Alibaba Cloud ApsaraDB for HBase

HBase Standard Edition

Issue: 20200521

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 **Notice:** If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings** > **Network** > **Set network type**.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
Courier font	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	switch {active|stand}

Contents

1 Product introduction

ApsaraDB for HBase Standard Edition is developed based on open source HBase. Alibaba Cloud has deeply optimized the stability and capability for operations and maintenance of ApsaraDB for HBase. Compared with HBase Community Edition, ApsaraDB for HBase Standard Edition is more stable and much easier to manage and maintain. In addition, it is completely compatible with HBase Community Edition. To compare ApsaraDB for HBase Standard Edition with other editions, see Benefits. ApsaraDB for HBase Standard Edition has two versions: V1.1 and V2.0.

ApsaraDB for HBase V1.1

ApsaraDB for HBase V1.1 is developed based on HBase Community Edition V1.1.2.

ApsaraDB for HBase V2.0

ApsaraDB for HBase V2.0 is a new distribution developed based on HBase Community Edition V2.0.0 released in 2018. ApsaraDB for HBase V2.0 has higher stability and performance than HBase Community EditionApsaraDB for HBase V1.1 and V2.0 comparison

2 Access preparation

Currently, HBase supports both internal access and Internet access. Internal network access is more stable and typically used in production. Internet access has a higher latency and is more suitable for development and testing purposes. For more information about how to access over the Internet, see Solutions for accessing ApsaraDB for HBase over a public network.

Step 1: Prepare an Elastic Compute Service (ECS) instance

Classic network

In the classic network, you can purchase an ECS instance in any zone of the region where ApsaraDB for HBase is deployed, and set a whitelist to connect to ApsaraDB for HBase.

For example, if ApsaraDB for HBase is deployed in Zone B of the China (Hangzhou) region , you can create an ECS instance in one of the following zones of the same region: Zone B, Zone C, Zone D, and Zone E.

Virtual Private Cloud (VPC)

In a VPC network, you can create an ECS instance in any zone if the ECS instance and ApsaraDB for HBase are in the same VPC. You can connect the ECS instance and ApsaraDB for HBase to the same VPC network by creating VSwitches in different zones.

1. First, create a VPC network in this region in the VPC console.

   For example, if you are in Hangzhou, create a VPC named `myvpc` in the China (Hangzhou) region.
2. In this VPC network, create a VSwitch in the zone where ApsaraDB for HBase is deployed.

   For example, if the HBase is deployed in China (Hangzhou) Zone E, we can create a VSwitch named `switch-zone-e` in Zone E of the same VPC network, and then specify a CIDR block as needed.
3. Set VPC to `myvpc` and `switch-zone-e` on the buy page of ApsaraDB for HBase.

4. Create a VSwitch that is used by the ECS instance.

   a. If you can purchase an ECS instance in the same zone where ApsaraDB for HBase is deployed such as Zone E, go to Step 5.

   b. If you cannot purchase an ECS instance in the same zone where ApsaraDB for HBase is deployed, check the available zones for the ECS instance, for example, Zone B. Create the corresponding VSwitch switch-zone-b in myvpc.

5. Create an ECS instance.

   a. If you select the same zone where ApsaraDB for HBase is deployed such as Zone E, you can purchase an ECS instance directly. Select the same VPC and VSwitch as ApsaraDB for HBase.

   b. If the ECS instances is not in the same zone, follow the previous step to select the zone. For example, create an ECS instance in Zone B and select the VPC myvpc and switch-zone-b that are previously created.

Step 2: Enable the whitelist

After creating the ECS instance, you need to add the internal IP address of the ECS instance to the whitelist in ApsaraDB for HBase. For more information, see Configure a whitelist.

Connectivity test

For more information, see Connectivity test.

3 Download the ApsaraDB for HBase clients

The server end of Alibaba Cloud ApsaraDB for HBase is customized by Alibaba Cloud, but the client is fully compatible with the community edition. However, we provide a dedicated HBase client based on the features of Alibaba Cloud products and make it open source. The Maven JAR package is uploaded to the Maven central repository, which means that you can download and use it anywhere. We do not change any syntax, but just provide some features and facilitate operations and maintenance. You can use the client provided by Alibaba Cloud ApsaraDB for HBase or open source client provided by the community.

Notes: 1. To access ApsaraDB for HBase over the Internet, you must use the client provided by Alibaba Cloud. To access HBase Performance-enhanced Edition, you must use the client provided by Alibaba Cloud or an additional plug-in. For more information, see Install the SDK for Java.

