monitor 	<p>One or more Gateway Hosts:</p> <ul style="list-style-type: none"> Hue HiveServer2 Flume Gateway configuration 	<p>200 - 500 Worker Hosts:</p> <ul style="list-style-type: none"> DataNode NodeManager Impalad Kudu tablet server (Recommended maximum number of tablet servers is 100.)

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
We recommend no more than three Kudu masters.			

500 -1000 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: <ul style="list-style-type: none"> NameNode JournalNode FailoverController ZooKeeper Kudu master Master Host 2: <ul style="list-style-type: none"> NameNode JournalNode FailoverController ZooKeeper Kudu master Master Host 3: <ul style="list-style-type: none"> YARN ResourceManager ZooKeeper JournalNode Kudu master Master Host 4: <ul style="list-style-type: none"> YARN ResourceManager ZooKeeper JournalNode Master Host 5: <ul style="list-style-type: none"> JobHistory Server Spark History Server ZooKeeper JournalNode <p>We recommend no more than three Kudu masters.</p>	Utility Host 1: <ul style="list-style-type: none"> Cloudera Manager Utility Host 2: <ul style="list-style-type: none"> Hive Metastore Impala Catalog Server Impala StateStore Oozie Utility Host 3: <ul style="list-style-type: none"> Activity Monitor Utility Host 4: <ul style="list-style-type: none"> Host Monitor Utility Host 5: <ul style="list-style-type: none"> Navigator Audit Server Utility Host 6: <ul style="list-style-type: none"> Navigator Metadata Server Utility Host 7: <ul style="list-style-type: none"> Reports Manager Utility Host 8: <ul style="list-style-type: none"> Service Monitor 	One or more Gateway Hosts: <ul style="list-style-type: none"> Hue HiveServer2 Flume Gateway configuration 	500 - 1000 Worker Hosts: <ul style="list-style-type: none"> DataNode NodeManager Impalad Kudu tablet server (Recommended maximum number of tablet servers is 100.)

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.

For more information about encrypting data at rest in an EDH, 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](#)

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 package repositories.

These repositories are effective solutions in most cases, but custom installation solutions are sometimes required. Using the Cloudera-hosted software repositories requires client access over the Internet. Typical installations use the latest available software. In some scenarios, these behaviors might not be desirable, such as:

- 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 cannot reach the Cloudera repository. (For a parcel installation, only the Cloudera Manager Server needs Internet access, but for a package installation, all cluster hosts require access to the Cloudera repository). Most organizations partition parts of their network from outside access. Isolating network segments improves security, but can add complexity to the installation process.

In both of these cases, using an internal repository allows you to meet the needs of your organization, whether that means installing specific versions of Cloudera software or installing Cloudera software on hosts without Internet access.

Introduction to 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. For detailed information about parcels, see [Parcels](#).

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel files and put them into a parcel repository. For more information, see [Configuring a Local Parcel Repository](#) on page 49.

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

Packages (`rpm` or `deb` files) help ensure that installations complete successfully by satisfying package dependencies. When you install a particular package, all other required packages are 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` (Ubuntu) are tools that can find and install required packages. For example, on a RHEL compatible system, you might run the command `yum install hadoop-0.20-hive`. The `yum` utility informs you that the Hive package requires `hadoop-0.20` and offers to install it for you. `zypper` and `apt-get` provide similar functionality.

Package Repositories

Package management tools rely on package repositories to install software and resolve any dependency requirements. For information on creating an internal repository, see [Configuring a Local Package Repository](#) on page 54.

Repository Configuration Files

Information about package repositories is stored in configuration files, the location of which varies according to the package management tool.

- **RHEL compatible (yum):** `/etc/yum.repos.d`
- **SLES (zypper):** `/etc/zypp/zypper.conf`
- **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:

```
ls -l /etc/yum.repos.d/
total 36
-rw-r--r--. 1 root root 1664 Dec  9  2015 CentOS-Base.repo
-rw-r--r--. 1 root root 1309 Dec  9  2015 CentOS-CR.repo
-rw-r--r--. 1 root root  649 Dec  9  2015 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  290 Dec  9  2015 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  630 Dec  9  2015 CentOS-Media.repo
-rw-r--r--. 1 root root 1331 Dec  9  2015 CentOS-Sources.repo
-rw-r--r--. 1 root root 1952 Dec  9  2015 CentOS-Vault.repo
-rw-r--r--. 1 root root  951 Jun 24  2017 epel.repo
-rw-r--r--. 1 root root 1050 Jun 24  2017 epel-testing.repo
```

The `.repo` files contain pointers to one or more repositories. There are similar pointers inside configuration files for `zypper` and `apt-get`. In the following excerpt 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=$basearch&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

Listing Repositories

You can list the enabled repositories by running one of the following commands:

- **RHEL compatible:** `yum repolist`
- **SLES:** `zypper repos`
- **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 the output of `yum repolist` on a CentOS 7 system:

repo id	repo name	status
base/7/x86_64	CentOS-7 - Base	9,591
epel/x86_64	Extra Packages for Enterprise Linux 7 - x86_64	12,382
extras/7/x86_64	CentOS-7 - Extras	392
updates/7/x86_64	CentOS-7 - Updates	1,962
repolist: 24,327		

Configuring a Local Parcel Repository

You can create a parcel repository for Cloudera Manager either by hosting an internal Web repository or by manually copying the repository files to the Cloudera Manager Server host for distribution to Cloudera Manager Agent hosts.

Using an Internally Hosted Remote Parcel Repository

The following sections describe how to use an internal Web server to host a parcel repository:

Setting Up a Web Server

To host an internal repository, you must install or use an existing Web server on an internal host that is reachable by the Cloudera Manager host, and then download the repository files to the Web server host. The examples on this page use Apache HTTP Server as the Web server. If you already have a Web server in your organization, you can skip to [Downloading and Publishing the Parcel Repository](#) on page 51.

1. Install Apache HTTP Server:

RHEL / CentOS

```
sudo yum install httpd
```

SLES

```
sudo zypper install httpd
```

Debian

```
sudo apt-get install httpd
```

2.



Warning: Skipping this step could result in an error message **Hash verification failed** when trying to download the parcel from a local repository, especially in Cloudera Manager 6 and higher.

Edit the Apache HTTP Server configuration file (`/etc/httpd/conf/httpd.conf` by default) to add or edit the following line in the `<IfModule mime_module>` section:

```
AddType application/x-gzip .gz .tgz .parcel
```

If the `<IfModule mime_module>` section does not exist, you can add it in its entirety as follows:



Note: This example configuration was modified from the default configuration provided after installing Apache HTTP Server on RHEL 7.

```
<IfModule mime_module>
#
# TypesConfig points to the file containing the list of mappings from
# filename extension to MIME-type.
#
TypesConfig /etc/mime.types

#
# AddType allows you to add to or override the MIME configuration
# file specified in TypesConfig for specific file types.
#
AddType application/x-gzip .tgz
#
# AddEncoding allows you to have certain browsers uncompress
# information on the fly. Note: Not all browsers support this.
#
AddEncoding x-compress .Z
AddEncoding x-gzip .gz .tgz
#
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz .parcel

#
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the server
# or added with the Action directive (see below)
```

```
#
# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)
#
#AddHandler cgi-script .cgi

# For type maps (negotiated resources):
#AddHandler type-map var

#
# Filters allow you to process content before it is sent to the client.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
</IfModule>
```

3. Start Apache HTTP Server:

RHEL 7

```
sudo systemctl start httpd
```

RHEL 6 or lower

```
sudo service httpd start
```

SLES 12, Ubuntu 16 or later, Debian 8

```
sudo systemctl start apache2
```

SLES 11, Ubuntu 14.04, Debian 7 or lower

```
sudo service apache2 start
```

Downloading and Publishing the Parcel Repository

1. Download `manifest.json` and the parcel files for the product you want to install:

CDH 6

Apache Impala, Apache Kudu, Apache Spark 2, and Cloudera Search are included in the CDH parcel. To download the files for the latest CDH 6.2 release, run the following commands on the Web server host:

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories
https://archive.cloudera.com/cdh6/6.2.0/parcels/ -P /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories
https://archive.cloudera.com/gplextras6/6.2.0/parcels/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cdh6
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/gplextras6
```

If you want to create a repository for a different CDH 6 release, replace `6.2.0` with the CDH 6 version that you want. For more information, see [CDH 6 Download Information](#).

CDH 5

Impala, Kudu, Spark 1, and Search are included in the CDH parcel. To download the files for a CDH release (CDH 5.14.4 in this example), run the following commands on the Web server host:

```
sudo mkdir -p /var/www/html/cloudera-repos
```

```
sudo wget --recursive --no-parent --no-host-directories  
https://archive.cloudera.com/cdh5/parcels/5.14.4/ -P /var/www/html/cloudera-repos
```

```
sudo wget --recursive --no-parent --no-host-directories  
https://archive.cloudera.com/gplextras5/parcels/5.14.4/ -P /var/www/html/cloudera-repos
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cdh5
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/gplextras5
```

If you want to create a repository for a different CDH release, replace 5.14.4 with the CDH version that you want. For more information, see [CDH Download Information](#).

Apache Accumulo for CDH

To download the files for an Accumulo release for CDH (Accumulo 1.7.2 in this example), run the following commands on the Web server host:

```
sudo mkdir -p /var/www/html/cloudera-repos  
sudo wget --recursive --no-parent --no-host-directories  
https://archive.cloudera.com/accumulo-c5/parcels/1.7.2/ -P /var/www/html/cloudera-repos  
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/accumulo-c5
```

If you want to create a repository for Accumulo 1.6.0 instead, replace 1.7.2 with 1.6.0.

CDS Powered By Apache Spark 2 for CDH

To download the files for a CDS release for CDH (CDS 2.3.0.cloudera3 in this example), run the following commands on the Web server host:

```
sudo mkdir -p /var/www/html/cloudera-repos  
sudo wget --recursive --no-parent --no-host-directories  
https://archive.cloudera.com/spark2/parcels/2.3.0.cloudera3/ -P  
/var/www/html/cloudera-repos  
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/spark2
```

If you want to create a repository for a different CDS release, replace 2.3.0.cloudera3 with the CDS version that you want. For more information, see [CDS Powered By Apache Spark Version Information](#).

Cloudera Navigator Key Trustee Server

Go to the Key Trustee Server [download page](#). Select **Parcels** from the **CHOOSE DOWNLOAD TYPE** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee Server parcels and manifest.json files in a .tar.gz file. Copy the file to your Web server, and extract the files with the `tar xvfz filename.tar.gz` command. This example uses Key Trustee Server 5.14.0:

```
sudo mkdir -p /var/www/html/cloudera-repos/keytrustee-server  
sudo tar xvfz /path/to/keytrustee-server-5.14.0-parcels.tar.gz -C  
/var/www/html/cloudera-repos/keytrustee-server --strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/keytrustee-server
```

Cloudera Navigator Key Trustee KMS and HSM KMS



Note: Cloudera Navigator HSM KMS is included in the Key Trustee KMS parcel.

Go to the Key Trustee KMS [download page](#). Select **Parcels** from the **CHOOSE DOWNLOAD TYPE** drop-down menu, and click **DOWNLOAD NOW**. This downloads the Key Trustee KMS parcels and `manifest.json` files in a `.tar.gz` file. Copy the file to your Web server, and extract the files with the `tar xvfz filename.tar.gz` command. This example uses Key Trustee KMS 5.14.0:

```
sudo mkdir -p /var/www/html/cloudera-repos/keytrustee-kms
sudo tar xvfz /path/to/keytrustee-kms-5.14.0-parcels.tar.gz -C
/var/www/html/cloudera-repos/keytrustee-kms --strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/keytrustee-kms
```

Sqoop Connectors

To download the parcels for a Sqoop Connector release, run the following commands on the Web server host. This example uses the latest available Sqoop Connectors:

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories
http://archive.cloudera.com/sqoop-connectors/parcels/latest/ -P
/var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/sqoop-connectors
```

If you want to create a repository for a different Sqoop Connector release, replace `latest` with the Sqoop Connector version that you want. You can see a list of versions in the [parcels](#) parent directory.

2. Visit the Repository URL `http://<Web_server>/cloudera-repos/` in your browser and verify the files you downloaded are present. If you do not see anything, your Web server may have been configured to not show indexes.

Configuring Cloudera Manager to Use an Internal Remote Parcel Repository

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

- **Navigation bar**

1. Click the parcel icon 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://<web_server>/cloudera-parcels/cdh6/6.2.0/`
4. Enter a **Reason for change**, and then 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 Admin Console 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. For instructions on downloading parcels, see [Downloading and Publishing the Parcel Repository](#) on page 51 above.
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-6.1.0-1.cdh6.1.0.p0.770702-e17.parcel:

```
shasum CDH-6.1.0-1.cdh6.1.0.p0.770702-e17.parcel | awk '{ print $1 }' >  
CDH-6.1.0-1.cdh6.1.0.p0.770702-e17.parcel.sha
```

7. Change the ownership of the parcel and hash files to `cloudera-scm`:

```
sudo chown -R cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/*
```

8. In the Cloudera Manager Admin Console, navigate to the **Parcels** page.
9. Click **Check for New Parcels** and verify that the new parcel appears.
10. Download, distribute, and activate the parcel.

