Cloudera Runtime 7.3.1

Application Access

Date published: 2023-09-07
Date modified: 2024-12-04



https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Application Access 4

Services support client RPMs for CDP Private Cloud Base 7.3.1 4

Prerequisites 5

Support for packages on RHEL, SLES, and Ubuntu 5

Package management tools 5

Repository configuration files 6

Select the repository strategy 6

Option 1 7
Option 2 7
Option 3 7
Creating an internal yum repository for RHEL packages 7

Downloading and configuring the client packages 9

RHEL 9
SLES 9
Ubuntu 9

Installing the client packages 10

Impala client packages 11
Sqoop client packages 13
Installing Phoenix client packages 15
Installing HBase client packages 15
Installing Solr client packages 16

Application Access

Learn more about the client RPMs /Debian packages, role requirements and how to access the services installed on the CDP cluster from your containerized application or edge nodes.

Note: This feature is available with limited availability.

Role Requirements

For users creating applications that connect to a CDP service(s), the Cluster Administrator must have provided the configurations required for you to connect to the services.

Client RPMs enable you to access services installed on the CDP cluster from your containerized application or edge nodes that are not managed by Cloudera Manager. You can use Client RPMs for the use cases that require thin client applications binding, without having to download and install the entire CDP parcel which requires over 10 GB of storage space on each platform node. Starting with the CDP Private Cloud Base 7.1.9 release, '-client' RPM/Debian packages are available.

Support for Application Access

Client libraries can be deployed on nodes that are not managed by Cloudera Manager. However, the burden of managing the configuration of those nodes such that the clients can connect and interact with the cluster falls outside the scope of support. Therefore, before contacting Cloudera support, the client issue must be replicated on a managed node by the customer. If the issue cannot be replicated on a managed node, then the issue is likely due to the configuration of the unmanaged node.

Services support client RPMs for CDP Private Cloud Base 7.3.1

Learn about the list of services that supports client RPMs for the CDP Private Cloud Base 7.3.1.

Services	Package
Hadoop	hadoop-client hadoop-libhdfs-client hadoop-mapreduce-client hadoop-yarn-client hadoop-hdfs-client
ZooKeeper	zookeeper-client zookeeper-native-client
Solr	solr-client search-client solr-crunch-client solr-mapreduce-client

Services	Package
Avro	avro-tools-client
HBase	hbase-client hbase-solr-client hbase_mcc-client
Impala	impala-shell-client impala-udf-devel-client
Oozie	oozie-client
Ozone	ozone-client
Phoenix	phoenix-client phoenix_queryserver-client
Spark	Cloudera recommends submitting spark jobs from edge nodes using Livy, see Submitting Spark applications using Livy.
Sqoop	sqoop-client

Prerequisites

Learn more about the configuration files and changes required to the edge node.

Copy all the required configuration files to the edge node and ensure that the configuration changes are available on the edge node.

Here are few examples:

- Copy the keytabs from the cm_host to the edge node host
- Copy the cloudera-scm-agent certificate from cm_host to edge node host
- Copy the krb5.conf files from cm_host to edge node host
- Copy the /etc/ configs files from cm_host to edge node host

Support for packages on RHEL, SLES, and Ubuntu

Understand about the support for the package on RHEL, SLES, and Ubuntu.

The client RPM package works on RHEL 7, RHEL 8, RHEL 9, SLES 12 and SLES 15. The Debian based packages should work on Ubuntu 20 and on all Debian-based Linux distributions.

Package management tools

Learn about the package management tools for RHEL, SLES, and Ubuntu.

Package management tools

- For RHEL 7 and RHEL 8, you can use the YUM package management tool and for RHEL 9, you can either use YUM or DNF package management tool.

  Note: Provide the .repo configuration file for YUM and DNF before the installation.

  The configuration file for YUM and related utilities is located in /etc/yum.conf
- For SLES 12 and SLES 15, you can use the zypper package management tool.
- For Ubuntu 20, you can use apt get install tool. The configuration file for the Debian package is called .list. The .list files contain configuration details for the Debian repository.