You can use the following methods to obtain the client of ApsaraDB for HBase:

Maven dependency

The ApsaraDB for HBase client of version 1.x corresponds to the HBase cluster of version 1.x. The ApsaraDB for HBase client of version 2.x corresponds to the HBase cluster of version 2.x. We recommend that you do not connect an HBase client of version 1.x to an HBase server of version 2.x unless you cannot upgrade the client. **If you connect an HBase server of version 1.x by using an HBase client of version 2.x, compatibility issues may occur.**

ApsaraDB for HBase version 1.x

```
<dependency>
  <groupId>com.aliyun.hbase</groupId>
  <artifactId>alihbase-client</artifactId>
  <version>1.1.10</version>
</dependency>
```

ApsaraDB for HBase version 2.x

```
<dependency>
  <groupId>com.aliyun.hbase</groupId>
  <artifactId>alihbase-client</artifactId>
  <version>2.0.5</version>
</dependency>
```

Download the tar package

Tar package:

```
wget http://public-hbase.oss-cn-hangzhou.aliyuncs.com/installpackage/alihbase-1.1.3-bin.tar.gz
```

Appendix

Release notes of clients

ApsaraDB for HBase version 1.x

- 1.1.1
  - Use version 1.8 for compilation.
  - Supports hybrid access such as Internet access and classic network access to a VPC network. You must configure the domain name provided by Alibaba Cloud ApsaraDB for HBase.
- 1.1.2
  - Use version 1.7 for compilation.
  - Hbase-client supports both guava12.0.1 and guava22.0 packages. Currently, the default dependency of hbase-client is guava12.0.1. You can remove the dependency as follows and then specify the guava dependency.

```
<dependency>
   <groupId>com.aliyun.hbase</groupId>
 <artifactId>alihbase-client</artifactId>
 <version>1.1.2</version>
   <exclusions>
     <exclusion>
        <artifactId>com.google.guava</artifactId>
        <groupId>guava</groupId>
     </exclusion>
  </exclusions>
</dependency>

<dependency>
   <artifactId>com.google.guava</artifactId>
 <groupId>guava</groupId>
 <version>22.0</version>
</dependency>
```

- 1.1.3
  - Use version 1.7 for compilation.
  - Optimized the Internet access feature to support access to multiple clusters over the Internet. For example, you can use CopyTable to synchronize data between two ApsaraDB for HBase clusters over the Internet.
  - . Fixed dependency compatibility issues for the Phoenix interface.
- 1.1.3.2
  - Use version 1.7 for compilation.
  - Based on the latest version 1.1.3 that is without security capability, ApsaraDB for HBase for Solr client module is supported.
  - Note that this version only supports the wget tar package. It is used for the command line operations of HBase for Solr. You can still use the jar package of version 1.1.3 for other HBase API operations.
- 1.1.4
  - Use version 1.7 for compilation.
  - Supported exporting data to OSS.
  - Note that this version only supports the wget tar package.
- 1.1.5
  - Use version 1.7 for compilation.
  - The HAS security feature is supported. Use the jdk8 runtime environment for security because the HAS security function only supports jdk8.
  - Note that this version only supports the wget tar package.
- 1.1.6
  - Use version 1.7 for compilation.
  - Added the HBase for Solr index management module.
  - Note that this version only supports the wget tar package.
- 1.1.8
  - Use version 1.7 for compilation.
  - Fixed the bugs caused by OOM (Out Of Memory) errors in clients.
- 1.1.9
  - Supported connecting to ApsaraDB for HBase Performance-enhanced Edition.

- 1.1.10
  - Supported connecting to the primary and secondary nodes in the same region.

HBase of version 2.x

- 2.0.0
  - Use version 1.8 for compilation.
  - Supported connecting to HBase version 2.0.0 over the Internet.
- 2.0.1
  - Use version 1.8 for compilation.
  - Supported the HAS security feature.
  - Note that this version only supports the wget tar package.
- 2.0.2
  - Use version 1.8 for compilation.
  - The HAS security feature is no longer supported.
  - Hot and cold data separation is optimized to make the automatic synchronization process transparent to applications.
- 2.0.3
  - Use version 1.8 for compilation.
  - Fixed the bugs caused by missing dependencies.
- 2.0.4
  - Supported connecting to ApsaraDB for HBase Performance-enhanced Edition.
- 2.0.5
  - Supported connecting to the primary and secondary nodes in the same region.

4 Solutions for accessing ApsaraDB for HBase over a public network

Currently, you can access ApsaraDB for HBase over a public or internal network free of charge.