Configuring a Local Package Repository

You can create a package repository for Cloudera Manager either by hosting an internal web repository or by manually copying the repository files to the Cloudera Manager Server host for distribution to Cloudera Manager Agent hosts.

Creating a Permanent Internal Repository

The following sections describe how to create a permanent internal repository using Apache HTTP Server:

Setting Up a Web server

To host an internal repository, you must install or use an existing Web server on an internal host that is reachable by the Cloudera Manager host, and then download the repository files to the Web server host. The examples in this section use Apache HTTP Server as the Web server. If you already have a Web server in your organization, you can skip to [Downloading and Publishing the Package Repository](#) on page 55.

1. Install Apache HTTP Server:

RHEL / CentOS

```
sudo yum install httpd
```

SLES

```
sudo zypper install httpd
```

Debian

```
sudo apt-get install httpd
```

2. Start Apache HTTP Server:

RHEL 7

```
sudo systemctl start httpd
```

RHEL 6 or lower

```
sudo service httpd start
```

SLES 12, Ubuntu 16 or later, Debian 8

```
sudo systemctl start apache2
```

SLES 11, Ubuntu 14.04, Debian 7 or lower

```
sudo service apache2 start
```

Downloading and Publishing the Package Repository

1. Download the package repository for the product you want to install:

Cloudera Manager 6

To download the files for the latest Cloudera Manager 6.2 release, run the following commands on the Web server host. Replace `<operating_system>` with the operating system you are using (redhat7, redhat6, sles12, ubuntu1604, or ubuntu1804):

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories
https://archive.cloudera.com/cm6/6.2.0/<operating_system>/ -P /var/www/html/cloudera-repos
sudo wget https://archive.cloudera.com/cm6/6.2.0/allkeys.asc -P
/var/www/html/cloudera-repos/cm6/6.2.0/
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cm6
```

If you want to create a repository for a different Cloudera Manager 6 release, replace 6.2.0 with the CDH 6 version that you want. For more information, see [Cloudera Manager 6 Version and Download Information](#).

CDH 6

To download the files for the latest CDH 6.2 release, run the following commands on the Web server host. Replace `<operating_system>` with the operating system you are using (redhat7, redhat6, sles12, ubuntu1604, or ubuntu1804):

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories
https://archive.cloudera.com/cdh6/6.2.0/<operating_system>/ -P
/var/www/html/cloudera-repos
```

```
sudo wget --recursive --no-parent --no-host-directories
https://archive.cloudera.com/gplextras6/6.2.0/<operating_system>/ -P
/var/www/html/cloudera-repos
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cdh6
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/gplextras6
```

If you want to create a repository for a different CDH 6 release, replace 6.2.0 with the CDH 6 version that you want. For more information, see [CDH 6 Download Information](#).

Cloudera Manager 5

To download the files for a Cloudera Manager release, download the repository tarball for your operating system.

```
sudo mkdir -p /var/www/html/cloudera-repos/cm5
```

Redhat/Centos:

```
wget https://archive.cloudera.com/cm5/repo-as-tarball/5.14.4/cm5.14.4-centos7.tar.gzd
```

```
tar xvfz cm5.14.4-centos7.tar.gz -C /var/www/html/cloudera-repos/cm5 --strip-components=1
```

Debian:

```
wget https://archive.cloudera.com/cm5/repo-as-tarball/5.14.4/cm5.14.4-debian-jessie.tar.gz
```

```
tar xvfz cm5.14.4-debian-jessie.tar.gz -C /var/www/html/cloudera-repos/cm5  
--strip-components=1
```

SLES:

```
wget https://archive.cloudera.com/cm5/repo-as-tarball/5.14.4/cm5.14.4-sles12.tar.gz
```

```
tar xvfz cm5.14.4-sles12.tar.gz -C /var/www/html/cloudera-repos/cm5 --strip-components=1
```

Ubuntu:

```
wget https://archive.cloudera.com/cm5/repo-as-tarball/5.14.4/cm5.14.4-ubuntul6-04.tar.gz
```

```
tar xvfz cm5.14.4-ubuntul6-04.tar.gz -C /var/www/html/cloudera-repos/cm5  
--strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cm5
```

If you want to create a repository for a different Cloudera Manager release or operating system, start in the [repo-as-tarball](#) parent directory, select the Cloudera Manager version you want to use, and then copy the `.tar.gz` link for your operating system.

CDH 5

To download the files for a CDH release, download the repository tarball for your operating system.

```
sudo mkdir -p /var/www/html/cloudera-repos/cdh5
```

Redhat/Centos:

```
wget https://archive.cloudera.com/cdh5/repo-as-tarball/5.14.4/cdh5.14.4-centos7.tar.gz
```

```
tar xvfz cdh5.14.4-centos7.tar.gz -C /var/www/html/cloudera-repos/cdh5  
--strip-components=1
```

Debian:

```
wget  
https://archive.cloudera.com/cdh5/repo-as-tarball/5.14.4/cdh5.14.4-debian-jessie.tar.gz
```

```
tar xvfz cdh5.14.4-debian-jessie.tar.gz -C /var/www/html/cloudera-repos/cdh5  
--strip-components=1
```

SLES:

```
wget https://archive.cloudera.com/cdh5/repo-as-tarball/5.14.4/cdh5.14.4-sles12.tar.gz
```

```
tar xvfz cdh5.14.4-sles12.tar.gz -C /var/www/html/cloudera-repos/cdh5 --strip-components=1
```


Ubuntu:

```
wget https://archive.cloudera.com/cdh5/repo-as-tarball/5.14.4/cdh5.14.4-ubuntu16-04.tar.gz
```

```
tar xvfz cdh5.14.4-ubuntu16-04.tar.gz -C /var/www/html/cloudera-repos/cdh5
--strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cdh5
```

If you want to create a repository for a different CDH release or operating system, start in the [repo-as-tarball](#) parent directory, select the CDH version you want to use, and then copy the `.tar.gz` link for your operating system.

Apache Accumulo for CDH

To download the files for an Accumulo release for CDH, run the following commands on the Web server host. Replace `<operating_system>` with the OS you are using (redhat, sles, debian, or ubuntu):

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories
https://archive.cloudera.com/accumulo-c5/<operating_system>/ -P
/var/www/html/cloudera-repos
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/accumulo-c5
```

Cloudera Navigator Key Trustee Server

Go to the Key Trustee Server [download page](#). Select **Packages** from the **CHOOSE DOWNLOAD TYPE** drop-down menu, select your operating system from the **CHOOSE AN OS** drop-down menu, and then click **DOWNLOAD NOW**. This downloads the Key Trustee Server package files in a `.tar.gz` file. Copy the file to your Web server, and extract the files with the `tar xvfz filename.tar.gz` command. This example uses Key Trustee Server 5.14.0:

```
sudo mkdir -p /var/www/html/cloudera-repos/keytrustee-server
sudo tar xvfz /path/to/keytrustee-server-5.14.0-parcels.tar.gz -C
/var/www/html/cloudera-repos/keytrustee-server --strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/keytrustee-server
```

Cloudera Navigator Key Trustee KMS and HSM KMS

Note: Cloudera Navigator HSM KMS is included in the Key Trustee KMS packages.

Go to the Key Trustee KMS [download page](#). Select **Package** from the **CHOOSE DOWNLOAD TYPE** drop-down menu, select your operating system from the **OPERATING SYSTEM** drop-down menu, and then click **DOWNLOAD NOW**. This downloads the Key Trustee KMS package files in a `.tar.gz` file. Copy the file to your Web server, and extract the files with the `tar xvfz filename.tar.gz` command. This example uses Key Trustee KMS 5.14.0:

```
sudo mkdir -p /var/www/html/cloudera-repos/keytrustee-kms
sudo tar xvfz /path/to/keytrustee-kms-5.14.0-parcels.tar.gz -C
/var/www/html/cloudera-repos/keytrustee-kms --strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/keytrustee-kms
```

2. Visit the Repository URL `http://<web_server>/cloudera-repos/` in your browser and verify the files you downloaded are present. If you do not see anything, your Web server may have been configured to not show indexes.

Creating a Temporary Internal Repository

You can quickly create a temporary remote repository to deploy packages on a one-time basis. Cloudera recommends using the same host that runs Cloudera Manager, or a gateway host. This example uses [Python SimpleHTTPServer](#) as the Web server to host the `/var/www/html` directory, but you can use a different directory.

- 1. Download the repository you need following the instructions in [Downloading and Publishing the Package Repository](#) on page 55.
- 2. Determine a port that your system is not listening on. This example uses port 8900.
- 3. Start a Python SimpleHTTPServer in the `/var/www/html` directory:

```
cd /var/www/html
python -m SimpleHTTPServer 8900
```

```
Serving HTTP on 0.0.0.0 port 8900 ...
```

- 4. Visit the Repository URL `http://<web_server>:8900/cloudera-repos/` in your browser and verify the files you downloaded are present.

Configuring Hosts to Use the Internal Repository

After establishing the repository, modify the client configuration to use it:

OS	Procedure
RHEL compatible	<div>Create <code>/etc/yum.repos.d/cloudera-repo.repo</code> files on cluster hosts with the following content, where <code><web_server></code> is the hostname of the Web server:<pre>[cloudera-repo] name=cloudera-repo baseurl=http://<web_server>/cm/5 enabled=1 gpgcheck=0</pre></div>
SLES	<div>Use the <code>zypper</code> utility to update client system repository information by issuing the following command:<pre>zypper addrepo http://<web_server>/cm <alias></pre></div>
Ubuntu	<div><div>Create <code>/etc/apt/sources.list.d/cloudera-repo.list</code> files on all cluster hosts with the following content, where <code><web_server></code> is the hostname of the Web server:<pre>deb http://<web_server>/cm <codename> <components></pre></div><div>You can find the <code><codename></code> and <code><components></code> variables in the <code>./conf/distributions</code> file in the repository.</div><div>After creating the <code>.list</code> file, run the following command:<pre>sudo apt-get update</pre></div></div>

Manually Install Cloudera Software Packages

This topic shows how to manually install Cloudera software packages, namely Cloudera Manager and CDH. This is useful for environments where it is not possible to use Cloudera Manager to install the required packages, such as organizations where password-less `sudo` is not permitted.

Although you can install the software packages manually, Cloudera does not support clusters that are not deployed and managed by Cloudera Manager.



Note: If you choose to install CDH manually using these instructions, you cannot use Cloudera Manager to install additional parcels. This can prevent you from using services that are only available via parcel.

Install Cloudera Manager Packages

1. On the Cloudera Manager Server host, type the following commands to install the Cloudera Manager packages.

OS	Command
RHEL, CentOS, Oracle Linux	<pre>sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre>
SLES	<pre>sudo zypper install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre>
Ubuntu	<pre>sudo apt-get install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre>

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

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:	<pre>\$ sudo yum install cloudera-manager-agent cloudera-manager-daemons</pre>
RHEL, if you're manually transferring RPMs:	<pre>\$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons</pre>
SLES	<pre>\$ sudo zypper install cloudera-manager-agent cloudera-manager-daemons</pre>
Ubuntu or Debian	<pre>\$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons</pre>

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

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

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

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04 and higher

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04

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

You should see the following:

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

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 Problems](#) on page 168. When the Agent hosts reboot, `cloudera-scm-agent` starts automatically.

Creating Virtual Images of Cluster Hosts

You can create virtual machine images, such as 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 [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 6 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:

```
shasum 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-6.0.0-1.cdh6.0.0.p0.309038 to /opt/cloudera/parcels/**CDH**, use the following command:

```
ln -s /opt/cloudera/parcels/CDH-6.0.0-1.cdh6.0.0.p0.309038 /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. If you have not already done so, [Step 1: Configure a Repository for Cloudera Manager](#) on page 96.

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
```

Configuring a Custom Java Home Location



Note: Cloudera strongly recommends installing the JDK at `/usr/java/jdk-version`, which allows Cloudera Manager to auto-detect and use the correct JDK version. If you install the JDK anywhere else, you must follow these instructions to configure Cloudera Manager with your chosen location. The following procedure changes the JDK location for Cloudera Management Services and CDH cluster processes only. It does not affect the JDK used by other non-Cloudera processes, or [gateway roles](#).

Although not recommended, the Oracle Java Development Kit (JDK), which Cloudera services require, may be installed at a custom location if necessary. These steps assume you have already installed the JDK as documented in [Step 2: Install Java Development Kit](#) on page 97.

