

# 云 HDFS

## 最佳实践

### 产品文档



腾讯云

---

**【版权声明】**

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

---

## 文档目录

### 最佳实践

使用 CHDFS 作为 Druid 的 Deep storage

原生 HDFS 数据迁移到腾讯云 CHDFS

使用 DataX 导入或导出 CHDFS

CDH 配置 CHDFS 指引

CHDFS Ranger 权限体系解决方案

# 最佳实践

## 使用 CHDFS 作为 Druid 的 Deep storage

最近更新时间：2022-03-30 09:30:26

### 环境依赖

- [CHDFS\\_JAR](#)。
- Druid 版本：Druid-0.12.1。

### 下载与安装

#### 获取 CHDFS JAR

在官方 Github 上下载 [CHDFS\\_JAR](#)。

#### 安装 CHDFS JAR

使用 CHDFS 作为 Druid 的 Deep Storage，需要借助 Druid-hdfs-extension 实现。

下载 CHDFS JAR 后，将 `chdfs_hadoop_plugin_network-1.7.jar` 拷贝到 Druid 安装路径 `extensions/druid-hdfs-storage` 以及 `hadoop-dependencies/hadoop-client/2.x.x` 下。

### 使用方法

#### 配置修改

1. 修改 Druid 安装路径的 `conf/druid/_common/common.runtime.properties` 文件，将 hdfs 的 extension 加入到 `druid.extensions.loadList` 中，同时指定 hdfs 为 Druid 的 deep storage，而路径则填写为 CHDFS 的路径：

```
properties
druid.extensions.loadList=["druid-hdfs-storage"]
druid.storage.type=hdfs
druid.storage.storageDirectory=ofs://<mountpoint>/<druid-path>
```

2. 在 `conf/druid/_common/` 这个目录下，新建一个 hdfs 的配置文件 `hdfs-site.xml`，填入 CHDFS 的配置信息等：

```
<!--?xml version="1.0" encoding="UTF-8"?-->
<!--?xml-stylesheet type="text/xsl" href="configuration.xsl"?-->
<!--
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. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.AbstractFileSystem.ofs.impl</name>
<value>com.qcloud.chdfs.fs.CHDFSDelegateFSAdapter</value>
</property>
<property>
<name>fs.ofs.impl</name>
<value>com.qcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter</value>
</property>
<!--本地 cache 的临时目录，对于读写数据，当内存 cache 不足时会写入本地硬盘，这个路径若不存在会自动创建-->
<property>
<name>fs.ofs.tmp.cache.dir</name>
<value>/data/chdfs_tmp_cache</value>
</property>
<!--appId 用户需要换成自己的 appId，可前往 https://console.tencentcloud.com/cam/cap
i 获取-->
<property>
<name>fs.ofs.user.appid</name>
<value>125000001</value>
</property>
</configuration>
```

上述配置的支持项与 CHDFS 官网文档描述完全一致，详情可参见 [挂载 CHDFS](#) 文档。

## 开始使用

依次启动 Druid 进程，Druid 数据就可加载到 CHDFS 中。

# 原生 HDFS 数据迁移到腾讯云 CHDFS

最近更新时间：2022-03-30 09:30:26

## 准备工作

1. 在腾讯云官网创建 CHDFS 文件系统和 CHDFS 挂载点，配置好权限信息。
2. 通过腾讯云 VPC 环境的 CVM 机器访问创建好的 CHDFS，详情请参见 [创建 CHDFS](#)。
3. 当挂载成功后，打开 hadoop 命令行工具，执行以下命令，验证 CHDFS 功能是否正常。

```
hadoop fs -ls ofs://f4xxxxxxxxxxxxxxxx.chdfs.ap-beijing.myqcloud.com/
```

如果能看到以下类似的输出，则表明云 HDFS 功能一切正常。

```
[hadoop@10 ~]$ hadoop fs -ls ofs://[redacted].chdfs.ap-beijing.myqcloud.com/
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/service/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/service/tez/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]
Found 31 items
drwxr-xr-x - root root 0 2019-12-30 17:03 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/0x
drwxr-xr-x - hadoop hadoop 0 2019-12-30 18:20 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/data
-rw-r--r-- 1 root root 1048576000 2019-12-13 23:20 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/dd_1G
drwxrwxrwx - hadoop hadoop 0 2019-12-18 12:08 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/emr
drwxrwxr-x - root root 0 2019-12-05 17:13 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-dir-0
drwxrwxr-x - root root 0 2019-12-05 17:13 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-dir-1
drwxrwxr-x - root root 0 2019-12-05 17:13 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-dir-2
drwxrwxr-x - root root 0 2019-12-05 17:24 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-dir-3
drwxr-xr-x - root root 0 2019-12-05 17:25 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-dir-4
-rwxrwxr-x 1 root root 0 2019-12-05 17:13 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-file-0
-rw-rw-r-- 1 root root 0 2019-12-05 17:13 ofs://[redacted].chdfs.ap-beijing.myqcloud.com/fuse-file-1
```

## 迁移

当准备工作就绪后，就可以使用 hadoop 社区标准的 Distcp 工具实现全量或者增量的 HDFS 数据迁移，详情请参见 [Distcp 官方指引文档](#)。

### 注意事项

在 hadoop distcp 工具中，提供了一些 CHDFS 不兼容的参数。如果指定如下表格中的一些参数，则不生效。

参数	描述	状态
-p[rbox]	r : replication, b : block-size, a : ACL, x : XATTR	不生效

### 示例说明

1. 当 CHDFS 准备就绪后，执行以下 hadoop 命令进行数据迁移。

```
hadoop distcp hdfs://10.0.1.11:4007/testcp ofs://f4xxxxxxxx-xxxx.chdfs.ap-beijing.myqcloud.com/
```

