Cloudera Runtime 7.1.0

Using Apache Phoenix to Store and Access Data

Date published: 2020-02-29
Date modified:

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. 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

Mapping Phoenix schemas to HBase namespaces.....4
Enable namespace mapping..... 4

Associating tables of a schema to a namespace.....5
Associate table in a customized Kerberos environment.....5
Associate a table in a non-customized environment without Kerberos.....6

Using SQL and APIs with Apache Phoenix.....6

Connecting to Apache Phoenix Query Server using the JDBC client..... 7
Connect to PQS directly.....7

Mapping Phoenix schemas to HBase namespaces

You can map a Phoenix schema to an HBase namespace to gain multitenancy features in Phoenix.

Important: You must configure this feature only in a CDP Private Cloud Base deployment. This feature is configured automatically in a CDP Public Cloud deployment.

HBase, the underlying storage engine for Phoenix, has namespaces to support multi-tenancy features. Multitenancy helps an HBase user or administrator to perform access control and quota management tasks. Also, namespaces enable tighter control of where a particular data set is stored on RegionsServers.

Enable namespace mapping

You can enable namespace mapping by configuring a set of properties using Cloudera Manager.

About this task

After you set the properties to enable the mapping of Phoenix schemas to HBase namespaces, reverting the property settings renders the Phoenix database unusable. Test or carefully plan the Phoenix to HBase namespace mappings before implementing them.

Important: Cloudera recommends that you enable namespace mapping. If you decide not to enable this feature, you can skip the following steps.

To enable Phoenix schema mapping to a non-default HBase namespace:

Procedure

1. Go to the HBase service.
2. Click the Configuration tab.
3. Select Scope (Service-Wide) .
4. Locate the HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml property or search for it by typing its name in the Search box.
5. Add the following property values:

   Name: phoenix.schema.isNamespaceMappingEnabled

   Description: Enables mapping of tables of a Phoenix schema to a non-default HBase namespace. To enable mapping of a schema to a non-default namespace, set the value of this property to true. The default setting for this property is false.

   Value: true

   Name: phoenix.schema.mapSystemTablesToNamespace

   Description: With true setting (default): After namespace mapping is enabled with the other property, all system tables, if any, are migrated to a namespace called system. With false setting: System tables are associated with the default namespace.

   Value: true
6. Select Scope Gateway .
7. Locate the HBase Client Advanced Configuration Snippet (Safety Valve) for hbase-site.xml property or search for it by typing its name in the Search box.

8. Add the following property values:

   Name: phoenix.schema.isNamespaceMappingEnabled

   Description: Enables mapping of tables of a Phoenix schema to a non-default HBase namespace. To enable mapping of the schema to a non-default namespace, set the value of this property to true. The default setting for this property is false.

   Value: true

   Name: phoenix.schema.mapSystemTablesToNamespace

   Description: With true setting (default): After namespace mapping is enabled with the other property, all system tables, if any, are migrated to a namespace called system.With false setting: System tables are associated with the default namespace.

   Value: true
9. Enter a Reason for change, and then click Save Changes to commit the changes.
10. Restart the role and service when Cloudera Manager prompts you to restart.

Note: If you do not want to map Phoenix system tables to namespaces because of compatibility issues with your current applications, set the phoenix.schema.mapSystemTablesToNamespace property to false.

Associating tables of a schema to a namespace

Important: You must use this feature only in a CDP Private Cloud Base deployment. This feature is configured automatically in a CDP Public Cloud deployment.

After you enable namespace mapping on a Phoenix schema that already has tables, you can migrate the tables to an HBase namespace. The namespace directory that contains the migrated tables inherits the schema name.

For example, if the schema name is store1, then the full path to the namespace is $hbase.rootdir/data/store1. System tables are migrated to the namespace automatically during the first connection after enabling namespace properties.

Associate table in a customized Kerberos environment

You can run a command to associate a table in a customized environment without Kerberos.

Before you begin

In a Kerberos-secured environment, you must have admin privileges (user hbase) to complete the following task.

Procedure

- Run a command to migrate a table of a schema to a namespace, using the following command syntax for the options that apply to your environment:

```
phoenix-psql
ZooKeeper_hostnames:2181
:zookeeper.znode.parent
:principal_name
:HBase_headless_keytab_location
;TenantId=tenant_Id
;CurrentSCN=current_SCN
-m
schema_name.table_name
```

Associate a table in a non-customized environment without Kerberos

You can run a command to associate a table in a non-customized environment without Kerberos.