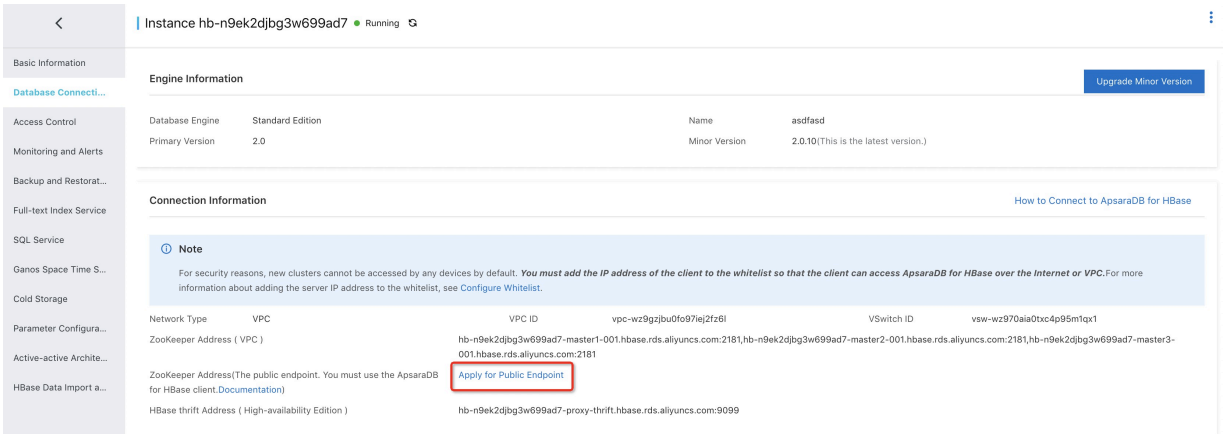
Access ApsaraDB for HBase over a public network ouside Alibaba Cloud

- This solution provides convenience for development and testing. In most cases, the test and development environment is offline, such as a personal computer.
- Customers can use ApsaraDB for HBase to synchronize data from on-premises databases to Alibaba Cloud.

Note: If you access ApsaraDB for HBase over a public network, Alibaba Cloud does not guarantee that the response latency and QPS can meet your requirements.

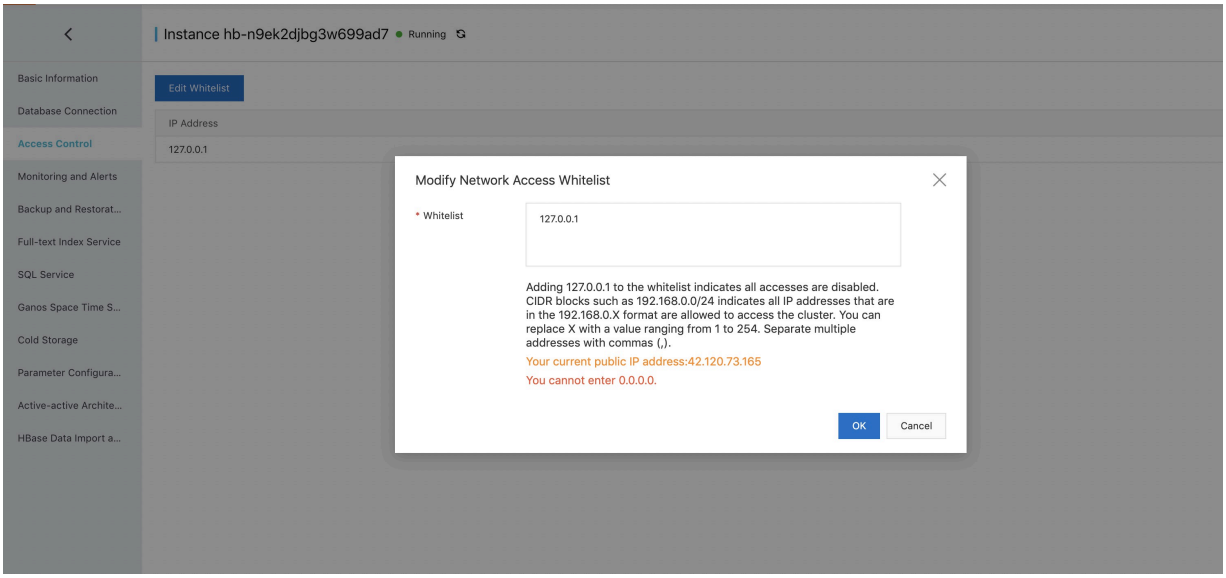
To access ApsaraDB for HBase over a public network, follow these steps:

1. Apply for a public endpoint.



ZooKeeper (ZK): use an endpoint that contains proxy-pub.

2. Configure the whitelist.



Access ApsaraDB for HBase in a VPC network from a classic network

- This solution is designed for data migration.

  This feature currently is unavailable in the ApsaraDB for HBase console. To use this feature, you must submit a ticket to the product team.

The preceding solutions require you to use the client customized by Alibaba Cloud.