其中 `f4xxxxxxxx-xxxx.chdfs.ap-beijing.myqcloud.com` 为挂载点域名，需要根据实际申请的挂载点信息进行替换。

2. Hadoop 命令执行完毕后，会在日志中打印出本次迁移的具体详情。如下示例所示：

```
2019-12-31 10:59:31 [INFO ] [main:13300] [org.apache.hadoop.mapreduce.Job:] [Job.java:1385]
Counters: 38
File System Counters
FILE: Number of bytes read=0
FILE: Number of bytes written=387932
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=1380
HDFS: Number of bytes written=74
HDFS: Number of read operations=21
HDFS: Number of large read operations=0
HDFS: Number of write operations=6
OFS: Number of bytes read=0
OFS: Number of bytes written=0
OFS: Number of read operations=0
OFS: Number of large read operations=0
OFS: Number of write operations=0
Job Counters
Launched map tasks=3
Other local map tasks=3
Total time spent by all maps in occupied slots (ms)=419904
Total time spent by all reduces in occupied slots (ms)=0
Total time spent by all map tasks (ms)=6561
Total vcore-milliseconds taken by all map tasks=6561
Total megabyte-milliseconds taken by all map tasks=6718464
Map-Reduce Framework
Map input records=3
Map output records=2
Input split bytes=408
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=179
CPU time spent (ms)=4830
Physical memory (bytes) snapshot=1051619328
```

```
Virtual memory (bytes) snapshot=12525191168
Total committed heap usage (bytes)=1383071744
File Input Format Counters
Bytes Read=972
File Output Format Counters
Bytes Written=74
org.apache.hadoop.tools.mapred.CopyMapper$Counter
BYTESSKIPPED=5
COPY=1
SKIP=2
```



# 使用 DataX 导入或导出 CHDFS

最近更新时间：2022-03-30 09:30:26

## 环境依赖

- [CHDFS\\_JAR](#)。
- DataX 版本：DataX-3.0。

## 下载与安装

### 获取 CHDFS JAR

在官方 Github 上下载 [CHDFS\\_JAR](#)。

### 获取 DataX 软件包

在官方 Github 上下载 [DataX](#)。

### 安装 CHDFS JAR

下载 CHDFS JAR 后，将 `chdfs_hadoop_plugin_network-1.7.jar` 拷贝到 Datax 解压路径 `plugin/reader/hdfsreader/libs/` 以及 `plugin/writer/hdfswriter/libs/` 下。

## 使用方法

### DataX 配置

#### 修改 `datax.py` 脚本

打开 DataX 解压目录下的 `bin/datax.py` 脚本，修改脚本中的 `CLASS_PATH` 变量为如下：

```
CLASS_PATH = ("%s/lib/*:%s/plugin/reader/hdfsreader/libs/*:%s/plugin/writer/hdfswriter/libs/*:..") % (DATAX_HOME, DATAX_HOME, DATAX_HOME)
```

#### 在配置 JSON 文件里配置 `hdfsreader` 和 `hdfswriter`

示例 JSON 如下：

```
{  
  "job": {
```

```
"setting": {
  "speed": {
    "byte": 10485760
  },
  "errorLimit": {
    "record": 0,
    "percentage": 0.02
  }
},
"content": [{
  "reader": {
    "name": "hdfsreader",
    "parameter": {
      "path": "testfile",
      "defaultFS": "ofs://f4xxxxxxxx-hxT9.chdfs.ap-beijing.myqcloud.com/",
      "column": ["*"],
      "fileType": "text",
      "encoding": "UTF-8",
      "hadoopConfig": {
        "fs.AbstractFileSystem.ofs.impl": "com.qcloud.chdfs.fs.CHDFSDelegateFSAdapter",
        "fs.ofs.impl": "com.qcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter",
        "fs.ofs.tmp.cache.dir": "/data/chdfs_tmp_cache",
        "fs.ofs.user.appid": "1250000000"
      }
    },
    "fieldDelimiter": ","
  }
},
  "writer": {
    "name": "hdfswriter",
    "parameter": {
      "path": "/user/hadoop/",
      "fileName": "testfile1",
      "defaultFS": "ofs://f4xxxxxxxx-hxT9.chdfs.ap-beijing.myqcloud.com/",
      "column": [{
        "name": "col",
        "type": "string"
      },
      {
        "name": "col1",
        "type": "string"
      },
      {
        "name": "col2",
        "type": "string"
      }
    ],
    "fileType": "text",
```

```
"encoding": "UTF-8",
"hadoopConfig": {
  "fs.AbstractFileSystem ofs.impl": "com.qcloud.chdfs.fs.CHDFSDelegateFSAdapter",
  "fs ofs.impl": "com.qcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter",
  "fs ofs.tmp.cache.dir": "/data/chdfs_tmp_cache",
  "fs ofs.user.appid": "1250000000"
},
"fieldDelimiter": ":",
"writeMode": "append"
}
}
}]
}
}
```

其中，hadoopConfig 配置为 CHDFS 所需要的配置，defaultFS 填写为 CHDFS 的路径。例如 ofs://f4xxxxxxxxx-hxT9.chdfs.ap-beijing.myqcloud.com/，其他配置同 hdfs 配置项即可。

## 执行数据迁移

将配置文件保存为 hdfs\_job.json，存放到 job 目录下，执行以下命令行：

```
bin/datax.py job/hdfs_job.json
```

观察屏幕正常输出如下：

```
2020-03-09 16:49:59.543 [job-0] INFO JobContainer -
[total cpu info] =>
averageCpu | maxDeltaCpu | minDeltaCpu
-1.00% | -1.00% | -1.00%

[total gc info] =>
NAME | totalGCCount | maxDeltaGCCount | minDeltaGCCount | totalGCtime | maxDeltaGC
Time | minDeltaGCtime
PS MarkSweep | 1 | 1 | 1 | 0.024s | 0.024s | 0.024s
PS Scavenge | 1 | 1 | 1 | 0.014s | 0.014s | 0.014s
2020-03-09 16:49:59.543 [job-0] INFO JobContainer - PerfTrace not enable!
2020-03-09 16:49:59.543 [job-0] INFO StandAloneJobContainerCommunicator - Total 2
records, 33 bytes | Speed 3B/s, 0 records/s | Error 0 records, 0 bytes | All Task
WaitWriterTime 0.000s | All Task WaitReaderTime 0.033s | Percentage 100.00%
2020-03-09 16:49:59.544 [job-0] INFO JobContainer -
任务启动时刻 : 2020-03-09 16:49:48
任务结束时刻 : 2020-03-09 16:49:59
任务总计耗时 : 11s
任务平均流量 : 3B/s
记录写入速度 : 0rec/s
```

---

读出记录总数 : 2  
读写失败总数 : 0

# CDH 配置 CHDFS 指引

最近更新时间：2022-03-30 09:30:26

## 简介

CDH（Cloudera's Distribution, including Apache Hadoop）是业界流行的 Hadoop 发行版本。本文指导如何在 CDH 环境下使用腾讯云 CHDFS 服务，以实现大数据计算与存储分离，提供灵活及低成本的大数据解决方案。