Procedure

- Run the following command to associate a table:

```
phoenix-psql ZooKeeper_hostname -m Schema_name.table_name
```

Using SQL and APIs with Apache Phoenix

You can create and interact with Apache HBase tables in the form of typical DDL/DML statements through its standard JDBC API. Apache Phoenix JDBC driver can be easily embedded in any app that supports JDBC.

Apache Phoenix has two kinds of JDBC drivers, *thick* and *thin*. The *thick* driver communicates directly with Apache ZooKeeper and Apache HBase and the *thin* client communicates with Apache HBase through Phoenix Query Server.

To connect to Apache Phoenix using the "thick" JDBC driver, you must use the JDBC URL syntax as shown here:

```
jdbc:phoenix:[comma-separated ZooKeeper Quorum [:port [:hbase root znode [:k
erberos_principal [:the path to kerberos keytab] ] ] ]
```

Example URL

```
jdbc:phoenix:zookeeper1.domain,zookeeper2.domain,
zookeeper3.domain:2181:/hbase:phoenix@EXAMPLE.COM:/etc/security/keytabs/ph
oenix.keytab
```

If you are running your client on the Apache Phoenix cluster, you do not have to specify the ZooKeeper parameter; the ZooKeeper quorum is retrieved automatically from hbase-site.xml configuration file.

You can use the standard JDBC APIs instead of the regular HBase client APIs to create tables, insert data, and query your HBase data.

The following drivers are currently supported

- JDBC driver
- ODBC driver
- Other non-Java drivers (example, Microsoft .Net driver and Python driver)

Obtaining a driver for application development

Based on your application development requirements, you can obtain one of the following drivers:

JDBC driver

Use the /opt/cloudera/parcels/CDH/lib/phoenix/*phoenix-5.0.0.7.2.0.0-128*-client.jar file in the Phoenix server-client repository.

ODBC driver

Use the Database Open Database Connectivity (ODBC) interface to access the operational database. The ODBC driver is provided by Cloudera as an additional download, but you can also use ODBC drivers from third-party providers.

You can download the Phoenix ODBC Driver from here: https://www.cloudera.com/downloads/hdp.html.

Note: You must have a Cloudera Enterprise Support Subscription to download the ODBC driver.

Other non-Java drivers

Other non-JDBC Drivers for Phoenix are available as add-ons and on other websites, but they are not currently supported by Cloudera. You can find compatible client drivers by searching on the web for avatica and the name of an application programming language that you want to use. For example: avatica python. For more information and links to driver download pages, see https://calcite.apache.org/avatica/docs/.

Microsoft .NET Driver

Download and install a NuGet package for Microsoft .NET Driver for Apache Phoenix. For more information, see https://www.nuget.org/packages/Microsoft.Phoenix.Client/.

Python Driver for Phoenix

Download the Python Driver for Apache Phoenix from the Apache Phoenix website. For more information, see http://phoenix.apache.org/python.html.

Connecting to Apache Phoenix Query Server using the JDBC client

You can interact with Apache Phoenix using your client and Apache Phoenix Query Server (PQS).

PQS is automatically configured when you create an Operational Database Data Hub cluster. There are two ways in which you can use the thin client to interact with Phoenix:

- Connect to PQS directly
- Connect to PQS using the Apache Knox Gateway (unsupported in this version of the runtime)

Connect to PQS directly

You can connect to the PQS using the JDBC thin client.

About this task

Procedure

- To connect to the PQS directly, you must use the JDBC URL syntax as shown here: jdbc:phoenix:thin:[key=value [;key=value...]]

  You must provide the correct URL, serialization, and authentication key-values to interact with the Phoenix Query Server. Only HTTP access is supported and auto-TLS is not supported.

  Important: Before you can connect to PQS, you must set the authentication, avatica-user, avatica-password, truststore, and truststore-password parameters in your client URL as described here: Client Reference. You have to use the certificate configured in PQS.

Example

jdbc:phoenix:thin:url=http://localhost:8765;serialization=PROTOBUF; authentication=SPENGO;principal=pqs@EXAMPLE.COM;keytab=/etc/security/keytabs/phoenix.keytab