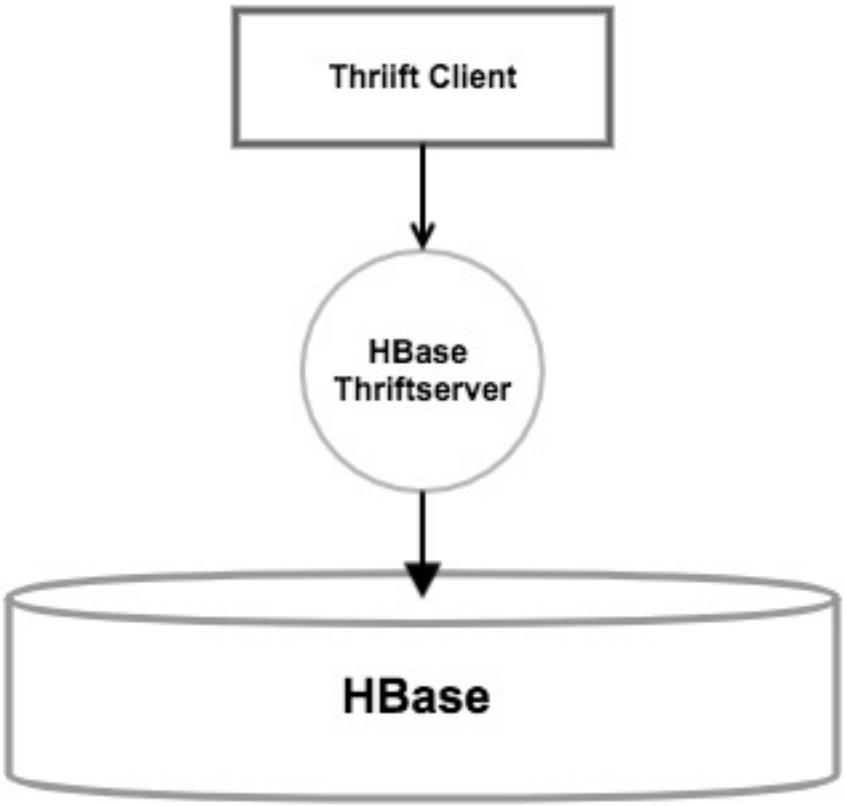
For more information about the Maven and tar packages for ApsaraDB for HBase, see: Download the ApsaraDB for HBase client

Virtual Private Network (VPN) solution

You can connect to the cluster by establishing a VPN from your office network to the VPC network.

5 Multi-language support (thrift)

Thrift allows you to create clients to access ApsaraDB for HBase using multiple languages. The supported language packages are listed on the official website of Apache Thrift: C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml, Delphi, and other languages. The Thrift client sends requests to the ThriftServer of ApsaraDB for HBase through the Thrift protocol. The ThriftServer then reroutes the requests to the storage servers of ApsaraDB for HBase. The following figure shows the architecture.



Note: If you use ApsaraDB for HBase Performance-enhanced Edition, see APIs for multiple languages.

Activate ApsaraDB for HBase ThriftServer:

Click Activate ThriftServer service (ThriftServer high-availability version) on the cluster basic information page of the ApsaraDB for HBase console to obtain an endpoint in the format of host:port.

2. Connect to ApsaraDB for HBase

2.1. Use the Thrift client

Typically, users use Python and PHP to connect to ApsaraDB for HBase. In the following example, PHP is used to access ApsaraDB for HBase.

2.1.1. Use PHP to access ApsaraDB for HBase through Thrift.

The version of Thrift used by ApsaraDB for HBase is 0.9.0. Therefore, we recommend that you use Thrift 0.9.0 to create a client. Click here to download Thrift 0.9.0. The downloaded source code package will be used later. You must install the Thrift compiling environment first. For more information, see Thrift official website.

Run the following command to check the version of Thrift installed.

```
thrift --version
```

2.1.2. Obtain the HBase Thrift file for the client.

Click here to download the Hbase.thrift file used to access ApsaraDB for HBase. ApsaraDB for HBase uses the thrift1 protocol.

The command is as follows:

```
 thrift --gen <language> Hbase.thrift
```

The language parameter in this command specifies the acronym of a language. The common languages are as follows:

```
thrift --gen php Hbase.thrift
thrift --gen cpp Hbase.thrift
thrift --gen py Hbase.thrift
```

Run the thrift --gen php Hbase.thrift command to obtain the required function package file gen-php under the directory.

```
thrift git:(last_dev) # ll
total 56
-rw-r--r--  1 xuanling.gc  staff    24K  3  5 15:06 Hbase.thrift
drwxr-xr-x  3 xuanling.gc  staff    96B  8  1 16:03 gen-php
```

Put the Thrift folder and the gen-php file in the src directory of the workload logic code. The Thrift folder is under /lib/php/lib of the downloaded Thrift source code folder. The directory is as follows with your own client.php code:

```
[root@xxxxxxxxxxx thrift_client]# ll
total 12
-rw-r--r--  1 zookeeper games 2743 Aug  2 11:16 client.php
drwxr-xr-x  3 zookeeper games 4096 Aug  2 01:22 gen-php
drwxr-xr-x 12 zookeeper games 4096 Aug  2 01:22 Thrift
```

Write PHP code.