To modify the Cloudera Manager configuration to ensure the JDK can be found:

1. Open the Cloudera Manager Admin Console.
2. In the main navigation bar, click the **Hosts** tab. If you are configuring the JDK location on a specific host only, click the link for that host.
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.

Creating a CDH Cluster Using a Cloudera Manager Template



Note: This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

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 64.

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 64

3. Create any local repositories required for the cluster.

See [Step 1: Configure a Repository for Cloudera Manager](#) on page 96.

4. Complete the `instantiator` section of the cluster configuration JSON document to create a template.

See [Creating the Template](#) on page 65.

5. Import the cluster template to the new cluster.

See [Importing the Template to a New Cluster](#) on page 68.

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.

To export the configuration:

1. Any [Host Templates](#) you have created are used to export the configuration. If you do not want to use those templates in the new cluster, delete them. In Cloudera Manager, go to **Hosts > Host Templates** and click **Delete** next to the Host Template you want to delete.
2. Delete any Host Templates created by the Cloudera Manager Installation Wizard. They typically have a name like `Template - 1`).
3. Run the following command to download the JSON configuration file to a convenient location for editing:

```
curl -u admin_username:admin_user_password
"http://Cloudera Manager URL/api/v12/clusters/Cluster name/export" >
path_to_file/file_name.json
```

For example:

```
curl -u adminuser:adminpass
"http://myCluster-1.myDomain.com:7180/api/v12/clusters/Cluster1/export" >
myCluster1-template.json
```



Note: Add the `?exportAutoConfig=true` parameter to the command above to include configurations made by [Autoconfiguration](#). These configurations are included for reference only and are not used when you import the template into a new cluster. For example:

```
curl -u admin_username:admin_user_password
"http://Cloudera Manager URL/api/v12/clusters/Cluster name/export"
>
path_to_file/file_name.json?exportAutoConfig=true
```

Preparing a New Cluster

The new cluster into which you import the cluster template must meet the following requirements:

- Database for Cloudera Manager is installed and configured.
- Cloudera Manager is installed and running.
- All required databases for CDH services are installed. See [Step 4: Install and Configure Databases](#) on page 101.
- 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 [Overview of Cloudera Manager Software Management](#).

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)
```

Service Dependencies in Cloudera Manager

The following tables list service dependencies that exist between various services in a Cloudera Manager deployment. As you configure services for Cloudera Manager, refer to the tables below for the appropriate version.

Service dependencies for Spark 2 on YARN and Cloudera Data Science Workbench are listed separately.

Version 6.2 Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		

Service	Dependencies	Optional Dependencies
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase • Kafka
HBase	<ul style="list-style-type: none"> • ZooKeeper • HDFS or Isilon 	
HDFS		<ul style="list-style-type: none"> • ADLS Connector or AWS S3 • KMS, Thales KMS, Key Trustee, or Luna KMS • ZooKeeper
Hive	YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper • Kudu
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • Kudu • YARN • ZooKeeper • Sentry • HBase
Kafka	ZooKeeper	Sentry
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	Sentry
Oozie	YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	Sentry
Spark on YARN	YARN	HBase
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 6.1 Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase • Kafka
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • ADLS Connector or AWS S3 • KMS, Thales KMS, Key Trustee, or Luna KMS • ZooKeeper
Hive	YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • Kudu • YARN • ZooKeeper • Sentry • HBase
Kafka	ZooKeeper	Sentry
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	Sentry
Kudu		
Oozie	YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	Sentry
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 6.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Accumulo C6	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
ADLS Connector		
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase Kafka
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> ADLS Connector or AWS S3 KMS, Thales KMS, Key Trustee, or Luna KMS ZooKeeper
Hive	YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Kafka	ZooKeeper	Sentry
Key Trustee		ZooKeeper
KMS		
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	Sentry
Kudu		
Luna KMS		ZooKeeper
Oozie	YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN

Service	Dependencies	Optional Dependencies
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Spark on YARN	YARN	HBase
Sqoop Client		
Thales KMS		ZooKeeper
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.16 Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase Kafka
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> ADLS Connector or AWS S3 KMS, Thales KMS, Key Trustee, or Luna KMS ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase 	Sentry

Service	Dependencies	Optional Dependencies
	<ul style="list-style-type: none"> Solr 	
Kudu		
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.15 Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase Kafka
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> ADLS Connector or AWS S3 KMS, Thales KMS, Key Trustee, or Luna KMS ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	Sentry
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.14 Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • ADLS Connector or AWS S3 • KMS, Thales KMS, Key Trustee, or Luna KMS • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Sentry • Impala • ZooKeeper • HBase

Service	Dependencies	Optional Dependencies
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	Sentry
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.13 Service Dependencies

Service	Dependencies	Optional Dependencies
AWS S3		
Data Context Connector		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> AWS S3 KMS, Thales KMS, Key Trustee, or Luna KMS ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.12 Service Dependencies

Service	Dependencies	Optional Dependencies
AWS S3		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> AWS S3 KMS, Thales KMS, Key Trustee, or Luna KMS ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
Thales KMS		ZooKeeper
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.11.2 Service Dependencies

Service	Dependencies	Optional Dependencies
AWS S3		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> AWS S3 KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.11.0 Service Dependencies

Service	Dependencies	Optional Dependencies
AWS S3		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> AWS S3 KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase

Service	Dependencies	Optional Dependencies
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
Kudu		
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Spark on YARN	YARN	HBase
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.10.3 Service Dependencies

Service	Dependencies	Optional Dependencies
AWS S3		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> AWS S3 KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.10 Service Dependencies

Service	Dependencies	Optional Dependencies
AWS S3		
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> AWS S3 KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase

Service	Dependencies	Optional Dependencies
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> Kudu YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.9 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN ZooKeeper Sentry

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.8 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • KMS or Key Trustee • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • YARN • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	

Service	Dependencies	Optional Dependencies
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	Sentry
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.7.1 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN

Service	Dependencies	Optional Dependencies
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.7.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	

Service	Dependencies	Optional Dependencies
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.6.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • KMS or Key Trustee • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive Hive	<ul style="list-style-type: none"> • YARN • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.5.2 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • KMS or Key Trustee • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • YARN • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.5.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase

Service	Dependencies	Optional Dependencies
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Spark on YARN	YARN	HBase
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.4.4 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	

Service	Dependencies	Optional Dependencies
HDFS		<ul style="list-style-type: none"> • KMS or Key Trustee • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • YARN • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper • Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.4.1 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • KMS or Key Trustee • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • Spark on YARN • HBase

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper Spark on YARN
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.4.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> Spark on YARN HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr

Service	Dependencies	Optional Dependencies
		<ul style="list-style-type: none"> • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • YARN • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
Spark on YARN	YARN	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.3.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • KMS or Key Trustee • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Sentry • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon 	<ul style="list-style-type: none"> • YARN

Service	Dependencies	Optional Dependencies
	<ul style="list-style-type: none"> Hive 	<ul style="list-style-type: none"> ZooKeeper Sentry HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> Hive ZooKeeper
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.2.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> KMS or Key Trustee ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> HBase Sentry ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Sentry Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN ZooKeeper Sentry HBase
Key Trustee		ZooKeeper
KMS		

Service	Dependencies	Optional Dependencies
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper
Sentry	HDFS or Isilon	ZooKeeper
Solr	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.1.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Flume		<ul style="list-style-type: none"> • Solr • HDFS or Isilon • HBase
HBase	<ul style="list-style-type: none"> • HDFS or Isilon • ZooKeeper 	
HDFS		<ul style="list-style-type: none"> • ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> • HBase • Sentry • ZooKeeper
Hue	<ul style="list-style-type: none"> • Oozie • Hive 	<ul style="list-style-type: none"> • Sqoop • Solr • Impala • ZooKeeper • HBase
Impala	<ul style="list-style-type: none"> • HDFS or Isilon • Hive 	<ul style="list-style-type: none"> • YARN • ZooKeeper • Sentry • HBase
Key-Value Store Indexer	<ul style="list-style-type: none"> • HBase • Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	<ul style="list-style-type: none"> • Hive • ZooKeeper
Sentry	HDFS or Isilon	ZooKeeper

Service	Dependencies	Optional Dependencies
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Sqoop	MapReduce or YARN	
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Version 5.0.0 Service Dependencies

Service	Dependencies	Optional Dependencies
Accumulo 16	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Flume		<ul style="list-style-type: none"> Solr HDFS or Isilon HBase
HBase	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
HDFS		<ul style="list-style-type: none"> ZooKeeper
Hive	MapReduce or YARN	<ul style="list-style-type: none"> HBase ZooKeeper
Hue	<ul style="list-style-type: none"> Oozie Hive 	<ul style="list-style-type: none"> Sqoop Solr Impala ZooKeeper HBase
Impala	<ul style="list-style-type: none"> HDFS or Isilon Hive 	<ul style="list-style-type: none"> YARN HBase
Isilon		
Kafka	ZooKeeper	
Key Trustee Server		
Key-Value Store Indexer	<ul style="list-style-type: none"> HBase Solr 	
MapReduce	HDFS or Isilon	ZooKeeper
Oozie	MapReduce or YARN	ZooKeeper
Solr	<ul style="list-style-type: none"> HDFS or Isilon ZooKeeper 	
Spark		HDFS or Isilon
Spark on YARN	YARN	
Sqoop	MapReduce or YARN	
Sqoop Client		

Service	Dependencies	Optional Dependencies
YARN	HDFS or Isilon	ZooKeeper
ZooKeeper		

Cloudera Data Science Workbench 1.5.0 Service Dependencies

Service	Dependencies	Optional Dependencies
CDSW/CDH6	<ul style="list-style-type: none"> • YARN • Spark 2 on YARN 	<ul style="list-style-type: none"> • HDFS • HBase • Hive • Solr • Sentry
CDSW/CDH5		

Spark 2 on YARN Service Dependencies

Spark 2 Version	Dependencies	Optional Dependencies
2.3.4	YARN	<ul style="list-style-type: none"> • HBase • Hive
2.2.4	YARN	Hive
2.1.4	YARN	Hive
2.0.2	YARN	Hive

Installing Cloudera Manager, CDH, and Managed Services

This procedure is recommended for installing Cloudera Manager and CDH for production environments. For a non-production "easy install," see [Installing a Proof-of-Concept Cluster](#).

Before you begin the installation, make sure you have reviewed the requirements and other considerations described in [Before You Install](#) on page 10.

The general steps in the installation procedure are as follows:

Step 1: Configure a Repository for Cloudera Manager

Cloudera Manager is installed using package management tools such as `yum` for RHEL compatible systems, `zypper` for SLES, and `apt-get` for Ubuntu. These tools depend on access to repositories to install software. Cloudera maintains Internet-accessible repositories for CDH and Cloudera Manager installation files. You can also create your own internal repository for hosts that do not have Internet access.

For more information on creating an internal repository for Cloudera Manager, [Configuring a Local Package Repository](#) on page 54.

To use the Cloudera repository:

RHEL compatible

1. Download the `cloudera-manager.repo` file for your OS version to the `/etc/yum.repos.d/` directory on the Cloudera Manager Server host.

You can find the URL in the **Repo File** column in the [Cloudera Manager 6 Version and Download Information](#) table for the Cloudera Manager version you want to install.

For example:

```
sudo wget <repo_file_url> -P /etc/yum.repos.d/
```

2. Import the repository signing GPG key:

- RHEL 7 compatible:

```
sudo rpm --import https://archive.cloudera.com/cm6/6.2.0/redhat7/yum/RPM-GPG-KEY-cloudera
```

- RHEL 6 compatible:

```
sudo rpm --import https://archive.cloudera.com/cm6/6.2.0/redhat6/yum/RPM-GPG-KEY-cloudera
```

3. Continue to [Step 2: Install Java Development Kit](#) on page 97.

SLES

1. Update your system package index by running:

```
sudo zypper refresh
```

2. Add the repo using `zypper addrepo`.

You can find the URL in the **Repo File** column in the [Cloudera Manager 6 Version and Download Information](#) table for the Cloudera Manager version you want to install.

For example:

```
sudo zypper addrepo -f
https://archive.cloudera.com/cm6/6.2.0/sles12/yum/cloudera-manager.repo
```

3. Import the repository signing GPG key:

```
sudo rpm --import https://archive.cloudera.com/cm6/6.2.0/sles12/yum/RPM-GPG-KEY-cloudera
```

4. Continue to [Step 2: Install Java Development Kit](#) on page 97.

Ubuntu

1. Download the `cloudera.list` file for your OS version to the `/etc/apt/sources.list.d/` directory on the Cloudera Manager Server host.

You can find the URL in the **Repo File** column in the [Cloudera Manager 6 Version and Download Information](#) table for the Cloudera Manager version you want to install.

2. Import the repository signing GPG key:

```
wget https://archive.cloudera.com/cm6/6.2.0/ubuntu1604/apt/archive.key
sudo apt-key add archive.key
```

3. Update your system package index by running:

```
sudo apt-get update
```

4. Continue to [Step 2: Install Java Development Kit](#) on page 97.

Step 2: Install Java Development Kit

For the JDK, you can either install the Oracle JDK version provided by Cloudera using Cloudera Manager, a different Oracle JDK directly from Oracle, or OpenJDK. Most Linux distributions supported by Cloudera include OpenJDK, but manual installation instructions are provided below if needed.

OpenJDK is supported with Cloudera Enterprise 6.1.0 and higher, and Cloudera Enterprise 5.16.0 and higher.

Requirements

- The JDK must be 64-bit. Do not use a 32-bit JDK.
- The installed JDK must be a supported version as documented in [Java Requirements](#).
- The *same version* of the JDK must be installed on each cluster host.
- The JDK must be installed at `/usr/java/jdk-version`.



Note: If you have installed the JDK in a different location, set the `JAVA_HOME` environment variable before installing Cloudera Manager. If you cannot set `JAVA_HOME` in your environment, create an empty file with the path `/etc/cloudera-pre-install/CLOUDERA_SKIP_JAVA_INSTALL_CHECK` on the Cloudera Manager server host. This will cause the installation process to skip any Java checks during installation of the Cloudera Manager Server and Daemon packages.



Important:

- The RHEL-compatible and Ubuntu operating systems supported by Cloudera Enterprise 6 all use AES-256 encryption by default for tickets. To support AES-256 bit encryption in JDK versions lower than 1.8u161, you must install the [Java Cryptography Extension \(JCE\) Unlimited Strength Jurisdiction Policy File](#) on all cluster and Hadoop user machines. Cloudera Manager can automatically install the policy files, or you can install them manually. For JCE Policy File installation instructions, see the `README.txt` file included in the `jce_policy-x.zip` file. JDK 1.8u161 and higher enable unlimited strength encryption by default, and do not require policy files.
- On SLES 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 Oracle JDK Using Cloudera Manager



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.

After completing [Step 1: Configure a Repository for Cloudera Manager](#) on page 96, you can install the Oracle JDK on the Cloudera Manager Server host using your package manager as follows:

• RHEL Compatible

```
sudo yum install oracle-j2sdk1.8
```

• SLES

```
sudo zypper install oracle-j2sdk1.8
```

• Ubuntu

```
sudo apt-get install oracle-j2sdk1.8
```

You can use Cloudera Manager to install the JDK on the remaining cluster hosts in an upcoming step. Continue to [Step 3: Install Cloudera Manager Server](#) on page 99.

Manually Installing Oracle JDK

The Oracle JDK installer is available both as an RPM-based installer for RPM-based systems, and as a `.tar.gz` file. These instructions are for the `.tar.gz` file.

1. Download the `.tar.gz` file for one of the 64-bit [supported versions](#) of the Oracle JDK from [Java SE 8 Downloads](#). (This link is correct at the time of writing, but can change.)



Note: If you want to download the JDK directly using a utility such as `wget`, you must accept the Oracle license by configuring headers, which are updated frequently. Blog posts and Q&A sites can be a good source of information on how to download a particular JDK version using `wget`.

2. Extract the JDK to `/usr/java/jdk-version`. For example:

```
tar xvfz /path/to/jdk-8u<update_version>-linux-x64.tar.gz -C /usr/java/
```

3. Repeat this procedure on all cluster hosts. After you have finished, continue to [Step 3: Install Cloudera Manager Server](#) on page 99.

Manually Installing OpenJDK

Before installing Cloudera Manager and CDH, perform the steps in this section to install [OpenJDK](#) on all hosts in your cluster(s).

When you install Cloudera Enterprise, Cloudera Manager includes an option to install Oracle JDK. De-select this option.

See [Supported JDKs](#) for information on which JDK versions are supported for Cloudera Enterprise releases.



Note: If you intend to enable [Auto-TLS](#), note the following:

You can specify a PEM file containing trusted CA certificates to be imported into the Auto-TLS truststore. If you want to use the certificates in the cacerts truststore that comes with OpenJDK, you must convert the truststore to PEM format first. However, OpenJDK ships with some intermediate certificates that cannot be imported into the Auto-TLS truststore. You must remove these certificates from the PEM file before importing the PEM file into the Auto-TLS truststore. This is not required when upgrading to OpenJDK from a cluster where Auto-TLS has already been enabled.

Log in to each host and run the following command:

RHEL

```
su -c yum install java-1.8.0-openjdk-devel
```

Ubuntu

```
sudo apt-get install openjdk-8-jdk
```

SLES

```
sudo zypper install java-1_8_0-openjdk-devel
```

Step 3: Install Cloudera Manager Server

In this step you install the Cloudera Manager packages on the Cloudera Manager Server host, and optionally enable auto-TLS.

Install Cloudera Manager Packages

1. On the Cloudera Manager Server host, type the following commands to install the Cloudera Manager packages.

OS	Command
RHEL, CentOS, Oracle Linux	<pre>sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre>
SLES	<pre>sudo zypper install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre>
Ubuntu	<pre>sudo apt-get install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre>

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

(Recommended) Enable Auto-TLS



Note: Auto-TLS supports two options:

- **Option 1:** Use Cloudera Manager to generate an internal Certificate Authority and corresponding certificates
- **Option 2:** Use an existing Certificate Authority and corresponding certificates

The following procedure demonstrates **Option 1**, enabling auto-TLS to use an internal certificate authority (CA) created and managed by Cloudera Manager. To use a trusted public CA (**Option 2**), you must first obtain the certificates for your cluster hosts. For more information, see [Configuring TLS Encryption for Cloudera Manager and CDH Using Auto-TLS](#).

Starting in Cloudera Manager 6.2, you can enable auto-TLS on existing clusters. If you do not want to enable auto-TLS right now, skip this section and continue to [Step 4: Install and Configure Databases](#) on page 101.

Auto-TLS greatly simplifies the process of enabling and managing TLS encryption on your cluster. It automates the creation of an internal *certificate authority* (CA) and deployment of certificates across all cluster hosts. It can also automate the distribution of existing certificates, such as those signed by a public CA. Adding new cluster hosts or services to a cluster with auto-TLS enabled automatically creates and deploys the required certificates.

To enable auto-TLS with an embedded Cloudera Manager CA, run the following command:

```
sudo JAVA_HOME=/usr/java/jdk1.8.0_181-cloudera /opt/cloudera/cm-agent/bin/certmanager
setup --configure-services
```



Note: The `certmanager` utility is included with Cloudera Manager Agent, but not Cloudera Manager Server. If you see an error about the `certmanager` command not being found, make sure you have installed the `cloudera-manager-agent` package as documented above.

Replace `jdk1.8.0_181-cloudera` with your JDK version. If you want to store the files in a directory other than the default (`/var/lib/cloudera-scm-server/certmanager`), add the `--location` option as follows:

```
sudo JAVA_HOME=/usr/java/jdk1.8.0_181-cloudera /opt/cloudera/cm-agent/bin/certmanager
--location /opt/cloudera/CMCA setup --configure-services
```

Check the `/var/log/cloudera-scm-agent/certmanager.log` log file to confirm that the `/var/lib/cloudera-scm-server/certmanager/*` directories were created.

That's it! When you start Cloudera Manager Server, it will have TLS enabled, and all hosts that you add to the cluster, as well as any [supported services](#), will automatically have TLS configured and enabled.

For more information about auto-TLS, see [Configuring TLS Encryption for Cloudera Manager and CDH Using Auto-TLS](#).

Install and Configure Databases

After installing the Cloudera Manager Server packages, continue to [Step 4: Install and Configure Databases](#) on page 101.

Step 4: Install and Configure Databases

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.

Although you can deploy different types of databases in a single environment, doing so can create unexpected complications. Cloudera recommends choosing one supported database provider for all of the Cloudera databases.

Cloudera recommends installing the databases on different hosts than the services. Separating databases from services can help isolate the potential impact from failure or resource contention in one or the other. It can also simplify management in organizations that have dedicated database administrators.

You can use your own PostgreSQL, MariaDB, MySQL, or Oracle database for the Cloudera Manager Server and other services that use databases. For information about planning, managing, and backing up Cloudera Manager data stores, see [Storage Space Planning for Cloudera Manager](#) on page 10 and [Backing Up Databases](#).

Required Databases

The following components all require databases: 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. The type of data contained in the databases and their relative sizes are as follows:

- Cloudera Manager Server - 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 cannot 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. For more information, see [Backing Up Databases](#).

- 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.

The Host Monitor and Service Monitor services use local disk-based datastores. For more information, see [Data Storage for Monitoring Data](#).

The JDBC connector for your database *must* be installed on the hosts where you assign the Activity Monitor and Reports Manager roles.

Installing and Configuring Databases

For instructions on installing and configuring databases for Cloudera Manager, CDH, and other managed services, see the instructions for the type of database you want to use:

Install and Configure MariaDB for Cloudera Software


To use a MariaDB database, follow these procedures. For information on compatible versions of MariaDB, see [Database Requirements](#).

Installing 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 102 to verify that your MariaDB configurations meet the requirements for Cloudera Manager.
- It is important that the `datadir` directory (`/var/lib/mysql` by default), is on a partition that has sufficient free space. For more information, see [Storage Space Planning for Cloudera Manager](#) on page 10.

1. Install MariaDB server:

OS	Command
RHEL compatible	<pre>sudo yum install mariadb-server</pre>
SLES	<pre>sudo zypper install mariadb-server</pre> <div> Note: Some SLES systems encounter errors when using the <code>zypper install</code> command. For more information on resolving this issue, see the Novell Knowledgebase topic, error running chkconfig.</div>
Ubuntu	<pre>sudo apt-get install mariadb-server</pre>

If these commands do not work, you might need to add a repository or use a different `yum install` command, particularly on RHEL 6 compatible operating systems. For more assistance, see the following topics on the MariaDB website:

- **RHEL compatible:** [Installing MariaDB with yum](#)
- **SLES:** [MariaDB Package Repository Setup and Usage](#)
- **Ubuntu:** [Installing MariaDB .deb Files](#)

Configuring and Starting the MariaDB Server



Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

1. Stop the MariaDB server if it is running:

- **RHEL 7 compatible:**

```
sudo systemctl stop mariadb
```

- **RHEL 6 compatible, Ubuntu, SLES:**

```
sudo service mariadb stop
```

2. If they exist, 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` (`/etc/my.cnf` by default).
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:
 - 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.
 - 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 do not need to 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]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security risks;
# to do so, uncomment this line:
symbolic-links = 0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

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

#In later versions of MariaDB, if you enable the binary log and do not set
#a server_id, MariaDB will not start. The server_id must be unique within
#the replicating group.
server_id=1

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

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

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 7 compatible	<code>sudo systemctl enable mariadb</code>
RHEL 6 compatible	<code>sudo chkconfig mariadb on</code>
SLES	<code>sudo chkconfig --add mariadb</code>
Ubuntu	<code>sudo chkconfig mariadb on</code> <div>  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. </div>

7. Start the MariaDB server:

- **RHEL 7 compatible:**

```
sudo systemctl start mariadb
```


- **RHEL 6 compatible, Ubuntu, SLES:**

```
sudo service mariadb start
```

8. Run `/usr/bin/mysql_secure_installation` to set the MariaDB root password and other security-related settings. In a new installation, the root password is blank. Press the **Enter** key when you're prompted for the root password. For the rest of the prompts, enter the responses listed below in **bold**:

```
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! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

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 any other hosts running services that require database access. For more information on Cloudera software that uses databases, see [Required Databases](#) on page 101.

Cloudera recommends that you consolidate all roles that require databases on a limited number of hosts, and install the driver on those hosts. Locating all such roles on the same hosts is recommended but not required. Make sure to 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	<div>  <p>Important: Using the <code>yum install</code> command to install the MySQL driver package before installing a JDK installs OpenJDK, and then uses the <code>Linux alternatives</code> command to set the system JDK to be OpenJDK. If you intend to use an Oracle JDK, make sure that it is installed before installing the MySQL driver using <code>yum install</code>.</p> <p>Alternatively, use the following procedure to manually install the driver.</p> </div>

OS	Command
	<p>1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html (in .tar.gz format). As of the time of writing, you can download version 5.1.46 using <code>wget</code> as follows:</p> <pre>wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.46.tar.gz</pre> <p>2. Extract the JDBC driver JAR file from the downloaded file. For example:</p> <pre>tar zxvf mysql-connector-java-5.1.46.tar.gz</pre> <p>3. Copy the JDBC driver, renamed, to <code>/usr/share/java/</code>. If the target directory does not yet exist, create it. For example:</p> <pre>sudo mkdir -p /usr/share/java/ cd mysql-connector-java-5.1.46 sudo cp mysql-connector-java-5.1.46-bin.jar /usr/share/java/ mysql-connector-java.jar</pre>
SLES	<pre>sudo zypper install mysql-connector-java</pre>
Ubuntu or Debian	<pre>sudo apt-get install libmysql-java</pre>

Creating Databases for Cloudera Software

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service in a CDH 5 cluster)
 - Reports Manager
- Hue
- Each Hive metastore
- Sentry Server
- Cloudera Navigator Audit Server
- Cloudera Navigator Metadata Server
- Oozie

1. Log in as the `root` user, or another user with privileges to create database and grant privileges:

```
mysql -u root -p
```

Enter password:

2. Create databases for each service deployed in the cluster using the following commands. You can use any value you want for the `<database>`, `<user>`, and `<password>` parameters. The **Databases for Cloudera Software** table, below lists the default names provided in the Cloudera Manager configuration settings, but you are not required to use them.