CHDFS 大数据组件支持情况如下：

组件名称	CHDFS 大数据组件支持情况	服务组件是否需要重启
Yarn	支持	重启 NodeManager
Yarn	支持	重启 NodeManager
Hive	支持	重启 HiveServer 及 HiveMetastore
Spark	支持	重启 NodeManager
Sqoop	支持	重启 NodeManager
Presto	支持	重启 HiveServer 及 HiveMetastore 和 Presto
Flink	支持	否
Impala	支持	否
EMR	支持	否
自建组件	后续支持	无
HBase	不推荐	无

## 版本依赖

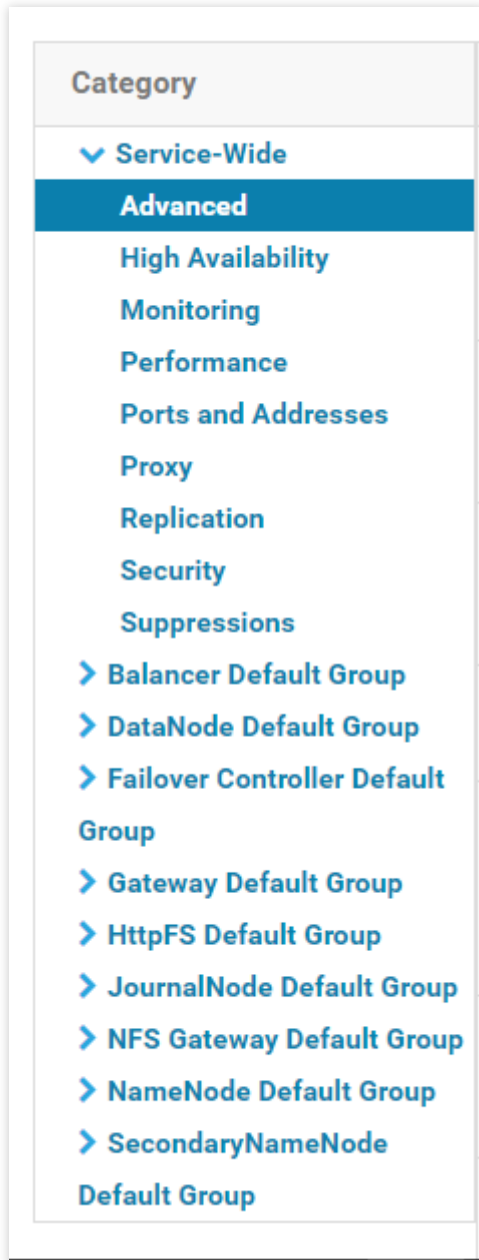
本文依赖的组件版本如下：

- CDH 5.16.1
- Hadoop 2.6.0

## 使用方法

### 存储环境配置

1. 登录 CDH 管理页面。
2. 在系统主页，选择**配置** > **服务范围** > **高级**，进入高级配置代码段页面，如下图所示：



3. 在 Cluster-wide Advanced Configuration Snippet(Safety Valve) for core-site.xml 的代码框中，填入 CHDFS 配置。

```
<property>
<name>fs.AbstractFileSystem.ofs.impl</name>
<value>com.qcloud.chdfs.fs.CHDFSDelegateFSAdapter</value>
</property>
```

```

<property>
<name>fs ofs.impl</name>
<value>com.qcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter</value>
</property>
<!--本地 cache 的临时目录, 对于读写数据, 当内存 cache 不足时会写入本地硬盘, 这个路径若不存在
会自动创建-->
<property>
<name>fs ofs.tmp.cache.dir</name>
<value>/data/emr/hdfs/tmp/chdfs/</value>
</property>
<!--appId-->
<property>
<name>fs ofs.user.appid</name>
<value>1250000000</value>
</property>
    
```

以下为必选的 CHDFS 配置项（需添加到 core-site.xml 中），CHDFS 其他配置可参见 [挂载 CHDFS](#)。

CHDFS 配置项	值	含义
fs.ofs.user.appid	1250000000	用户 appId
fs.ofs.tmp.cache.dir	/data/emr/hdfs/tmp/chdfs/	本地 cache 的临时目
fs.ofs.impl	com.qcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter	chdfs 对 FileSystem com.qcloud.chdfs.fs.
fs.AbstractFileSystem.ofs.impl	com.qcloud.chdfs.fs.CHDFSDelegateFSAdapter	chdfs 对 AbstractFile com.qcloud.chdfs.fs.

- 对 HDFS 服务进行操作，单击部署客户端配置，此时以上 core-site.xml 配置会更新到集群里的机器上。
- 将 CHDFS 最新的 SDK 包，放置到 CDH HDFS 服务的 jar 包路径下，请根据实际值进行替换，示例如下：

```

cp chdfs_hadoop_plugin_network-2.0.jar /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16
.1.p0.3/lib/hadoop-hdfs/
    
```

注意：

在集群中的每台机器都需要在相同的位置放置 SDK 包。

## 数据迁移

使用 Hadoop Distcp 工具将 CDH HDFS 数据迁移到 CHDFS，详情请参见[原生 HDFS 数据迁移到腾讯云 CHDFS](#)。

## 大数据套件使用 CHDFS

### 1. MapReduce

#### 操作步骤

- (1) 按照 [数据迁移](#) 章节，配置好 HDFS 的相关配置，并将 CHDFS 的 SDK jar 包，放置到 HDFS 相应的目录。
- (2) 在 CDH 系统主页，找到 YARN，重启 NodeManager 服务（TeraGen 命令可以不用重启，但是 TeraSort 由于业务内部逻辑，需要重启 NodeManger，建议都统一重启 NodeManager 服务）。

#### 示例

下面以 Hadoop 标准测试中的 TeraGen 和 TeraSort 为例：

```
hadoop jar ./hadoop-mapreduce-examples-2.7.3.jar teragen -Dmapred.map.tasks=4 109 9 cosn://examplebucket-1250000000/teragen_5/
hadoop jar ./hadoop-mapreduce-examples-2.7.3.jar terasort -Dmapred.map.tasks=4 cosn://examplebucket-1250000000/teragen_5/ cosn://examplebucket-1250000000/result14
```