The preceding Thrift folder and the gen-php folder can be renamed as needed. In this example, we keep the default names to make it easy for you to distinguish the folders. The following PHP code shows the program that creates a table named new in ApsaraDB for HBase.

```
<? php
ini_set('display_errors', E_ALL);
$GLOBALS['THRIFT_ROOT'] = "/root/thrift_client";
/* Dependencies. In the proper order. */
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Transport/TTransport.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Transport/TSocket.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Protocol/TProtocol.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Protocol/TBinaryProtocol.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Protocol/TBinaryProtocolAccelerated.
php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Transport/TBufferedTransport.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Type/TMessageType.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Factory/TStringFuncFactory.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/StringFunc/TStringFunc.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/StringFunc/Core.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Type/TType.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Exception/TException.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Exception/TTransportException.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/Thrift/Exception/TProtocolException.php';

require_once $GLOBALS['THRIFT_ROOT'] . '/gen-php/Hbase/Types.php';
require_once $GLOBALS['THRIFT_ROOT'] . '/gen-php/Hbase/Hbase.php';

use Thrift\Protocol\TBinaryProtocol;
use Thrift\Transport\TBufferedTransport;
use Thrift\Transport\TSocket;
use Hbase\HbaseClient;
use Hbase\ColumnDescriptor;
use Hbase\Mutation;

$host='hb-bp12pt6alr1788y35-001.hbase.rds.aliyuncs.com';
$port=9099;

$socket = new TSocket($host, $port);

$socket->setSendTimeout(10000); // Request sending timeout in milliseconds.
$socket->setRecvTimeout(20000); // Response reception timeout in milliseconds.
```

```
$transport = new TBufferedTransport($socket);
$protocol = new TBinaryProtocol($transport);
$client = new HbaseClient($protocol);

$transport->open();

#### List tables ####
echo "----list tables----\n";
$tables = $client->getTableNames();
foreach ($tables as $name) {
    var_dump($tables);
}

$tablename='new';
#### Write data ####
echo "----write data----\n";
$row = 'key';
$value = 'value';
$atrribute = array();
$mutations = array(
    new Mutation(array(
        'column' => 'info:cn1',
        'value' => $value
    )),
);

try {
    $client->mutateRow($tablename, $row, $mutations, $atrribute);
} catch (Exception $e) {
    var_dump($e);// Generate a debugging log
}

### Read data ####
echo "---read data---\n";
$result = $client->getRow($tablename, $row, $atrribute);
var_dump($result);

### Delete data ####
echo "---delete data---\n";
$client->deleteAllRow($tablename, $row, $atrribute);
echo "---get data---\n";
$result = $client->getRow($tablename, $row, $atrribute);
var_dump($result);

### Scan data ###
$row = 'ID1';
$value = 'v1';
$mutations = array(
   new Mutation(array(
        'column' => 'info:c1',
        'value' => $value
    )),
);
try {
    $client->mutateRow($tablename, $row, $mutations, $atrribute);
} catch (Exception $e) {
    var_dump($e);
}

$row = 'ID2';
$value = 'v2';
$mutations = array(
   new Mutation(array(
        'column' => 'info:c1',
```

```
        'value' => $value
    )),
);
try {
    $client->mutateRow($tablename, $row, $mutations, $atrribute);
} catch (Exception $e) {
    var_dump($e);
}

$row = 'ID3';
$value = 'v3';
$mutations = array(
  new Mutation(array(
        'column' => 'info:c1',
        'value' => $value
    )),
);
try {
    $client->mutateRow($tablename, $row, $mutations, $atrribute);
} catch (Exception $e) {
    var_dump($e);
}

echo 'prefix scan';
$scan = $client->scannerOpenWithPrefix($tablename, 'ID', null, null);

$nbRows = 100;
$arr = $client->scannerGetList($scan, $nbRows);
echo 'count of result :'.count($arr)."\n";
var_dump($arr);
foreach ($arr as $k => $TRowResult) {
  echo "\trow:$TRowResult->row\tcolumns(array):";
  foreach ($TRowResult->columns as $key => $value) {
    echo "key:$key\tvalue:$value->value\ttimestamp:$value->timestamp\n";
  }
}

echo 'range scan';
$scan = $client->scannerOpenWithStop($tablename, 'ID0', 'ID2', null, null);

$nbRows = 100;
$arr = $client->scannerGetList($scan, $nbRows);
echo 'count of result :'.count($arr)."\n";
var_dump($arr);
foreach ($arr as $k => $TRowResult) {
  echo "\trow:$TRowResult->row\tcolumns(array):";
  foreach ($TRowResult->columns as $key => $value) {
    echo "key:$key\tvalue:$value->value\ttimestamp:$value->timestamp\n";
  }
}

### Increment operation ###
echo "do increment on a new row";
$row = 'ID4';
try {
$newCount = $client->atomicIncrement($tablename, $row, 'info:c1', 1234);
} catch (Exception $e) {
    var_dump($e);
}

echo "new count $newCount\n";

### Write long-type data to ApsaraDB for HBase ###
```

```
$row = 'ID5';
$value = pack("J", 4567);

$mutations = array(
   new Mutation(array(
       'column' => 'info:c1',
       'value' => $value
   )),
);
try {
    $client->mutateRow($tablename, $row, $mutations, null);
} catch (Exception $e) {
    var_dump($e);
}

echo "---read data and print it as long ---\n";
$result = $client->getRow($tablename, $row, null);

foreach ($result[0]->columns as $key => $value) {
  $count = unpack("J*mycount", $value->value);
  var_dump($count);
}

? >
```