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.

```
CREATE DATABASE <database> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
GRANT ALL ON <database>.* TO '<user>'@'%' IDENTIFIED BY '<password>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

Table 16: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Activity Monitor	amon	amon
Reports Manager	rman	rman
Hue	hue	hue
Hive Metastore Server	metastore	hive
Sentry Server	sentry	sentry
Cloudera Navigator Audit Server	nav	nav
Cloudera Navigator Metadata Server	navms	navms
Oozie	oozie	oozie

3. Confirm that you have created all of the databases:

```
SHOW DATABASES;
```

You can also confirm the privilege grants for a given user by running:

```
SHOW GRANTS FOR '<user>'@'%' ;
```

Setting Up the Cloudera Manager Database

After completing the above instructions to install and configure MariaDB databases for Cloudera software, continue to [Step 5: Set up the Cloudera Manager Database](#) on page 130 to configure a database for Cloudera Manager.

Install and Configure MySQL for Cloudera Software

To use a MySQL database, follow these procedures. For information on compatible versions of the MySQL database, see [Database Requirements](#).


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 108 to verify that your MySQL configurations meet the requirements for Cloudera Manager.
- For MySQL 5.6 and 5.7, you must install the *MySQL-shared-compat* or *MySQL-shared* package. This is required for the Cloudera Manager Agent package installation.
- 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	<p>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.</p> <pre>wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm</pre> <pre>sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm</pre> <pre>sudo yum update</pre> <pre>sudo yum install mysql-server</pre> <pre>sudo systemctl start mysqld</pre>
SLES	<pre>sudo zypper install mysql libmysqlclient_r17</pre> <div>Note: Some SLES systems encounter errors when using the preceding <code>zypper install</code> command. For more information on resolving this issue, see the Novell Knowledgebase topic, error running chkconfig.</div>
Ubuntu	<pre>sudo apt-get install mysql-server</pre>

Configuring and Starting the MySQL Server



Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

1. Stop the MySQL server if it is running.

OS	Command
RHEL 7 Compatible	<code>sudo systemctl stop mysqld</code>
RHEL 6 Compatible	<code>sudo service mysqld stop</code>
SLES, Ubuntu	<code>sudo service mysql stop</code>

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` (`/etc/my.cnf` by default).
4. 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:
 - 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.
 - 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 do not need to 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]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
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

#In later versions of MySQL, if you enable the binary log and do not set
#a server_id, MySQL will not start. The server_id must be unique within
#the replicating group.
server_id=1

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

```

5. 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.
6. Ensure the MySQL server starts at boot:

OS	Command
RHEL 7 compatible	<code>sudo systemctl enable mysqld</code>
RHEL 6 compatible	<code>sudo chkconfig mysqld on</code>
SLES	<code>sudo chkconfig --add mysql</code>
Ubuntu	<code>sudo chkconfig mysql on</code> <div>  Note: chkconfig may not be available on recent Ubuntu releases. You may need to use Upstart to configure MySQL to start automatically when the system boots. For more information, see the Ubuntu documentation or the Upstart Cookbook. </div>

7. Start the MySQL server:

OS	Command
RHEL 7 Compatible	<code>sudo systemctl start mysqld</code>
RHEL 6 Compatible	<code>sudo service mysqld start</code>
SLES, Ubuntu	<code>sudo service mysql start</code>

8. Run `/usr/bin/mysql_secure_installation` to set the MySQL root password and other security-related settings. In a new installation, the root password is blank. Press the **Enter** key when you're prompted for the root password. For the rest of the prompts, enter the responses listed below in **bold**:

```
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 any other hosts running services that require database access. For more information on Cloudera software that uses databases, see [Required Databases](#) on page 101.




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 5.1 driver version 5.1.26 or higher.

Cloudera recommends that you consolidate all roles that require databases on a limited number of hosts, and install the driver on those hosts. Locating all such roles on the same hosts is recommended but not required. Make sure to 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	<div>  Important: Using the <code>yum install</code> command to install the MySQL driver package before installing a JDK installs OpenJDK, and then uses the <code>Linux alternatives</code> command to set the system JDK to be OpenJDK. If you intend to use an Oracle JDK, make sure that it is installed before installing the MySQL driver using <code>yum install</code>. Alternatively, use the following procedure to manually install the driver. </div> <ol style="list-style-type: none"> Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html (in <code>.tar.gz</code> format). As of the time of writing, you can download version 5.1.46 using <code>wget</code> as follows: <div> <pre>wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.46.tar.gz</pre> </div> Extract the JDBC driver JAR file from the downloaded file. For example: <div> <pre>tar zxvf mysql-connector-java-5.1.46.tar.gz</pre> </div> Copy the JDBC driver, renamed, to <code>/usr/share/java/</code>. If the target directory does not yet exist, create it. For example: <div> <pre>sudo mkdir -p /usr/share/java/ cd mysql-connector-java-5.1.46 sudo cp mysql-connector-java-5.1.46-bin.jar /usr/share/java/ mysql-connector-java.jar</pre> </div>
SLES	<div> <pre>sudo zypper install mysql-connector-java</pre> </div>
Ubuntu or Debian	<div> <pre>sudo apt-get install libmysql-java</pre> </div>

Creating Databases for Cloudera Software

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service in a CDH 5 cluster)
 - Reports Manager
- Hue
- Each Hive metastore
- Sentry Server
- Cloudera Navigator Audit Server
- Cloudera Navigator Metadata Server
- Oozie

1. Log in as the `root` user, or another user with privileges to create database and grant privileges:

```
mysql -u root -p
```

```
Enter password:
```


2. Create databases for each service deployed in the cluster using the following commands. You can use any value you want for the `<database>`, `<user>`, and `<password>` parameters. The **Databases for Cloudera Software** table, below lists the default names provided in the Cloudera Manager configuration settings, but you are not required to use them.

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.

```
CREATE DATABASE <database> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
GRANT ALL ON <database>.* TO '<user>'@'%' IDENTIFIED BY '<password>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

Table 17: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Activity Monitor	amon	amon
Reports Manager	rman	rman
Hue	hue	hue
Hive Metastore Server	metastore	hive
Sentry Server	sentry	sentry
Cloudera Navigator Audit Server	nav	nav
Cloudera Navigator Metadata Server	navms	navms
Oozie	oozie	oozie

3. Confirm that you have created all of the databases:

```
SHOW DATABASES;
```

You can also confirm the privilege grants for a given user by running:

```
SHOW GRANTS FOR '<user>'@'%' ;
```

4. 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.

Setting Up the Cloudera Manager Database

After completing the above instructions to install and configure MySQL databases for Cloudera software, continue to [Step 5: Set up the Cloudera Manager Database](#) on page 130 to configure a database for Cloudera Manager.

Install and Configure PostgreSQL for Cloudera Software



Note: The following instructions are for a dedicated PostgreSQL database for use in production environments, and are unrelated to the embedded PostgreSQL database provided by Cloudera for [non-production](#) installations.

To use a PostgreSQL database, follow these procedures. For information on compatible versions of the PostgreSQL database, see [Database Requirements](#).

Installing 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 115 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 supports the use of a custom schema name for the Cloudera Manager Server database, but not the CDH component databases (such as Hive, Hue, Sentry, and so on). For more information, see <https://www.postgresql.org/docs/current/static/ddl-schemas.html>.

Install the PostgreSQL packages as follows:

RHEL:

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

SLES:

```
sudo zypper install --no-recommends postgresql96-server
```



Note: This command installs PostgreSQL 9.6. If you want to install a different version, you can use `zypper search postgresql` to search for an available supported version. See [Database Requirements](#).

Ubuntu:

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

Installing the `psycopg2` Python Package

Hue in CDH 6 requires version 2.5.4 or higher of the `psycopg2` Python package for connecting to a PostgreSQL database. The `psycopg2` package is automatically installed as a dependency of Cloudera Manager Agent, but the version installed is often lower than 2.5.4.

If you are installing or upgrading to CDH 6 and using PostgreSQL for the Hue database, you must install `psycopg2` 2.5.4 or higher on all Hue hosts as follows. These examples install version 2.7.5 (2.6.2 for RHEL 6):

RHEL 7 Compatible

1. Install the `python-pip` package:

```
sudo yum install python-pip
```

2. Install `psycopg2` 2.7.5 using `pip`:

```
sudo pip install psycopg2==2.7.5 --ignore-installed
```

RHEL 6 Compatible

1. Make sure that you have [installed Python 2.7](#). You can verify this by running the following commands:

```
source /opt/rh/python27/enable
python --version
```

2. Install the `python-pip` package:

```
sudo yum install python-pip
```

3. Install the `postgresql-devel` package:

```
sudo yum install postgresql-devel
```

4. Install the `gcc*` packages:

```
sudo yum install gcc*
```

5. Install `psycopg2 2.6.2` using `pip`:

```
sudo bash -c "source /opt/rh/python27/enable; pip install psycopg2==2.6.2
--ignore-installed"
```

Ubuntu / Debian

1. Install the `python-pip` package:

```
sudo apt-get install python-pip
```

2. Install `psycopg2 2.7.5` using `pip`:

```
sudo pip install psycopg2==2.7.5 --ignore-installed
```

SLES 12

Install the `python-psycopg2` package:

```
sudo zypper install python-psycopg2
```

Configuring and Starting the PostgreSQL Server

Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

By default, PostgreSQL only accepts connections on the loopback interface. You must reconfigure PostgreSQL to accept connections from the fully qualified domain names (FQDN) of the hosts hosting the services for which you are configuring databases. If you do not make these changes, the services cannot connect to and use the database on which they depend.

1. Make sure that `LC_ALL` is set to `en_US.UTF-8` and initialize the database as follows:

- **RHEL 7:**

```
echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
sudo su -l postgres -c "postgresql-setup initdb"
```

- **RHEL 6:**

```
echo 'LC_ALL="en_US.UTF-8"' >> /etc/default/locale
sudo service postgresql initdb
```

- **SLES 12:**

```
sudo su -l postgres -c "initdb --pgdata=/var/lib/pgsql/data --encoding=UTF-8"
```

- **Ubuntu:**

```
sudo service postgresql start
```

2. Enable MD5 authentication. Edit `pg_hba.conf`, which is usually found in `/var/lib/pgsql/data` or `/etc/postgresql/<version>/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:
 - `max_connection` - 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` - 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:

$$\text{max_wal_size} = (3 * \text{checkpoint_segments}) * 16\text{MB}$$

- `checkpoint_completion_target` - 0.9.

4. Configure the PostgreSQL server to start at boot.

OS	Command
RHEL 7 compatible	<code>sudo systemctl enable postgresql</code>
RHEL 6 compatible	<code>sudo chkconfig postgresql on</code>
SLES	<code>sudo chkconfig --add postgresql</code>
Ubuntu	<code>sudo chkconfig postgresql on</code> <div>  <p>Note: <code>chkconfig</code> may not be available on recent Ubuntu releases. You may need to use Upstart to configure PostgreSQL to start automatically when the system boots. For more information, see the Ubuntu documentation or the Upstart Cookbook.</p> </div>

5. Restart the PostgreSQL database:

- **RHEL 7 Compatible:**

```
sudo systemctl restart postgresql
```

- **All Others:**

```
sudo service postgresql restart
```

Creating Databases for Cloudera Software

Create databases and service accounts for components that require databases:

- Cloudera Manager Server

- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service in a CDH 5 cluster)
 - Reports Manager
- Hue
- Each Hive metastore
- Sentry Server
- Cloudera Navigator Audit Server
- Cloudera Navigator Metadata Server
- Oozie

The databases must be configured to support the PostgreSQL UTF8 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. Create databases for each service you are using from the below table:

```
CREATE ROLE <user> LOGIN PASSWORD '<password>';
```

```
CREATE DATABASE <database> OWNER <user> ENCODING 'UTF8';
```

You can use any value you want for *<database>*, *<user>*, and *<password>*. The following examples are the default names provided in the Cloudera Manager configuration settings, but you are not required to use them:

Table 18: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Activity Monitor	amon	amon
Reports Manager	rman	rman
Hue	hue	hue
Hive Metastore Server	metastore	hive
Sentry Server	sentry	sentry
Cloudera Navigator Audit Server	nav	nav
Cloudera Navigator Metadata Server	navms	navms
Oozie	oozie	oozie

Record the databases, usernames, and passwords chosen because you will need them later.

3. For PostgreSQL 8.4 and higher, set `standard_conforming_strings=off` for the Hive Metastore and Oozie databases:

```
ALTER DATABASE <database> SET standard_conforming_strings=off;
```

Setting Up the Cloudera Manager Database

After completing the above instructions to install and configure PostgreSQL databases for Cloudera software, continue to [Step 5: Set up the Cloudera Manager Database](#) on page 130 to configure a database for Cloudera Manager.

Install and Configure Oracle Database for Cloudera Software

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



Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

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.conf
# 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.

Installing Cloudera Manager, CDH, and Managed Services

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 any other hosts that use a database.

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 (or `ojdbc8.jar`, for Oracle Database 12.2).

2. Copy the Oracle JDBC JAR file to `/usr/share/java/oracle-connector-java.jar`. The Cloudera Manager databases and the Hive Metastore 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 Cloudera Software

Create schema and user accounts for components that require databases:

- Cloudera Manager Server

- Cloudera Management Service roles:
 - Activity Monitor (if using the MapReduce service in a CDH 5 cluster)
 - Reports Manager
- Hue
- Each Hive metastore
- Sentry Server
- Cloudera Navigator Audit Server
- Cloudera Navigator Metadata Server
- Oozie

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 user and schema for each service you are using from the below table:

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

You can use any value you want for *<schema>*, *<user>*, and *<password>*. The following examples are the default names provided in the Cloudera Manager configuration settings, but you are not required to use them:

Table 19: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Activity Monitor	amon	amon
Reports Manager	rman	rman
Hue	hue	hue
Hive Metastore Server	metastore	hive
Sentry Server	sentry	sentry
Cloudera Navigator Audit Server	nav	nav
Cloudera Navigator Metadata Server	navms	navms
Oozie	oozie	oozie

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

```
ALTER USER <user> quota 100m on <tablespace>;
```

or for unlimited space:

```
ALTER USER username quota unlimited on <tablespace>;
```

4. Set the following additional privileges for Oozie:

```
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 <tablespace>;
```



Important:

For security reasons, *do not* grant `select` any table privileges to the Oozie user.

5. Set the following additional privileges for the Cloudera Navigator Audit Server database:

```
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.

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

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 [Cloudera Installation Guide](#) on page 9 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**.


Parcel Name	Version	Status	Actions
ORACLE_INSTANT_CLIENT	11.2-1.oracleinstantclient1.0.0.p0.130 	Distributed, Activated	<button>Deactivate</button>

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

- Select **Actions > Synchronize Database**
- 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](#)

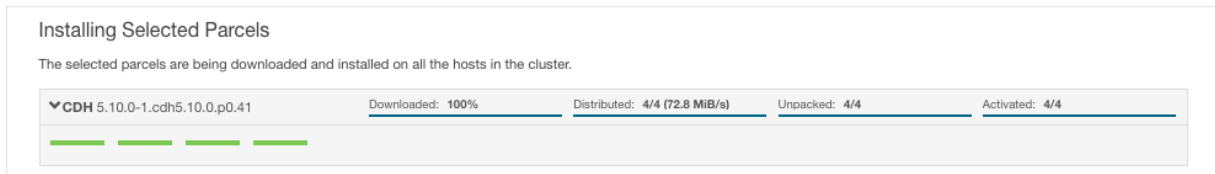
Connect Hue Service to Oracle

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 [Cloudera Installation Guide](#) on page 9 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/<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 an Oracle Database for Cloudera Manager

After completing the above instructions to install and configure Oracle databases for Cloudera software, continue to [Step 5: Set up the Cloudera Manager Database](#) on page 130 to configure a database for Cloudera Manager.

Configuring an External Database for Sqoop 2



Note: This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

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 [Database Requirements](#) 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 the PostgreSQL documentation to install it.

See [Install and Configure PostgreSQL for Cloudera Software](#) on page 113.

Create the Sqoop 2 User and Sqoop 2 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 2 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 2 User and Sqoop 2 Database](#) on page 129.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

7. Restart the service.

Step 5: Set up the Cloudera Manager Database

Cloudera Manager Server includes a script that can create and configure a database for itself. The script can:

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

Although the script can create a database, the following procedures assume that you have already created the database as described in [Step 4: Install and Configure Databases](#) on page 101.

The following sections describe the syntax for the script and demonstrate how to use it:

Syntax for `scm_prepare_database.sh`

The syntax for the `scm_prepare_database.sh` script is as follows:

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh [options] <databaseType>
<databaseName> <databaseUser> <password>
```



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

To create a new database, you must specify the `-u` and `-p` parameters for a user with privileges to create databases. If you have already created the database as instructed in [Step 4: Install and Configure Databases](#) on page 101, do not specify these options.

The following tables describe the parameters and options for the `scm_prepare_database.sh` script:

Table 20: Parameters

Parameter (Required in bold)	Description
<databaseType>	One of the supported database types: <ul style="list-style-type: none"> • MariaDB: <code>mysql</code> • MySQL: <code>mysql</code> • Oracle: <code>oracle</code> • PostgreSQL: <code>postgresql</code>
<databaseName>	The name of the Cloudera Manager Server database to use. For MySQL, MariaDB, and PostgreSQL databases, the script can create the specified database if you specify the <code>-u</code> and <code>-p</code> options with the credentials of a user that has privileges to create databases and grant privileges. The default database name provided in the Cloudera Manager configuration settings is <code>scm</code> , but you are not required to use it.
<databaseUser>	The username for the Cloudera Manager Server database to create or use. The default username provided in the Cloudera Manager configuration settings is <code>scm</code> , but you are not required to use it.

Parameter (Required in bold)	Description
<code><password></code>	<p>The password for the <code><databaseUser></code> to create or use. If you do not want the password visible on the screen or stored in the command history, do not specify the password, and you are prompted to enter it as follows:</p> <pre>Enter SCM password:</pre>

Table 21: Options

Option	Description
<code>-? --help</code>	Display help.
<code>--config-path</code>	The path to the Cloudera Manager Server configuration files. The default is <code>/etc/cloudera-scm-server</code> .
<code>-f --force</code>	If specified, the script does not stop if an error occurs.
<code>-h --host</code>	The IP address or hostname of the host where the database is installed. The default is to use <code>localhost</code> .
<code>-p --password</code>	<p>The admin password for the database application. Use with the <code>-u</code> option. The default is no password. Do not put a space between <code>-p</code> and the password (for example, <code>-phunter2</code>). If you do not want the password visible on the screen or stored in the command history, use the <code>-p</code> option without specifying a password, and you are prompted to enter it as follows:</p> <pre>Enter database password:</pre> <p>If you have already created the database, do not use this option.</p>
<code>-P --port</code>	The port number to use to connect to the database. The default port is 3306 for MariaDB, 3306 for MySQL, 5432 for PostgreSQL, and 1521 for Oracle. This option is used for a remote connection only.
<code>--scm-host</code>	The hostname where the Cloudera Manager Server is installed. If the Cloudera Manager Server and the database are installed on the same host, do not use this option or the <code>-h</code> option.
<code>--scm-password-script</code>	A script to execute whose <code>stdout</code> provides the password for user SCM (for the database).
<code>-u --user</code>	The admin username for the database application. Use with the <code>-p</code> option. Do not put a space between <code>-u</code> and the username (for example, <code>-uroot</code>). If this option is supplied, the script creates a user and database for the Cloudera Manager Server. If you have already created the database, do not use this option.

Preparing the Cloudera Manager Server Database

1. Run the `scm_prepare_database.sh` script on the Cloudera Manager Server host, using the database name, username, and password you created in [Step 4: Install and Configure Databases](#) on page 101:

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh <databaseType> <databaseName> <databaseUser>
```

When prompted, enter the password.

2. If it exists, remove the embedded PostgreSQL properties file:

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

Installing Cloudera Manager, CDH, and Managed Services

The following examples demonstrate the syntax and output of the `scm_prepare_database.sh` script for different scenarios:

Example 1: Running the script when MySQL or MariaDB 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`:

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh mysql scm scm
```

```
Enter SCM password:
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/java/jdk1.8.0_141-cloudera/bin/java -cp
/usr/share/java/mysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share/java/postgresql-connector-java.jar:/opt/cloudera/cm/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 2: Running the script when MySQL or MariaDB is installed on another host

This example demonstrates how to run the script on the Cloudera Manager Server host (`cm01.example.com`) and connect to a remote MySQL or MariaDB host (`db01.example.com`):

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh mysql -h db01.example.com --scm-host
cm01.example.com scm scm
```

```
Enter database password:
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/java/jdk1.8.0_141-cloudera/bin/java -cp
/usr/share/java/mysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share/java/postgresql-connector-java.jar:/opt/cloudera/cm/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 to configure Oracle

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh -h cm-oracle.example.com oracle
orcl sample_user sample_pass
```

```
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/java/jdk1.8.0_141-cloudera/bin/java -cp
/usr/share/java/mysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share/java/postgresql-connector-java.jar:/opt/cloudera/cm/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!
```

Installing CDH

After configuring the Cloudera Manager Server database, continue to [Step 6: Install CDH and Other Software](#) on page 133.

Step 6: Install CDH and Other Software

After setting up the Cloudera Manager database, start Cloudera Manager Server, and log in to the Cloudera Manager Admin Console:

1. Start Cloudera Manager Server:

- **RHEL 7 compatible, Ubuntu, SLES:**

```
sudo systemctl start cloudera-scm-server
```

- **RHEL 6 compatible:**

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

2. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run the following on the Cloudera Manager Server host:

```
sudo tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

When you see this log entry, the Cloudera Manager Admin Console is ready:

```
INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server.
```

If the Cloudera Manager Server does not start, see [Troubleshooting Installation Problems](#) on page 168.

3. In a web browser, go to `http://<server_host>:7180`, where `<server_host>` is the FQDN or IP address of the host where the Cloudera Manager Server is running.



Note: If you enabled [auto-TLS](#), you are redirected to `https://<server_host>:7183`, and a security warning is displayed. You might need to indicate that you trust the certificate, or click to proceed to the Cloudera Manager Server host.

4. Log into Cloudera Manager Admin Console. The default credentials are:

Username: admin

Password: admin



Note: 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.

After logging in, the installation wizard launches. The following sections guide you through each step of the installation wizard:

Welcome

The **Welcome** page provides a brief overview of Cloudera Manager, and links to the release notes for the version you are installing. Click **Continue** to proceed with the installation.

Accept License

The **Accept License** page provides the **End User License Terms and Conditions**. Read the license agreement and click the checkbox labeled **Yes, I accept the End User License Terms and Conditions** if you accept the terms and conditions of the license agreement.

Click **Continue** to proceed.

Select Edition

On the **Select Edition** 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 Cloudera Enterprise 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:
 - Essentials Edition
 - Data Science and Engineering Edition
 - Operational Database Edition
 - Data Warehouse Edition
 - Enterprise Data Hub Edition

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

2. If you select Cloudera Enterprise, install a license:

- a. Click the **Select License File** field.
- b. Browse to the location of your license file, click the file, and click **Open**.
- c. 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.

Welcome (Add Cluster - Installation)

The **Welcome** page of the **Add Cluster - Installation** wizard provides a brief overview of the installation and configuration procedure, as well as some links to relevant documentation.

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

Cluster Basics

The **Cluster Basics** page allows you to specify the **Cluster Name** and select the **Cluster Type**:

- **Regular Cluster:** A Regular Cluster contains storage nodes, compute nodes, and other services such as metadata and security collocated in a single cluster.
- **Compute Cluster:** A Compute Cluster consists of only compute nodes. To connect to existing storage, metadata or security services, you must first choose or create a Data Context on a Base Cluster.

For new installations, **Regular Cluster** is the only option. You cannot add a compute cluster if you do not have an existing base cluster.

For more information on regular and compute clusters, and data contexts, see [Virtual Private Clusters and Cloudera SDX](#).

Enter a cluster name and then click **Continue**.

Setup Auto-TLS



Important: Auto-TLS is only available with an Enterprise license.

The **Setup Auto-TLS** page provides instructions for initializing the certificate manager for auto-TLS if you have not done so already. If you already initialized the certificate manager in [Step 3: Install Cloudera Manager Server](#) on page 99, the wizard displays a message indicating that auto-TLS has been initialized. Click **Continue** to proceed with the installation.

If you have not already initialized the certificate manager, and you want to enable auto-TLS, follow the instructions provided on the page before continuing. When you reload the page as instructed, you are redirected to `https://<server_host>:7183`, and a security warning is displayed. You might need to indicate that you trust the certificate, or click to proceed to the Cloudera Manager Server host. You might also be required to log in again and re-complete the previous steps in the wizard.

For more information, see [Configuring TLS Encryption for Cloudera Manager and CDH Using Auto-TLS](#).

If you do not want to enable auto-TLS at this time, click **Continue** to proceed.

Specify Hosts

Choose which hosts will run CDH and other 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 in the **Hostnames** field. You can specify hostname and IP address ranges as follows:

Expansion Range	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].example.com	host1.example.com, host2.example.com, host3.example.com
host[07-10].example.com	host07.example.com, host08.example.com, host09.example.com, host10.example.com



Important: Unqualified hostnames (short names) must be unique in a Cloudera Manager instance. For example, you cannot have both *host01.example.com* and *host01.standby.example.com* managed by the same Cloudera Manager Server.

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. Only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. You can enter an address range that spans over unused addresses and then clear the nonexistent hosts later in the procedure, but wider ranges require more time to scan.

2. Click **Search**. 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**. You can modify the search pattern and repeat the search as many times as you need until you see all of the expected hosts.



Note: 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, and that firewalls and SELinux are not blocking access.

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 or where you do not want to install services.
4. Click **Continue**.

The **Select Repository** screen displays.

Select Repository



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

The **Select Repository** page allows you to specify repositories for **Cloudera Manager Agent** and **CDH and other software**.

In the **Cloudera Manager Agent** section:

1. Select either **Public Cloudera Repository** or **Custom Repository** for the Cloudera Manager Agent software.
2. If you select **Custom Repository**, do not include the operating system-specific paths in the URL. For instructions on setting up a custom repository, see [Configuring a Local Package Repository](#) on page 54.

In the **CDH and other software** section:

1. Select the repository type to use for the installation. In the **Install Method** section select one of the following:
 - **Use Parcels (Recommended)**

A parcel is a binary distribution format containing the program files, along with additional metadata used by Cloudera Manager. Parcels are required for rolling upgrades. For more information, see [Parcels](#).
 - **Use Packages**

A package is a standard binary distribution format that contains compiled code and meta-information such as a package description, version, and dependencies. Packages are installed using your operating system package manager.
2. Select the version of CDH to install.
 - a. If you selected **Use Parcels** and you do not see the version you want to install, click the **More Options** button to add the repository URL for your version. Repository URLs for CDH 6 version are documented in [CDH 6 Download Information](#). After adding the repository, click **Save Changes** and wait a few seconds for the version to appear. If your Cloudera Manager host uses an HTTP proxy, click the **Proxy Settings** button to configure your proxy.



Note: Cloudera Manager only displays CDH versions it can support. If an available CDH version is too new for your Cloudera Manager version, it is not displayed.

- b. If you selected **Use Packages**, and the version you want to install is not listed, you can select **Custom Repository** to specify a repository that contains the desired version. Repository URLs for CDH 6 version are documented in [CDH 6 Download Information](#).
3. If you selected **Use Parcels**, specify any **Additional Parcels** you want to install. If you are installing CDH 6, do not select the **KAFKA**, **KUDU**, or **SPARK** parcels, because they are included in CDH 6.
 4. Click **Continue**.

Accept JDK License



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.

To allow Cloudera Manager to automatically install the Oracle JDK on cluster hosts, read the JDK license and check the box labeled **Install Oracle Java SE Development Kit (JDK8)** if you accept the terms. If you installed your own Oracle JDK version in [Step 2: Install Java Development Kit](#) on page 97, leave the box unchecked.

If you allow Cloudera Manager to install the JDK, a second checkbox appears, labeled **Install Java Unlimited Strength Encryption Policy Files**. These policy files are required to enable AES-256 encryption in JDK versions lower than 1.8u161. JDK 1.8u161 and higher enable unlimited strength encryption by default, and do not require policy files.

After reading the license terms and checking the applicable boxes, click **Continue**.

Enter Login Credentials

1. Select **root** for the `root` account, or select **Another user** and enter the username for an account that has password-less `sudo` privileges.
2. 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 modify the default SSH port if necessary.

3. Specify the maximum number of host installations to run at once. The default and recommended value is 10. You can adjust this based on your network capacity.
4. Click **Continue**.

The **Install Agents** page displays.

Install Agents

The **Install Agents** page displays the progress of the installation. You can click on the **Details** link for any host to view the installation log. If the installation is stalled, you can click the **Abort Installation** button to cancel the installation and then view the installation logs to troubleshoot the problem.

If the installation fails on any hosts, you can click the **Retry Failed Hosts** to retry all failed hosts, or you can click the **Retry** link on a specific host.

If you selected the option to manually install agents, see [Manually Install Cloudera Manager Agent Packages](#) for the procedure and then continue with the next steps on this page.

After installing the Cloudera Manager Agent on all hosts, click **Continue**.

If you are using parcels, the **Install Parcels** page displays. If you chose to install using packages, the **Inspect Cluster** page displays.

Install Parcels

If you selected parcels for the installation method, the **Install Parcels** page reports the installation progress of the parcels you selected earlier. After the parcels are downloaded, progress bars appear representing each cluster host. You can click on an individual progress bar for details about that host.

After the installation is complete, click **Continue**.

The **Inspect Cluster** page displays.

Inspect Cluster

The **Inspect Cluster** page provides a tool for inspecting network performance as well as the [Host Inspector](#) to search for common configuration problems. Cloudera recommends that you run the inspectors sequentially:

1. Run the **Inspect Network Performance** tool. You can click **Advanced Options** to customize some `ping` parameters.
2. After the network inspector completes, click **Show Inspector Results** to view the results in a new tab.
3. Address any reported issues, and click **Run Again** (if applicable).
4. Click **Inspect Hosts** to run the Host Inspector utility.
5. After the host inspector completes, click **Show Inspector Results** to view the results in a new tab.
6. Address any reported issues, and click **Run Again** (if applicable).

If the reported issues cannot be resolved in a timely manner, and you want to abandon the cluster creation wizard to address them, select the radio button labeled **Quit the wizard and Cloudera Manager will delete the temporarily created cluster** and then click **Continue**.

Otherwise, after addressing any identified problems, select the radio button labeled **I understand the risks, let me continue with cluster creation**, and then click **Continue**.

This completes the **Cluster Installation** wizard and launches the **Add Cluster - Configuration** wizard.

Continue to [Step 7: Set Up a Cluster Using the Wizard](#) on page 138.

Step 7: Set Up a Cluster Using the Wizard

After completing the **Add Cluster - Installation** wizard, the **Add Cluster - Configuration** wizard automatically starts. The following sections guide you through each page of the wizard:

Select Services

The **Select Services** page allows you to select the services you want to install and configure. Make sure that you have the appropriate license key for the services you want to use. You can choose from:

Essentials

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, and Hue

Data Engineering

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Spark

Data Warehouse

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Impala

Operational Database

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and HBase

All Services (Cloudera Enterprise Data Hub)

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer

Custom Services

Choose your own services. Services required by chosen services will automatically be included. Flume can be added after your initial cluster has been set up.

To include [Cloudera Navigator data management](#), check the box labeled **Include Cloudera Navigator**.

After selecting the services you want to add, click **Continue**. The **Assign Roles** page displays.

Assign Roles

The **Assign Roles** page suggests role assignments for the hosts in your cluster. You can click on the hostname for a role to select a different host. You can also click the **View By Host** button to see all the roles assigned to a host.

To review the recommended role assignments, see [Recommended Cluster Hosts and Role Distribution](#) on page 43.

After assigning all of the roles for your services, click **Continue**. The **Setup Database** page displays.

Setup Database

On the **Setup Database** page, you can enter the database hosts, names, usernames, and passwords you created in [Step 4: Install and Configure Databases](#) on page 101. For services that support it, you can add finer-grained customizations using a JDBC URL override.



Important: The Hive service is currently the only service that supports the JDBC URL override.

Select the database type and enter the database name, username, and password for each service. For MariaDB, select **MySQL**.

For services that support it, to specify a JDBC URL override, select **Yes** in the **Use JDBC URL Override** dropdown menu. For information on the JDBC URL format, see [Specifying a JDBC URL Override for Database Connections](#). You must also specify the database type, username, and password.

Click **Test Connection** to validate the settings. If the connection is successful, a green checkmark and the word **Successful** appears next to each service. If there are any problems, the error is reported next to the service that failed to connect.

After verifying that each connection is successful, click **Continue**. The **Review Changes** page displays.

Review Changes

The **Review Changes** page lists default and suggested settings for several configuration parameters, including data directories.



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

Review and make any necessary changes, and then click **Continue**. The **Command Details** page displays.

Command Details

The **Command Details** page lists the details of the **First Run** command. You can expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting **Show All Steps**, **Show Only Failed Steps**, or **Show Only Running Steps**.

After the **First Run** command completes, click **Continue** to go to the **Summary** page.

Summary

The **Summary** page reports the success or failure of the setup wizard. Click **Finish** to complete the wizard. The installation is complete.

Cloudera recommends that you change the default password as soon as possible by clicking the logged-in username at the top right of the home screen and clicking **Change Password**.

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 Cloudera Manager](#).



Note: See [Product Compatibility Matrix for Cloudera Navigator](#) 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.	<code>/var/lib/cloudera-scm-navigator</code>

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 might 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 might 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 [Navigator Metadata Server Tuning](#) 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 [Database Requirements](#) 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 [Step 4: Install and Configure Databases](#) on page 101.

Adding Cloudera Navigator Roles During the Cloudera Manager Installation Process

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

1. Install Cloudera Manager as detailed in [Cloudera Installation Guide](#) on page 9.
2. On the first page of the Cloudera Manager installation wizard, choose one of the license options that supports Cloudera Navigator:
 - Data Science and Engineering Edition
 - Operational Database Edition
 - Data Warehouse Edition
 - Enterprise Data Hub Edition
3. Upload the license:
 - a. Click **Upload License**.
 - b. Click the document icon to the left of the **Select a License File** text field.
 - c. Go to the location of your license file, click the file, and click **Open**.
 - d. Click **Upload**.
4. Click **Continue** to proceed with the installation.
5. In the first page of the **Add Services** procedure, click the **Include Cloudera Navigator** checkbox.
6. 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. For more information, see:

- [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.
-----	---

Installing the Cloudera Navigator Data Management Component

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 Navigator 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. Topics include Auditing , Metadata , Lineage Diagrams , Cloudera Navigator and the Cloud , Services and Security Management , and more.
Upgrade	Upgrading Cloudera Manager (Cloudera Navigator is upgraded along with Cloudera Manager.)
Security	Configuring Authentication for Cloudera Navigator
	Configuring TLS/SSL for Navigator Audit Server
	Configuring TLS/SSL for Navigator Metadata Server
Release Notes	Cloudera Navigator Data Management Release Notes

Installing Cloudera Navigator Encryption Components

The following sections demonstrate how to install the Cloudera Navigator encryption components, used for encrypting data at rest in Cloudera Enterprise:

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](#) 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:

- [Configuring a Local Parcel Repository](#) on page 49
- [Configuring a Local Package Repository](#) on page 54

Installing Key Trustee Server



Important: This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See [Managing Licenses](#) for more information.

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 144.

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 143 to Cloudera Manager following the instructions in [Configuring Cloudera Manager Server Parcel Settings](#).
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](#).



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 146.

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 143.

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 [Configuring Hosts to Use the Internal Repository](#) on page 58 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 [Configuring a Local Package Repository](#) on page 54.

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 146.

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 29 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 [Configuring a Local Package Repository](#) on page 54.

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 148 above for more information.

2. Install the Key HSM repository

Add the local Key HSM repository you created in Step 1. See [Configuring a Local Package Repository](#) on page 54 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 [Configuring a Local Package Repository](#) on page 54.

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 143 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 2: Installing JCE Policy File for AES-256 Encryption](#).

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 [Configuring a Local Parcel Repository](#) on page 49 if you are using parcels, or [Configuring a Local Package Repository](#) on page 54 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 Cloudera Manager to Use an Internal Remote Parcel Repository](#) on page 53 for more information.
3. Download, distribute, and activate the Key Trustee KMS parcel. See [Managing Parcels](#) 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 149, configure the Key Trustee KMS host to use the repository. See [Configuring Hosts to Use the Internal Repository](#) on page 58 for more information.
2. Because the `keytrustee-keyprovider` package depends on the `hadoop-kms` package, you must add the CDH repository. See [Configuring a Local Package Repository](#) on page 54 for instructions.
3. Install the `keytrustee-keyprovider` package using the appropriate command for your operating system:

Installing Cloudera Navigator Encryption Components

- **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 143
- [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 nSolo, nConnect 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 [Configuring a Local Parcel Repository](#) on page 49 if you are using parcels, or [Configuring a Local Package Repository](#) on page 54 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 Cloudera Manager to Use an Internal Remote Parcel Repository](#) on page 53 for more information.
3. Download, distribute, and activate the Navigator HSM KMS parcel. See [Managing Parcels](#) 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 HSM is `KEYTRUSTEE`.

4. If you are newly installing Thales HSM KMS to a 6.0.0 system, then you must set the port to a non-default value before adding the HSM KMS backed by Thales service in Cloudera Manager. The recommended port is 11501. The non-privileged port default is 9000 (which you do not have to change). To change the privileged port, log into the Thales HSM KMS machine(s), and run the following commands:

```
# sudo /opt/nfast/bin/config-serverstartup --enable-tcp --enable-privileged-tcp
--privport=11501
[server_settings] change successful; you must restart the hardserver for this to take
effect
# sudo /opt/nfast/sbin/init.d-ncipher restart
-- Running shutdown script 90ncsnmpd

-- Running shutdown script 60raserv

...

'ncsnmpd' server now running
```

Installing Navigator HSM KMS Backed by Thales HSM Using Packages

1. After [Setting Up an Internal Repository](#) on page 151, configure the Navigator KMS Services backed by Thales HSM host to use the repository. See [Configuring Hosts to Use the Internal Repository](#) on page 58 for more information.
2. Because the `keytrustee-keyprovider` package depends on the `hadoop-kms` package, you must add the CDH repository. See [Configuring a Local Package Repository](#) on page 54 for instructions.
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
```

Installing Cloudera Navigator Encryption Components

4. If you are newly installing Thales HSM KMS to a 6.0.0 system, then you must set the port to a non-default value before adding the HSM KMS backed by Thales service in Cloudera Manager. The recommended port is 11501. The non-privileged port default is 9000 (which you do not have to change). To change the privileged port, log into the Thales HSM KMS machine(s), and run the following commands:

```
# sudo /opt/nfast/bin/config-serverstartup --enable-tcp --enable-privileged-tcp
--privport=11501
[server_settings] change successful; you must restart the hardserver for this to take
effect
# sudo /opt/nfast/sbin/init.d-ncipher restart
-- Running shutdown script 90ncsnmpd

-- Running shutdown script 60raserv

...

'ncsnmpd' server now running
```

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 [Navigator HSM KMS: Recommended Hardware and Supported Distributions](#).

Before performing the Luna HSM KMS setup, run the `vtl 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 [Configuring a Local Parcel Repository](#) on page 49 if you are using parcels, or [Configuring a Local Package Repository](#) on page 54 if 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 Cloudera Manager to Use an Internal Remote Parcel Repository](#) on page 53 for more information.

- Download, distribute, and activate the Navigator HSM KMS parcel. See [Managing Parcels](#) 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

- After [Setting Up an Internal Repository](#) on page 152, configure the Navigator HSM KMS backed by Luna HSM host to use the repository. See [Configuring Hosts to Use the Internal Repository](#) on page 58 for more information.
- Because the `keytrustee-keyprovider` package depends on the `hadoop-kms` package, you must add the CDH repository. See [Configuring a Local Package Repository](#) on page 54 for instructions.
- 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 5](#) 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 [Configuring a Local Package Repository](#) on page 54.

Installing Navigator Encrypt (RHEL-Compatible)



Note: For details about supported Linux Operating Systems, refer to the [Table 5](#).

1. Install the Cloudera Repository

Add the internal repository you created. See [Configuring Hosts to Use the Internal Repository](#) on page 58 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 Phone](#).
- 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/repoforge/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 [Configuring Hosts to Use the Internal Repository](#) on page 58 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
```

Installing Cloudera Navigator Encryption Components

- **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 only) Run `systemctl daemon-reload`

Due to [changes](#) in SLES 12, 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 [Configuring Hosts to Use the Internal Repository](#) on page 58 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
```

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 ]
```

Installing Cloudera Navigator Encryption Components

```
[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 158.



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 153.

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.

After Installation

The following topics describe post-installation actions, such as deploying client configuration and some simple tests to validate the installation and confirm that everything is working as expected.

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 might need to redeploy the client configuration files. If a service's status is "Client configuration redeployment required," you need to redeploy those files.

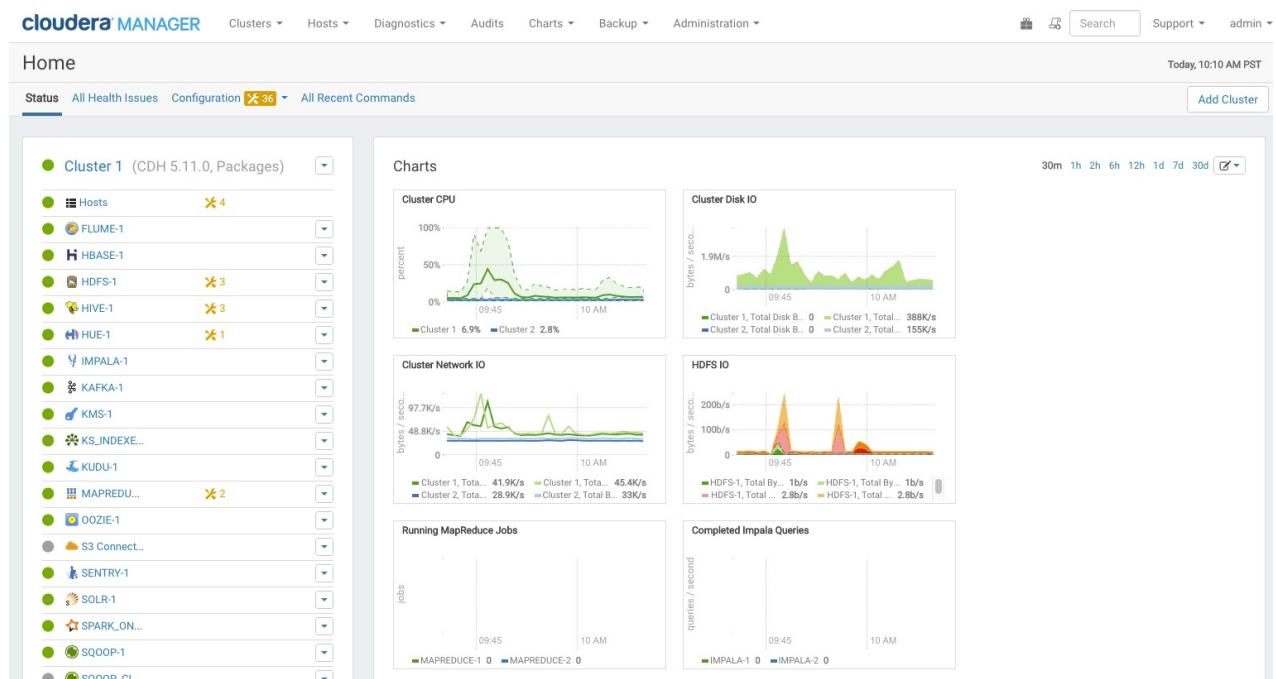
See [Client Configuration Files](#) for information on downloading client configuration files, or redeploying them through Cloudera Manager.


Testing the Installation



Note: This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

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`
3. Depending on whether your cluster is configured to run MapReduce jobs on the YARN or MapReduce service, view the results of running the job by selecting one of the following from the top navigation bar in the Cloudera Manager Admin Console :
 - **Clusters > *ClusterName* > yarn Applications**
 - **Clusters > *ClusterName* > mapreduce Activities**

If you run the PiEstimator job on the YARN service (the default) you will see an entry like the following in **yarn Applications**:

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

Testing with Hue

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

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

For more information, see the [Hue User Guide](#).

Installing the GPL Extras Parcel

GPL Extras contains functionality for [compressing data](#) using the LZO compression algorithm.

To install the GPL Extras parcel:

1. Add the appropriate repository to the Cloudera Manager list of [parcel repositories](#). Specify the repository in Cloudera Manager as follows:
 - **CDH 6:** `https://archive.cloudera.com/gplextras6/6.x.y/parcels/`

- **CDH 5:** <https://archive.cloudera.com/gplextras5/parcels/5.x.y/>

Replace x.y with the minor and maintenance version (for example, 5.14.1 or 6.2.0). If you are using LZO with Impala, make sure that you match the GPL Extras parcel version to the CDH version.

2. Download, distribute, and activate the parcel.
3. The LZO parcels require that the underlying operating system has the native LZO packages installed. If they are not installed on all cluster hosts, you can install them as follows:

RHEL compatible:

```
sudo yum install lzo
```

Debian or Ubuntu:

```
sudo apt-get install liblzo2-2
```

SLES:

```
sudo zypper install liblzo2-2
```

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 the CDH Cluster](#).

Download, Distribute, and Activate Parcels

1. In the Cloudera Manager Admin Console, click the parcel icon 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](#) page. 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 [Configuring a Local Parcel Repository](#) on page 49 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. If you have enabled [high availability for HDFS](#), you can choose [Rolling Restart](#) instead to minimize cluster downtime. 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, Kudu, and Search**

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

- **Including Impala, Kudu, and Search**

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

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

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

4. If your Hue service uses the embedded SQLite 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.

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 can 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 the CDH Cluster](#).

Install CDH and Managed Service Packages

Choose a Repository Strategy

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

- Public 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 [Custom Installation Solutions](#) on page 48. When using an internal repository, you must copy the `.repo` or `.list` file to the Cloudera Manager Server host and update the repository properties to point to internal repository URLs.

Install 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 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 kudu llama oozie pig pig-udf-datafu search sentry solr-mapreduce
spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2
```



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 kudu llama oozie pig pig-udf-datafu search sentry solr-mapreduce
spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2
```



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

OS Version	Package Link
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 kudu llama oozie pig pig-udf-datafu search sentry solr-mapreduce
spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2
```




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

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. If you have enabled [high availability for HDFS](#), you can choose **Rolling Restart** instead to minimize cluster downtime. 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**.

Secure Your Cluster

After completing your Cloudera Enterprise installation and making sure that everything is working properly, secure your cluster by enabling authentication, authorization, auditing, and encryption.

For comprehensive instructions on securing your cluster, see [Cloudera Security](#).

Troubleshooting Installation Problems

This topic describes common installation issues and suggested solutions.

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`.

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 173.

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 [Install and Configure MySQL for Cloudera Software](#) on page 107.

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.

Cluster hosts do not appear

Some cluster hosts do not appear when you click **Find Hosts** in install or update wizard.

Possible Reasons

You might 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.

"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 [Configure Network Names](#) on page 21.

Troubleshooting Installation Problems

- 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 [Step 4: Install and Configure Databases](#) on page 101.

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 63 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 [Install and Configure MySQL for Cloudera Software](#) on page 107.

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 6 and CDH 6, then uninstall Cloudera Manager and CDH, and then attempt to install CDH 5 and Cloudera Manager 5, incorrect cached information might 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

```
sudo 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;
```

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.

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 176, `/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 might 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

For each cluster managed by Cloudera Manager:

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.

2. 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.

Deactivate and Remove Parcels

If you installed using packages, skip this step and go to [Uninstall the Cloudera Manager Server](#) on page 174; you will remove packages in [Uninstall Cloudera Manager Agent and Managed Software](#) on page 174. 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.

Uninstalling Cloudera Manager and Managed Software

There might 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/cmfd/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 7, SLES 12, Debian 8, Ubuntu 16.04 and higher

```
sudo systemctl stop supervisord
```

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04

```
sudo service cloudera-scm-agent hard_stop
```

2. Uninstall software:

OS	Parcel Install	Package Install
RHEL	<pre>\$ sudo yum remove 'cloudera-manager-*</pre>	<ul style="list-style-type: none"> • CDH 5 <pre>\$ sudo yum remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-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 oozie pig pig-udf-datafu search sentry solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>
SLES	<pre>\$ sudo zypper remove 'cloudera-manager-*</pre>	<ul style="list-style-type: none"> • CDH 5 <pre>\$ sudo zypper remove 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama oozie pig pig-udf-datafu search sentry solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>
Debian/Ubuntu	<pre>\$ sudo apt-get purge 'cloudera-manager-*</pre>	<ul style="list-style-type: none"> • CDH 5 <pre>\$ sudo apt-get purge 'cloudera-manager-*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hadoop-kms hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama oozie pig pig-udf-datafu search sentry solr-mapreduce spark-core spark-master spark-worker spark-history-server spark-python sqoop sqoop2 hue-common oozie-client solr solr-doc sqoop2-client zookeeper</pre>

3. Run the clean command:

RHEL

```
sudo yum clean all
```

SLES

```
sudo zypper clean
```

Uninstalling Cloudera Manager and Managed Software

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/cmfs /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
```

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.

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.

Appendix: Apache License, Version 2.0

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.
```