说明：

`ofs:// schema` 后面请替换为用户 CHDFS 的挂载点路径。

### 2. Hive

#### 2.1 MR 引擎

#### 操作步骤

- (1) 按照 [数据迁移](#) 章节，配置好 HDFS 的相关配置，并且将 CHDFS 的 SDK jar 包，放置到 HDFS 相应的目录。
- (2) 在 CDH 主页面，找到 HIVE 服务，重启 Hiveserver2 及 HiverMetastore 角色。

#### 示例

某用户的真实业务查询，例如执行 Hive 命令行，创建一个 Location，作为在 CHDFS 上的分区表：

```
CREATE TABLE `report.report_o2o_pid_credit_detail_grant_daily` (
  `cal_dt` string,
  `change_time` string,
  `merchant_id` bigint,
  `store_id` bigint,
  `store_name` string,
```



```
`wid` string,
`member_id` bigint,
`meber_card` string,
`nickname` string,
`name` string,
`gender` string,
`birthday` string,
`city` string,
`mobile` string,
`credit_grant` bigint,
`change_reason` string,
`available_point` bigint,
`date_time` string,
`channel_type` bigint,
`point_flow_id` bigint)
PARTITIONED BY (
`topicdate` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
'cosn://examplebucket-1250000000/user/hive/warehouse/report.db/report_o2o_pid_cre
dit_detail_grant_daily'
TBLPROPERTIES (
'last_modified_by'='work',
'last_modified_time'='1589310646',
'transient_lastDdlTime'='1589310646')
```

执行 sql 查询：

```
select count(1) from report.report_o2o_pid_credit_detail_grant_daily;
```

观察结果如下：

```
Time taken: 1.589 seconds
hive> select count(1) from report.report_o2o_pid_credit_detail_grant_daily;
Query ID = root_20200713121818_3a911a0c-2496-4e7e-b59d-b2a26266a6ab
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1594351711155_0014, Tracking URL = http://hadoop01:8088/proxy/application_1594351711155_0014/
Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job -kill job_1594351711155_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-07-13 12:18:19,189 Stage-1 map = 0%, reduce = 0%
2020-07-13 12:18:25,391 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 8.89 sec
2020-07-13 12:18:30,544 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 10.8 sec
MapReduce Total cumulative CPU time: 10 seconds 800 msec
Ended Job = job_1594351711155_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 10.8 sec HDFS Read: 17383
HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 800 msec
OK
27677
Time taken: 19.128 seconds, Fetched: 1 row(s)
hive>
```

## 2.2 Tez 引擎

Tez 引擎需要将 CHDFS 的 jar 包导入到 Tez 的压缩包内，下面以 apache-tez.0.8.5 为例进行说明：

### 操作步骤

- (1) 找到 CDH 集群安装的 tez 包，然后解压，例如/usr/local/service/tez/tez-0.8.5.tar.gz。
- (2) 将 CHDFS 的 jar 包放置到解压后的目录下，然后重新压缩输出一个压缩包。
- (3) 将新的压缩包上传到 tez.lib.uris 指定的路径下（如果之前存在路径则直接替换即可）。
- (4) 在 CDH 主页面，找到 HIVE，重启 hiveserver 和 hivemetastore。

## 3. Spark

### 操作步骤

- (1) 按照 [数据迁移](#) 章节，配置好 HDFS 的相关配置，并且将 CHDFS 的 SDK jar 包，放置到 HDFS 相应的目录。
- (2) 重启 NodeManager 服务。

### 示例

以 CHDFS 进行 Spark example word count 测试为例。

```
spark-submit --class org.apache.spark.examples.JavaWordCount --executor-memory 4g
--executor-cores 4 ./spark-examples-1.6.0-cdh5.16.1-hadoop2.6.0-cdh5.16.1.jar cos
n://examplebucket-1250000000/wordcount
```

执行结果如下：

```
20/07/17 18:35:05 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 3 ms
20/07/17 18:35:05 INFO executor.Executor: Finished task 0.0 in stage 1.0 (TID 1). 1870 bytes result sent to driver
20/07/17 18:35:05 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 31 ms on localhost (executor driver) (1
20/07/17 18:35:05 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
20/07/17 18:35:05 INFO scheduler.DAGScheduler: ResultStage 1 (collect at JavaWordCount.java:68) finished in 0.031 s
20/07/17 18:35:05 INFO scheduler.DAGScheduler: Job 0 finished: collect at JavaWordCount.java:68, took 0.548912 s
And: 4
day.: 1
God: 4
Let: 1
light.: 1
it: 1
light,: 1
evening: 1
was: 2
that: 1
light.: 1
be: 1
Day,: 1
said,: 1
darkness.: 1
he: 1
first: 1
there: 2
light: 2
Night.: 1
morning: 1
darkness: 1
were: 1
saw: 1
divided: 1
and: 4
good.: 1
from: 1
called: 2
the: 8
```

## 4. Sqoop

### 操作步骤

- (1) 按照 [数据迁移](#) 章节，配置好 HDFS 的相关配置，并且将 CHDFS 的 SDK jar 包，放置到 HDFS 相应的目录。
- (2) CHDFS 的 SDK jar 包还需要放到 sqoop 目录下（例如/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/sqoop/）。
- (3) 重启 NodeManager 服务。

### 示例

以导出 MYSQL 表到 CHDFS 为例，可参考 [关系型数据库和 HDFS 的导入导出](#) 文档进行测试。

```
sqoop import --connect "jdbc:mysql://IP:PORT/mysql" --table sqoop_test --username
root --password 123 --target-dir cosn://examplebucket-1250000000/sqoop_test
```

执行结果如下：

```
20/07/17 18:48:33 INFO mapreduce.Job: map 100% reduce 0%
20/07/17 18:48:33 INFO mapreduce.Job: Job job_1594976906551_0011 completed successfully
20/07/17 18:48:33 INFO mapreduce.Job: Counters: 35
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=526689
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=295
    HDFS: Number of bytes written=0
    HDFS: Number of read operations=3
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=0
    OFS: Number of bytes read=0
    OFS: Number of bytes written=104
    OFS: Number of read operations=0
    OFS: Number of large read operations=0
    OFS: Number of write operations=3
  Job Counters
    Launched map tasks=3
    Other local map tasks=3
    Total time spent by all maps in occupied slots (ms)=36308
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=9077
    Total vcore-milliseconds taken by all map tasks=9077
    Total megabyte-milliseconds taken by all map tasks=37179392
  Map-Reduce Framework
    Map input records=3
    Map output records=3
    Input split bytes=295
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=277
    CPU time spent (ms)=6430
    Physical memory (bytes) snapshot=1371717632
    Virtual memory (bytes) snapshot=18832379904
    Total committed heap usage (bytes)=6655311872
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=104
20/07/17 18:48:33 INFO mapreduce.ImportJobBase: Transferred 0 bytes in 13.0961 seconds (0 bytes/sec)
20/07/17 18:48:33 INFO mapreduce.ImportJobBase: Retrieved 3 records.
20/07/17 18:48:33 INFO fs.CHDFSHadoopFileSystemAdapter: actual-file-system-close usedTime: 13
20/07/17 18:48:33 INFO fs.CHDFSHadoopFileSystemAdapter: end-close time: 1594982913402, total-used-time: 13650
```

## 5. Presto

### 操作步骤

- (1) 按照 [数据迁移](#) 章节，配置好 HDFS 的相关配置，并且将 CHDFS 的 SDK jar 包，放置到 HDFS 相应的目录。
- (2) CHDFS 的 SDK jar 包还需要放到 presto 目录下（例如/usr/local/services/cos\_presto/plugin/hive-hadoop2）。
- (3) 由于 presto 不会加载 hadoop common 下的 gson-2...jar，需将 gson-2...jar 也放到 presto 目录下（例如 /usr/local/services/cos\_presto/plugin/hive-hadoop2，仅 CHDFS 依赖 gson）。
- (4) 重启 HiveServer、HiveMetaStore 和 Presto 服务。

### 示例

以 HIVE 创建 Location 为 CHDFS 的表查询为例：

```
select * from chdfs_test_table where bucket is not null limit 1;
```

说明：

chdfs\_test\_table 为 location 是 ofs scheme 的表。

查询结果如下：

```
presto:inventory_search> select * from chdfs_test_table_0720 where bucket is not null limit 1;
  appid | bucket | path | size |
-----+-----+-----+-----+
  125   | ssuupv80105841qq206 | 444600684/20190316/3/01a9ee49bd4045179c92e319ff03b810 | 3800424 |
(1 row)

Query 20200720 112833 00061 thb89, FINISHED, 7 nodes
```

# CHDFS Ranger 权限体系解决方案

最近更新时间：2022-03-30 09:30:26

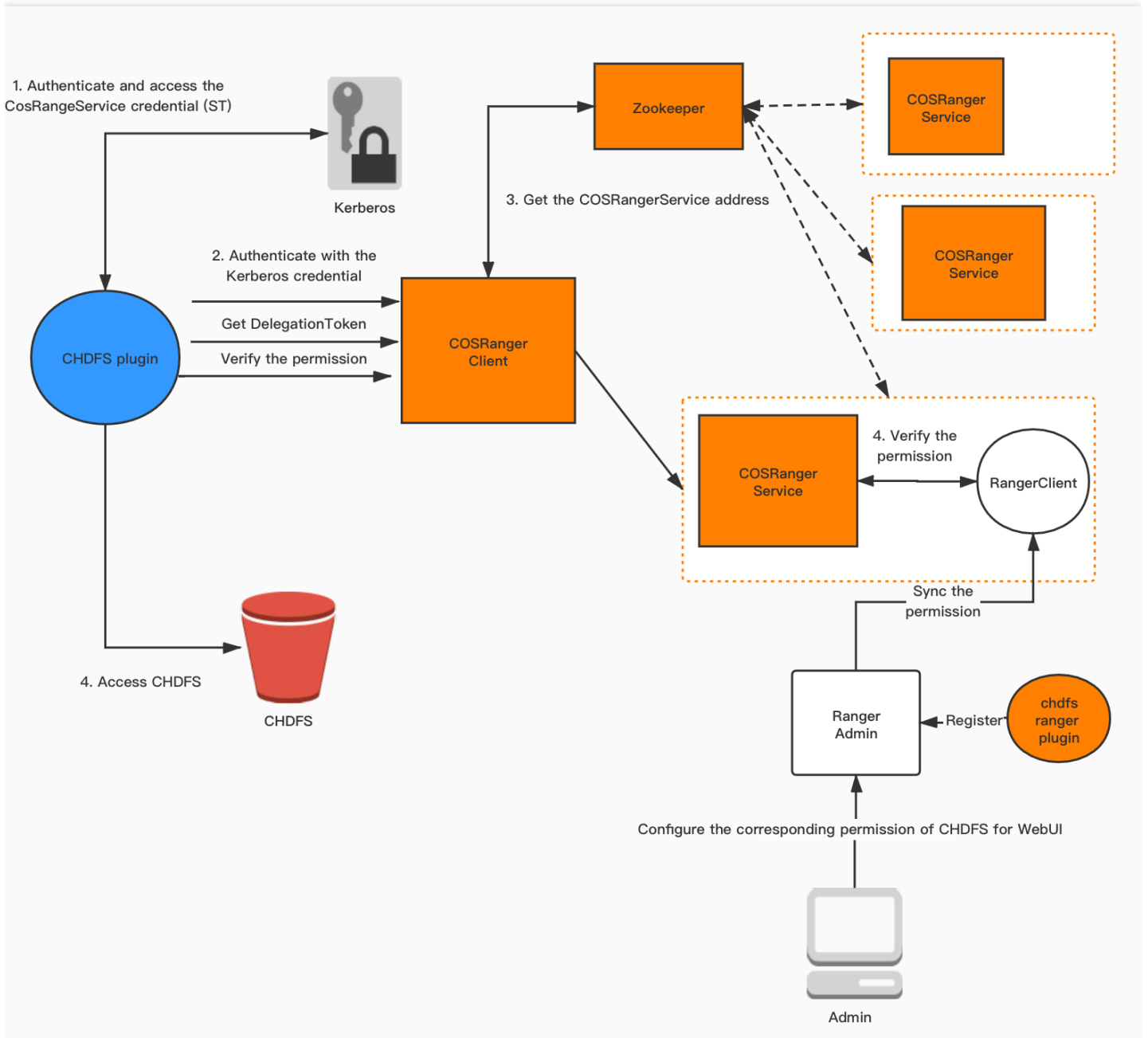
## 背景

大数据用户使用存算分离后，将数据托管在云 HDFS（Cloud HDFS，CHDFS）上。CHDFS 提供了类似 HDFS 的权限体系管控。Hadoop Ranger 在 HDFS 权限基础上，提供了更精细的权限管控，包括用户组权限设置，针对某个前缀的权限设置。同时 Hadoop Ranger 作为一站式的权限体系解决方案，不仅支持存储端权限管控，还支持 YARN，Hive 等组件权限管控。因此，为了维持方便客户的使用习惯，我们提供了 CHDFS 的 Ranger 接入解决方案，方便客户使用 Ranger 来进行 CHDFS 的权限管控。

## 优势

- 细粒度的权限控制，兼容 Hadoop 权限习惯。
- 用户统一管理大数据组件与云端托管存储的权限。

## 解决方案架构



Hadoop 权限体系中，认证由 Kerberos 提供，授权鉴权由 Ranger 负责。在此基础上，我们提供以下组件，来支持 CHDFS 的 Ranger 权限方案。

- **CHDFS-Ranger-Plugin**：提供 Ranger 服务端的服务定义插件。它们提供了 Ranger 侧的 CHDFS 服务描述，部署了该插件后，用户即可在 Ranger 的控制页面上，填写相应的权限策略。
- **COSRangerService**：该服务集成了 Ranger 的客户端，周期性从 Ranger 服务端同步权限策略，在收到客户的鉴权请求后，在本地进行权限校验。同时它提供了 Hadoop 中 DelegationToken 相关的生成，续租等接口，所有的接口都是通过 Hadoop IPC 定义。
- **CosRangerClient**：COSN 插件对其进行动态加载，把权限校验的请求转发给 CosRangerService。

## 部署环境

- Hadoop 环境。
- ZooKeeper、Ranger、Kerberos 服务（如果有认证需求，则部署）。

说明：

以上服务由于是成熟的开源组件，因此客户可自行安装。

## 部署组件

- 部署CHDFS-Ranger-Plugin
- 部署COS-Ranger-Service
- 部署COS-Ranger-Client
- 部署CHDFS

CHDFS-Ranger-Plugin 拓展了 Ranger Admin 控制台上的服务种类，用户可在 Ranger 控制台上，设置和 CHDFS 相关的操作权限。

### 代码地址

可前往 [Github](#) 的 ranger-plugin 目录下获取。

### 版本

V1.2版本及以上。

### 部署步骤

1. 在 Ranger 的服务定义目录下新建 COS 目录（注意：目录权限需要保证至少有 x 与 r 权限）。
2. 腾讯云的 EMR 环境，路径是 ranger/ews/webapp/WEB-INF/classes/ranger-plugins。
3. 自建的 hadoop 环境，可以通过在ranger目录下 find hdfs 等方式找到已经接入到 ranger 服务的组件，查找 ranger-plugins 目录位置。

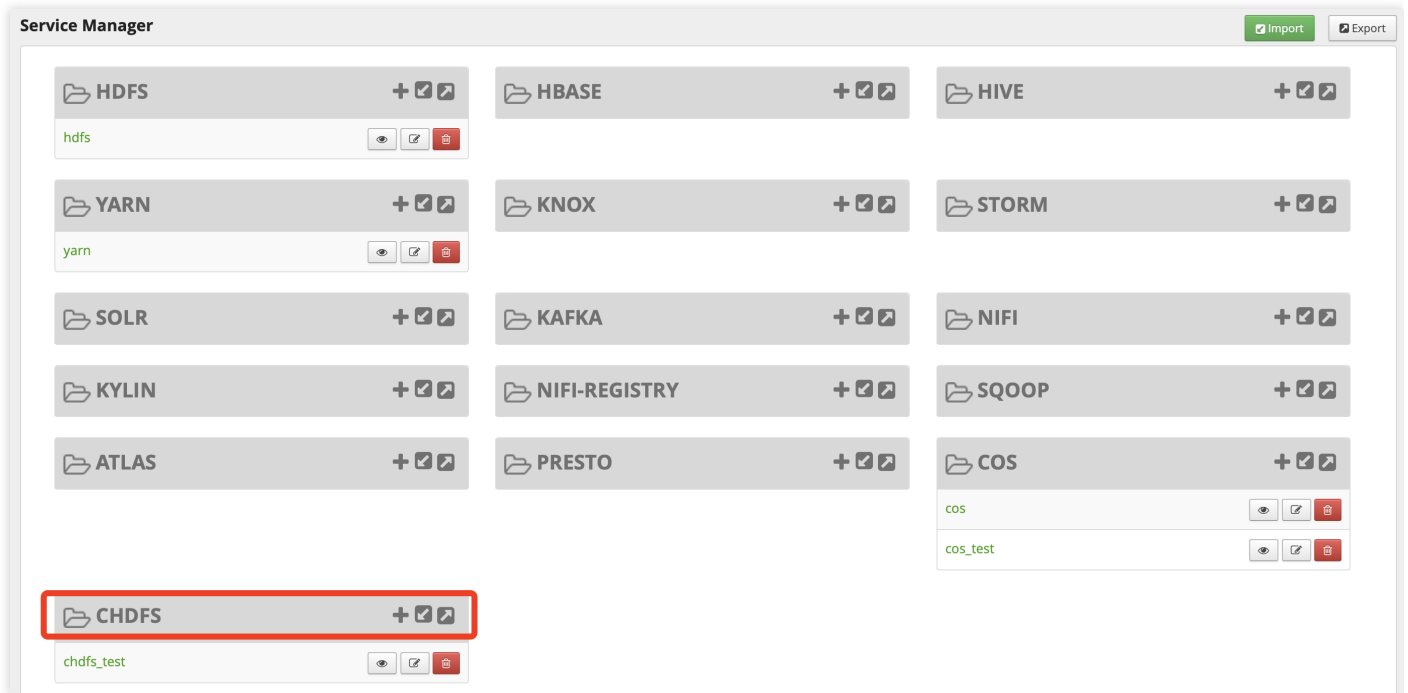


```
[hadoop@10 ranger-plugins]$ ll
total 68
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 atlas
drwxr-xr-x 2 hadoop hadoop 4096 Dec 15 10:57 chdfs
drwxr-xr-x 2 hadoop hadoop 4096 Dec 15 10:57 cos
drwxr-xr-x 2 hadoop hadoop 4096 Feb 25 2020 hbase
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 hdfs
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 hive
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 kafka
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 kms
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 Knox
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 kylin
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 nifi
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 nifi-registry
drwxr-xr-x 2 hadoop hadoop 4096 Aug 5 2020 presto
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 solr
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 sqoop
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 storm
drwxr-xr-x 2 hadoop hadoop 4096 Feb 19 2020 yarn
```

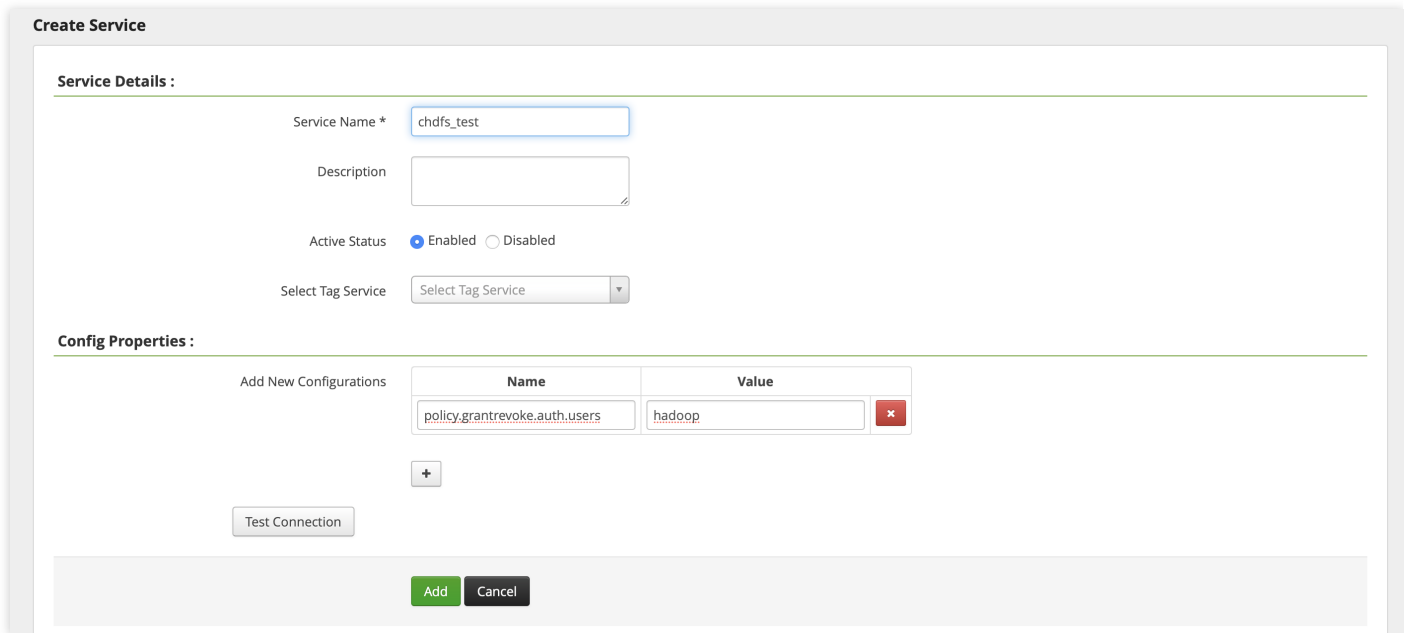
4. 在 CHDFS 目录下，放入 cos-chdfs-ranger-plugin-xxx.jar（注意：jar 包至少有 r 权限）。
5. 重启 Ranger 服务。
6. 在 Ranger 上注册 CHDFS Service。可参考如下命令：

```
##生成服务,需传入 Ranger 管理员账号密码,以及 Ranger 服务的地址。
##对于腾讯云 EMR 集群,管理员用户是 root,密码是构建 emr 集群时设置的 root 密码,ranger
服务的 IP 换成 EMR 的 master 节点 IP。
adminUser=root
adminPasswd=xxxxxxx
rangerServerAddr=10.0.0.1:6080
curl -v -u${adminUser}:${adminPasswd} -X POST -H "Accept:application/json" -H
"Content-Type:application/json" -d @./chdfs-ranger.json http://${rangerServerA
ddr}/service/plugins/definitions
##如果要删除刚定义的服务,则传入刚刚创建服务时,返回的服务 ID
serviceId=102
curl -v -u${adminUser}:${adminPasswd} -X DELETE -H "Accept:application/json" -
H "Content-Type:application/json" http://${rangerServerAddr}/service/plugins/d
efinitions/${serviceId}
```

7. 创建服务成功后，可在 Ranger 控制台看到 CHDFS 服务。如下所示：

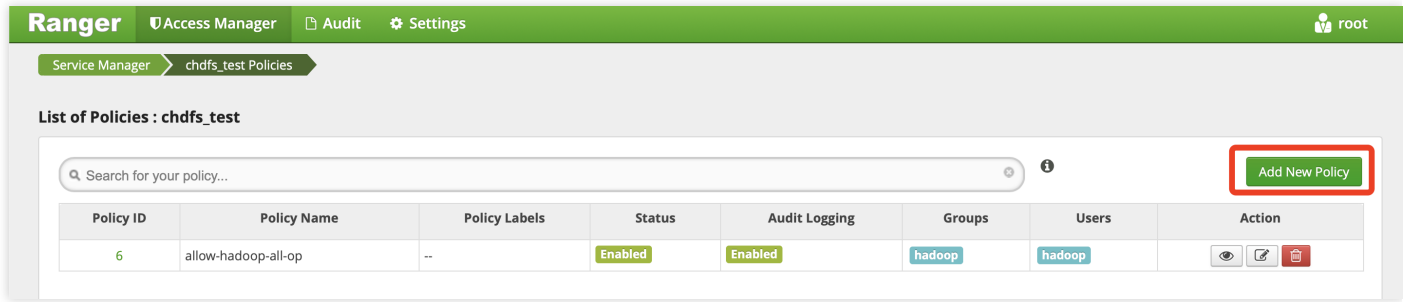


8. 在 CHDFS 服务侧单击【+】，定义新服务实例，服务实例名可自定义，例如 `chdfs` 或者 `chdfs_test`。服务的配置如下所示：

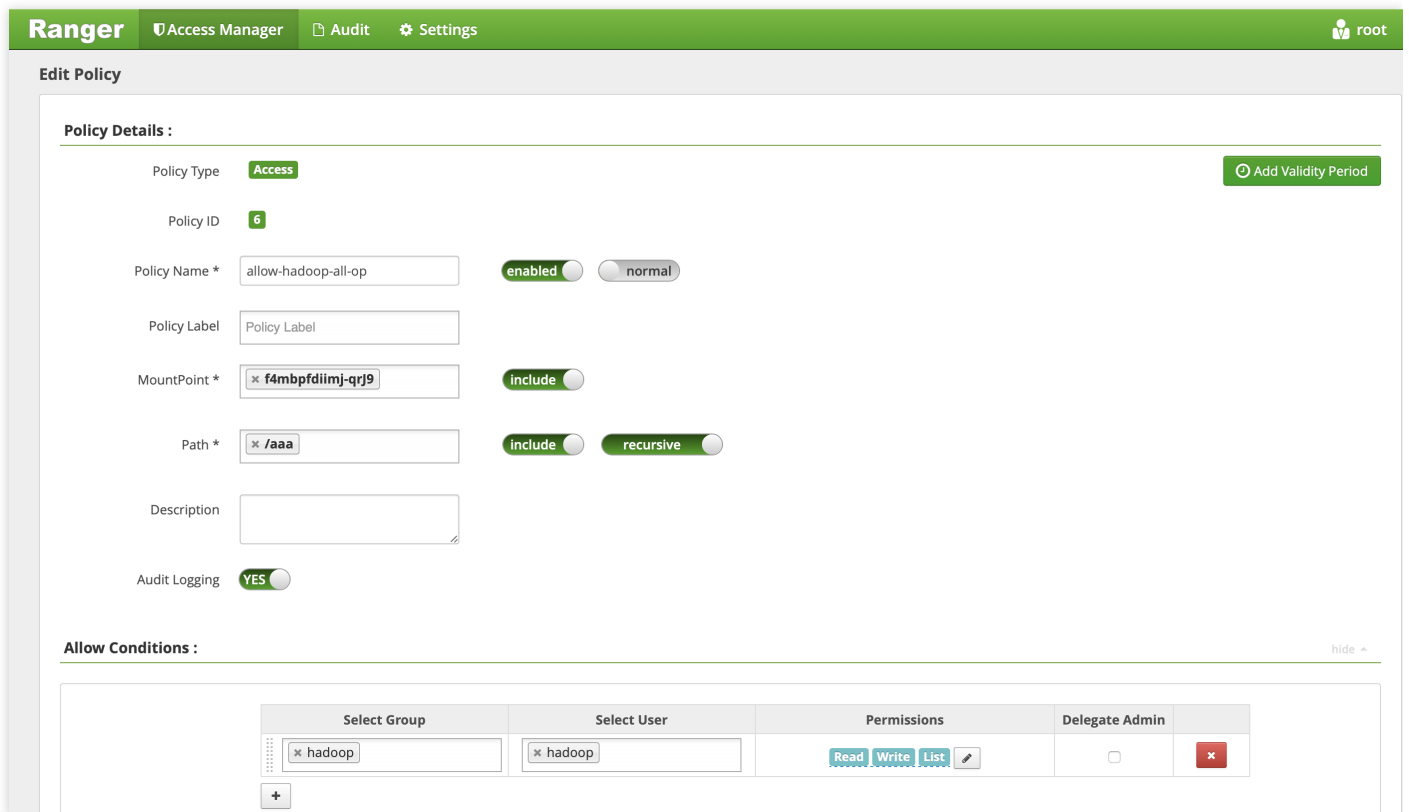


其中 `policy.grantrevoke.auth.users` 需设置后续启动 COSRangerService 服务的用户名。通常建议设置成 `hadoop`，后续 COSRangerService 可使用此用户名进行启动。

9. 单击新生成的 CHDFS 服务实例，添加 policy。如下所示：



10. 在跳转界面中，配置以下参数。具体说明如下：



- **MountPoint:** 挂载点的名称，格式为 f4mxxxxxx-yyyy 等样式，可登录 [CHDFS 控制台](#) 查看。
- **path:** CHDFS 路径。注意 CHDFS 路径必须以 / 开始。
  - include：表示设置的权限适用于 path 本身，还是除了 path 以外的其他路径。
  - recursive：表示权限不仅适用于 path，还适用于 path 路径下的子成员（即递归子成员）。通常用于 path 设置为目录的情况。
- **user/group：** 用户名和用户组。这里是或的关系，即用户名或者用户组满足其中一个，即可拥有对应的操作权限。
- **Permissions：**
  - Read：读操作。对应于对象存储里面的 GET、HEAD 类操作，包括下载对象、查询对象元数据等。

- Write：写操作。对应于对象存储里面的 PUT 类等修改操作，例如上传对象。
- Delete：删除操作。对应于对象存储里删除 Object。对于 Hadoop 的 Rename 操作，需要有对原路径的删除操作权限，对新路径的写入操作权限。
- List：遍历权限。对应于对象存储里面的 List Object。

## 验证

1. 使用 `hadoop cmd` 执行访问 `chdfs` 的相关操作。如下范例所示：

```
# 将挂载点，路径等替换为自己的实际信息。
hadoop fs -ls ofs://f4mxxxxxyyyy-zzzz.chdfs.ap-guangzhou.myqcloud.com/doc
hadoop fs -put ./xxx.txt ofs://f4mxxxxxyyyy-zzzz.chdfs.ap-guangzhou.myqcloud.com/doc/
hadoop fs -get ofs://f4mxxxxxyyyy-zzzz.chdfs.ap-guangzhou.myqcloud.com/exampleobject.txt
hadoop fs -rm ofs://f4mxxxxxyyyy-zzzz.chdfs.ap-guangzhou.myqcloud.com/exampleobject.txt
```

2. 使用 MR Job 进行验证，验证前需重启相关的服务，例如 Yarn、Hive 等。

## 常见问题

### kerberos 是否必须安装？

如果所在的集群，用户都是可信的，例如仅内部使用的集群，可安装 Kerberos 满足认证的需求。如果用户仅进行鉴权操作，为了避免无权限的客户误操作，那么可以不安装 Kerberos，只使用 ranger 进行鉴权。

安装 Kerberos 会引入一些性能损耗，请用户综合自己的安全需求与性能需求进行考量。如果需要认证，开启 Kerberos 后，需要设置 COS Ranger Service 和 COS Ranger Client 相关的配置项。

### 如果开启了 Ranger，但未配置任何 Policy，或者未匹配到任何 Policy，会如何操作？

如果未匹配上任何 policy，会默认拒绝该操作。

### 开启了 ranger 后，CHDFS 端是否还会进行 POSIX 鉴权？

Ranger 鉴权是在客户端环境进行的，经过 ranger 鉴权的请求，会发给 CHDFS 服务端，服务端默认会进行 POSIX 鉴权。因此如果权限都在 Ranger 端进行控制，请在 CHDFS 控制台关闭 POSIX 权限。