The execution result is as follows:

```
[root@xxxxxxxxxxx thrift_client]# php client.php
----list tables----
array(1) {
  [0]=>
  string(3) "new"
}
----write data----
---read data---
array(1) {
  [0]=>
  object(Hbase\TRowResult)#8 (3) {
    ["row"]=>
    string(3) "key"
    ["columns"]=>
    array(1) {
      ["info:cn1"]=>
      object(Hbase\TCell)#10 (2) {
        ["value"]=>
        string(5) "value"
        ["timestamp"]=>
        int(1533179795969)
      }
    }
    ["sortedColumns"]=>
    NULL
  }
}
---delete data---
---get data---
array(0) {
}
```

2.2. Use Python to connect to ApsaraDB for HBase

Some users may access the ThriftServer by using a third-party library in Python such as HappyBase. The third-party library includes Thrift. For more information, see Installation guide. There are various of Python libraries, and you can install Thrift and access the HBase Thrift library through pip. Assuming that you have installed Python and pip, the execution process is as follows:

```
pip install thrift // Install the latest version of Thrift by default.
pip install hbase-thrift // Install the HBase Thrift interface library.
```

You can write the code to access ApsaraDB for HBase after the preceding two steps are complete.

```
import sys
import time
import os

from thrift import Thrift
from thrift.transport import TSocket, TTransport
from thrift.protocol import TBinaryProtocol
from thrift.protocol.TBinaryProtocol import TBinaryProtocolAccelerated
from hbase import ttypes
from hbase.Hbase import Client, ColumnDescriptor, Mutation

def printRow(entry):
  print "row: " + entry.row + ", cols:",
  for k in sorted(entry.columns):
    print k + " => " + entry.columns[k].value,
  print


transport = TSocket.TSocket('hb-bp12pt6alr1788y35-001.hbase.rds.aliyuncs.com', 9099)
transport = TTransport.TBufferedTransport(transport)
protocol = TBinaryProtocol.TBinaryProtocolAccelerated(transport)
client = Client(protocol)
transport.open()

print "---list table--"
print client.getTableNames()

table="new"
row="key"

print "---write data---"
mutations = [Mutation(column="info:cn1", value="value")]
client.mutateRow(table, row, mutations)

print "---get data----"
printRow(client.getRow(table, row)[0])

print "---delete data---"
client.deleteAllRow(table, row)
print "---end----"
```

```
transport.close()
```

The execution result is as follows:

```
[root@Test ~]# python Hbase_client.py
---list table--
['new']
---write data---
---get data----
row: key, cols: info:cn1 => value
---delete data---
---end----
```

Access ApsaraDB for HBase using Go

Download the optimized Thrift access package and decompress it into $GOPATH/src directory. You can also choose to download the original version from https://github.com/sdming/goh.

```
wget http://public-hbase.oss-cn-hangzhou.aliyuncs.com/thrift/goh.tar.gz
tar -xvzf goh.tar.gz
mv github.com $GOPATH/src
```

For more information about the sample code, see $GOPATH/src/github.com/sdming/goh/demo/client.go that contains DDL, and sample code for reading and writing data.

6 Access ApsaraDB for HBase HDFS

In some scenarios such as using bulk loads to import data to ApsaraDB for HBase, you must enable the HDFS ports for your ApsaraDB for HBase cluster.

- Note: If you enable the HDFS ports, Alibaba Cloud is not responsible for any data loss in HDFS caused by user mistakes. Make sure that you are familiar with the HDFS operations.
- Contact the ApsaraDB for HBase Q&A DingTalk group to activate HDFS. Caution: After you activate HDFS, your cluster is exposed to malicious attacks, which may cause performance instability or even data loss. To ensure data security, you are not allowed to activate HDFS as needed. You must contact our ApsaraDB for HBase Q&A DingTalk group to activate HDFS. We will disable it after you complete your tasks.