Repository configuration files

Learn about the repository configuration files.

Repository configuration files

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

These are the location of the different configuration files.

- RHEL compatible (yum): /etc/yum.repos.d
- SLES (zypper): /etc/zypp/repos.d/
- Ubuntu (apt-get): /etc/apt/sources.list.d/(Additional repositories are specified using .list files in the /etc/apt/sources .list.d/ directory.)



Important: The client RPMs are signed with the Cloudera GPG key. You must import the repository signing GPG key, and use it before starting the installation process.

Select the repository strategy

Learn about the different repository strategy and choose the preferred one.

You can choose one of the following repository strategies to access the client packages.

Method 1

If you have internet access, you can access the Cloudera public repositories that contain the client RPM packages and available through the Cloudera provided repo location.

For example: https://archive.cloudera.com/p/cdh_client/7.3.1.0/

Ensure that you add the required repository location information to your selected .repo file present in your system and access the client packages. Alteratively, you can also make your own .repo file.

Method 2

For the internally hosted repositories, you can download the RHEL, SLES, or Ubuntu tarball to your local machine and create a local repository to access the client packages. You can use one of the following options to access the client packages.

Related Information

Option 1

Option 2

Option 3

Option 1

For the internally hosted repositories, you can download the RHEL, SLES, or Ubuntu tarball in your local machine and create a local repository to access the client packages. Learn about the option 1.

You can download the tarball and host all the client packages in a locallly shared system such as NFS. You can use the shared location that contains the yum repository and RPM content and access it across multiple locations using the mount point method.

Related Information

Select the repository strategy

Option 2

For the internally hosted repositories, you can download the RHEL, SLES, or Ubuntu tarball in your local machine and create a local repository to access the client packages. Learn about the option 2.

You can download the tarball and host all the client packages in a local file system and then use it.

Related Information

Select the repository strategy

Option 3

For the internally hosted repositories, you can download the RHEL, SLES, or Ubuntu tarball in your local machine and create a local repository to access the client packages. Learn about the option 3 and how to create an internal yum repository, temporary internal repository, and configure hosts to use the internal repository.

You can have a centralized Apache server running and host all the content on that server.

Related Information

Select the repository strategy

Creating an internal yum repository for RHEL packages

Learn more about how to create an internal yum internal repository using the 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 and then download the repository files to the Web server host. The examples in this section use Apache HTTP Server as the Web server.

1. Install Apache HTTP Server:

```
sudo yum install httpd
```

2. Start Apache HTTP Server:

```
sudo systemctl start httpd
```

Downloading and publishing the package repository for the CDH client

1. Download the package repository of CDH client that you want to install.
2. Download the files for a CDP Private Cloud Base release:
   - Run the following command to create a local repository directory for the Cloudera package repository:

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

- Run the following command to modify the file permission that allows you to download the files under the local repository directory:

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

- Copy the files to the yum directory.

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

Creating a temporary internal repository

You can create a temporary remote repository to deploy packages on a one-time basis. Cloudera recommends using the same host that runs CDP Private Cloud Base 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 to follow the instructions in "Downloading and publishing the package repository for the CDH Client".
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
```

4. Serving HTTP on 0.0.0.0 port 8900 ...
5. 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 creating the repository, modify the client configuration to use it:

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

Operating System	Procedure
	After creating the .list file, run the following command: sudo apt-get update

Downloading and configuring the client packages

Know about the process on how to download and configure the client packages for RHEL, SLES and Ubuntu.

RHEL

Know about the process on how to download and configure the client packages for RHEL

Note: Contact the Cloudera Account Team to get access to the CDP 7.3.1 RHEL client packages.

To setup the repository for RHEL client package, you can follow the below example.

For example:

```
Create /etc/yum.repos.d/cloudera-repo.repo files on cluster hosts with
 the following content, where <web_server> is the hostname of the Web
server:
[cloudera-repo]
name=cloudera-repo
baseurl=http://<web_server>/<cdh-client >
enabled=1
gpgcheck=1
```

SLES

Know about the process on how to download and configure the client packages for SLES.

Note: Contact the Cloudera Account Team to get access to the CDP 7.3.1 SLES client packages.

To set up the repository for SLES client package, you can follow the below example.

For example:

```
sles15-builds-5gkjt:~ # cat /etc/zypp/repos.d/cdh-client.repo
[CDH-7.3.1.0-xxx]
name=CDH Version - CDH-7.3.1.x.xxxx
baseurl=file:///grid/0/jenkins/zypp/
gpgcheck=1
gpgkey=file:///grid/0/jenkins/zypp/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
enabled=1
priority=1
type=rpm-md
sles15-builds-5gkjt:~ #
```

Ubuntu

Know about the process on how to download and configure the client packages for Ubuntu.

Note: Contact the Cloudera Account Team to get access to the CDP 7.3.1 Ubuntu client packages.

Creating an APT repository for Ubuntu Packages

1. Create the following file:

```
/etc/apt/sources.list.d/cloudera-client.list
```

2. Add the client path to the arch packages.

Note: Modify the GBN value as per the builds that are being validated

3. Check if the following signature error occur:

```
The following signatures could not be verified because the public key is
not available
```

4. Run the following command to fix the signature error:

```
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys B9733A7A07513CAD
```

5. Update the repository.

```
apt update
```

6. Install the package.

```
apt install <package>
```

Installing the client packages

Know more about installing the client packages for RHEL, SLES, and Ubuntu.

RHEL

Important: You must import the repository signing GPG key and use it before starting the installation process.

You can install the client RPM for the supported service and yum update to update the newly added repository.

```
yum install <service-client>
```

Here are some examples:

```
yum install spark-python-client
yum install ozone-client
```

SLES

Important: You must import the repository signing GPG key and use it before starting the installation process.

You can install the client RPM for the supported service.

```
Zypper install <service-client>
```

Here are some examples:

```
Zypper install spark-python-client
```

```
Zypper install ozone-client
```

Ubuntu

You can install the client package for the supported service.

```
apt install <service-client>
```

Here are some examples:

```
apt install spark-python-client
apt install ozone-client
```

Impala client packages

Learn about how to install the Impala shell, a command-line interface for running SQL queries, and the necessary development files for building custom C++ User-Defined Functions (UDFs).

Installing the Impala Shell Client

The Impala shell is available as a client package for various operating systems, such as RPMs for Red Hat/CentOS/SUSE and DEBs for Ubuntu. It can be installed using native package managers and supports multiple connection scenarios, including various security protocols.

Prerequisites: You must ensure that the official Cloudera repository for your operating system is properly configured on your target host.

1. Install the RPM.
   - For Red Hat / CentOS 7:

     ```
     sudo yum install impala-shell-client
     ```
   - For Red Hat 9:

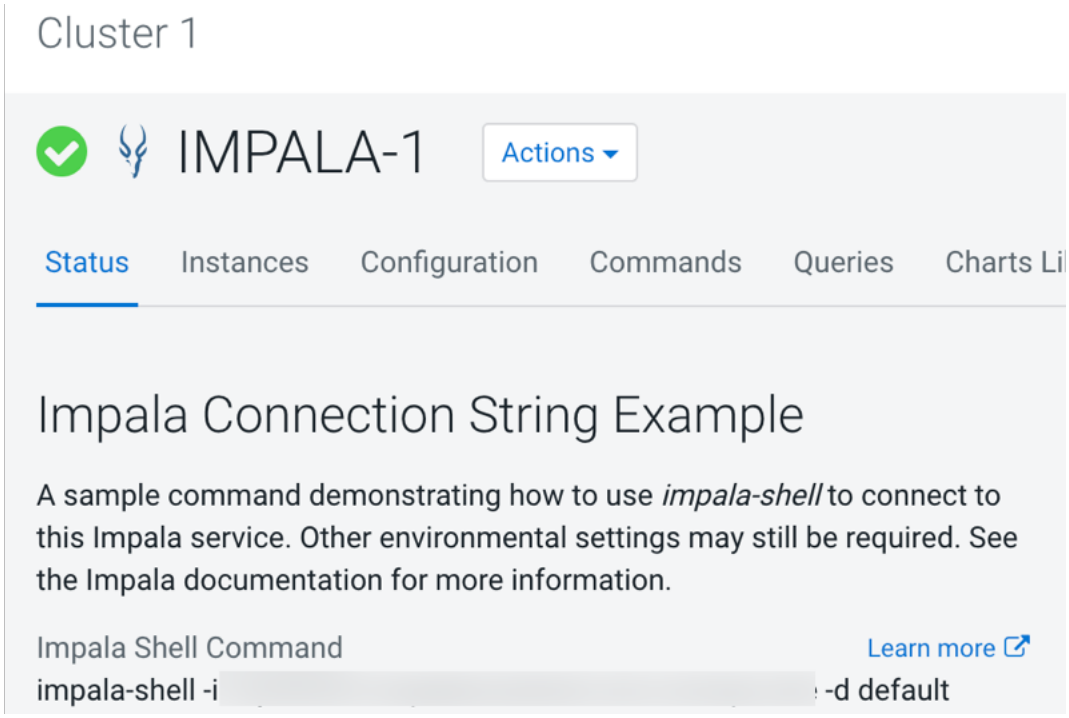
     ```
     sudo dnf install impala-shell-client
     ```
   - For SUSE 12 / SUSE 15:

     ```
     sudo zypper install impala-shell-client
     ```
   - For Ubuntu 20 / Ubuntu 22:

     ```
     sudo apt-get update
     sudo apt-get install impala-shell-client
     ```
2. Verify the installation:

   ```
   which impala-shell
   ```

   ```
   impala-shell --version
   ```
3. Connect to an Impala Coordinator.
   - If the client node is within the Cloudera Manager cluster:
     a. Find the connection string example from the **Status** tab on your Impala cluster page within Cloudera Manager.



b. Run the shell command on the client node, replacing <impala-coordinator-host> with your coordinator's hostname:

```
impala-shell -i <impala-coordinator-host> -d default
```

Result:

```
Starting Impala Shell with no authentication using Python <python-ve
rsion> Opened TCP connection to <impala-coordinator-host>:21050 Conn
ected to <impala-coordinator-host>:21050 Server version: impalad ver
sion <build-version> Query: use `default` ************************
*************************************************** Welcome to
 the Impala shell. ... You can run a single query from the command l
ine using the '-q' option. ***************************************
***************************************** Query: use `default` [<im
pala-coordinator-host>:21050] default>
```

c. If Kerberos is enabled, ensure a valid Kerberos ticket has been obtained for the Impala user on the client node. For more, see *Enabling Kerberos Authentication for CDP*.

d. If SSL is enabled, ensure that the CA certificate file exists at the configured path on the client node.

- If the client node is outside the Cloudera Manager cluster:

a. Without SSL and Kerberos:

If the Impala cluster does not use SSL or kerberos, you can connect directly with a simple command:

```
impala-shell -i <impala-coordinator-host> -d default
```

b. With SSL and Kerberos:

If SSL and kerberos are enabled, direct access to Impala from outside the Cloudera Manager cluster may not be possible due to security restrictions.

c. Using Knox as a gateway (if enabled):

If Knox is available, you can connect through Knox + LDAP authentication using the following command:

```
impala-shell -i <impala-coordinator-host>:8443 \
```

```
  --protocol=hs2-http \
  --ssl \
  -l -u <username> \
  --ldap_password_cmd='echo -n <password>' \
  --http_path=gateway/cdp-proxy-api/impala
```

This approach uses HS2 HTTP protocol through Knox gateway, with LDAP credentials and SSL encryption.

Installing the Impala UDF Development Client

To develop Impala User-Defined Functions (UDFs) in C++, you need to install the Impala development headers and library files. The following steps guide you through downloading, installing, and verifying these files on your target host.

Prerequisites: You must ensure that the official Cloudera repository for your operating system is properly configured on your target host.

1. Install the downloaded package.
   - For Red Hat / CentOS 7

     ```
     sudo yum install impala-udf-devel-client
     ```
   - For Red Hat 9

     ```
     sudo dnf install impala-udf-devel-client
     ```
   - For SUSE 12 / SUSE 15

     ```
     sudo zypper install impala-udf-devel-client
     ```
   - For Ubuntu 20 / Ubuntu 22

     ```
     sudo apt-get update
     sudo apt-get install impala-udf-dev-client
     ```
2. Verify the installation by checking for the presence of UDF headers and library files.

   ```
   ls /usr/include/impala_udf/udf.h
   ls /usr/lib64/libImpalaUdf*
   # Or, on some systems:
   ls /usr/lib/libImpalaUdf*
   ```

   Result: The Impala UDF development headers (/usr/include/impala_udf/udf.h) and library files (/usr/lib64/libImpalaUdf*) are successfully installed.

To learn how to use these headers and libraries to develop Impala UDFs in C++, see *User-defined functions (UDFs)*

Related Information

Enabling Kerberos Authentication for CDP

User-defined functions (UDFs)

Sqoop client packages

Use these steps to install client RPMs and configure Sqoop for connecting to a Cloudera cluster from an unmanaged node.

Prerequisites

- Infrastructure: Unmanaged host running one of these operating systems:

  - RHEL or RHEL-compatible (using yum)
  - SUSE Linux Enterprise Server (SLES) (using zypper)
  - Ubuntu or Debian (using apt)
- Java Development Kit: Full JDK (not just JRE) installed on the unmanaged node, matching the version used on the managed cluster.

Step 1: Set up Kerberos and Java Key Stores

If your cluster uses Kerberos for authentication (highly recommended for secure environments):

1. Install Java Key Stores and Trust Stores.

   Import your Java key-store (.jks) and trust-store files as required by your cluster security configuration.
2. Validate Kerberos Client Installation

   Ensure the unmanaged node has Kerberos utilities (krb5-workstation or appropriate packages) installed and configured to communicate with the cluster's Kerberos KDC.

Step 2: Obtain and Copy Configuration Files

To ensure consistent configuration, copy the necessary service configuration files from a managed Cloudera Manager host to unmanaged nodes.

1. On a managed node with the related components installed (typically under /etc), locate the following config directories:
   - /etc/hadoop/conf
   - /etc/hive/conf (if using Hive with Sqoop)
   - /etc/hbase/conf (if using HBase with Sqoop)
2. Copy these directories, preserving ownership and permissions, to the same locations on the unmanaged node.

Step 3: Sqoop-Specific configuration and setup

3.1 Install Java Development Kit (JDK)

- Install the full Java Development Kit (JDK) on the unmanaged node, matching the version used in the managed cluster.
- The JDK is required for Sqoop to function properly.

3.2 Install Sqoop and Dependencies

- Use the unmanaged node's default package manager to install the Sqoop client along with its dependencies:
  - For RHEL/CentOS:

    ```
    sudo yum install sqoop-client
    ```
  - For Ubuntu/Debian:

    ```
    sudo apt-get install sqoop-client
    ```
  - For SLES:

    ```
    sudo zypper install sqoop-client
    ```

3.3 Install JDBC Drivers

Place all necessary JDBC driver files required by Sqoop into:

```
/usr/lib/sqoop/lib/
```

Note: If configurations are modified via Cloudera Manager on the managed hosts, you must manually update the configuration files on the unmanaged node to reflect these changes.

Related Information

Enabling Kerberos Authentication for CDP

Installing Phoenix client packages

Know how to install client RPMs and configure Phoenix to connect to a Cloudera cluster.

Before you begin

The Java Development Kit must be installed and configured on your system.

Procedure

1. Download and configure the client RPMs for the RHEL, SUSE Linux Enterprise Server (SLES), and Ubuntu operating systems. For more information, see *Downloading and configuring the client packages*.

   Additionally, configure the following gpay key location for RHEL.

   ```
   gpgkey={{ baseurl }} RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
   ```

   The following is an example of the baseurl.

   ```
   baseurl=http://cloudera-build-2-us-west-2.vpc.cloudera.com/s3/build/6898
   0718/cdh_client/7.x/redhat8/yum/
   ```

2. Install the Phoenix client packages.

   For example,

   ```
   yum install phoenix-client -y
   ```

   For more information, see *Installing the client packages*.

3. Download the hbase/conf directory to the node.

   For example,

   ```
   get -r /etc/hbase /etc
   get -r /etc/hbase/conf /etc/hbase
   ```

4. Launch the Apache Phoenix client using the following command.

   ```
   phoenix-sqlline
   ```

Related Information

Downloading and configuring the client packages
Installing the client packages

Installing HBase client packages

Know how to install client RPMs and configure HBase to connect to a Cloudera cluster.

Before you begin

The Java Development Kit must be installed and configured on your system.

Procedure

1. Download and configure the client RPMs for the RHEL, SUSE Linux Enterprise Server (SLES), and Ubuntu operating systems. For more information, see *Downloading and configuring the client packages*.

   Additionally, configure the following gpay key location for RHEL.

   ```
   gpgkey={{ baseurl }} RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
   ```

   The following is an example of the baseurl.

   ```
   baseurl=http://cloudera-build-2-us-west-2.vpc.cloudera.com/s3/build/6898
   0718/cdh_client/7.x/redhat8/yum/
   ```

2. Install the HBase client packages.

   For example,

   ```
   yum install hbase-client -y
   ```

   For more information, see *Installing the client packages*.

3. Download the hbase/conf directory to the node.

   For example,

   ```
   get -r /etc/hbase /etc
   get -r /etc/hbase/conf /etc/hbase
   ```

4. Launch the Apache HBase client using the following command.

   ```
   hbase shell
   ```

Related Information

Downloading and configuring the client packages
Installing the client packages

Installing Solr client packages

Know how to install client RPMs and configure Solr client tools to connect to a Cloudera cluster.

About this task

Cloudera supports the following Search client packages for edge nodes and application hosts:

- solr-client — general Solr client tooling, including solrctl
- solr-crunch-client — Solr Crunch indexing tools
- solr-mapreduce-client — MapReduceIndexerTool client dependencies
- hbase-solr-client — HBase–Solr and Lily HBase Indexer client tooling

Install only the packages required for your workload. Package names are the same across RHEL, SLES, and Ubuntu; use yum, zypper, or apt as described in *Installing the client packages*.

Before you begin

The Java Development Kit must be installed and configured on your system.

Procedure

1. Download and configure the client RPMs for the RHEL, SUSE Linux Enterprise Server (SLES), and Ubuntu operating systems. For more information, see *Downloading and configuring the client packages*.

Additionally, configure the following gpg key location for RHEL.

```
gpgkey={{ baseurl }} RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
```

The following is an example of the baseurl.

```
baseurl=http://cloudera-build-2-us-west-2.vpc.cloudera.com/s3/build/6898
0718/cdh_client/7.x/redhat8/yum/
```

2. Install the Solr client packages you need.

   For example, on RHEL:

   ```
   yum install solr-client -y
   yum install solr-crunch-client -y
   yum install solr-mapreduce-client -y
   yum install hbase-solr-client -y
   ```

   For more information, see *Installing the client packages*.
3. Copy the Solr configuration from a cluster host to the client node.

   For example, copy /etc/solr/conf from a host that runs a Solr role, including solr-env.sh and related files referenced by solrctl.

   ```
   # Example only — confirm paths with your cluster administrator
   scp -r <solr-host>:/etc/solr/conf /etc/solr/conf
   ```
4. Verify the installation.

   Ensure SOLR_ZK_ENSEMBLE is set in /etc/solr/conf/solr-env.sh, then run a solrctl command. For example:

   ```
   solrctl collection --list
   ```

   On Kerberized clusters, run kinit and use solrctl --jaas as described in *Enabling Solr clients to authenticate with a secure Solr*.

   On clusters with Solr HTTPS or TLS-enabled Solr ZooKeeper, configure additional client settings as described in the Search security documentation before you run solrctl or indexer tools from the edge node.

Related Information

Downloading and configuring the client packages

Installing the client packages

Enabling Solr clients to authenticate with a secure Solr