Verification

- Test the HDFS ports by using an HDFS client to connect to the HDFS cluster in ApsaraDB for HBase.
- Create an Hadoop client configuration folder named conf. If the folder already exists, you do not need to create a new one.
- Add the following HDFS configuration files to the folder. For more information about how to set the hosts {hbase-header-1-host} and {hbase-header-1-host}, consult the ApsaraDB for HBase Q&A DingTalk group.

  - core-site.xml

```
<configuration>
 <property>
   <name>fs.defaultFS</name>
   <value>hdfs://hbase-cluster</value>
 </property>
</configuration>
```

- hdfs-site.xml

```
<configuration>
<property>
      <name>dfs.nameservices</name>
      <value>hbase-cluster</value>
</property>
 <property>
 <name>dfs.client.failover.proxy.provider.hbase-cluster</name>
 <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider
</value>
</property>
<property>
 <name>dfs.ha.automatic-failover.enabled.hbase-cluster</name>
```

```
 <value>true</value>
</property>
<property>
   <name>dfs.namenode.http-address.hbase-cluster.nn1</name>
   <value>{hbase-header-1-host}:50070</value>
</property>
 <property>
   <name>dfs.namenode.http-address.hbase-cluster.nn2</name>
   <value>{hbase-header-2-host}:50070</value>
 </property>
<property>
      <name>dfs.ha.namenodes.hbase-cluster</name>
      <value>nn1,nn2</value>
</property>
<property>
      <name>dfs.namenode.rpc-address.hbase-cluster.nn1</name>
      <value>{hbase-header-1-host}:8020</value>
</property>
<property>
      <name>dfs.namenode.rpc-address.hbase-cluster.nn2</name>
      <value>{hbase-header-2-host}:8020</value>
</property>
</configuration>
```

- Add conf to the classpath of the Hadoop client.
- Test the HDFS ports by reading data from HDFS and writing data to HDFS.

```
echo "hdfs port test"  >/tmp/test
hadoop dfs -put /tmp/test  /
hadoop dfs -cat /test
```

7 Use Hive to read and write an ApsaraDB for HBase table

ApsaraDB for HBase allows you to use Hive to read and write a table.

- Before you perform this task, you must add the IP addresses of all the nodes in the Hadoop cluster where Hive runs to the whitelist of ApsaraDB for HBase.
- Retrieve the ZooKeeper endpoint of ApsaraDB for HBase. You can find the endpoint in the ApsaraDB for HBase console.

Modify the configuration

- Switch to the Hive configuration directory `/etc/ecm/hive-conf/`.
- Open the `hbase-site.xml` file and set the value of the `hbase.zookeeper.quorum` parameter to the ZooKeeper endpoint of ApsaraDB for HBase.

```
<property>
   <name>hbase.zookeeper.quorum</name>
   <value>hb-xxx-001.hbase.rds.aliyuncs.com,hb-xxx-002.hbase.rds.aliyuncs.com,hb-xxx-003.hbase.rds.aliyuncs.com</value>
</property>
```

Read and write an ApsaraDB for HBase table in Hive

If you have not created a table in ApsaraDB for HBase, you can create an external table for ApsaraDB for HBase in Hive.

- Launch Hive CLI
- 

- Create a Hive external table for ApsaraDB for HBase.

```
CREATE TABLE hive_hbase_table(key int, value string)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,cf1:val")
TBLPROPERTIES ("hbase.table.name" = "hive_hbase_table", "hbase.mapred.output.outputtable" = "hive_hbase_table");
```

- Insert data to the table in Hive.

```
insert into hive_hbase_table values(212,'bab');
```

```
hive> insert into hive_hbase_table values(212,'bab');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_20181014173030_a0e99198-9aa5-4d29-b011-dc7b36365a20
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1536221485395_0084, Tracking URL = http://emr-header-1.cluster-74778:20888/proxy/application_1536221485395_0084/
Kill Command = /usr/lib/hadoop-current/bin/hadoop job  -kill job_1536221485395_0084
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2018-10-14 17:30:40,833 Stage-3 map = 0%,  reduce = 0%
2018-10-14 17:30:47,252 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 3.66 sec
MapReduce Total cumulative CPU time: 3 seconds 660 msec
Ended Job = job_1536221485395_0084
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 3.66 sec   HDFS Read: 11867 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 660 msec
OK
Time taken: 17.385 seconds
```

- A same table is created in ApsaraDB for HBase, and the same data is inserted into the table.

```
[root@iZbp16ku9i9clejitib6dzZ ~]# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apps/t-apsara-hbase-1.4.6.3/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/apps/t-emr-hadoop-2.7.2.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.4.6.3, r89ac288a5add370c07548ec3ce25f6e1f3210d23, Fri Jul  6 14:13:46 CST 2018

hbase(main):001:0> list
TABLE
A_TABLE
BAKE
BASE_TABLE
B_IDX
B_IDX1
B_IDX2
DEFAULT.TEST_IDX
MY_TABLE
PROD_METRICS
SYSTEM.MUTEX
SYSTEM:CATALOG
SYSTEM:FUNCTION
SYSTEM:SEQUENCE
SYSTEM:STATS
[illegible]
hive_hbase_table
[illegible]
tv
18 row(s) in 0.2110 seconds
```

```
hbase(main):004:0* scan 'hive_hbase_table'
ROW                                                    COLUMN+CELL
 212                                                   column=cf1:val, timestamp=1539509446271, value=bab
1 row(s) in 0.0950 seconds
```

- Insert data into the table in ApsaraDB for HBase, and show the table in Hive.

```
hbase(main):005:0> put 'hive_hbase_table','132','cf1:val','acb'
0 row(s) in 0.0430 seconds
```

Show the table in Hive:

```
hive> select * from hive_hbase_table;
OK
132     acb
212     bab
Time taken: 0.273 seconds
```

- If you delete the external table in Hive, the associated table in ApsaraDB for HBase is also deleted.

```
hive>
    >
    >
    > drop table hive_hbase_table;
OK
Time taken: 6.307 seconds
```

Show the table in ApsaraDB for HBase. An error message is displayed indicating that the table does not exist.

```
hbase(main):008:0* scan 'hive_hbase_table'
ROW                                                    COLUMN+CELL

ERROR: Unknown table hive_hbase_table!
```

If you have already created a table in ApsaraDB for HBase before you create the external table, you can associate the external table with the table in ApsaraDB for HBase. In this case, if you delete the external table in Hive, the table in ApsaraDB for HBase will not be deleted.

- Create a table in ApsaraDB for HBase, and insert data into the table.

```
hbase(main):020:0* create 'hbase_table','f'
0 row(s) in 1.3010 seconds

=> Hbase::Table - hbase_table
hbase(main):021:0> put 'hbase_table','1122','f:col1','hello'
0 row(s) in 0.0190 seconds

hbase(main):022:0> put 'hbase_table','1122','f:col2','hbase'
0 row(s) in 0.0110 seconds
```

- Create an external table in Hive, associate it with the table in ApsaraDB for HBase, and show the external table.

```
hive> create external table hbase_table(key int, col1 string, col2 string)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES ("hbase.columns.mapping" = "f:col1,f:col2")
    > TBLPROPERTIES("hbase.table.name" = "hbase_table", "hbase.mapred.output.outputtable" = "hbase_table");
OK
Time taken: 0.129 seconds
hive> select * from hbase_table;
OK
1122    hello   hbase
Time taken: 0.181 seconds, Fetched: 1 row(s)
hive>
```

- If you delete the external table in Hive, the table in ApsaraDB for HBase will not be deleted.

```
hive> drop table hbase_table;
OK
Time taken: 0.102 seconds
hive>
```

```
hbase(main):023:0> scan 'hbase_table'
ROW                                                    COLUMN+CELL
 1122                                                  column=f:col1, timestamp=1539510170256, value=hello
 1122                                                  column=f:col2, timestamp=1539510181752, value=hbase
1 row(s) in 0.0160 seconds
```

For more information about how to use Hive to interact with ApsaraDB for HBase, visit https://cwiki.apache.org/confluence/display/Hive/HBaseIntegration.

If you are using a Hive service running in a MapReduce cluster deployed on an Elastic Compute Service (ECS) instance, you can use Hive to interact with ApsaraDB for HBase in the same way as using a Hive service running in an E-MapReduce cluster. However, some of the configurations in the Hive configuration file hbase-site.xml are different from those of ApsaraDB for HBase. In this case, you only need to ensure that the `hbase.zookeeper.quorum` settings are consistent.

8 Modify configurations

After you modify the configurations, you must **restart the cluster**. The cluster is restarted by using the rolling restart method, which has minor impacts on the cluster if the cluster size is small.

ApsaraDB for HBase generates logs after each modification.

Subsequent optimization: You can change some parameters without restarting the cluster.

The following table lists some of the user-configurable parameters. For more information, check the configurations in the console.

Parameter	Default	Description
hbase.hregion.majorcompaction	604800000	The default value is 604800000 milliseconds (seven days). You can set the value to 0 to disable automatic major compaction . Subsequent products of ApsaraDB for HBase will support triggering major compaction periodically or within a time period.
hbase.ipc.server.callqueue.read.ratio	0	Adjusts the numbers of read and scan queues to control read/write splitting.
hbase.ipc.server.callqueue.scan.ratio	0	Controls read/write splitting.
hbase.regionserver.global.memstore.lowerLimit	0.3	The proportion of memstore.
hbase.regionserver.global.memstore.size	0.35	The proportion of memstore.
hbase.rpc.timeout	60000	The timeout for sending Remote Procedure Call (RPC) requests, which can be extended.

Parameter	Default	Description
hfile.block.cache.size	0.4	If more read requests are received than scan requests , you can set this parameter to 0.5.

See: