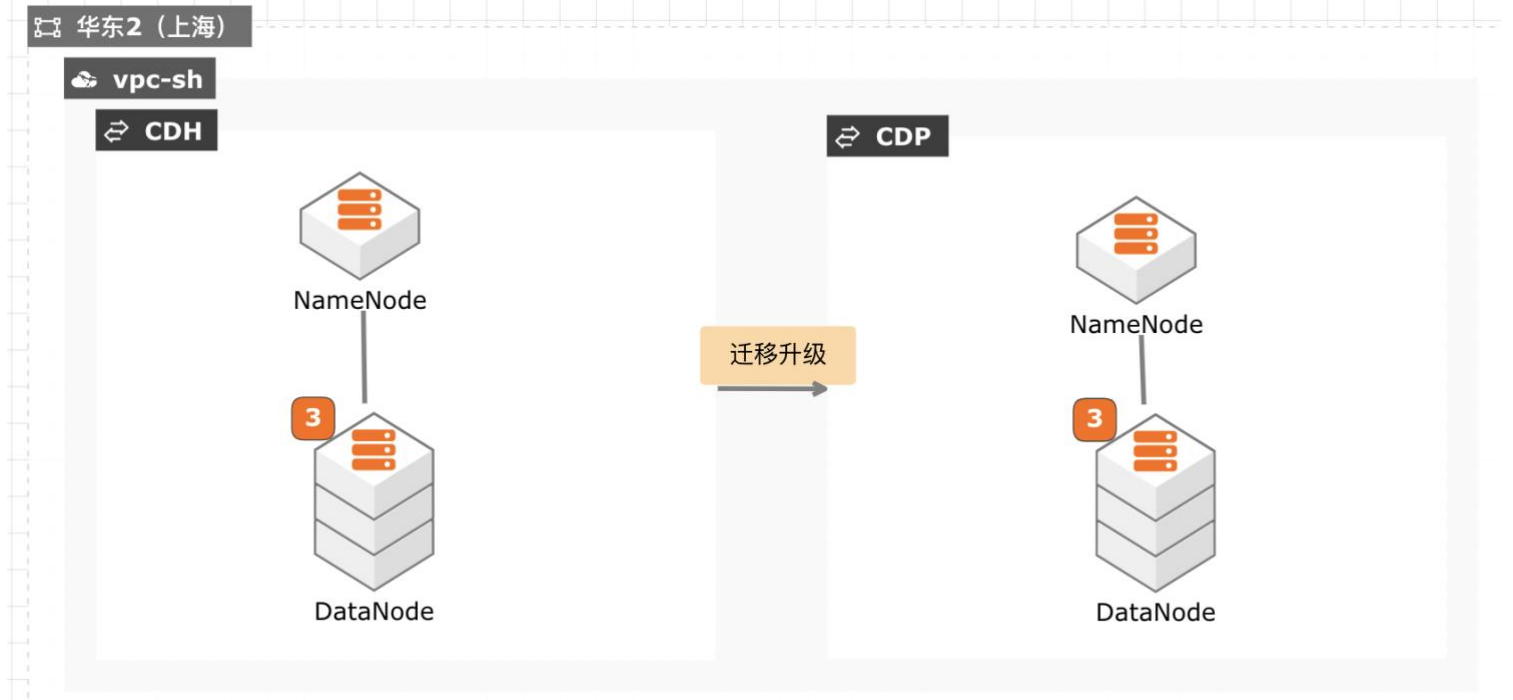


CDH 迁移升级 CDP 最佳实践

业务架构



场景描述

当前 CDH 免费版停止下载，终止服务，针对需要企业版服务能力并且 CDH 升级过程对业务影响较小的客户，通过安装新的 CDP 集群，将现有数据拷贝至新集群，然后将新集群切换为生产集群，升级过程没有数据丢失风险，停机时间较短，适合大部分互联网客户升级使用。

解决的问题

- CDH 升级至 CDP
- 要求升级过程无数据丢失风险
- 要求升级宕机时间及短
- 数据和组件迁移
- 数据一致性校验

产品列表

VPC、ECS、EIP、CDP、CDH

ALIBABA CLOUD

文档版本：20210819

文档版本信息

文本信息

属性	内容
文档名称	CDH 迁移升级 CDP 最佳实践
文档编号	
文档版本	V1.0
版本日期	2021-08-19
文档状态	外部发布
制作人	弦望
审阅人	

文档变更记录

版本编号	日期	作者	审核人	说明
V1.0	2021-08-19	玮休、哲谦、 弦望、棕泽、 王雪峰		创建



法律声明

阿里云提醒您在使用或阅读本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

前言

概述

当前 CDH 免费版停止下载，终止服务，针对需要企业版服务能力并且 CDH 升级过程对业务影响较小的客户，通过安装新的 CDP 集群，将现有数据拷贝至新集群，然后将新集群切换为生产集群，升级过程没有数据丢失风险，停机时间较短，适合大部分互联网客户升级使用。

名词解释

- CDH: hadoop 是一个开源项目，所以很多公司在这个基础进行商业化，Cloudera 对 hadoop 做了相应的改变。Cloudera 公司的发行版，我们将该版本称为 CDH(Cloudera Distribution Hadoop)。
- CDP: Cloudera 对 hadoop 做了商业化版本，面向 IT 和企业的大数据平台 Cloudera Data Platform (CDP)，是 CDH 的升级版。
- 云服务器 ECS (Elastic Compute Service) :是一种弹性可伸缩的计算服务，助您降低 IT 成本，提升运维效率，使您更专注于核心业务创新。更多信息，请参见 ECS 简介 (<https://www.aliyun.com/product/ecs>)。

目录

文档版本信息	I
法律声明	II
前言	III
目录	IV
最佳实践概述	1
前置条件	3
1. 基础环境搭建	4
1.1. 使用 CADT 创建资源	4
1.2. 部署 CDH	6
1.2.1. 分布式集群配置	6
1.2.2. 安装 CDH	13
1.2.3. 安装 Kudo	20
1.2.4. 升级 JDK 到 1.8 版本	24
1.2.5. 安装 Spark2	26
1.3. 假数据构造	32
1.3.1. 构造 HDFS 和 Hive 测试数据	32
1.3.2. 构造 Impala 测试数据	35
1.3.3. 构造 Kudu 测试数据	37
1.3.4. 构造 Hbase 测试数据	38
1.4. 部署 CDP	40
1.4.1. 购买部署 CDP	40
1.4.2. 安装 Impala	43
1.4.3. 安装 Hbase	46
1.4.4. 安装 kudu	49
1.4.5. 安装 Spark	53
2. 数据迁移	58
2.1. HDFS 迁移	58
2.2. Hive/Impala 迁移	63
2.3. Kudo 迁移	69
2.4. Hbase 迁移	74
3. 组件迁移	79
3.1. Sentry 迁移	79
3.1.1. 概述	79
3.1.2. 迁移方法	79
3.1.2.1. 复制策略的 Sentry 权限部分	79
3.1.3. 转换规则	80
3.1.4. 风险提示	81
3.1.5. 迁移步骤	82
3.1.5.1. 导出 Sentry 权限	82
3.1.5.2. 导入权限到 Ranger	88
3.2. Navigator 迁移	92

3.2.1.	概述	92
3.2.2.	迁移方法	93
3.2.3.	转换规则	94
3.2.4.	风险提示	94
3.2.5.	迁移步骤	95
3.3.	YARN Fair Scheduler 迁移	96
3.3.1.	概述	96
3.3.2.	迁移方法	97
3.3.3.	转换规则	97
3.3.4.	调度程序迁移的限制	101
3.3.4.1.	不能在同一级别上创建静态和动态队列	101
3.3.4.2.	放置规则和映射规则不同	101
3.3.4.3.	动态队列的容量值是固定的	102
3.3.4.4.	风险提示	102
3.3.5.	迁移步骤	107
3.3.5.1.	fs2cs 转换工具的 CLI 选项	109
3.3.6.	手动配置调度程序属性	110
3.3.6.1.	使用 YARN 队列管理器 UI 配置调度程序属性	111
3.3.7.	使用 Cloudera Manager 安全阀配置调度程序属性	112
4.	附录：代码改造指南	114
4.1.	Hive 代码改造	114
4.2.	Impala 代码改造	115
4.3.	Spark 代码改造	118
4.4.	Hbase 代码改造	119
4.5.	Kudo 代码改造	121
4.6.	Kafka 代码改造	122
4.7.	Solr 代码改造	122

最佳实践概述

概述

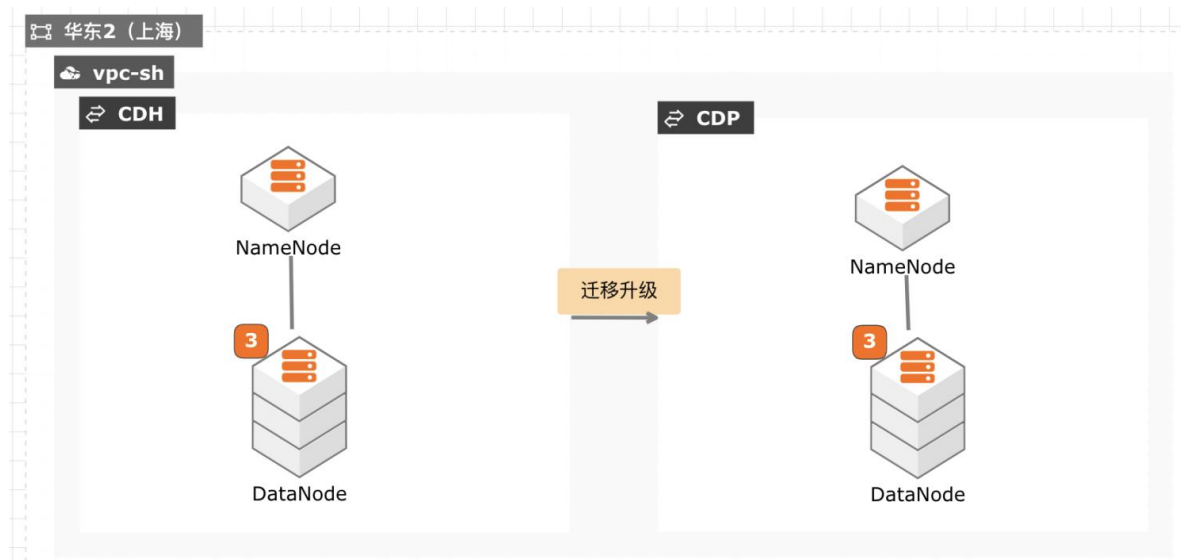
当前 CDH 免费版停止下载，终止服务，针对需要企业版服务能力并且 CDH 升级过程对业务影响较小的客户，通过安装新的 CDP 集群，将现有数据拷贝至新集群，然后将新集群切换为生产集群，升级过程没有数据丢失风险，停机时间较短，适合大部分互联网客户升级使用。

适用场景

- 较短的停服时间升级 CDH 至 CDP
- CDH 的数据迁移至 CDP
- CDH 组件升级至最新的 CDP 组件

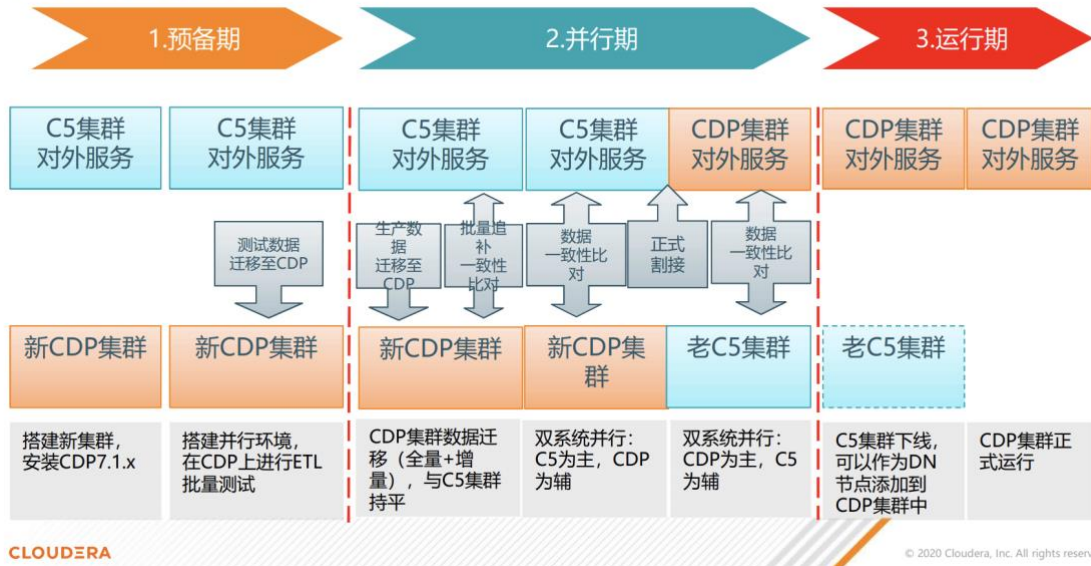
技术架构

本实践方案基于如下图所示的技术架构和主要流程编写操作步骤：



迁移实施流程如下：

拷贝升级实施流程



方案优势

- 没有数据丢失风险。
- 较短的服务停机时间。

前置条件

在进行本文操作前，您需要完成以下准备工作：

- 注册阿里云账号，并完成实名认证。您可以登录阿里云控制台，并前往实名认证页面（<https://account.console.aliyun.com/v2/#/authc/home>）查看是否已经完成实名认证。
- 阿里云账户余额大于 100 元。您可以登录阿里云控制台，并前往账户总览页面（<https://expense.console.aliyun.com/#/account/home>）查看账户余额。
- 开通 ECS、购买 CDP 服务。

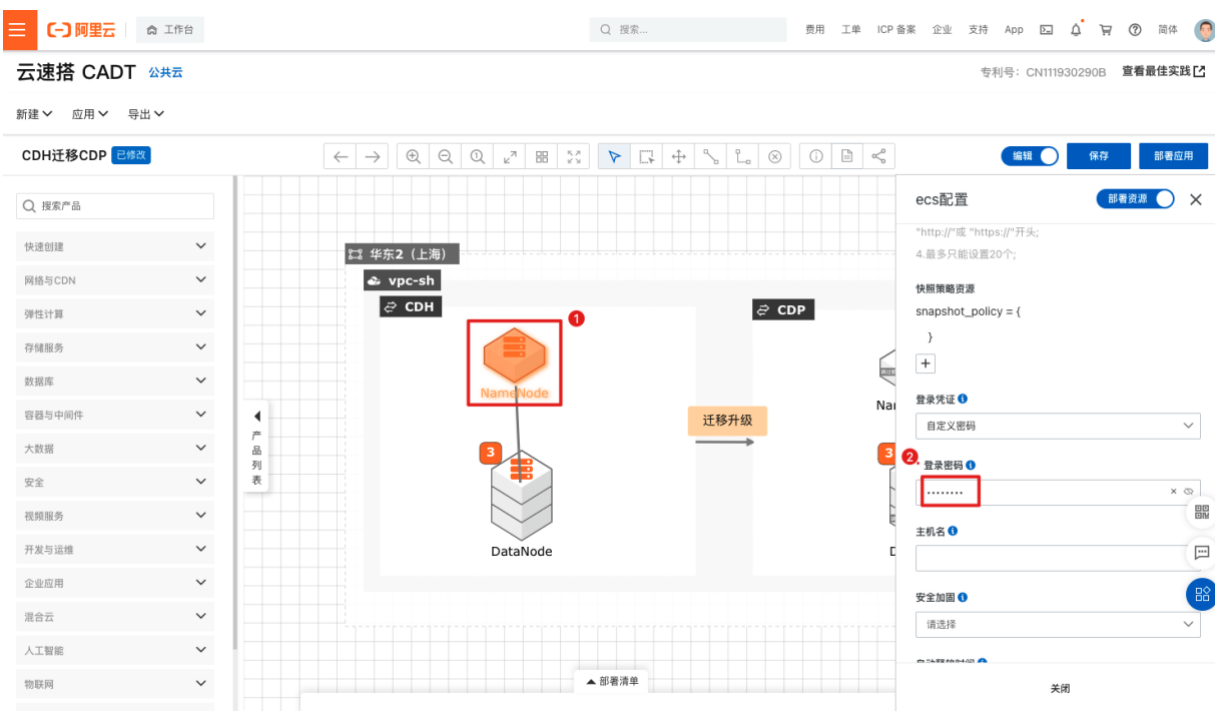
1. 基础环境搭建

1.1. 使用 CADT 创建资源

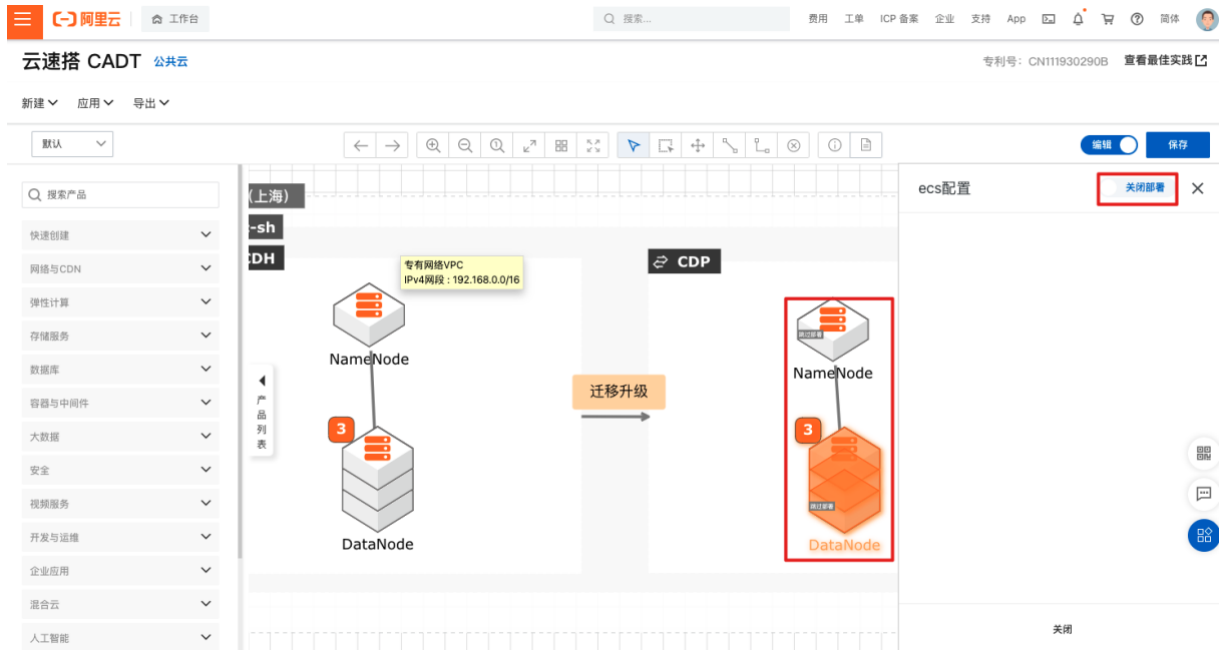
步骤1 登录 CADT 控制台 (<https://bpstudio.console.aliyun.com/>)，选择官方模板新建。



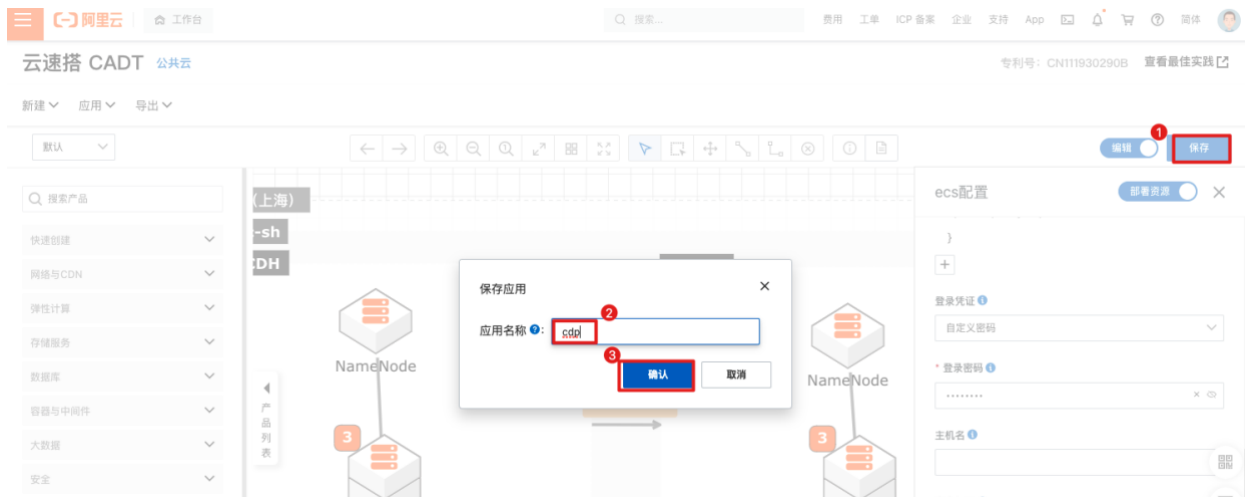
步骤2 设置 CDH 集群相关的 ECS 密码。

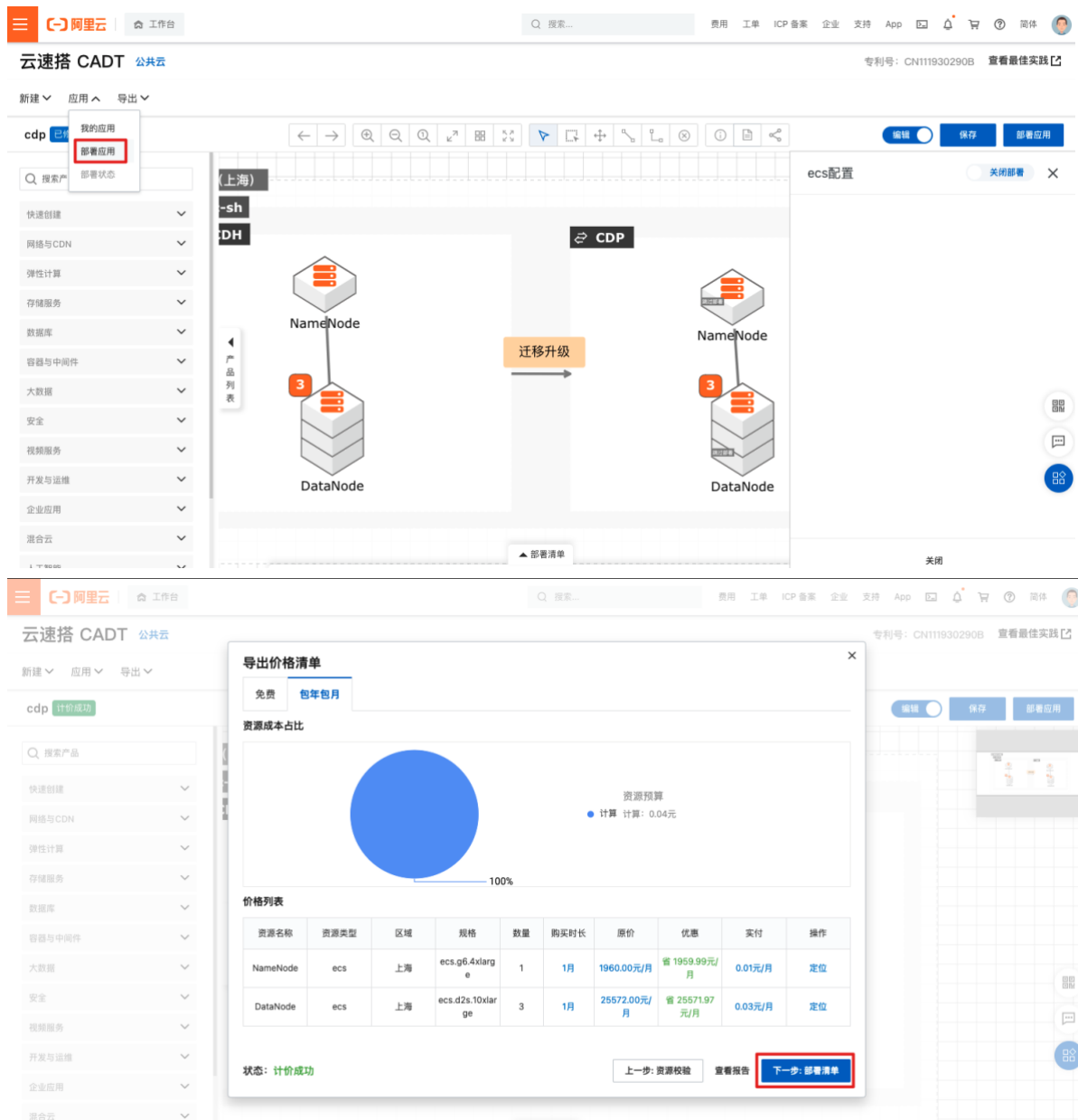


步骤3 由于当前 CDP 集群通过云市场购买，所以 CDP 集群不通过 CADT 部署，关闭部署。



步骤4 保存应用。开始部署应用，等待部署完成。





1.2. 部署 CDH

1.2.1. 分布式集群配置

步骤1 修改 CDH 集群各节点的主机名(先在 ECS 控制台编辑各实例名称)。

通过 SSH 分别远程登录 Hadoop 集群所在的 4 台 ECS 实例，并执行下面的命令：

ECS 实例名	执行的命令
master	<code>hostnamectl set-hostname master</code>
slave1	<code>hostnamectl set-hostname slave1</code>
slave2	<code>hostnamectl set-hostname slave2</code>

ECS 实例名	执行的命令
slave3	hostnamectl set-hostname slave3

说明： 建议您在实践过程中，Hadoop 集群主机名保持和上表规划的一致，这样您在后续步骤中无需修改 CDH 的配置文件即可直接使用，最大限度降低基础环境搭建的工作量。

磁盘扩容请参考：https://help.aliyun.com/document_detail/111738.htm

步骤2 进入 master 节点，配置 **/etc/hosts** 文件。

收集 3 个节点的私网 IP 地址和对应的主机名，如下所示（请根据您的环境的实际 IP 地址进行修改）：

192.168.10.205 master

192.168.10.207 slave1

192.168.10.208 slave2

192.168.10.206 slave3

通过 SSH 远程登录 **master** 节点所在的 ECS 实例，通过 vim 编辑器将上面的信息粘贴到 **/etc/hosts** 文件中，保存后退出。

通过 **scp** 命令将 **/etc/hosts** 文件远程拷贝到 slave1~3 节点上：

```
scp /etc/hosts root@slave1:/etc/hosts
```

```
scp /etc/hosts root@slave2:/etc/hosts
```

```
scp /etc/hosts root@slave3:/etc/hosts
```

步骤3 配置 SSH 无密码访问

1. 在 master 节点执行下面 3 个命令，生成密钥对。

其中第 1 个命令在 master 节点本地执行；第 2、3 个命令在 master 节点通过 SSH 远程登录到 slave1、slave2 和 slave3 节点远程执行，命令执行过程中提示输入的地方都直接回车即可。

```
ssh-keygen -t rsa
```

```
[root@master ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:sJEZr6eYRrHr2tnKLFqMyCqCm2pFrZSWagrq+yZMQA8 root@master
The key's randomart image is:
+----[RSA 2048]-----+
|
| .       =
| E      =
| . o    = .
| . B + =
| . = + o S
| +* + + o
|@. + = .
|B*.B o
|%=*+B.
+----[SHA256]-----+
```

```
ssh slave1 "ssh-keygen -t rsa"
```

```
ssh slave2 "ssh-keygen -t rsa"
```

```
ssh slave3 "ssh-keygen -t rsa"
```

2. 通过下面命令将 master、slave1、slave2 和 slave3 这 4 个节点上的公钥文件 **/root/.ssh/id_rsa.pub** 内容 拷贝到 master 节点的 **/root/.ssh/authorized_keys** 文件中。

```
cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
```

```
ssh slave1 "cat /root/.ssh/id_rsa.pub" >> /root/.ssh/authorized_keys
```

```
ssh slave2 "cat /root/.ssh/id_rsa.pub" >> /root/.ssh/authorized_keys
```

```
ssh slave3 "cat /root/.ssh/id_rsa.pub" >> /root/.ssh/authorized_keys
```

3. 查看 master 节点生产的文件。

```
cat /root/.ssh/authorized_keys
```

4. 通过 **scp** 命令将 authorized_keys 文件远程拷贝到 slave1、slave2 和 slave3 节点。

```
scp /root/.ssh/authorized_keys root@slave1:/root/.ssh/
```

```
scp /root/.ssh/authorized_keys root@slave2:/root/.ssh/
```

```
scp /root/.ssh/authorized_keys root@slave3:/root/.ssh/
```

5. 在 master 节点验证无密码登录。

```
ssh slave1
```

```
exit
```

```
ssh slave2
```

```
exit
```

```
ssh slave3
```

```
exit
```

步骤4 在 master 及 slave1~3 节点上安装 oss util 工具。

```
wget wget http://gosspublic.alicdn.com/ossutil/1.7.6/ossutil64
```

```
chmod 755 ossutil64
```


./ossutil64 config

```
[root@master ~]# ./ossutil64 config
The command creates a configuration file and stores credentials.

Please enter the config file name,the file name can include path(default /root/.ossutilconfig, carriage return will use the default file. If you specified this option to other file, you should specify --config-file option to the file when you use other commands):
No config file entered, will use the default config file /root/.ossutilconfig

For the following settings, carriage return means skip the configuration. Please try "help config" to see the meaning of the settings
Please enter language(CH/EN, default is:EN, the configuration will go into effect after the command successfully executed):CH
Please enter endpoint:http://oss-cn-hangzhou-internal.aliyuncs.com
Please enter accessKeyID:
Please enter accessKeySecret:
Please enter stsToken:
```

说明：杭州地域的用户可以直接使用内网 endpoint，其他地域请替换为公网 endpoint

步骤5 在 master 及 slave1~3 节点上下载 CDH 安装包及工具。

mkdir cdh

```
./ossutil64 sync oss://cdh2cdp/ ./cdh
```

```
[root@master ~]# ./ossutil64 sync oss://cdh2cdp/ ./cdh
Succeed: Total num: 23, size: 3,802,321,119. OK num: 23(download 23 objects).
```

步骤6 安装 JDK。

在 master、slave1~3 节点执行下面命令，完成 4 个节点的 JDK 的安装。

```
cd cdh/rpms/x86_64/
rpm -ivh jdk-6u31-linux-amd64.rpm
rpm -ivh oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm
ln -sf /usr/java/jdk1.7.0_67-cloudera/ /usr/java/jdk
echo 'export JAVA_HOME=/usr/java/jdk'>> /etc/profile
echo 'export PATH=$PATH:JAVA_HOME/bin'>> /etc/profile
source /etc/profile
```

步骤7 在所有节点上设置时区

```
timedatectl set-timezone Asia/Shanghai
```

查看时区

```
timedatectl status
```

```
root@master x86_64]# timedatectl status
Local time: Thu 2021-08-26 16:10:20 CST
Universal time: Thu 2021-08-26 08:10:20 UTC
RTC time: Thu 2021-08-26 16:10:19
Time zone: Asia/Shanghai (CST, +0800)
NTP enabled: yes
TP synchronized: yes
RTC in local TZ: yes
DST active: n/a

Warning: The system is configured to read the RTC time in the local time zone.
This mode can not be fully supported. It will create various problems
with time zone changes and daylight saving time adjustments. The RTC
time is never updated, it relies on external facilities to maintain it.
If at all possible, use RTC in UTC by calling
'timedatectl set-local-rtc 0'.
```

步骤8 在所有节点安装 ntp。

```
yum install -y ntp
```

步骤9 以 master 为 ntp 的主节点，使用以下命令编辑。

```
vi /etc/ntp.conf
```

在 ntp.conf 中添加以下内容：

```
server 127.127.1.0 iburst local clock
```

```
restrict 192.168.10.0 mask 255.255.255.0 nomodify notrap
```

步骤10 开启 ntp 服务并查看状态。

```
systemctl start ntpd
```

```
systemctl status ntpd
```

```
[root@master x86_64]# systemctl status ntpd
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-08-25 17:35:15 CST; 22h ago
     Process: 9738 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 9739 (ntpd)
      CGroup: /system.slice/ntpd.service
              └─9739 /usr/sbin/ntpd -u ntp:ntp -g

Aug 25 17:35:15 master systemd[1]: Starting Network Time Service...
Aug 25 17:35:15 master ntpd[9738]: ntpd 4.2.6p5@1.2349-o Tue Jun 23 15:38:18 UTC 2020 (1)
Aug 25 17:35:15 master systemd[1]: Started Network Time Service.
Aug 25 17:35:15 master ntpd[9739]: proto: precision = 0.044 usec
Aug 25 17:35:15 master ntpd[9739]: 0.0.0.0 c01d 0d kern kernel time sync enabled
Aug 25 17:35:15 master ntpd[9739]: line 60 column 27 syntax error, unexpected T_String, exp..._EOC
Aug 25 17:35:15 master ntpd[9739]: syntax error in /etc/ntp.conf line 60, column 27
hint: Some lines were ellipsized, use -l to show in full.
```

步骤11 分布设置 slave1~3 和 master 节点时钟同步。

```
#设置 slave1 和 master 时钟同步
```

```
ssh slave1 systemctl stop ntpd
```

```
ssh slave1 systemctl disable ntpd
```

```
ssh slave1 ntpdate master
```

```
#设置 slave2 和 master 时钟同步
```

```
ssh slave2 systemctl stop ntpd
```

```
ssh slave2 systemctl disable ntpd
```

```
ssh slave2 ntpdate master
```

```
##设置 slave3 和 master 时钟同步
```

```
ssh slave3 systemctl stop ntpd
```

```
ssh slave3 systemctl disable ntpd
```

```
ssh slave3 ntpdate master
```

```
[root@master x86_64]# ssh slave1 ntpdate master
26 Aug 16:20:34 ntpdate[21356]: adjust time server 192.168.10.205 offset 0.000155 sec
```

步骤12 添加定时任务，每天 0 点从节点同步主节点时间。

```
ssh slave1
```

```
crontab -e
```

```
00 00 *** /usr/sbin/ntpdate master
```

```
exit
```

```
00 00 * * * /usr/sbin/ntpdate master
~
~
~
~
~
```

说明：分布登录 slave1 slave2 slave3 执行。

步骤13 在 master 节点添加第三方依赖。

```
yum -y install chkconfig python bind-utils psmisc libxslt zlib portmap sqlite cyrus-sasl-plain cyrus-sasl-gssapi fuse fuse-libs redhat-lsb httpd mod_ssl openssl-devel python-psycopg2 MySQL-python postgresql-server libaio autoconf
```

步骤14 为了避免冲突，检测是否有 mariadb 和 mysql。

```
rpm -qa | grep mariadb
rpm -qa | grep -i mysql
如果有先移除，如
rpm -e --nodeps mariadb-libs-5.5.60-1.el7_5.x86_64
```

步骤15 在 master 节点安装 mysql。

```
rpm -ivh MySQL-server-5.6.47-1.el7.x86_64.rpm
rpm -ivh MySQL-client-5.6.47-1.el7.x86_64.rpm
```

步骤16 加入到系统服务并设置开启启动。

```
chkconfig --add mysql
chkconfig mysql on
```

步骤17 启动 mysql。

```
service mysql start
```

步骤18 查看原始密码。

```
cat /root/.mysql_secret
[root@master x86_64]# cat /root/.mysql_secret
# The random password set for the root user at Tue Aug 24 20:10:24 2021 (local time): gLyc59mPIjzE
tmrM
```

步骤19 设置 mysql。

```
mysql_secure_installation
```

```

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]
... Success!

```

步骤20 登录 mysql。

```
mysql -uroot -pTest1234!
```

```

[root@master x86_64]# mysql -uroot -pTest1234!
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 952
Server version: 5.6.47 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

步骤21 创建相关数据库和授权。

```
Grant all privileges on . to 'root'@'%' identified by 'Test1234!' with grant option;
flush privileges;
```

```
create database if not exists amon default charset utf8 collate utf8_general_ci;
create database if not exists rman default charset utf8 collate utf8_general_ci;
create database if not exists nav default charset utf8 collate utf8_general_ci;
create database if not exists navms default charset utf8 collate utf8_general_ci;
create database if not exists hue default charset utf8 collate utf8_general_ci;
create database if not exists sentry default charset utf8 collate utf8_general_ci;
create database if not exists hive;
create database if not exists oozie;
create database if not exists scm;
grant all on hive.* to 'hive'@'%' identified by 'Test1234!' with grant option;
```

```
grant all on oozie.* to 'oozie'@'%' identified by 'Test1234!' with grant option;
grant all on hue.* to 'hue'@'%' identified by 'Test1234!' with grant option;
grant all on amon.* to 'amon'@'%' identified by 'Test1234!' with grant option;
grant all on rman.* to 'rman'@'%' identified by 'Test1234!' with grant option;
grant all on nav.* to 'nav'@'%' identified by 'Test1234!' with grant option;
grant all on navms.* to 'navms'@'%' identified by 'Test1234!' with grant option;
grant all on sentry.* to 'sentry'@'%' identified by 'Test1234!' with grant option;
grant all on *.* to 'scm'@'%' identified by 'Test1234!' with grant option;
flush privileges;
```

步骤22 退出 mysql，创建 jdbc 驱动共享目录，并拷贝驱动到共享目录（注意改名）。

```
mkdir -p /usr/share/java
cp mysql-connector-java-5.1.47.jar /usr/share/java/mysql-connector-java.jar
```

1.2.2. 安装 CDH

步骤1 在 master 节点上安装 cloudera manager 的 server 端。

```
rpm -ivh cloudera-manager-daemons-5.16.2-1.cm5162.p0.7.el7.x86_64.rpm
rpm -ivh cloudera-manager-server-5.16.2-1.cm5162.p0.7.el7.x86_64.rpm
rpm -ivh cloudera-manager-agent-5.16.2-1.cm5162.p0.7.el7.x86_64.rpm
```

步骤2 修改数据库配置文件。

```
cd /etc/cloudera-scm-server
vi db.properties
```

```
Copyright (c) 2012 Cloudera, Inc. All rights reserved.
#
# This file describes the database connection.
#
# The database type
# Currently 'mysql', 'postgres' and 'oracle' are valid databases.
com.cloudera.cmf.db.type=mysql
# The database host
# If a non standard port is needed, use 'hostname:port'
com.cloudera.cmf.db.host=master:3306
# The database name
com.cloudera.cmf.db.name=scm
# The database user
com.cloudera.cmf.db.user=scm
# The database user's password
com.cloudera.cmf.db.password=Test1234!
# The db setup type
# By default, it is set to INIT
# If scm-server uses Embedded DB then it is set to EMBEDDED
# If scm-server uses External DB then it is set to EXTERNAL
com.cloudera.cmf.db.setupType=EXTERNAL
~
```

说明：参考上图配置。

步骤3 启动服务。

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

步骤4 查看 cm 日志，大概过 1 分多钟可以看到服务已启动（可以看到 7180 端口）。

```
tail -F /var/log/cloudera-scm-server/cloudera-scm-server.log
```

```
2:14.286Z
INFO WebServerImpl:org.mortbay.log: jetty-6.1.26.cloudera.4
INFO WebServerImpl:org.mortbay.log: Started SelectChannelConnector[0.0.0.0:7180]
INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server.
```

步骤5 在 slave1~3 节点安装 agent（进入 rpm 包所在目录）。

```
rpm -ivh cloudera-manager-daemons-5.16.2-1.cm5162.p0.7.el7.x86_64.rpm
```

```
rpm -ivh cloudera-manager-agent-5.16.2-1.cm5162.p0.7.el7.x86_64.rpm
```

步骤6 在 slave1~3 节点上修改配置文件。

```
vi /etc/cloudera-scm-agent/config.ini
```

```
root@slave1:~# xianwang
[[General]]
# Hostname of the CM server.
server_host=master

# Port that the CM server is listening on.
server_port=7182

## It should not normally be necessary to modify these.
# Port that the CM agent should listen on.
```

说明：所有 server 指向 master。

步骤7 在 slave1~3 节点上启动 agent。

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

```
root@slave1:~# xianwang
[[General]]
# Hostname of the CM server.
server_host=master

# Port that the CM server is listening on.
server_port=7182

## It should not normally be necessary to modify these.
# Port that the CM agent should listen on.
```

步骤8 在 master 的 /var/www/html 下创建 cdh5_parcel 目录。

```
mkdir /var/www/html/cdh5_parcel
```

步骤9 进入 cdh/cdh5 目录，移动文件到 cdh5_parcel 目录。

```
mv CDH-5.16.2-1.cdh5.16.2.p0.8-el7.parcel /var/www/html/cdh5_parcel/
```

```
mv CDH-5.16.2-1.cdh5.16.2.p0.8-el7.parcel.sha /var/www/html/cdh5_parcel/
```

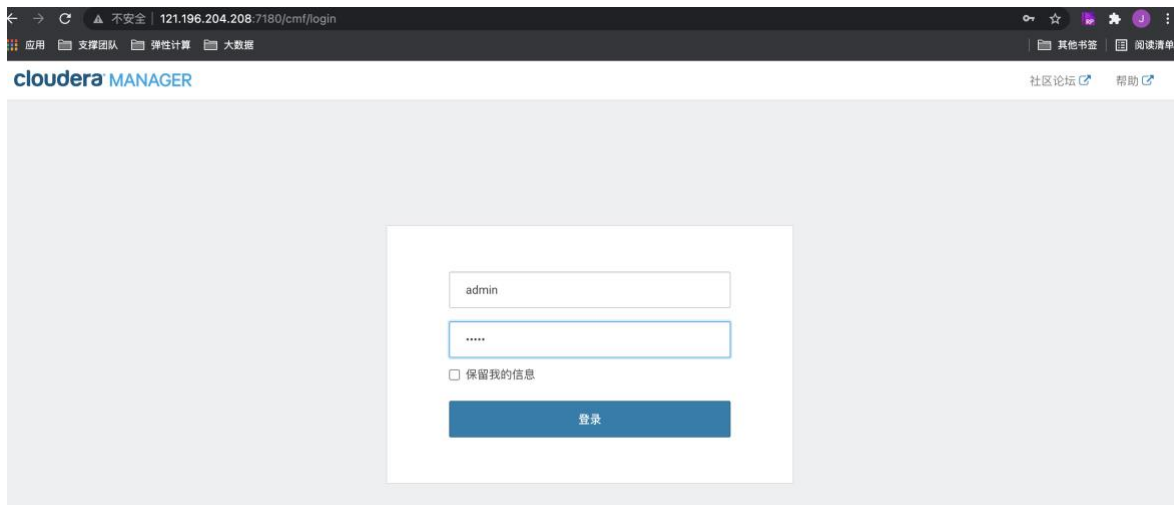
```
el.sha
```

```
mv manifest.json /var/www/html/cdh5_parcel/
```

步骤10 启动 httpd 服务。

```
service httpd start
```

步骤11 在浏览器中使用 master 节点 ip:7180，账号 admin，密码 admin 登录。



步骤12 接受协议进入安装流程。

Welcome to Cloudera Manager

End User License Terms and Conditions

Cloudera Standard License
Version 2016-05-26

END USER LICENSE TERMS AND CONDITIONS

THESE TERMS AND CONDITIONS (THESE "TERMS") APPLY TO YOUR USE OF THE PRODUCTS (AS DEFINED BELOW) PROVIDED BY CLOUDERA, INC. ("CLOUDERA").

PLEASE READ THESE TERMS CAREFULLY.

IF YOU ("YOU" OR "CUSTOMER") PLAN TO USE ANY OF THE PRODUCTS ON BEHALF OF A COMPANY OR OTHER ENTITY, YOU REPRESENT THAT YOU ARE THE EMPLOYEE OR AGENT OF SUCH COMPANY (OR OTHER ENTITY) AND YOU HAVE THE AUTHORITY TO ACCEPT ALL OF THE TERMS AND CONDITIONS SET FORTH IN AN ACCEPTED REQUEST (AS DEFINED BELOW) AND THESE TERMS (COLLECTIVELY, THE "AGREEMENT") ON BEHALF OF SUCH COMPANY (OR OTHER ENTITY).

BY USING ANY OF THE PRODUCTS, YOU ACKNOWLEDGE AND AGREE THAT:

(A) YOU HAVE READ ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT;
(B) YOU UNDERSTAND ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT;
(C) YOU AGREE TO BE LEGALLY BOUND BY ALL OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT

IF YOU DO NOT AGREE WITH ANY OF THE TERMS OR CONDITIONS OF THESE TERMS, YOU MAY NOT USE ANY PORTION OF THE PRODUCTS.

Yes, I accept the End User License Terms and Conditions. [接受协议](#)

If your download and use of Cloudera Manager are on behalf of a company that has an existing agreement with Cloudera for the use of the software, your action does not modify that existing agreement.

Thank you for choosing Cloudera Manager and CDH.

This installer will install **Cloudera Express 5.16.1** and enable you to later choose packages for the services below (there may be some license implications).

- Apache Hadoop (Common, HDFS, MapReduce, YARN)
- Apache HBase
- Apache ZooKeeper
- Apache Oozie
- Apache Hive
- Hue (Apache licensed)
- Apache Flume
- Apache Impala
- Apache Sentry
- Apache Sqoop
- Cloudera Search (Apache licensed)
- Apache Spark

You are using Cloudera Manager to install and configure your system. You can learn more about Cloudera Manager by clicking on the **Support** menu above.

Before you proceed, be sure to checkout the [CDH and Cloudera Manager Requirements and Supported Versions](#)

- [Supported Operating Systems](#)
- [Supported Databases](#)
- [Supported JDK Versions](#)

步骤13 勾选所有 4 个节点，进入下一步。

Specify hosts for your CDH cluster installation.

New Hosts Currently Managed Hosts (4)

These hosts do not belong to any clusters. Select some to form your cluster.

<input checked="" type="checkbox"/>	Name	IP	Rack	CDH Version
	<input type="text" value="Any Name"/>	<input type="text" value="Any IP"/>	<input type="text" value="Any Rack"/>	<input type="text" value="All"/>
<input checked="" type="checkbox"/>	...master	172.21.119.114	/default	None
<input checked="" type="checkbox"/>	...slave1	172.21.119.112	/default	None
<input checked="" type="checkbox"/>	...slave2	172.21.119.113	/default	None

步骤14 配置自己的 http 仓库。

Cluster Installation

Select Repository

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Choose Method

Use Packages

Use Parcels (Recommended) [More Options](#) [Proxy Settings](#)

CDH Version

CDH-5.16.2-1.cdh5.16.2.p0.8

CDH-4.7.1-1.cdh4.7.1.p0.47

Versions of CDH that are too new for this version of Cloudera Manager (5.16.1) will not be shown.

Additional Parcels

ACCUMULO-1.7.2-5.5.0.ACCUMULO5.5.0.p0.8

ACCUMULO-1.4.4-1.cdh4.5.0.p0.65

None

IMPALA-2.1.0-1.impala2.0.0.p0.1995

None

KAFKA-4.1.0-1.4.1.0.p0.4

None

KUDU-1.4.0-1.cdh5.12.2.p0.8

None

SOLR-1.3.0-1.cdh4.5.0.p0.9

None

SPARK-0.9.0-1.cdh4.6.0.p0.98

None

配置我们自己的http 仓库

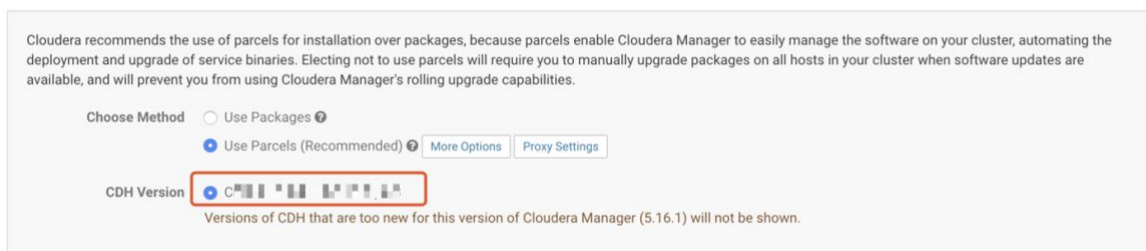
步骤15 配置自己的 http 仓库地址。



步骤16 选择 5.16.2 版本进行安装。

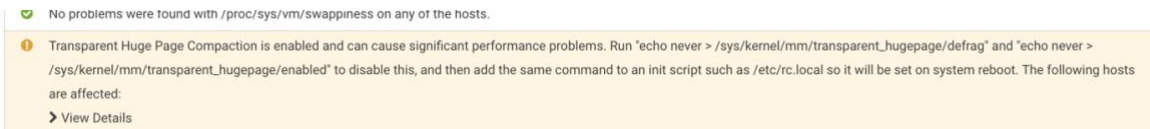
Cluster Installation

Select Repository



进入下一步安装，如果遇到警告，执行以下命令：

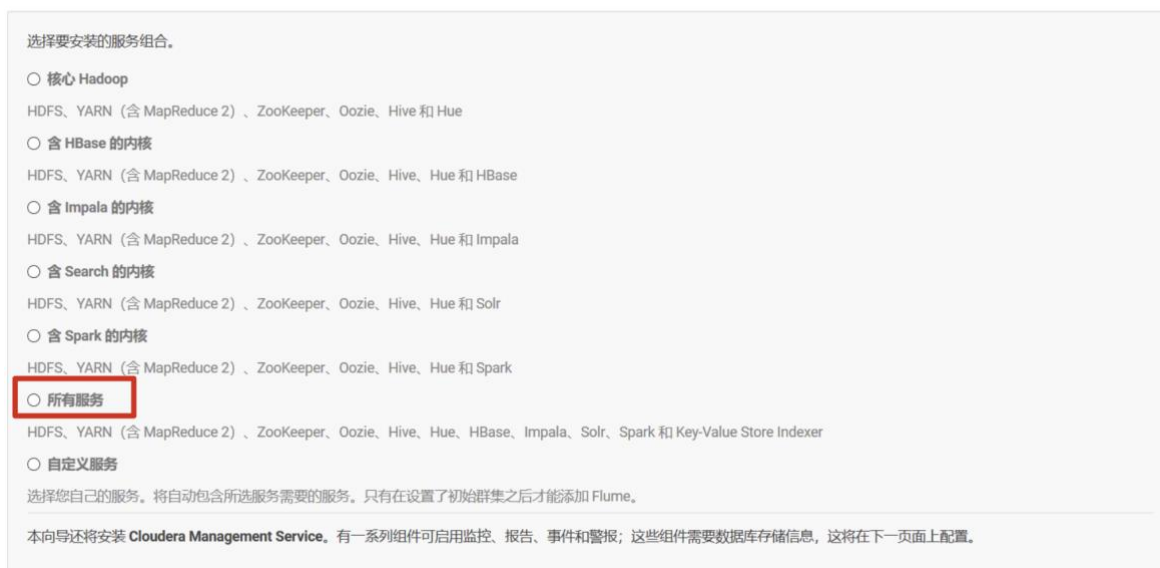
```
echo never > /sys/kernel/mm/transparent_hugepage/defrag  
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```



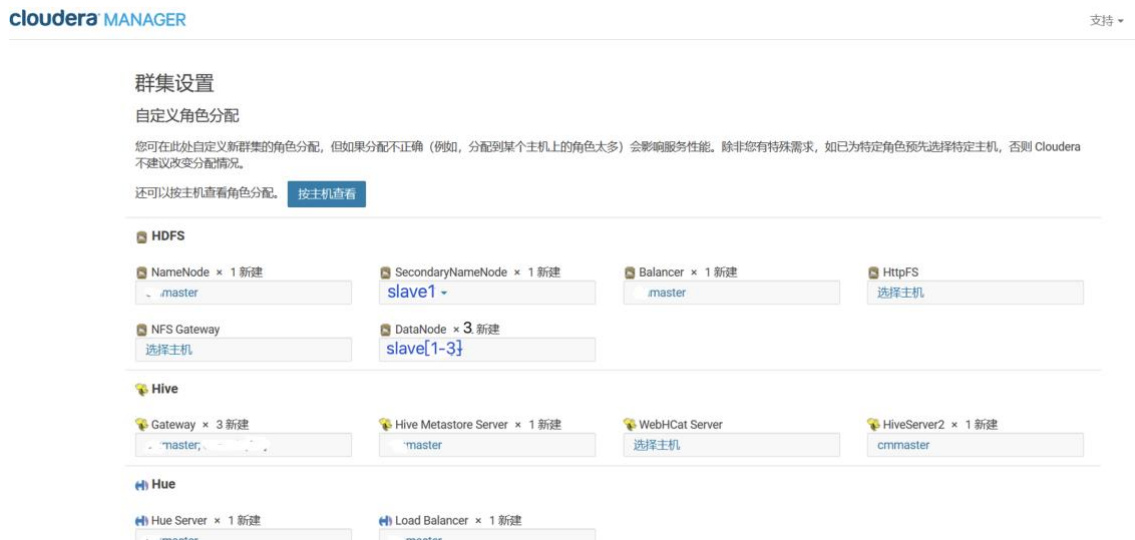
步骤17 选择安装所有服务。

群集设置

Select Services

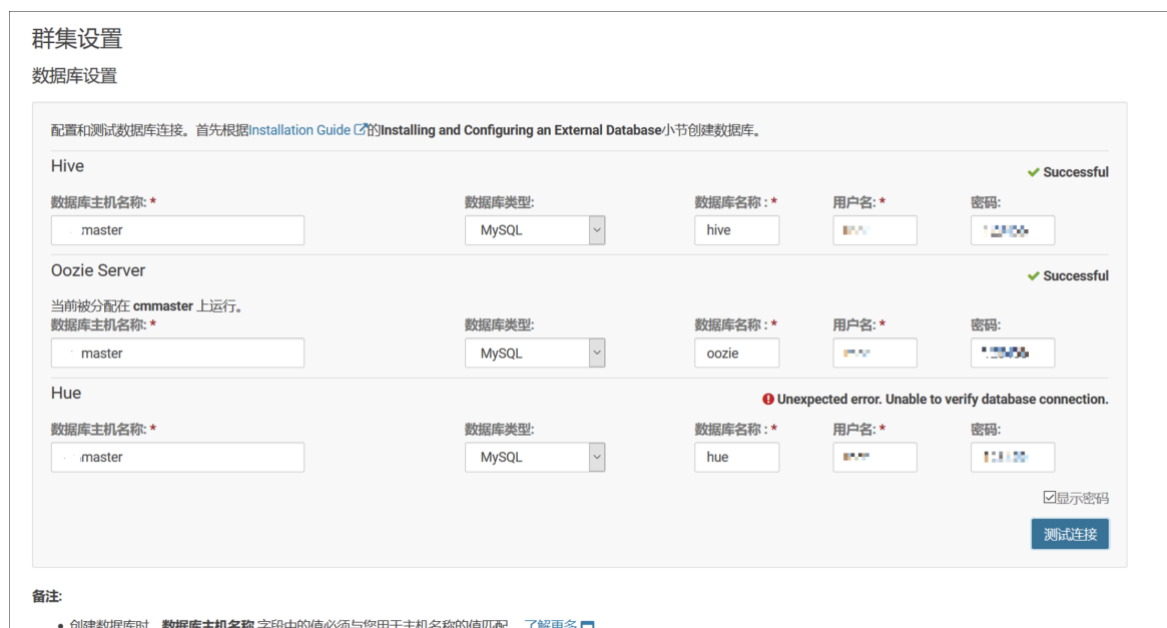


步骤18 进行集群设置，和数据库相关的配置必须安装在 master 上。



步骤19 配置数据库信息。

如果 hue 数据库无法连接，请参考 <https://www.cnblogs.com/wuning/p/11738697.html> 解决。



步骤20 采用默认配置安装。

cloudera MANAGER 支持 ▾

群集设置

审核更改

HDFS 块大小 <small>dfs.block.size, dfs.blocksize</small>	Cluster 1 > HDFS (服务范围) ?
<input type="text" value="128"/> 兆字节 ▾	
接受的 DataNode 失败的卷 <small>dfs.datanode.failed.volumes.tolerated</small> 编辑单个值	Cluster 1 > DataNode Default Group ...和另 1 个 ?
<input type="text" value="0"/>	
DataNode 数据目录 <small>dfs.data.dir, dfs.datanode.data.dir</small> 编辑单个值	Cluster 1 > DataNode Default Group ...和另 1 个 ?
<input type="text" value="/dfs/dn"/> ⊞	
NameNode 数据目录 <small>dfs.name.dir, dfs.namenode.name.dir</small>	Cluster 1 > NameNode Default Group ?
<input type="text" value="/dfs/nn"/> ⊞	
HDFS 检查点目录 <small>fs.checkpoint.dir, dfs.namenode.checkpoint.dir</small>	Cluster 1 > SecondaryNameNode Default Group ?
<input type="text" value="/dfs/snn"/> ⊞	

步骤21 采用默认配置安装。

cloudera MANAGER 支持 ▾

群集设置

审核更改

HDFS 块大小 <small>dfs.block.size, dfs.blocksize</small>	Cluster 1 > HDFS (服务范围) ?
<input type="text" value="128"/> 兆字节 ▾	
接受的 DataNode 失败的卷 <small>dfs.datanode.failed.volumes.tolerated</small> 编辑单个值	Cluster 1 > DataNode Default Group ...和另 1 个 ?
<input type="text" value="0"/>	
DataNode 数据目录 <small>dfs.data.dir, dfs.datanode.data.dir</small> 编辑单个值	Cluster 1 > DataNode Default Group ...和另 1 个 ?
<input type="text" value="/dfs/dn"/> ⊞	
NameNode 数据目录 <small>dfs.name.dir, dfs.namenode.name.dir</small>	Cluster 1 > NameNode Default Group ?
<input type="text" value="/dfs/nn"/> ⊞	
HDFS 检查点目录 <small>fs.checkpoint.dir, dfs.namenode.checkpoint.dir</small>	Cluster 1 > SecondaryNameNode Default Group ?
<input type="text" value="/dfs/snn"/> ⊞	

步骤22 进行安装，等待完成。

群集设置

首次运行 命令

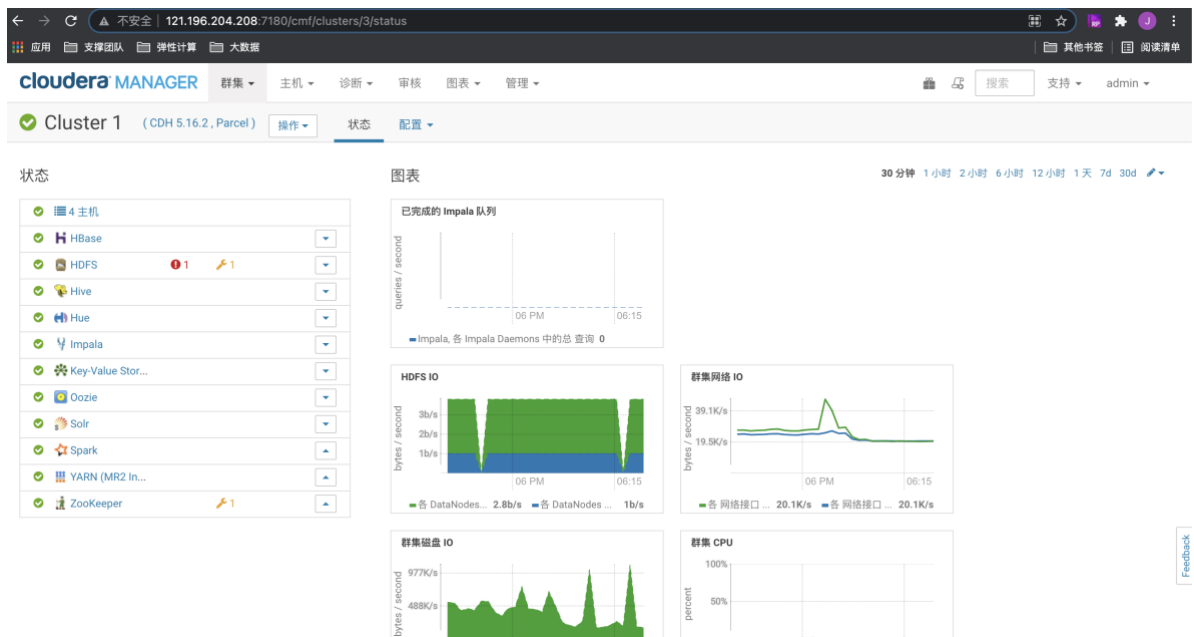
状态 正在运行 1月19, 1:42:58 凌晨

✓ 已完成 0 个步骤 (共 8 个)。

 Show All Steps Show Only Failed Steps Show Running Steps

<input type="radio"/> Ensuring that the expected software releases are installed on hosts. 已完成 0/1 个步骤。 <input type="radio"/> 正在部署客户端配置	1月19, 1:42:58 凌晨
<input type="radio"/> 启动 Cloudera Management Service, ZooKeeper	
<input type="radio"/> 启动 HDFS	
<input type="radio"/> 启动 YARN (MR2 Included)	
<input type="radio"/> 启动 Hive	
<input type="radio"/> 启动 Oozie	
<input type="radio"/> 启动 Hue	

步骤23 完成安装后，进入主页查看集群运行情况。

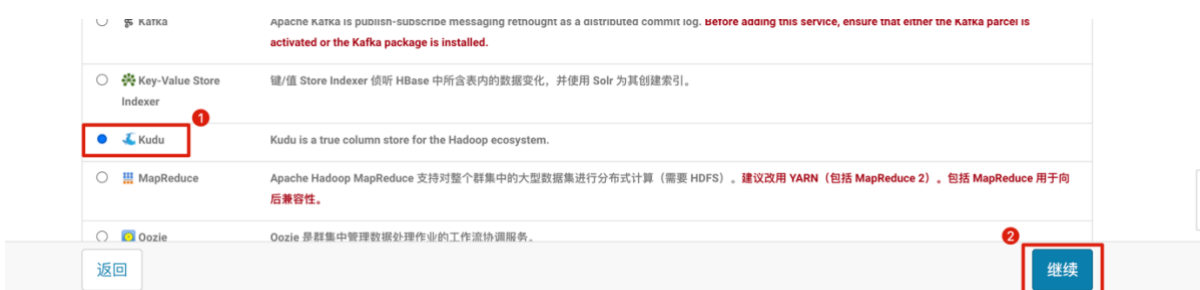


1.2.3. 安装 Kudu

步骤1 由于 CDH 默认没有安装 Kudu 组件，先添加服务。



步骤2 选择 kudu 继续安装。



步骤3 选择 master 节点作为 Master，Slave1~3 作为 table server。



步骤4 配置 Kudu 相关的目录地址。

审核更改

Kudu Master WAL Directory fs_wal_dir Master Default Group 1
/home/data/kudu/wal
Missing required value: Kudu Master WAL Directory

Kudu Master Data Directories fs_data_dirs Master Default Group 2
/home/data/kudu/data

Kudu Tablet Server WAL Directory fs_wal_dir Tablet Server Default Group 3
/home/data/kudu/wal
Missing required value: Kudu Tablet Server WAL Directory

Kudu Tablet Server Data Directories fs_data_dirs Tablet Server Default Group 4
/home/data/kudu/data
Missing required value: Kudu Tablet Server Data Directories

为 Master 启用 TLS/SSL Master Default Group

Master TLS/SSL 服务器私钥文件 (PEM 格式) webserver_private_key_file Master Default Group 5

返回 1 2 3 4 继续

步骤5 进行安装等待完成。

将 Kudu 服务添加到 Cluster 1

首次运行 命令

状态 ● 已完成 📅 8月30, 6:13:45 晚上 🕒 22.38s

Finished First Run of the following services successfully: Kudu.

✓ 已完成 2 个步骤 (共 2 个)

Show All Steps Show Only Failed Steps Show Running Steps

Ensuring that the expected software releases are installed on hosts.	8月30, 6:13:45 晚上	15ms
启动 Kudu	Kudu	8月30, 6:13:45 晚上 22.36s

返回 1 2 3 4 继续

步骤6 完成安装。

将 Kudu 服务添加到 Cluster 1

恭喜您!

您的新服务已安装在群集中并已配置。

注意：您可能仍必须启动新服务。建议您在执行此操作之前，先重启所有配置过时的依赖关系服务。可以通过单击下面的完成在主页面上执行这些操作。

[返回](#)

1 2 3 4

[完成](#)

Feedback

```
1/09/06 10:29:58 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 23.0 (TID 51, slave1, executor 1, partition 0, NODE_LOCAL, 8782 bytes)
1/09/06 10:29:59 INFO storage.BlockManagerInfo: Added broadcast_23_piece0 in memory on slave1:46691 (size: 69.2 KB free: 366.0 MB)
1/09/06 10:29:59 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 23.0 (TID 51) in 115 ms on slave1 (executor 1) (1/1)
1/09/06 10:29:59 INFO cluster.YarnScheduler: Removed TaskSet 23.0, whose tasks have all completed, from pool
1/09/06 10:29:59 INFO scheduler.DAGScheduler: ResultStage 23 (save at KuduBackup.scala:96) finished in 0.131 s
1/09/06 10:29:59 INFO scheduler.DAGScheduler: Job 23 finished: save at KuduBackup.scala:96, took 0.132379 s
1/09/06 10:29:59 INFO datasources.FileFormatWriter: Write Job 358b4cfa-deab-4f41-bde1-3d072c907405 committed.
1/09/06 10:29:59 INFO datasources.FileFormatWriter: Finished processing stats for write job 358b4cfa-deab-4f41-bde1-3d072c907405.
1/09/06 10:29:59 INFO backup.BackupIO: Writing metadata to hdfs://kudu-backups/6f0186e896dc43d6848e668b4ece809a-impala%3A%3Atpcds_kudu_2.web_site/1630895350692/.kudu-metadata.json
1/09/06 10:29:59 INFO backup.KuduBackups: Successfully backed up table impala:tpcds_kudu_2.web_site
1/09/06 10:29:59 INFO server.AbstractConnector: Stopped Spark@622ef26a{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
1/09/06 10:29:59 INFO ui.SparkUI: Stopped Spark web UI at http://slave1:4040
1/09/06 10:29:59 INFO cluster.YarnClientSchedulerBackend: Interrupting monitor thread
1/09/06 10:29:59 INFO cluster.YarnClientSchedulerBackend: Shutting down all executors
1/09/06 10:29:59 INFO cluster.YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
1/09/06 10:29:59 INFO cluster.SchedulerExtensionServices: Stopping SchedulerExtensionServices
serviceOption=None,
services=List(),
started=false)
```

```
[root@slave1 ~]# sudo -u hdfs hadoop distcp hdfs:///kudu-backups/* hdfs://192.168.10.203/kudu-backups/
21/09/06 10:46:57 INFO tools.OptionsParser: parseChunkSize: blocksperchunk false
21/09/06 10:46:58 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=false, append=false, useDiff=false, useRdiff=false, fromSnapshot=null, toSnapshot=null, skipCRC=false, blocking=true, numListStatusThreads=0, maxMaps=20, mapBandwidth=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[hdfs://kudu-backups/*], targetPath=hdfs://192.168.10.203/kudu-backups, targetPathExists=false, filtersFile='null', blocksPerChunk=0, copyBufferSize=8192}
21/09/06 10:46:58 INFO client.RMProxy: Connecting to ResourceManager at master/192.168.10.205:8032
21/09/06 10:46:58 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 148; dirCnt = 48
21/09/06 10:46:58 INFO tools.SimpleCopyListing: Build file listing completed.
21/09/06 10:46:58 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
21/09/06 10:46:58 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
21/09/06 10:46:59 INFO tools.DistCp: Number of paths in the copy list: 148
21/09/06 10:46:59 INFO tools.DistCp: Number of paths in the copy list: 148
21/09/06 10:46:59 INFO client.RMProxy: Connecting to ResourceManager at master/192.168.10.205:8032
21/09/06 10:46:59 INFO mapreduce.JobSubmitter: number of splits:22
21/09/06 10:46:59 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1630669007611_0008
21/09/06 10:46:59 INFO impl.YarnClientImpl: Submitted application application_1630669007611_0008
21/09/06 10:46:59 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1630669007611_0008/
21/09/06 10:46:59 INFO tools.DistCp: DistCp job-id: job_1630669007611_0008
21/09/06 10:46:59 INFO mapreduce.Job: Running job: job_1630669007611_0008
21/09/06 10:47:04 INFO mapreduce.Job: Job job_1630669007611_0008 running in uber mode : false
21/09/06 10:47:04 INFO mapreduce.Job: map 0% reduce 0%
21/09/06 10:47:08 INFO mapreduce.Job: map 9% reduce 0%
21/09/06 10:47:09 INFO mapreduce.Job: map 50% reduce 0%
21/09/06 10:47:12 INFO mapreduce.Job: map 64% reduce 0%
21/09/06 10:47:13 INFO mapreduce.Job: map 100% reduce 0%

Bytes Read=63927
File Output Format Counters
Bytes Written=0
DistCp Counters
  Bytes Copied=856175431
  Bytes Expected=856175431
  Files Copied=148
```

1.2.4. 升级 JDK 到 1.8 版本

说明：本方案使用的 Kudu 迁移工具依赖 Spark2，Spark2.4 版本依赖 JDK1.8，故先对集群的 JDK 进行升级，如果原集群已经是 JDK1.8 的可略过此步骤。

步骤1 在/usr/java/目录下创建 jdk1.8.0_301 目录。

```
mkdir /usr/java/jdk1.8.0_301
```

步骤2 拷贝前面已经下载好的安装包到/usr/java/jdk1.8.0_301 目录。

```
cp -r cdh/jdk1.8.0_301/* /usr/java/jdk1.8.0_301
```

步骤3 把 JDK1.8 分发的所有 slave1 节点上。

```
ssh slave1 mkdir /usr/java/jdk1.8.0_301
scp -r cdh/jdk1.8.0_301/* root@slave1:/usr/java/jdk1.8.0_301
ssh slave2 mkdir /usr/java/jdk1.8.0_301
scp -r cdh/jdk1.8.0_301/* root@slave2:/usr/java/jdk1.8.0_301
ssh slave3 mkdir /usr/java/jdk1.8.0_301
scp -r cdh/jdk1.8.0_301/* root@slave3:/usr/java/jdk1.8.0_301
```

步骤4 给与/bin/java 执行权限。

```
chmod +x /usr/java/jdk1.8.0_301/bin/java
```

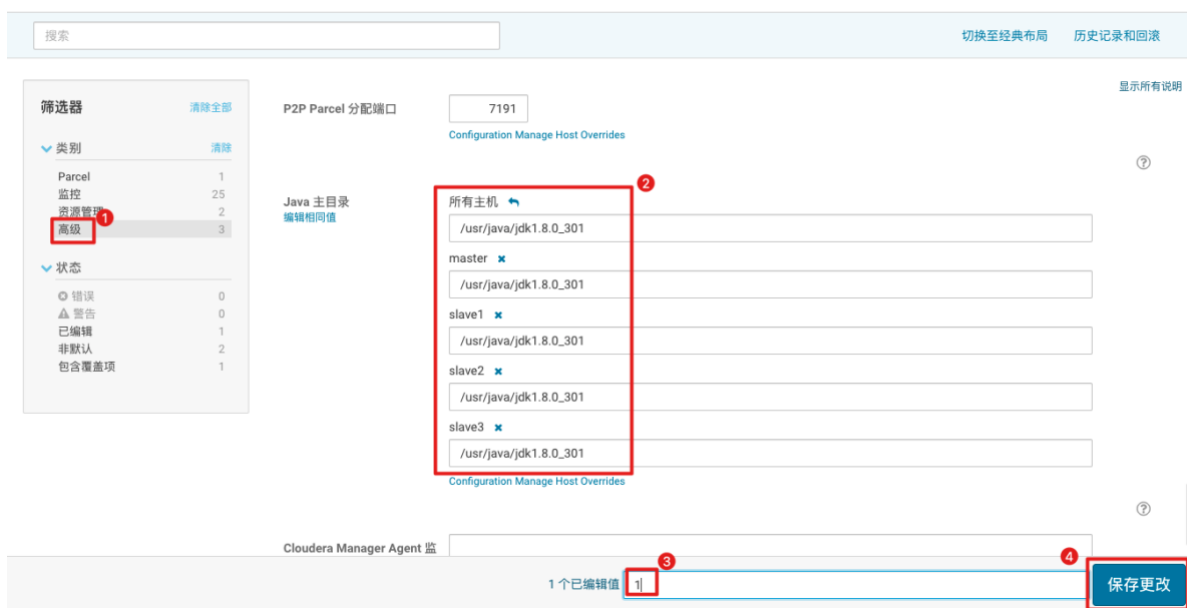


```
ssh slave1 chmod +x /usr/java/jdk1.8.0_301/bin/java  
ssh slave2 chmod +x /usr/java/jdk1.8.0_301/bin/java  
ssh slave3 chmod +x /usr/java/jdk1.8.0_301/bin/java
```

步骤5 进入主机菜单配置 JDK 所在目录。



步骤6 配置所有主机的 JAVA 主目录为/usr/java/jdk1.8.0_301。



步骤7 重启服务，代各组件恢复正常工作。



1.2.5. 安装 Spark2

说明：由于 Kudo 迁移工具需 Spark2 支持，CDH5 可以支持 Spark 和 Spark2 两个环境同时运行，如果集群上没有 Spark2 请先进行安装。已经安装的略过此步骤。

另外，kudu 迁移也可以使用 kudu table copy 来实现，也不需要安装 Spark2。

步骤8 把 Sparks2 的安装包移动到/opt/cloudera/csd/目录下。

```
mv cdh/spark2/* /opt/cloudera/csd/
```

步骤9 查看文件以及成功拷贝到/opt/cloudera/csd/目录下。

```
[root@master csd]# ls
manifest.json
SPARK2-2.4.0.cloudera2-1.cdh5.13.3.p0.1041012-el7.parcel
SPARK2-2.4.0.cloudera2-1.cdh5.13.3.p0.1041012-el7.parcel.sha1
SPARK2_ON_YARN-2.4.0.cloudera2.jar
```

步骤10 给文件授权。

```
[root@master csd]# chown -R cloudera-scm:cloudera-scm /opt/cloudera/csd/SPARK2_ON_YARN-2.4.0.cloudera2.jar
[root@master csd]# chmod -R 644 /opt/cloudera/csd/SPARK2_ON_YARN-2.4.0.cloudera2.jar
```

步骤11 重启 scm 服务。

```
sudo systemctl restart cloudera-scm-server
```

```
root@master csd]# sudo systemctl restart cloudera-scm-server
```

步骤12 新建并拷贝安装包到 spark2_parcel 目录。

```
sudo mkdir /var/www/html/spark2_parcel
sudo mv *.parcel* /var/www/html/spark2_parcel
sudo mv manifest.json /var/www/html/spark2_parcel
sudo chmod -R ugo+rX /var/www/html/spark2_parcel
```

```
root@master csd]# sudo mkdir /var/www/html/spark2_parcel
root@master csd]# sudo mv *.parcel* /var/www/html/spark2_parcel
root@master csd]# sudo mv manifest.json /var/www/html/spark2_parcel
root@master csd]# sudo chmod -R ugo+rX /var/www/html/spark2_parcel
```

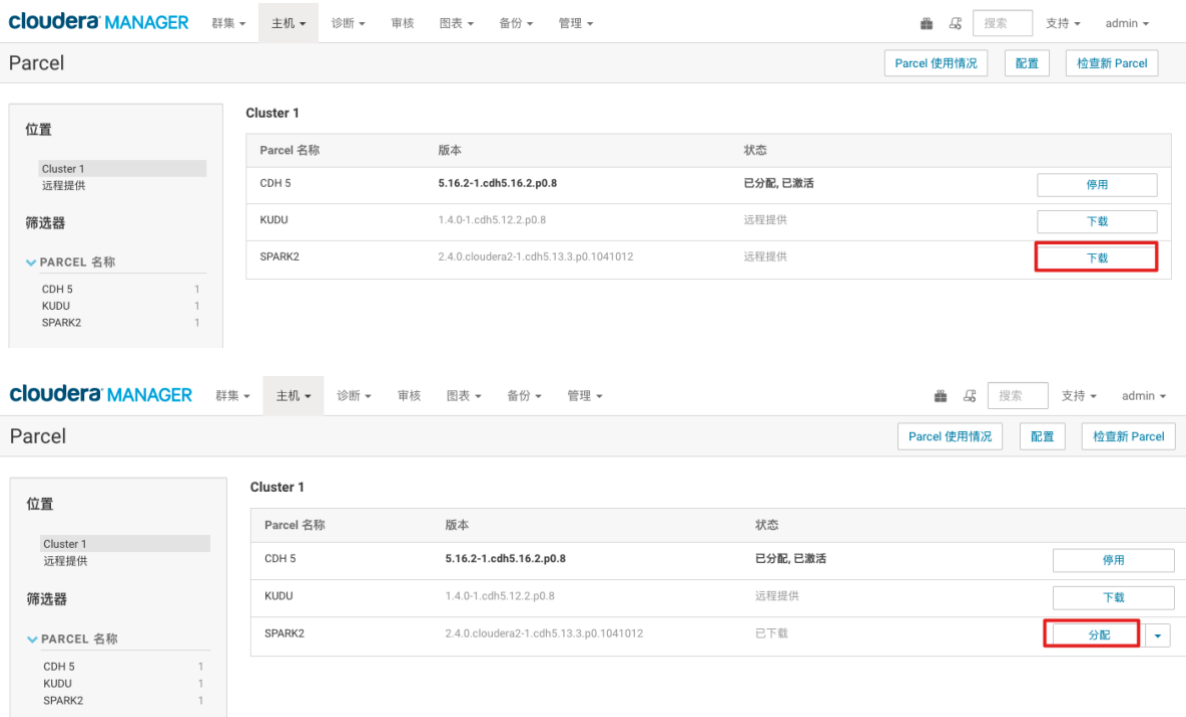
步骤13 配置 Parcel，填写前面建立的目录作为 Parcel 存储 URL。

The screenshot shows the Cloudera Manager interface. In the top navigation menu, the '主机' (Hosts) dropdown is open, and the 'Parcel' option is highlighted. Below this, the 'Parcel' configuration page is displayed. The 'Parcel 设置' (Parcel Settings) modal is open, showing the following configuration fields:

- Parcel 目录: /opt/cloudera/parcels
- 本地 Parcel 存储库路径: /opt/cloudera/parcel-repo
- Parcel 更新频率: 1 小时
- 远程 Parcel 存储库 URL: http://master/spark2_parcel/ (highlighted with a red box)
- Other URLs: http://archive.cloudera.com/spark2/parcels/2.4.0.cloudera2/, https://archive.cloudera.com/accumulo/parcels/1.4/, https://archive.cloudera.com/accumulo-c5/parcels/latest/

Buttons at the bottom include '更改原因...', '取消', and '保存更改'.

步骤14 下载并分配 Spark2 安装包到各节点



步骤15 激活安装。



步骤16 开始添加 Spark2 服务。



Cloudera Manager service selection interface. The 'Spark 2' option is selected and highlighted with a red box and a '1' callout. A '继续' (Continue) button is highlighted with a red box and a '2' callout.

步骤17 选择节点和配置进行安装。

Cloudera Manager interface showing the configuration for adding Spark 2 service to Cluster 1. The page title is "将 Spark 2 服务添加到 Cluster 1" and the subtitle is "为新的 Spark 2 选择一组依赖关系".

HBase	HDFS	Hive	Spark	YARN (MR2 Included)	ZooKeeper	
<input checked="" type="radio"/>	HDFS			YARN (MR2 Included)	ZooKeeper	
<input type="radio"/>	HBase	HDFS	Hive	Spark	YARN (MR2 Included)	ZooKeeper
<input type="radio"/>	HBase	HDFS		YARN (MR2 Included)	ZooKeeper	

The '继续' (Continue) button is highlighted with a red box.

cloudera MANAGER

将 Spark 2 服务添加到 Cluster 1

自定义 Spark 2 的角色分配

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确（例如，分配到某个主机上的角色太多），性能受到影响。

还可以按主机查看角色分配。 [按主机查看](#)

History Server × 1 新建

Gateway × 1 新建

[返回](#) 1 2 3 4 5 6 [继续](#)

Feedback

cloudera MANAGER

支持 admin

将 Spark 2 服务添加到 Cluster 1

审核更改

为 History Server 启用 TLS/SSL
spark.ssl.historyServer.enabled ?

History Server TLS/SSL 服务器 JKS Keystore 文件位置
spark.ssl.historyServer.keyStore ?

History Server TLS/SSL 服务器 JKS Keystore 文件密码
 ?

[返回](#) 1 2 3 4 5 [继续](#)

Feedback

将 Spark 2 服务添加到 Cluster 1

首次运行 命令

状态 ✅ 已完成 自 9月 3, 7:40:01 晚上 ⌚ 56.14s

Finished First Run of the following services successfully: Spark 2.

✅ 已完成 4 个步骤 (共 4 个)。

Show All Steps Show Only Failed Steps Show Running Steps

▶️ ✅ Ensuring that the expected software releases are installed on hosts.		9月 3, 7:40:01 晚上	92ms
▶️ ✅ 正在部署客户端配置 Successfully deployed all client configurations.	Cluster 1	9月 3, 7:40:02 晚上	16.47s
▶️ ✅ 并行执行 (spark_on_yarn,hdfs,hive,hbase,solr,yarn,spark2_on_yarn) 的 DeployClusterClientConfig.		9月 3, 7:40:02 晚上	16.47s
▶️ ✅ 依次运行 2 步骤 在服务 Spark 2 上成功执行命令 Create Spark History Log Dir		9月 3, 7:40:18 晚上	17.18s
▶️ ✅ 执行服务 Spark 2 上的命令 Create Spark User Dir	Spark 2	9月 3, 7:40:18 晚上	8.48s
▶️ ✅ 执行服务 Spark 2 上的命令 Create Spark History Log Dir Successfully created HDFS directory.	Spark 2	9月 3, 7:40:27 晚上	8.68s
▶️ ✅ 使用用户 spark 组 spark 和权限 1777 创建 DFS 目录 /user/spark/spark2ApplicationHistory	NameNode (master)	9月 3, 7:40:27 晚上	8.68s
▶️ ✅ 启动 Spark 2 Successfully started service.	Spark 2	9月 3, 7:40:35 晚上	22.22s
▶️ ✅ 正在启动服务上的 1 角色		9月 3, 7:40:35 晚上	22.22s

返回

1 2 3 4 5

继续

步骤18 完成安装并查看 Spark2 服务是否正常。

将 Spark 2 服务添加到 Cluster 1

恭喜您!

您的新服务已安装在群集中并已配置。

注意: 您可能仍必须启动新服务。建议您在执行此操作之前, 先重启所有配置过时的依赖关系服务。可以通过单击下面的完成在主页面上执行这些操作。

返回

1 2 3 4 5

完成

1.3. 假数据构造

1.3.1. 构造 HDFS 和 Hive 测试数据

步骤1 使用以下命令安装 maven 环境（建议从 democode.txt 中拷贝）。

```
yum -y install gcc gcc-c++  
wget http://archive.apache.org/dist/maven/maven-3/3.6.0/binaries/apache-maven-3.6.0-bin.tar.gz  
tar xzf apache-maven-3.6.0-bin.tar.gz  
mv apache-maven-3.6.0 /usr/local/maven3
```

步骤2 使用以下命令配置 maven 环境变量。

```
export M2_HOME=/usr/local/maven3  
export PATH=$PATH:$M2_HOME/bin  
source /etc/profile  
可以通过 mvn -version 查看 maven 已成功安装。
```

```
[root@master ~]# mvn -version  
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5fffb20918da4f719f3; 2018-10-25T02:41:47+08:00)  
Maven home: /usr/local/maven3  
Java version: 1.7.0_67, vendor: Oracle Corporation, runtime: /usr/java/jdk1.7.0_67-cloudera/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "3.10.0-957.21.3.el7.x86_64", arch: "amd64", family: "unix"
```

步骤3 使用以下命令下载 tpchds 测试环境安装包。

```
yum -y install git  
git clone https://github.com/hortonworks/hive-testbench.git  
http://maven.aliyun.com/nexus/content/groups/public/
```

步骤4 进入 tpchds-gen 所在目录。

```
cd hive-testbench/tpchds-gen  
可能原来的国外地址无法下载，替换为阿里云的 mvn 仓库  
http://maven.aliyun.com/nexus/content/groups/public/
```



```

<pluginRepositories>
  <pluginRepository>
    <id>central</id>
    <name>Central Repository</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public</url>
    <layout>default</layout>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
    <releases>
      <updatePolicy>never</updatePolicy>
    </releases>
  </pluginRepository>
</pluginRepositories>
<repositories>
  <repository>
    <id>central</id>
    <name>Central Repository</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public</url>
    <layout>default</layout>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>

```

步骤5 使用以下命令编译 tpcds 测试环境。

```

cd ..
export HADOOP_USER_NAME=hdfs
yum -y install unzip patch
./tpcds-build.sh

```

```

[root@master hive-testbench]# ./tpcds-build.sh
Building TPC-DS Data Generator
make: Nothing to be done for `all'.
TPC-DS Data Generator built, you can now use tpcds-setup.sh to generate data.

```

步骤6 使用以下命令生产测试数据。

```

#打开调试信息, 可以排查执行错误问题
export DEBUG_SCRIPT=X
#生产 orc 表, 2 表示 2G 数据, 后面的目录为 hdfs 路径
./tpcds-setup.sh 2 /extwarehouse/tpcds
#生产 textfile 表, 2 表示 2G 数据, 后面的目录为 hdfs 路径
FORMAT=textfile ./tpcds-setup.sh 2 /extwarehouse/tpcds

```

步骤7 Hadoop fs 命令查看测试数据相关的文件目录都已经生成。

```

hadoop fs -ls /extwarehouse/tpcds/2

```

```
[root@master hive-testbench]# hadoop fs -ls /extwarehouse/tpcds/2
Found 26 items
-rwxrwxrwx  3 hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/ SUCCESS
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/call_center
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/catalog_page
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/catalog_returns
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/catalog_sales
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/customer
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/customer_address
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/customer_demographics
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/date_dim
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/dbgen_version
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/household_demographics
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/income_band
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/inventory
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/item
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/promotion
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/reason
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/ship_mode
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/store
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/store_returns
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/store_sales
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/time_dim
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/warehouse
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/web_page
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/web_returns
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/web_sales
drwxrwxrwx  - hdfs supergroup    0 2021-08-30 10:46 /extwarehouse/tpcds/2/web_site
```

步骤8 进一步查看文件中的演示数据，都已正常生成。

```
hadoop fs -cat /extwarehouse/tpcds/2/call_center/data-m-00001
```

```
[root@master hive-testbench]# hadoop fs -cat /extwarehouse/tpcds/2/call_center/data-m-00001
1|AAAAAAAAABAAAAAAAA|1998-01-01||2450952|NY Metro|large|23|13087|8AM-4PM|Bob Belcher|6|More than other authori|Shared others could not c
ount fully dollars. New members ca|Julius Tran|3|pri|6|cally|730|Ash Hill|Boulevard|Suite 0|Midway|Williamson County|TN|31904|United St
ates|-5|0.11|
2|AAAAAAAAACA|1998-01-01|2000-12-31||2450806|Mid Atlantic|medium|6|2268|8AM-8AM|Felipe Perkins|2|A bit narrow forms matter animals
. Consist|Largely blank years put substantially deaf, new others. Question|Julius Durham|5|anti|1|ought|984|Center Hill|Way|Suite 70|Mi
dway|Williamson County|TN|31904|United States|-5|0.12|
3|AAAAAAAAACA|2001-01-01||2450806|Mid Atlantic|medium|6|4134|8AM-4PM|Mark Hightower|2|Wrong troops shall work sometimes in a opti
|Largely blank years put substantially deaf, new others. Question|Julius Durham|1|ought|2|able|984|Center Hill|Way|Suite 70|Midway|Will
iamson County|TN|31904|United States|-5|0.01|
4|AAAAAAAAAE|1998-01-01|2000-01-01||2451063|North Midwest|medium|1|649|8AM-4PM|Larry Mccray|2|Dealers make most historical, direc
t students|Rich groups catch longer other fears; future,|Matthew Clifton|4|ese|3|pri|463|Pine Ridge|RD|Suite U|Midway|Williamson County
|TN|31904|United States|-5|0.05|
5|AAAAAAAAAE|2000-01-02|2001-12-31||2451063|North Midwest|small|10|2650|8AM-8AM|Larry Mccray|2|Dealers make most historical, dire
ct students|Blue, due beds come. Politicians would not make far thoughts. Specifically new horses partic|Gary Colburn|4|ese|3|pri|463|P
ine Ridge|RD|Suite U|Midway|Williamson County|TN|31904|United States|-5|0.12|
6|AAAAAAAAAE|2002-01-01||2451063|North Midwest|medium|14|7028|8AM-4PM|Larry Mccray|5|Silly particles could pro|Blue, due beds co
me. Politicians would not make far thoughts. Specifically new horses partic|Gary Colburn|5|anti|3|pri|463|Pine Ridge|RD|Suite U|Midway|
Williamson County|TN|31904|United States|-5|0.11|
7|AAAAAAAAAH|1998-01-01||2451024|Pacific Northwest|small|8|2216|8AM-4PM|Alden Snyder|6|Major, formal states can suppor|Reduced,
subsequent bases could not lik|Frederick Weaver|5|anti|4|ese|415|Jefferson Tenth|Court|Suite 180|Midway|Williamson County|TN|31904|Unit
ed States|-5|0.00|
8|AAAAAAAAAT|1998-01-01|2000-12-31||2450808|California|small|6|3096|8AM-12AM|Wayne Ray|6|Here possible notions arrive only. Ar|Co
mmon, free creditors should exper|Daniel Weller|5|anti|2|able|550|Cedar Elm|Ct.|Suite I|Fairview|Williamson County|TN|35709|United Stat
es|-5|0.06|
```

步骤9 进入 hive 命令行，查看 hive 表及 hive 数据，可见正常查看。

```
hive
use tpcds_text_2;
show tables;
select * from call_center limit 10;
```

```

Time taken: 1.144 seconds, Fetched: 0 row(s)
hive> use tpcds_text_2;
OK
Time taken: 0.017 seconds
hive> show tables;
OK
call_center
catalog_page
catalog_returns
catalog_sales
customer
customer_address
customer_demographics
date_dim
hbase_hive_tabale_item
household_demographics
income_band
inventory
item
promotion
reason
ship_mode
store
store_returns
store_sales
time_dim
warehouse
web_page
web_returns
web_sales
web_site
Time taken: 0.015 seconds, Fetched: 25 row(s)

```

```

hive> select * from call_center limit 10;
OK
1  AAAAAAAAAAAAAAAAAA  1998-01-01  NULL  NULL  2450952 NY Metro  large 23 13087 8AM-4PM  B
ob Belcher 6 More than other authori Shared others could not count fully dollars. New member
s ca Julius Tran 3 pri 6 cally 730 Ash Hill B
oulevard Suite 0 Midway Williamson County TN 31904 United States -5 0.11
2  AAAAAAAAAACAAAAAAA  1998-01-01  2000-12-31  NULL  2450806 Mid Atlantic medium 6 2268 8AM-8AM
Felipe Perkins 2 A bit narrow forms matter animals. Consist Largely blank years put substantially deaf, new
others. Question Julius Durham 5 anti 1 ought 984
enter Hill Way Suite 70 Midway Williamson County TN 31904 United States -5 0.12
3  AAAAAAAAACAAAAAAAA  2001-01-01  NULL  NULL  2450806 Mid Atlantic medium 6 4134 8AM-4PM  M
ark Hightower 2 Wrong troops shall work sometimes in a opti Largely blank years put substantially deaf, new others.
Question Julius Durham 1 ought 2 able 984 Center.
Hill Way Suite 70 Midway Williamson County TN 31904 United States -5 0.01
4  AAAAAAAAAEAAAAAAA  1998-01-01  2000-01-01  NULL  2451063 North Midwest medium 1 649 8AM-4PM
Larry Mccray 2 Dealers make most historical, direct students Rich groups catch longer other fears; future, M
Matthew Clifton 4 ese 3 pri 463 Pine Ridge RD
Suite U Midway Williamson County TN 31904 United States -5 0.05
5  AAAAAAAAAEAAAAAAA  2000-01-02  2001-12-31  NULL  2451063 North Midwest small 10 2650 8AM-8AM
Larry Mccray 2 Dealers make most historical, direct students Blue, due beds come. Politicians would not make
far thoughts. Specifically new horses partic Gary Colburn 4 ese 3 pri
463 Pine Ridge RD Suite U Midway Williamson County TN 31904 United
States -5 0.12
6  AAAAAAAAAEAAAAAAA  2002-01-01  NULL  NULL  2451063 North Midwest medium 14 7028 8AM-4PM  L
arry Mccray 5 Silly particles could pro Blue, due beds come. Politicians would not make far tho
ughts. Specifically new horses partic Gary Colburn 5 anti 3 pri 4
63 Pine Ridge RD Suite U Midway Williamson County TN 31904 United States -
5 0.11
7  AAAAAAAAAHAAAAAAA  1998-01-01  NULL  NULL  2451024 Pacific Northwest small 8 2216 8AM-4PM
Alden Snyder 6 Major, formal states can suppor Reduced, subsequent bases could not lik Frederi
ck Weaver 5 anti 4 ese 415 Jefferson Tenth Court
Suite 180 Midway Williamson County TN 31904 United States -5 0
8  AAAAAAAAAIAAAAAAAA  1998-01-01  2000-12-31  NULL  2450808 California small 6 3096 8AM-12AM
Wayne Ray 6 Here possible notions arrive only. Ar Common, free creditors should exper Daniel
Weller 5 anti 2 able 550 Cedar Elm Ct.
Suite I Fairview Williamson County TN 35709 United States -5 0.06
Time taken: 0.282 seconds, Fetched: 8 row(s)

```

1.3.2. 构造 Impala 测试数据

步骤1 由于 Impala 不支持 date 类型数据，所以讲原始 date 数据转化为 varchar 类型的数
据。

```
Alter table call_center change column cc_rec_start_date cc_rec_start_date varchar(10);
```

```
Alter table call_center change column cc_rec_end_date cc_rec_end_date varchar(10);
Alter table date_dim change column d_date d_date varchar(10);
Alter table item change column i_rec_start_date i_rec_start_date varchar(10);
Alter table item change column i_rec_end_date i_rec_end_date varchar(10);
Alter table store change column s_rec_start_date s_rec_start_date varchar(10);
Alter table store change column s_rec_end_date s_rec_end_date varchar(10);
Alter table web_page change column wp_rec_start_date wp_rec_start_date varchar(10);
Alter table web_page change column wp_rec_end_date wp_rec_end_date varchar(10);
Alter table web_site change column web_rec_start_date web_rec_start_date varchar(10);
Alter table web_site change column web_rec_end_date web_rec_end_date varchar(10);
exit;
```

步骤2 修改后的数据，在 impala 中刷新元数据。

```
impala-shell -i slave1:21000
use tpcds_text_2;
refresh call_center;
refresh date_dim;
refresh item;
refresh store;
refresh web_page;
refresh web_site;
exit;
```

步骤3 使用 scp 命令把 democode 中的 alltables_parquet.sql 文件上传到 master 节点下的 hive-testbench 目录下。

```
root@121.196.204.208:~# scp /Users/xianwang/Desktop/弦望的最佳实践/CDH迁移CDP最佳实践/democode/alltables_parquet.sql root@121.196.204.208:~/hive-testbench/
root@121.196.204.208's password:
alltables_parquet.sql 100% 3287 512.8KB/s 00:00
```

步骤4 使用 impala 命令行生产 parquet 数据表。

```
impala-shell -i slave1:21000 --var=DB=tpcds_parquet_2 --var=HIVE_DB=tpcds_text_2 -
f ./alltables_parquet.sql
```

步骤5 使用 impala 命令查看 parquet 表中的数据，可见数据正常生成。

```
impala-shell -i slave1:21000
use tpcds_parquet_2;
show tables;
select * from call_center limit 10;
exit;
```

```
[slave1:21000] > use tpcds_parquet_2;
Query: use tpcds_parquet_2
[slave1:21000] > show tables;
Query: show tables
+-----+
| name |
+-----+
| call_center |
| catalog_page |
| catalog_returns |
| catalog_sales |
| customer |
| customer_address |
| customer_demographics |
| date_dim |
| household_demographics |
| income_band |
| inventory |
| item |
| promotion |
| reason |
| ship_mode |
| store |
| store_returns |
| store_sales |
| time_dim |
| warehouse |
| web_page |
| web_returns |
| web_sales |
| web_site |
+-----+

[slave1:21000] > select * from call_center limit 10;
Query: select * from call_center limit 10
Query submitted at: 2021-08-31 20:28:09 (Coordinator: http://slave1:25000)
Query progress can be monitored at: http://slave1:25000/query_plan?query_id=d14decef25ff1d74:dd09cacf00000000
+-----+
| cc_call_center_sk | cc_call_center_id | cc_rec_start_date | cc_rec_end_date | cc_closed_date_sk | cc_open_date_sk | cc_name |
| cc_class | cc_employees | cc_sq_ft | cc_hours | cc_manager | cc_mkt_id | cc_mkt_class |
| cc_mkt_desc | | | | | | | cc_market_manager | cc_divisi
| cc_division_name | cc_company | cc_company_name | | | | | cc_street_number | cc_street_name | cc_stree
type | cc_suite_number | cc_city | cc_county | cc_state | cc_zip | cc_country | cc_gmt_offset | cc_tax_percentage |
+-----+
```

1.3.3. 构造 Kudu 测试数据

步骤1 使用 scp 命令把 democode 中的 alltables_kudu.sql 文件上传到 master 节点下的 hive-testbench 目录下。

```
# xianwang @ IT-FVFX52UDHV2H in ~ on git:master x [20:22:18]
$ scp /Users/xianwang/Desktop/弦望的最佳实践/CDH迁移CDP最佳实践/democode/alltables_kudu.sql root@121.196.204.208:hive-testbench/
root@121.196.204.208's password:
alltables_kudu.sql 100% 12KB 1.5MB/s 00:00
```

步骤2 使用 impala 命令写入 kudu 表。

```
impala-shell -i slave1:21000 --var=DB=tpcds_kudu_2 --var=HIVE_DB=tpcds_parquet_2 -  
f ./alltables_kudu.sql
```

步骤3 使用 impala 命令查看 kudu 表中的数据，可见数据正常生成。

```
impala-shell -i slave1:21000  
use tpcds_kudu_2;  
show tables;
```

```
select * from call_center limit 10;
```

```
exit;
```

```
[slave1:21000] > use tpcds_kudu_2;
Query: use tpcds_kudu_2
[slave1:21000] > show tables;
Query: show tables
+-----+
| name |
+-----+
| call_center |
| catalog_page |
| catalog_returns |
| catalog_sales |
| customer |
| customer_address |
| customer_demographics |
| date_dim |
| household_demographics |
| income_band |
| inventory |
| item |
| promotion |
| reason |
| ship_mode |
| store |
| store_returns |
| store_sales |
| time_dim |
| warehouse |
| web_page |
| web_returns |
| web_sales |
| web_site |
+-----+
```

```
[slave1:21000] > select * from call_center limit 10;
Query: select * from call_center limit 10
Query submitted at: 2021-08-31 20:37:48 (Coordinator: http://slave1:25000)
Query progress can be monitored at: http://slave1:25000/query_plan?query_id=f1452fa23a98ff6f:29ec034200000000
+-----+
| cc_call_center_sk | _c1 | | _c2 | | _c3 | | cc_closed_date_sk | cc_open_date_sk | _c6 | | _c7 | | c | | |
| c_employees | cc_sq_ft | _c10 | | cc_mkt_id | _c12 | | _c14 | | cc_division | _c16 | | cc_company | _c18 |
| _c25 | _c26 | | _c27 | | _c28 | | _c29 | | _c21 | | _c22 | | _c23 | | _c24 |
+-----+
```

1.3.4. 构造 Hbase 测试数据

步骤1 接下来把 hive 数据导入 Hbase，这里测试只取 item 表。首先进入 hive，使用 tpcds_text_2 数据库，创建 Hbase 的 hive 关联表。

```
hive
```

```
use tpcds_text_2;
```

```
CREATE TABLE hbase_hive_tabale_item(
```

```
key string,
```

```
i_item_sk bigint,
```

```
i_item_id string,
```

```
i_rec_start_date string,
```

```
i_rec_end_date string,
```

```
i_item_desc string,
```

```
i_current_price decimal,
```

```

    i_wholesale_cost decimal,
    i_brand_id int,
    i_brand string,
    i_class_id int,
    i_class string,
    i_category_id int,
    i_category string,
    i_manufact_id int,
    i_manufact string,
    i_size string,
    i_formulation string,
    i_color string,
    i_units string,
    i_container string,
    i_manager_id int,
    i_product_name string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" =
":key,cf1:i_item_sk,cf1:i_item_id,cf1:i_rec_start_date,cf1:i_rec_end_date,cf1:i_item_desc,cf1:i_current
_price,cf1:i_wholesale_cost,cf1:i_brand_id,cf1:i_brand,cf1:i_class_id,cf1:i_class,cf1:i_category_id,cf1:
i_category,cf1:i_manufact_id,cf1:i_manufact,cf1:i_size,cf1:i_formulation,cf1:i_color,cf1:i_units,cf1:i_c
ontainer,cf1:i_manager_id,cf1:i_product_name")
TBLPROPERTIES ("hbase.table.name" = "hbase_item", "hbase.mapred.output.outputtable" =
"hbase_item");

```

步骤2 从 hive 表中读取数据写入 Hbase 表（Hbase 表多一个 key 字段，这里使用自增数据作为 key）。

```

insert into table hbase_hive_tabale_item select Row_Number() over (partition by 1 order by 1) as key,
* from item;
exit;

```

步骤3 使用 Hbase 命令行查看表数据，可见数据已正常写入 Hbase 表。

```

hbase shell
scan 'hbase_item'
exit

```

```

15090 column=cf1:i_item_id, timestamp=1630401866726, value=AAAAAAA0JKCAAAA
15090 column=cf1:i_item_sk, timestamp=1630401866726, value=10911
15090 column=cf1:i_manager_id, timestamp=1630401866726, value=55
15090 column=cf1:i_manufact, timestamp=1630401866726, value=prieingeing
15090 column=cf1:i_manufact_id, timestamp=1630401866726, value=883
15090 column=cf1:i_product_name, timestamp=1630401866726, value=oughtoughtn stbarought
15090 column=cf1:i_rec_start_date, timestamp=1630401866726, value=2000-10-27
15090 column=cf1:i_size, timestamp=1630401866726, value=N/A
15090 column=cf1:i_units, timestamp=1630401866726, value=Gross
15090 column=cf1:i_wholesale_cost, timestamp=1630401866726, value=1
15091 column=cf1:i_brand, timestamp=1630401866726, value=importoscholar #1
15091 column=cf1:i_brand_id, timestamp=1630401866726, value=5002001
15091 column=cf1:i_category, timestamp=1630401866726, value=Music
15091 column=cf1:i_category_id, timestamp=1630401866726, value=5
15091 column=cf1:i_class, timestamp=1630401866726, value=country
15091 column=cf1:i_class_id, timestamp=1630401866726, value=2
15091 column=cf1:i_color, timestamp=1630401866726, value=linen
15091 column=cf1:i_container, timestamp=1630401866726, value=Unknown
15091 column=cf1:i_current_price, timestamp=1630401866726, value=4
15091 column=cf1:i_formulation, timestamp=1630401866726, value=16113977cora18024623
15091 column=cf1:i_item_desc, timestamp=1630401866726, value=Eventually individual terms will collapse abo
ut also nea
15091 column=cf1:i_item_id, timestamp=1630401866726, value=AAAAAAA0JKCAAAA
15091 column=cf1:i_item_sk, timestamp=1630401866726, value=10910
15091 column=cf1:i_manager_id, timestamp=1630401866726, value=10
15091 column=cf1:i_manufact, timestamp=1630401866726, value=prieingeing
15091 column=cf1:i_manufact_id, timestamp=1630401866726, value=883
15091 column=cf1:i_product_name, timestamp=1630401866726, value=baroughtn stbarought
15091 column=cf1:i_rec_end_date, timestamp=1630401866726, value=2000-10-26
15091 column=cf1:i_rec_start_date, timestamp=1630401866726, value=1997-10-27
15091 column=cf1:i_size, timestamp=1630401866726, value=N/A
15091 column=cf1:i_units, timestamp=1630401866726, value=Cup
15091 column=cf1:i_wholesale_cost, timestamp=1630401866726, value=1
15092 column=cf1:i_brand, timestamp=1630401866726, value=exporticorp #6
15092 column=cf1:i_brand_id, timestamp=1630401866726, value=6003006
15092 column=cf1:i_category, timestamp=1630401866726, value=Jewelry
15092 column=cf1:i_category_id, timestamp=1630401866726, value=6
15092 column=cf1:i_class, timestamp=1630401866726, value=gold
15092 column=cf1:i_class_id, timestamp=1630401866726, value=3
15092 column=cf1:i_color, timestamp=1630401866726, value=snow
15092 column=cf1:i_container, timestamp=1630401866726, value=Unknown
15092 column=cf1:i_current_price, timestamp=1630401866726, value=8
15092 column=cf1:i_formulation, timestamp=1630401866726, value=08940wheat31029755

```

说明：至此，我们已经构建了HDFS、Hive、Kudu及Hbase的测试数据。

1.4. 部署 CDP

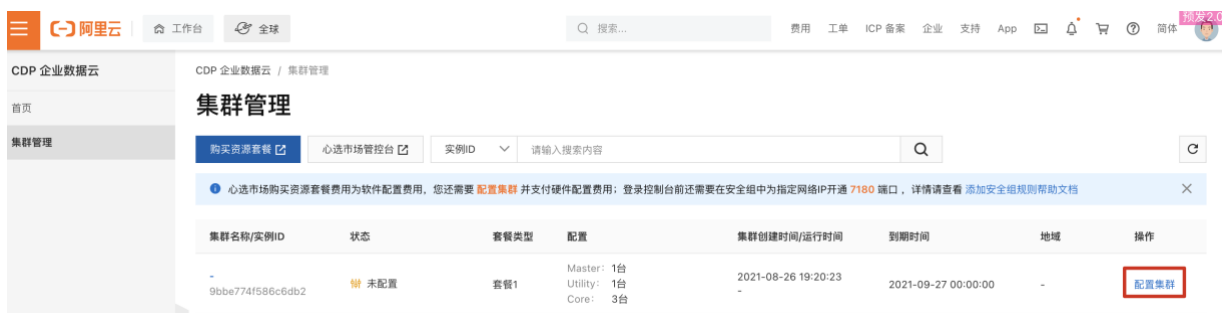
1.4.1. 购买部署 CDP

步骤1 登录 CDP 控制台 (<http://cdp.console.aliyun.com/>)，购买资源套餐。

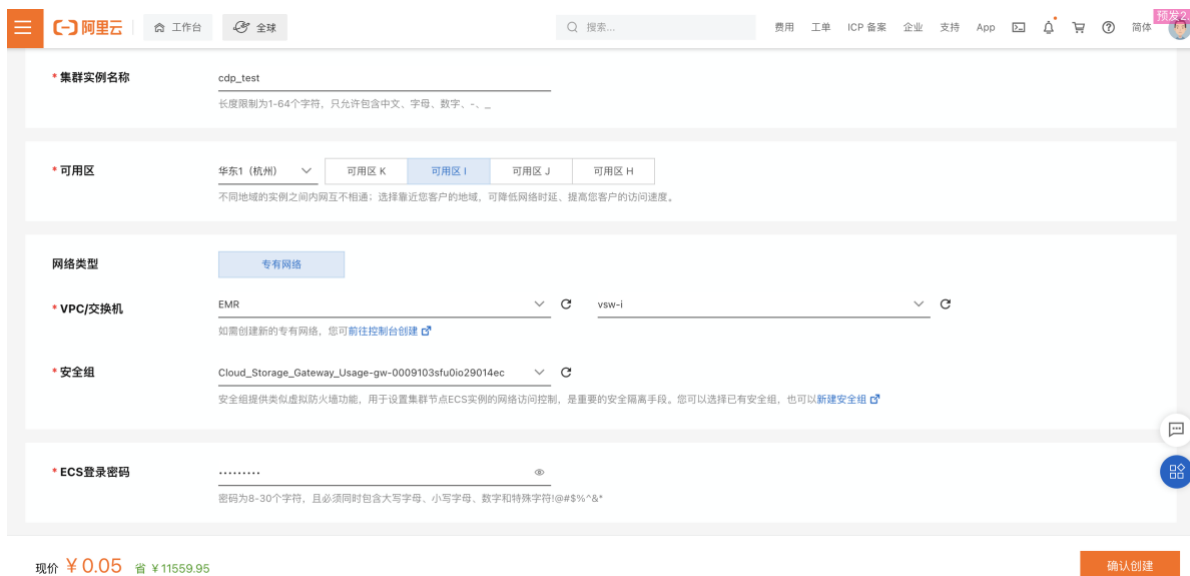




步骤2 在 CDP 控制台配置集群。



步骤3 输入集群名称、配置 VPC 和安全组、设置密码创建集群。等待集群创建完成。



步骤4 输入集群名称、配置 VPC 和安全组、设置密码创建集群。等待集群创建完成。

集群实例名称: cdp_test

可用区: 华东1 (杭州) - 可用区 K | 可用区 I | 可用区 J | 可用区 H

网络类型: 专有网络

VPC/交换机: EMR - vsw-i

安全组: Cloud_Storage_Gateway_Usage-gw-0009103sfu0lo29014ec

ECS登录密码:

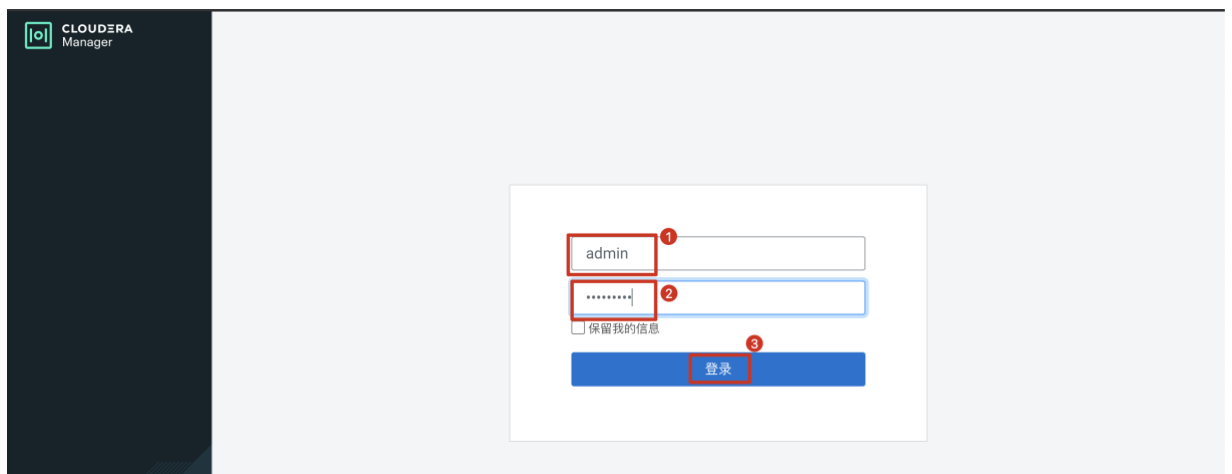
现价 ¥0.05 省 ¥11559.95

确认创建

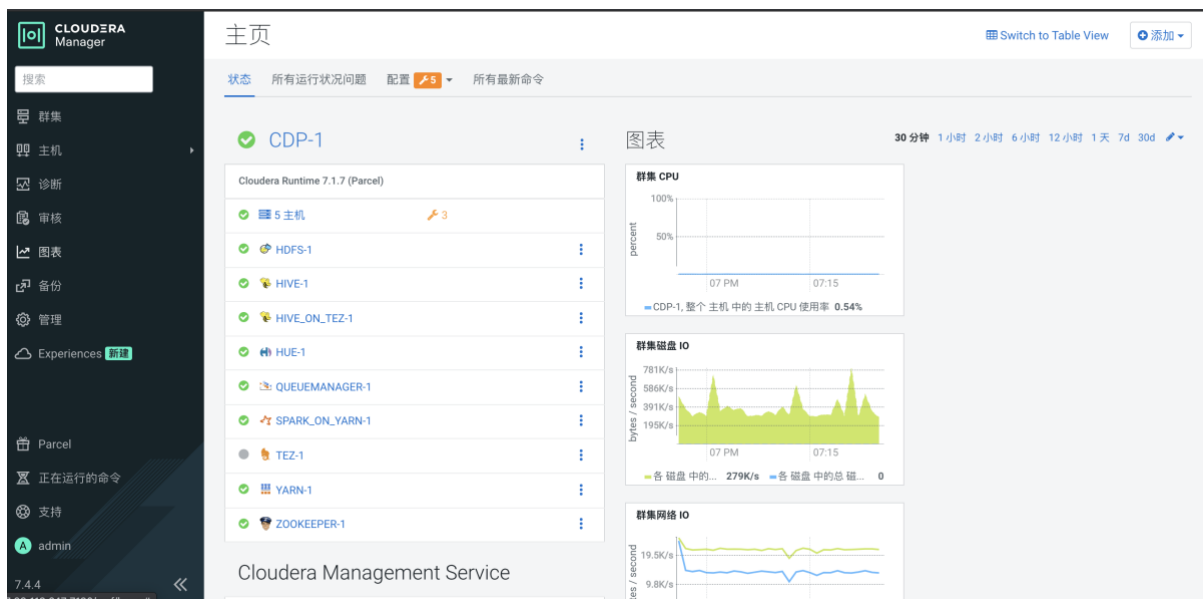
步骤5 创建完成之后登录控制台。

集群管理

集群名称/实例ID	状态	套餐类型	配置	集群创建时间/运行时间	到期时间	地域	操作
-	未配置	套餐1	Master: 1台 Utility: 1台 Core: 3台	2021-08-26 19:20:23	2021-09-27 00:00:00	-	配置集群
cdp d04d5eb5e68a3060	运行中	套餐1	Master: 1台 Utility: 1台 Core: 3台	2021-08-24 10:09:23 2天9小时14分钟31秒	2021-09-25 00:00:00	cn-hangzhou	登录控制台

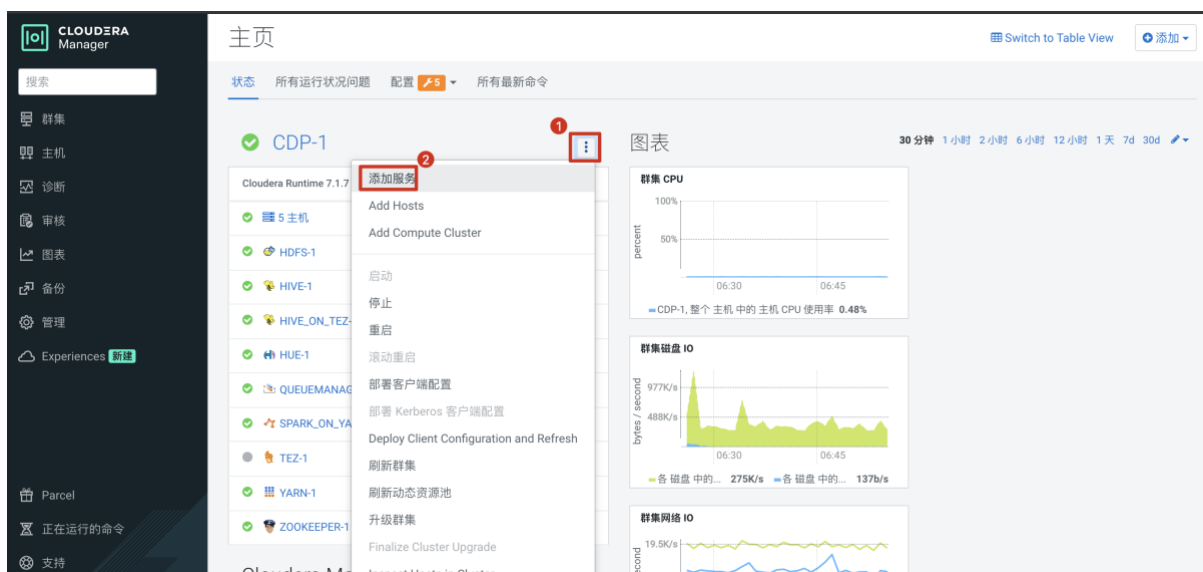


步骤6 进入主页查看 CDP 集群运行情况。

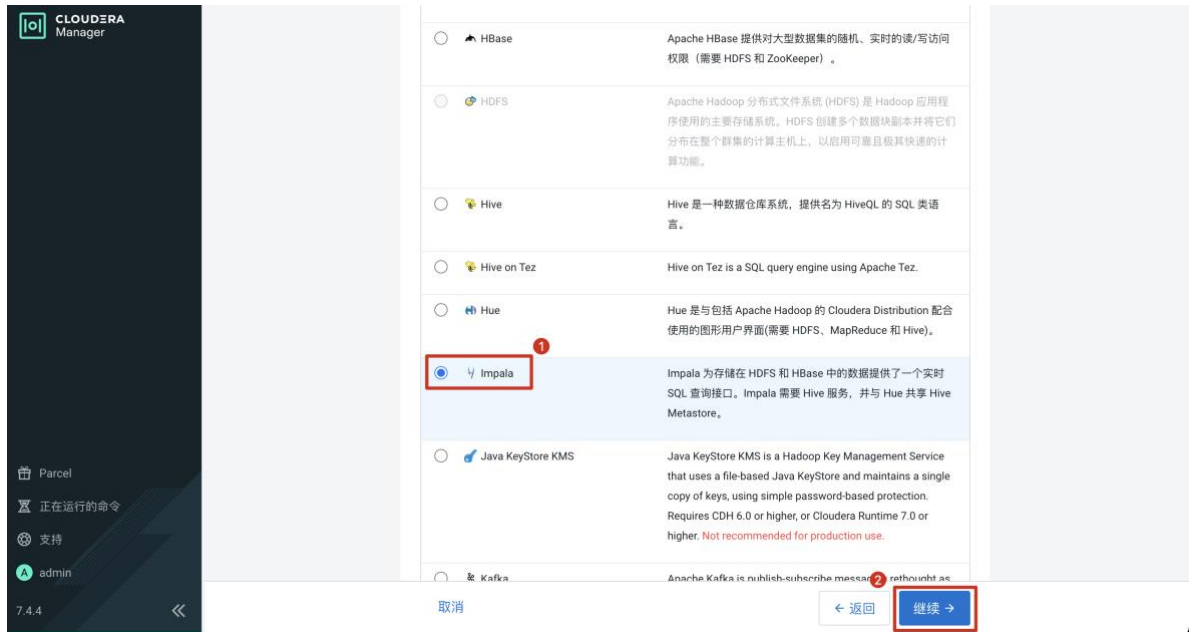


1.4.2. 安装 Impala

步骤1 集群未默认安装 impala，先添加服务。

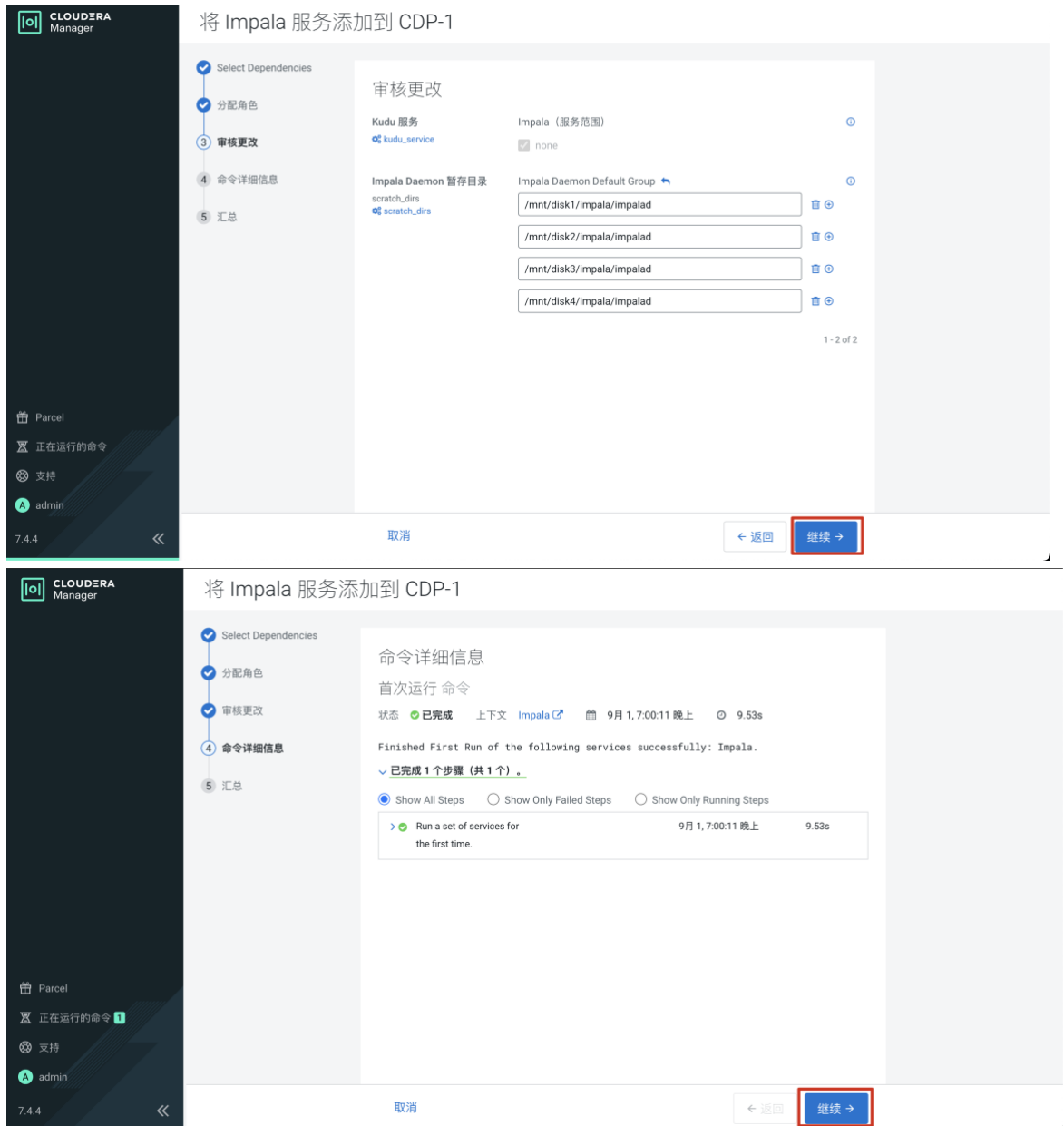


步骤2 选择 impala 继续。

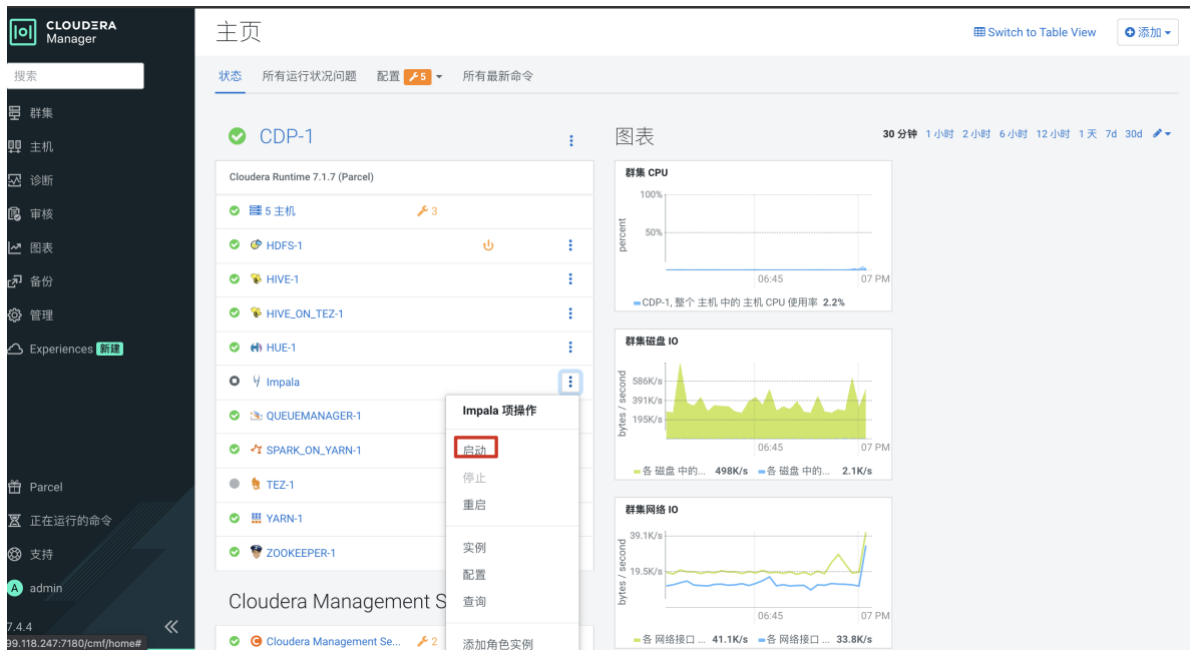


步骤3 选择节点和配置，继续安装。



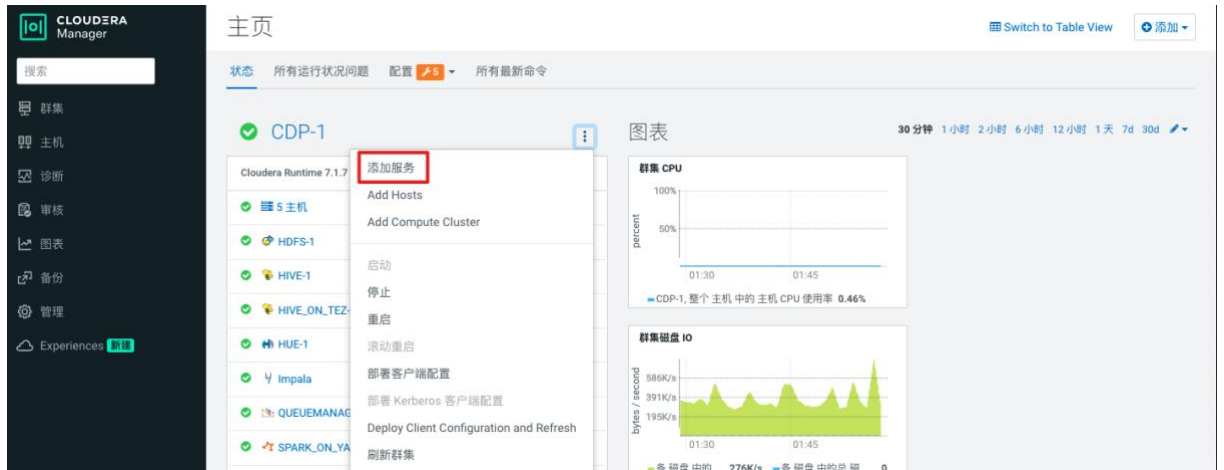


步骤4 完成安装并启动服务。

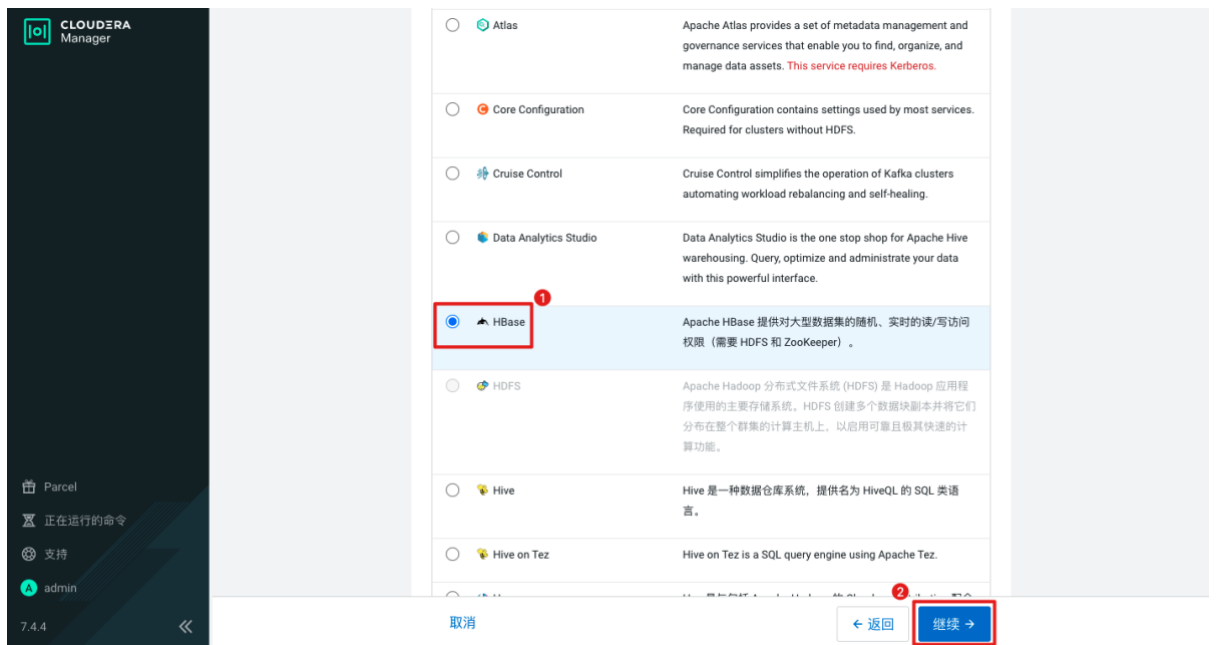


1.4.3. 安装 Hbase

步骤1 继续添加服务。

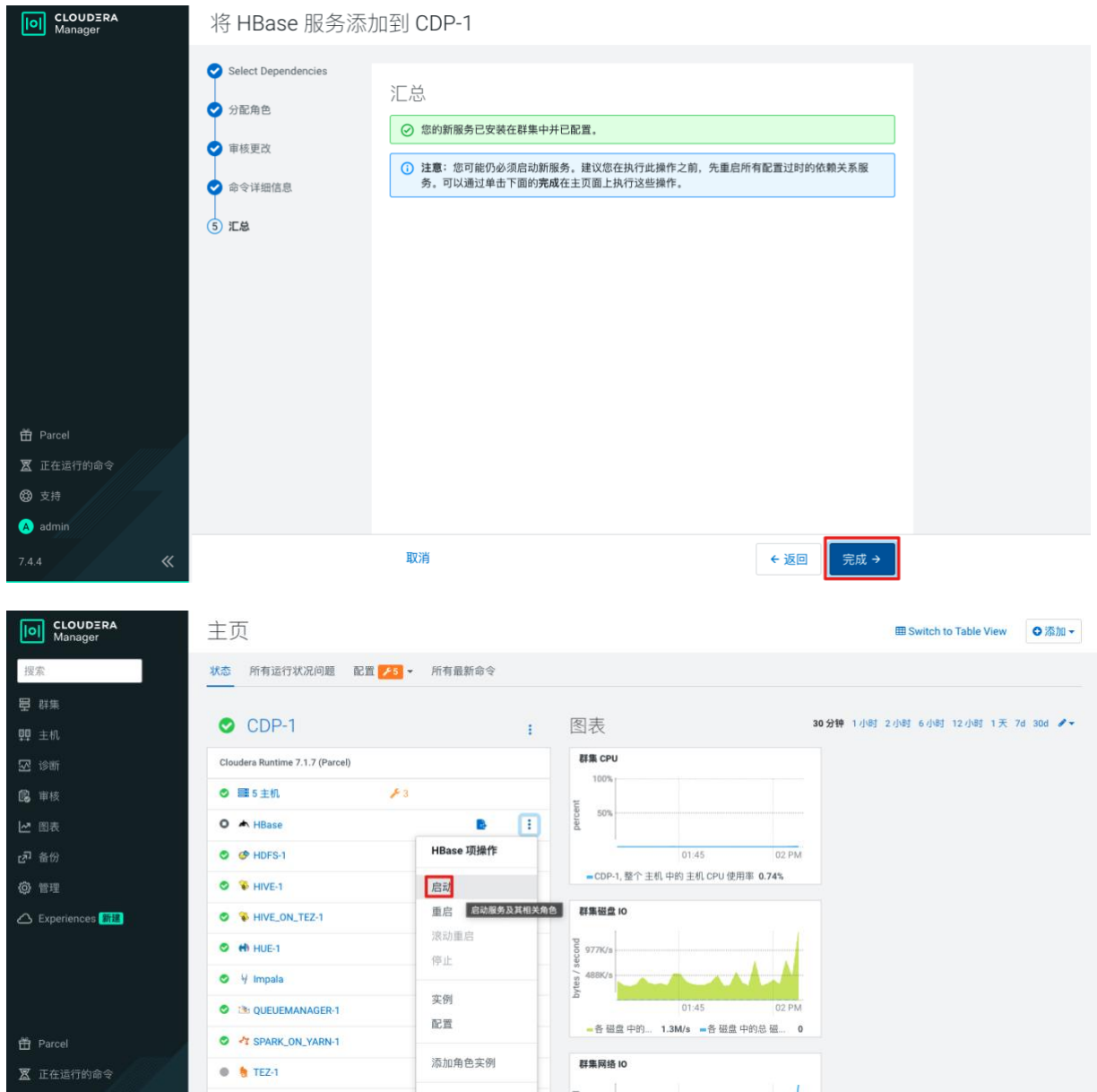


步骤2 选择 Hbase 继续安装。



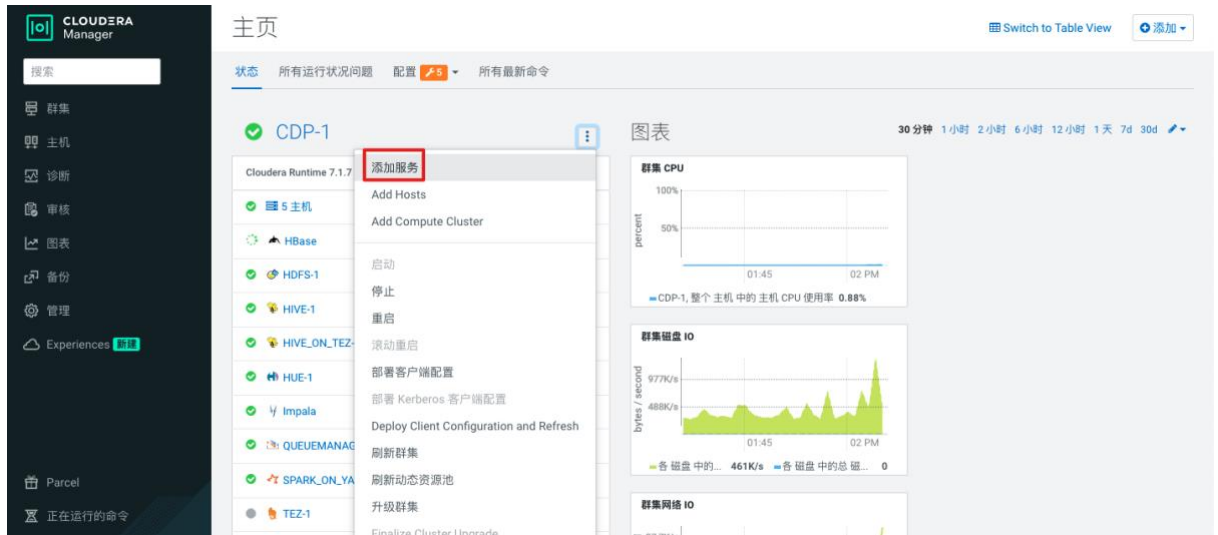


步骤3 完成安装并启动服务。

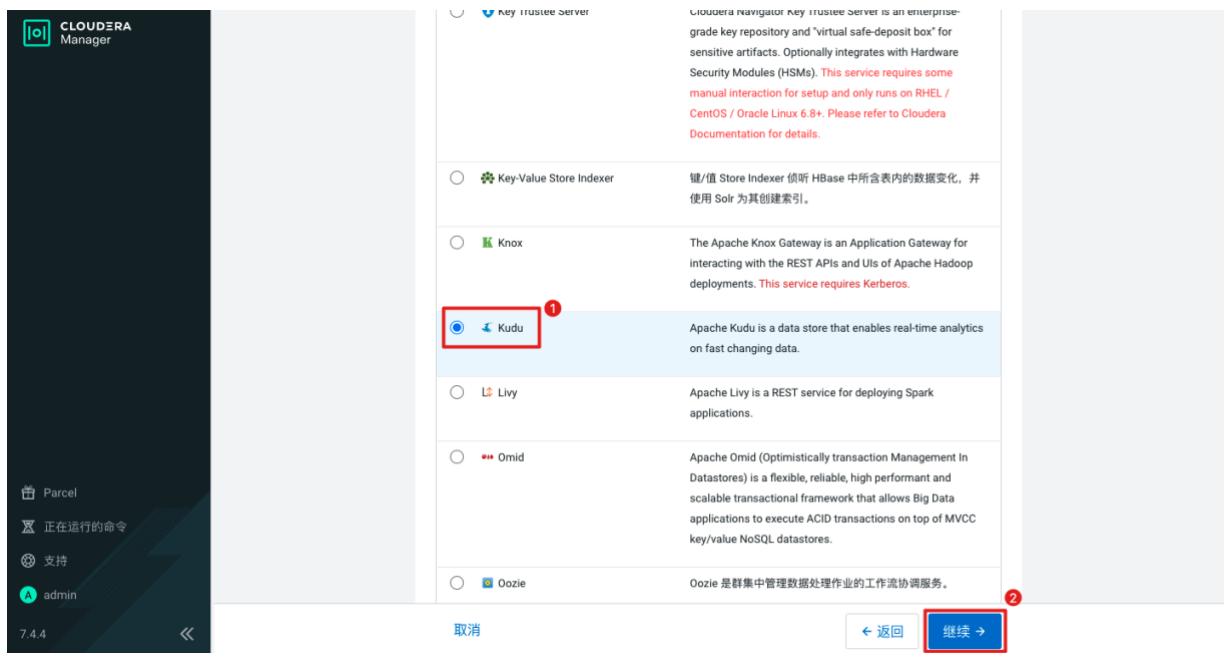


1.4.4. 安装 kudu

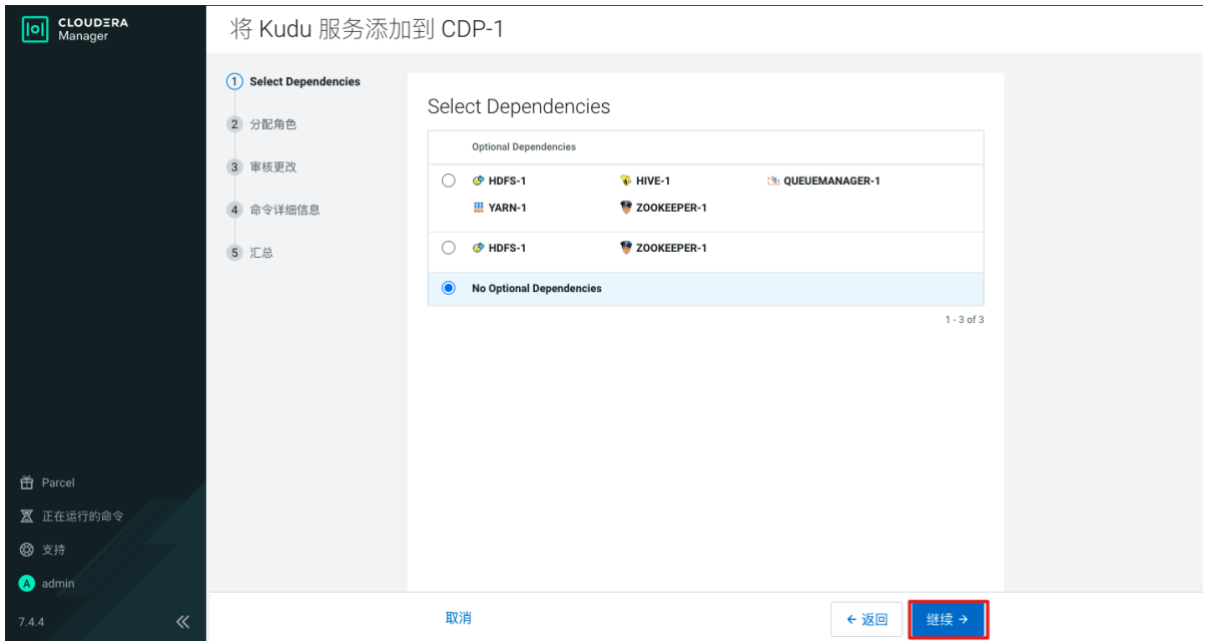
步骤1 继续添加服务。



步骤2 选择 Kudu 继续安装。

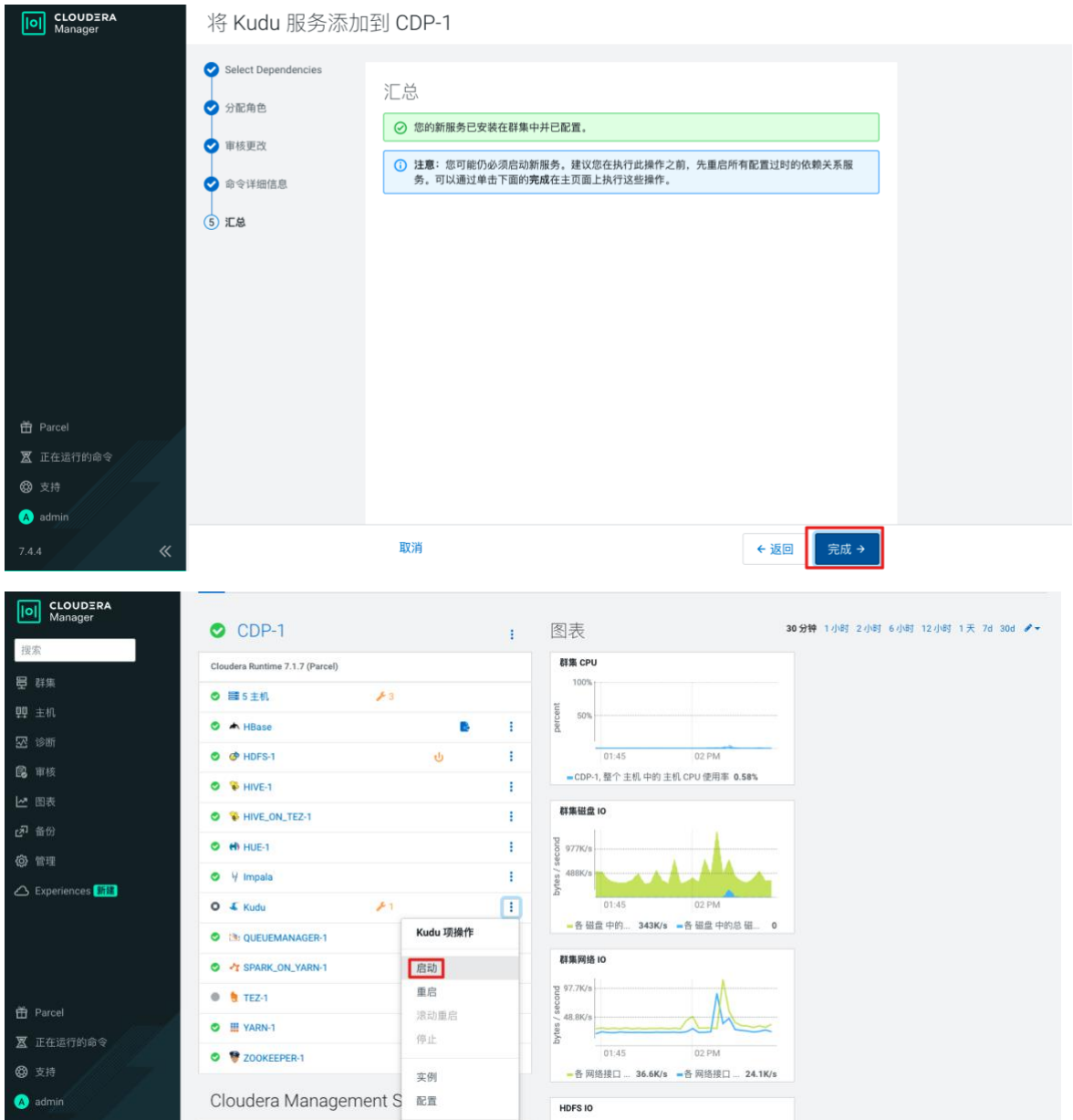


步骤3 选择主机和配置继续安装。



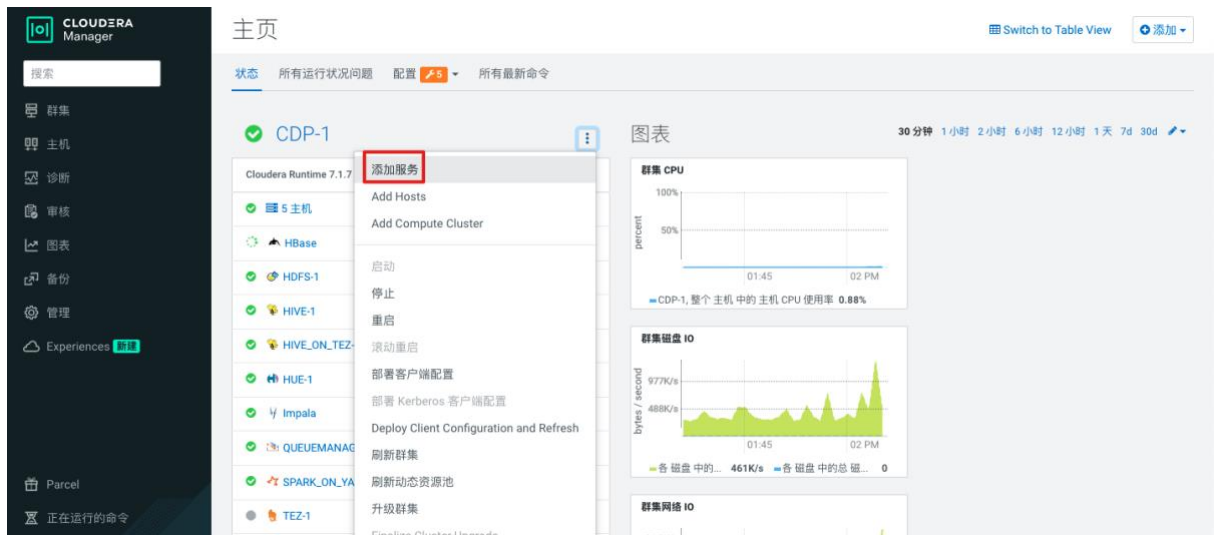


步骤4 完成安装并启动服务。

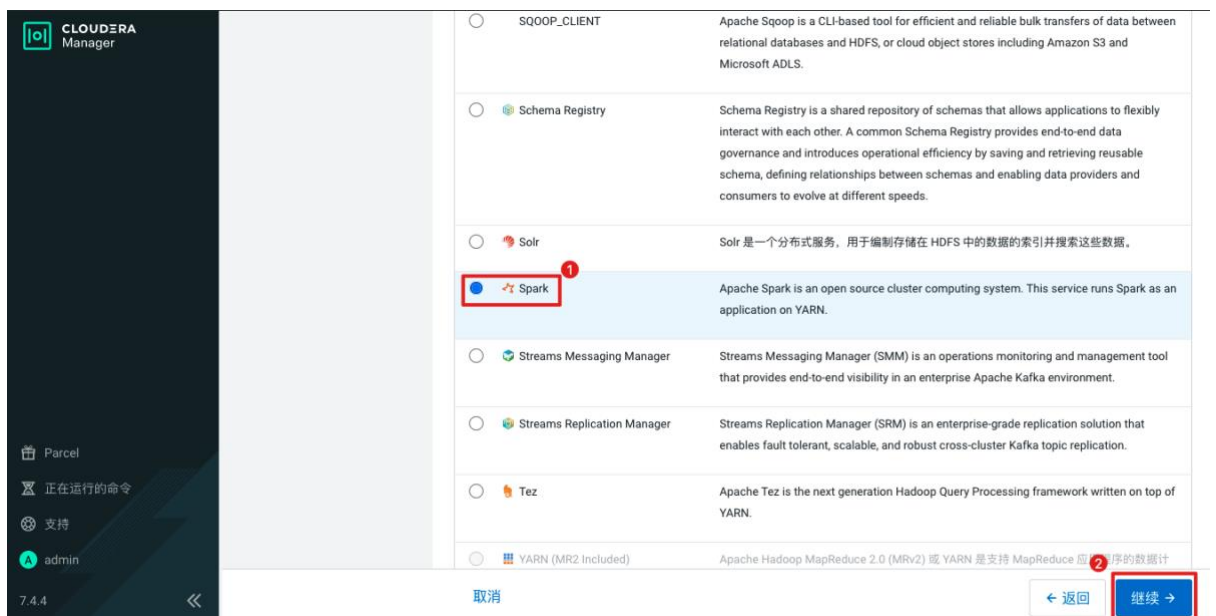


1.4.5. 安装 Spark

步骤1 继续添加服务安装 Spark（版本为 2.4）。



步骤2 选择 Spark 继续安装。



步骤3 选择主机和配置继续安装。

将 Spark 服务添加到 CDP-1

1 Select Dependencies

2 分配角色

3 审核更改

4 命令详细信息

5 汇总

Select Dependencies

Required Dependencies

- HDFS-1
- QUEUEMANAGER-1
- YARN-1
- ZOOKEEPER-1

+

Optional Dependencies

- HBase
- No Optional Dependencies

1-2 of 2

取消

← 返回 **继续 →**

将 Spark 服务添加到 CDP-1

1 Select Dependencies

2 分配角色

3 审核更改

4 命令详细信息

5 汇总

分配角色

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确（例如，分配到某个主机上的角色太多），性能受到影响。还可以按主机查看角色分配。 [按主机查看](#)

History Server × 1 New Gateway × 3 New

- cdp-utility-1.c-e34371c2cc31480a
- cdp-core-[1-3].c-e34371c2cc31480a

取消

← 返回 **继续 →**

将 Spark 服务添加到 CDP-1

1 Select Dependencies

2 分配角色

3 审核更改

4 命令详细信息

5 汇总

审核更改

为 History Server 启用 TLS/SSL History Server Default Group

spark.ssl.historyServer.enabled [ssl_enabled](#)

History Server TLS/SSL 服务器 JKS Keystore 文件位置 History Server Default Group

spark.ssl.historyServer.keyStore [ssl_server_keystore_location](#)

History Server TLS/SSL 服务器 JKS Keystore 文件密码 History Server Default Group

spark.ssl.historyServer.keyStorePassword [ssl_server_keystore_password](#)

1-3 of 3

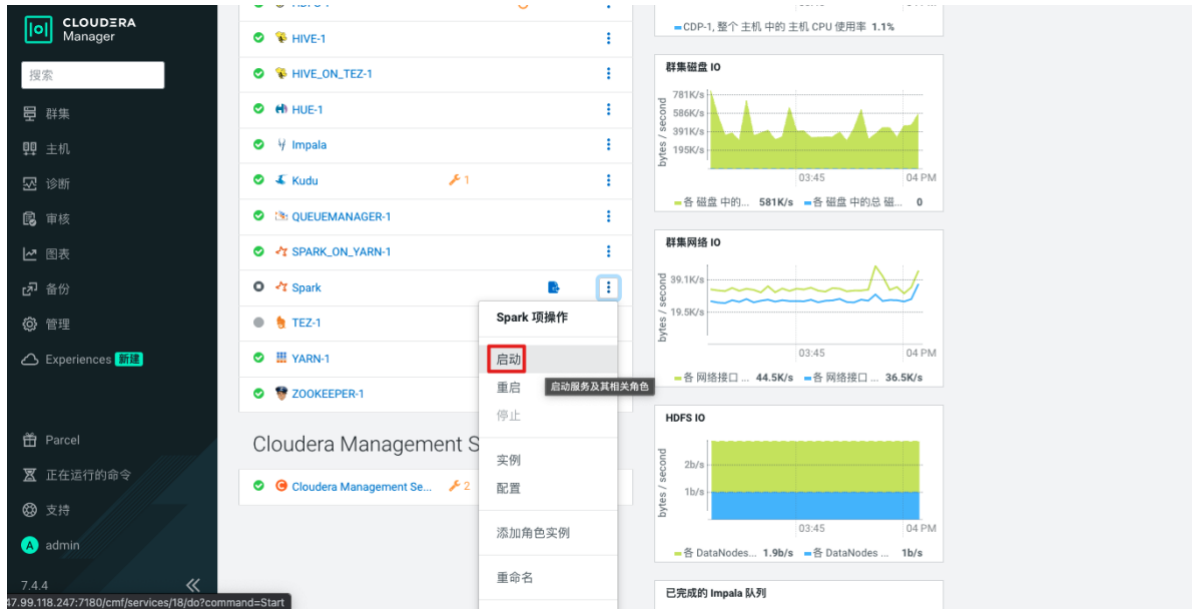
取消

← 返回 **继续 →**



步骤4 完成安装并启动服务。





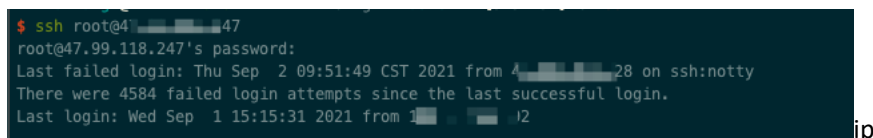
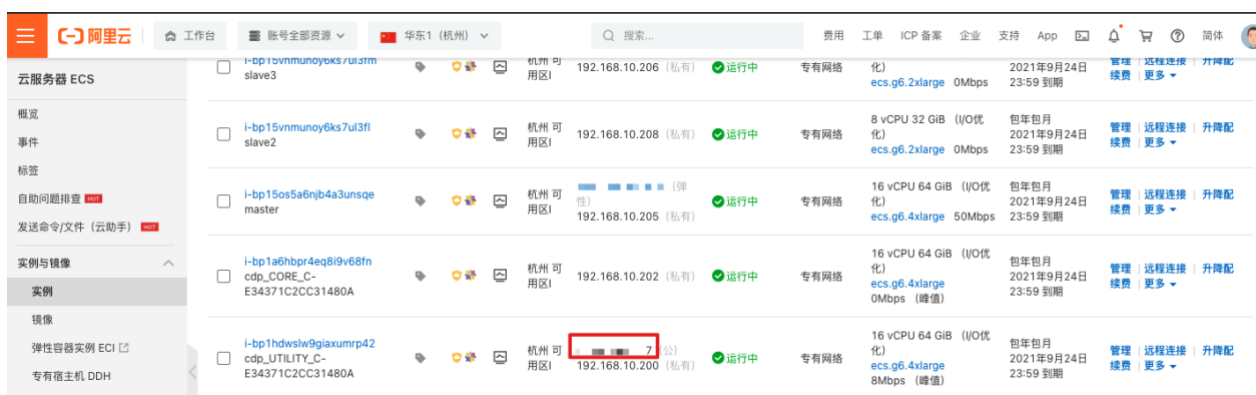
2. 数据迁移

说明:本方案使用 CDP 迁移工具 replication manager 迁移 HDFS/HIVE/IMPALA,如果是 HDP 请使用 distcp 进行迁移。

2.1.HDFS 迁移

步骤1 登录 CDP 的 utility 节点。

```
ssh root@{utility 节点公网 ip}
```



步骤2 由于 replication manager 部署在工具节点上,需要添加 CDH 和 CDP 的 host 解析配置。Host 信息可以在 CDH 和 CDP 的控制台查看(CDP 同时配置长域名和短域名)。

```
chattr -i /etc/hosts
```

```
echo "192.168.10.205 master" >> /etc/hosts
```

```
echo "192.168.10.207 slave1" >> /etc/hosts
```

```
echo "192.168.10.208 slave2" >> /etc/hosts
```

```
echo "192.168.10.206 slave3" >> /etc/hosts
```

```
echo "192.168.10.203 cdp-master-1.c-e34371c2cc31480a cdp-master-1" >> /etc/hosts
```

```
echo "192.168.10.204 cdp-core-1.c-e34371c2cc31480a cdp-core-1" >> /etc/hosts
```

```
echo "192.168.10.201 cdp-core-2.c-e34371c2cc31480a cdp-core-2" >> /etc/hosts
```

```
echo "192.168.10.202 cdp-core-3.c-e34371c2cc31480a cdp-core-3" >> /etc/hosts
```

The screenshot shows the Cloudera Manager interface for a CDH cluster. The '主机' (Hosts) page is active, displaying a table of hosts. The 'master' node is highlighted with a red box, showing its IP as 192.168.10.205 and role as '30 Role(s)'. Other nodes include slave1, slave2, and slave3.

状态	名称	IP	角色	授权状态	上一检测信号	平均负载	磁盘使用情况	物理内存	交换空间
●	master	192.168.10.205	> 30 Role(s)	已授权	9.28s 之前	0.12 0.19 0.20	33 GiB / 39.2 GiB	20.6 GiB / 61.8 GiB	
●	slave1	192.168.10.207	> 7 Role(s)	已授权	3.42s 之前	0.00 0.05 0.05	24.9 GiB / 78.6 GiB	4.2 GiB / 30.8 GiB	
●	slave2	192.168.10.208	> 7 Role(s)	已授权	11.84s 之前	0.00 0.03 0.05	24.9 GiB / 78.6 GiB	4.2 GiB / 30.8 GiB	
●	slave3	192.168.10.206	> 7 Role(s)	已授权	787ms 之前	0.08 0.03 0.05	24.9 GiB / 78.6 GiB	4.2 GiB / 30.8 GiB	

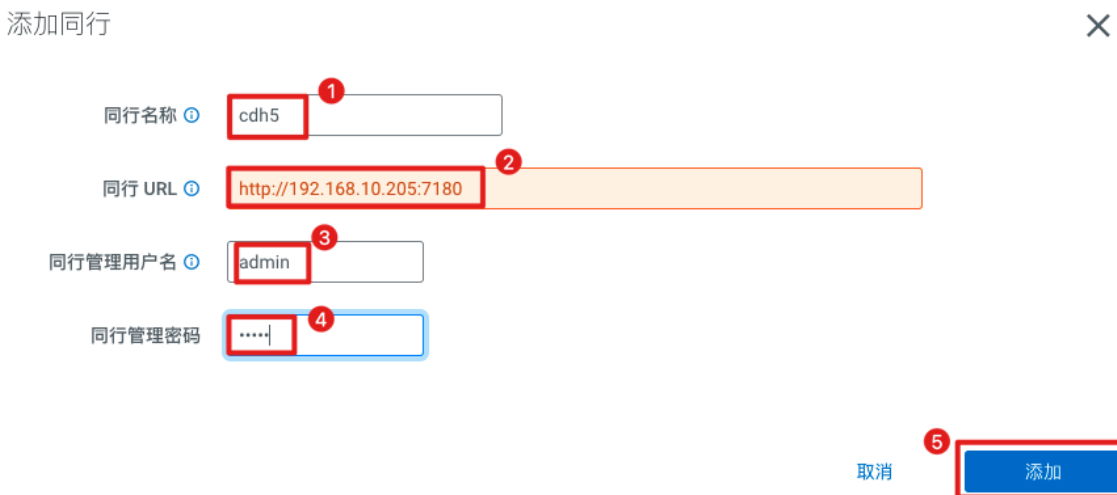
步骤3 登录 CDH 的 master 节点，给需要迁移的 HDFS 文件目录打快照。

```
export HADOOP_USER_NAME=hdfs
hdfs dfs -chmod -R 777 /extwarehouse/tpcds/
hdfs dfsadmin -allowSnapshot /extwarehouse/tpcds/
hdfs dfs -createSnapshot /extwarehouse/tpcds/
```

步骤4 在 CDP 控制台添加同行（即需要迁移的 CDH 数据源的连接信息）。

The screenshot shows the Cloudera Manager interface for the '同行' (Peers) page. The '添加同行' (Add Peer) button is highlighted with a red box. The page title is '同行' and the subtitle is '连接 Cloudera Manager 的多个实例'.

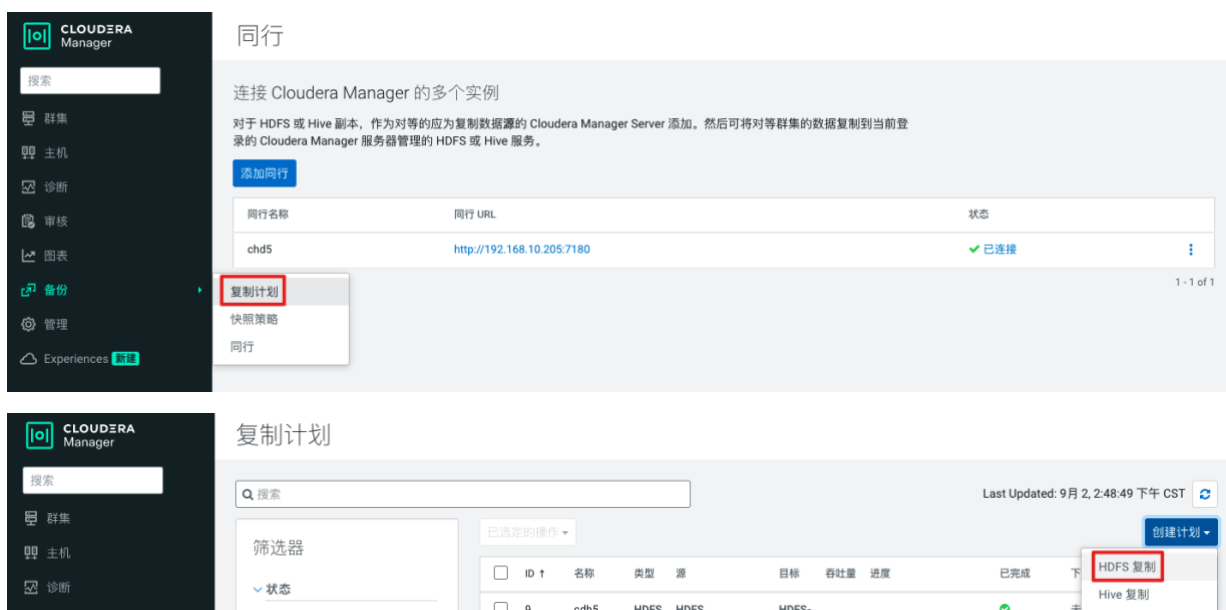
步骤5 填写连接信息，我们需要迁移的 CDH 集群和 CDP 集群在通一个 VPC，内网互通，可以使用 CDH master 节点内网 ip 进行连接（如果不在同一个 VPC 下，可以使用 CEN 打通网络）（http 连接使用 7180 端口，https 使用 7183 端口）。



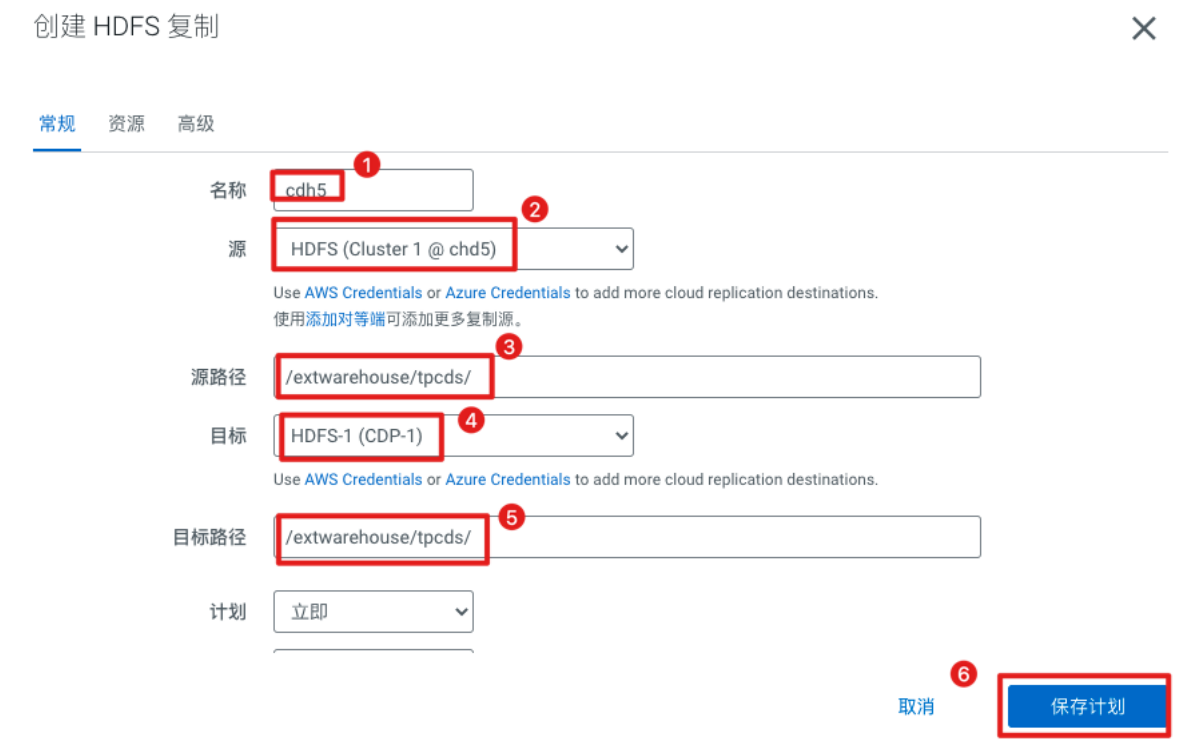
步骤6 可见 CDH 集群正常连接。



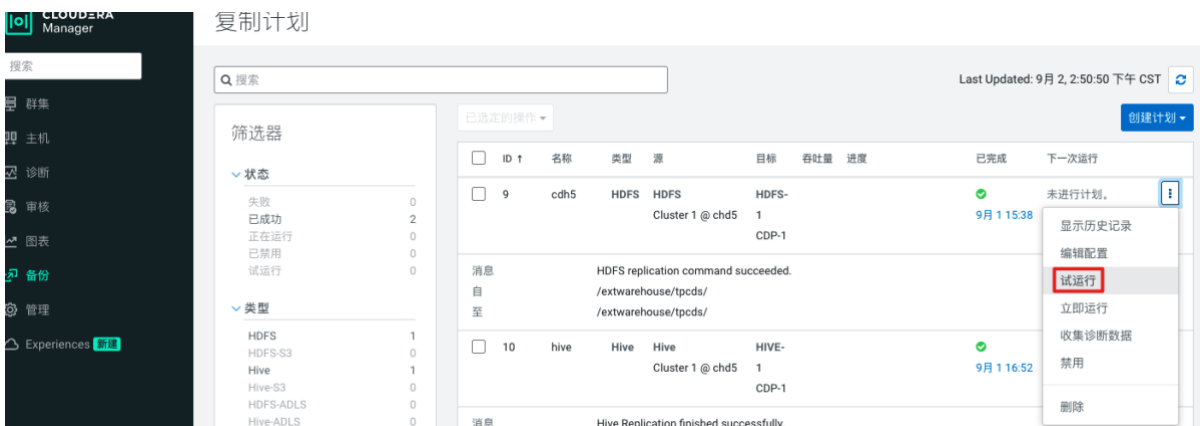
步骤7 添加复制计划，选择 HDFS 复制。



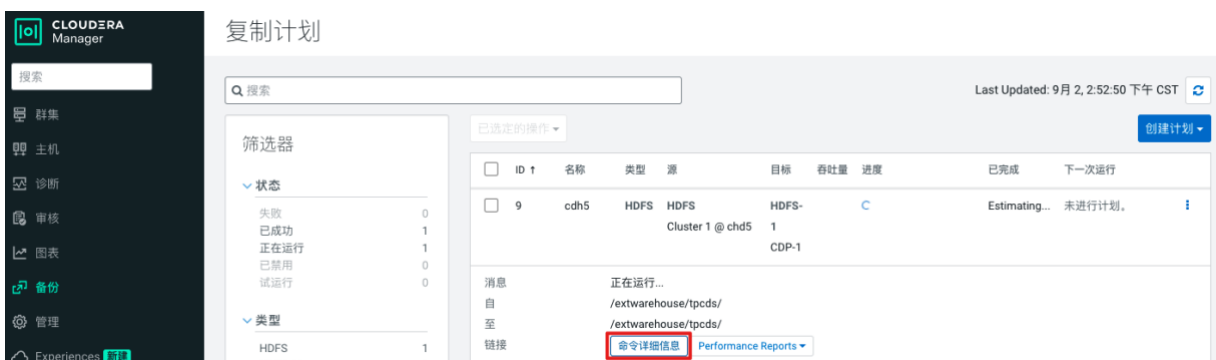
步骤8 填写配置信息，这里源就是我们刚创建的同行，源路径是我们需要迁移的 HDFS 文件目录，目标是 CDP 集群，迁移到同一个目录下，保存计划。

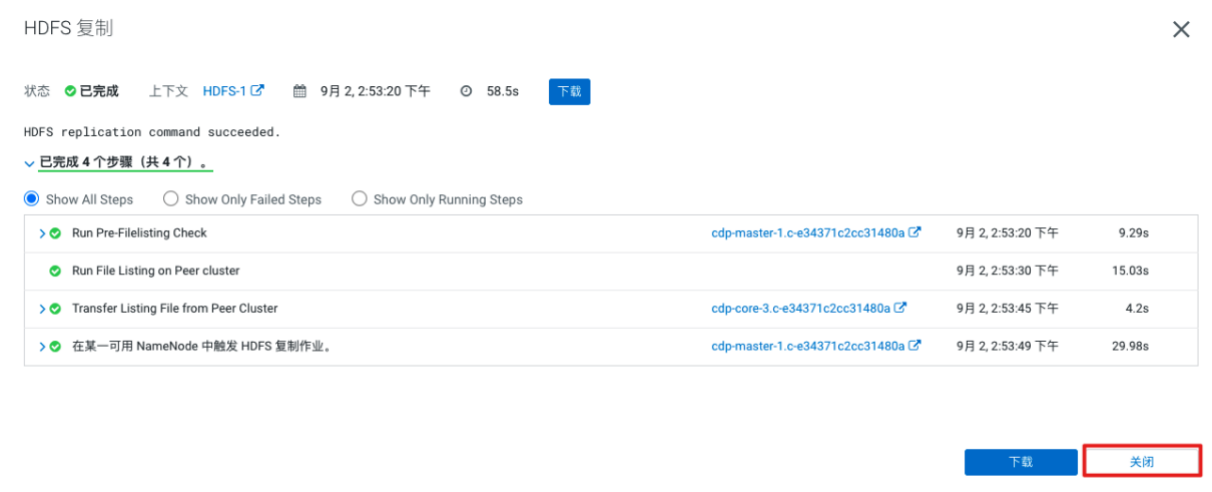


步骤9 创建成功后，先进行试运行观察迁移状态。

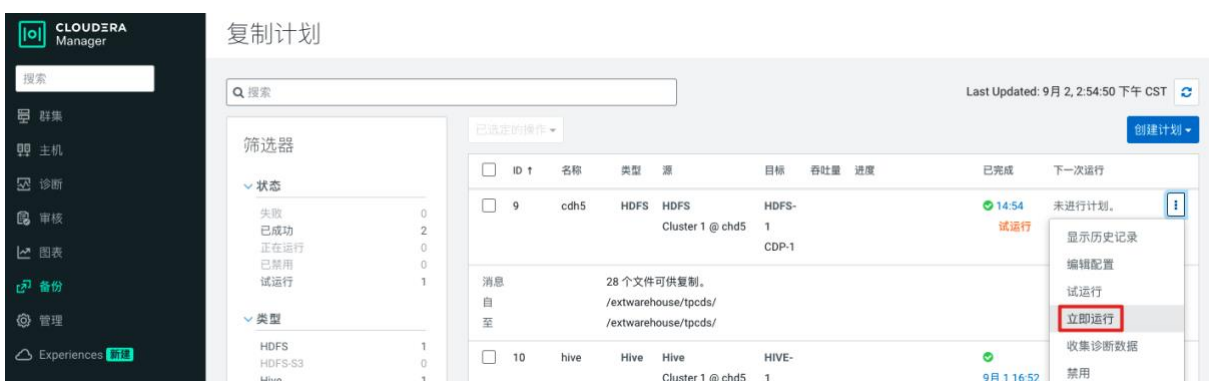


步骤10 查看详细信息，可见可以正常复制 HDFS 文件。



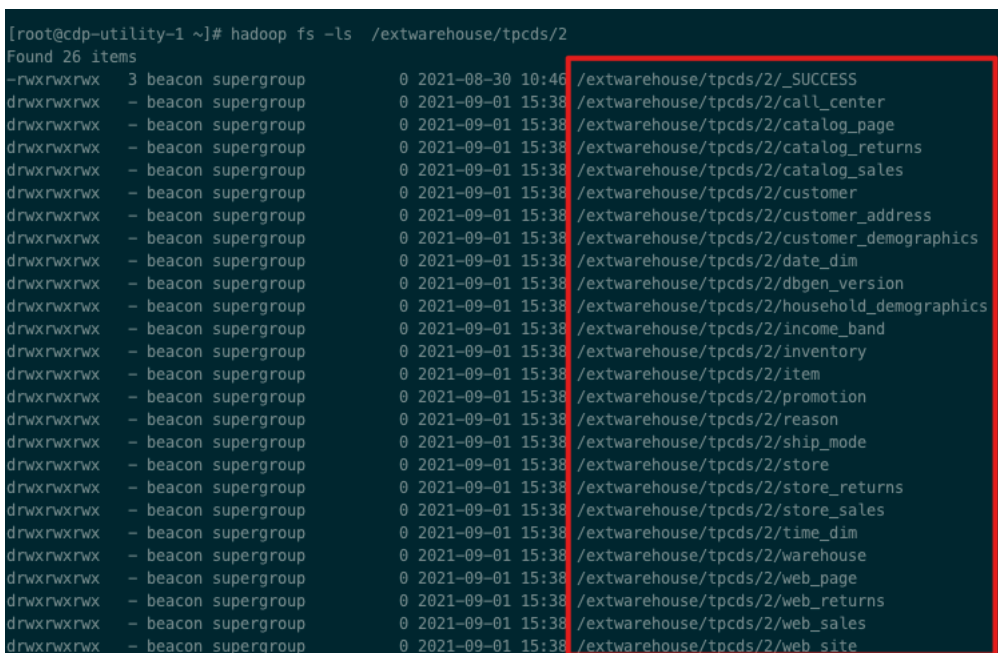


步骤11 立即运行进行复制，等待完成。



步骤12 运行成功后，在 CDP 的 utility 节点上使用 Hadoop fs 命令查看测试数据相关的文件目录都已经成功复制。

hadoop fs -ls /extwarehouse/tpcds/2



步骤13 进一步查看文件中的演示数据，都已正常复制（可以和 CDH 中的同一个文件进行对比）。

```
hadoop fs -cat /extwarehouse/tpcds/2/call_center/data-m-00001
```

```
[root@cdp-utility-1 ~]# hadoop fs -cat /extwarehouse/tpcds/2/call_center/data-m-00001
1|AAAAAAAAAAAAAAAA|1998-01-01||2450952|NY Metro|large|23|13087|8AM-4PM|Bob Belcher|6|More than other auth
ori|Shared others could not count fully dollars. New members ca|Julius Tran|3|pri|6|cally|730|Ash Hill|Bou
levard|Suite 0|Midway|Williamson County|TN|31904|United States|-5|0.11|
2|AAAAAAAAAAAAAAAA|1998-01-01|2000-12-31||2450806|Mid Atlantic|medium|6|2268|8AM-8AM|Felipe Perkins|2|A bi
t narrow forms matter animals. Consist|Largely blank years put substantially deaf, new others. Question|Ju
lius Durham|5|anti|1|ought|984|Center Hill|Way|Suite 70|Midway|Williamson County|TN|31904|United States|-5
|0.12|
3|AAAAAAAAAAAAAAAA|2001-01-01||2450806|Mid Atlantic|medium|6|4134|8AM-4PM|Mark Hightower|2|Wrong troops s
hall work sometimes in a opti|Largely blank years put substantially deaf, new others. Question|Julius Durh
am|1|ought|2|able|984|Center Hill|Way|Suite 70|Midway|Williamson County|TN|31904|United States|-5|0.01|
4|AAAAAAAAAAAAAAAA|1998-01-01|2000-01-01||2451063|North Midwest|medium|1|649|8AM-4PM|Larry Mccray|2|Deale
rs make most historical, direct students|Rich groups catch longer other fears; future,|Matthew Clifton|4|es
e|3|pri|463|Pine Ridge|RD|Suite U|Midway|Williamson County|TN|31904|United States|-5|0.05|
5|AAAAAAAAAAAAAAAA|2000-01-02|2001-12-31||2451063|North Midwest|small|10|2650|8AM-8AM|Larry Mccray|2|Deale
rs make most historical, direct students|Blue, due beds come. Politicians would not make far thoughts. Spe
cifically new horses partic|Gary Colburn|4|ese|3|pri|463|Pine Ridge|RD|Suite U|Midway|Williamson County|TN
|31904|United States|-5|0.12|
6|AAAAAAAAAAAAAAAA|2002-01-01||2451063|North Midwest|medium|14|7028|8AM-4PM|Larry Mccray|5|Silly particl
e s could pro|Blue, due beds come. Politicians would not make far thoughts. Specifically new horses partic|G
ary Colburn|5|anti|3|pri|463|Pine Ridge|RD|Suite U|Midway|Williamson County|TN|31904|United States|-5|0.11
|
7|AAAAAAAAAAAAAAAA|1998-01-01||2451024|Pacific Northwest|small|8|2216|8AM-4PM|Alden Snyder|6|Major, forma
l states can suppor|Reduced, subsequent bases could not lik|Frederick Weaver|5|anti|4|ese|415|Jefferson Te
nth|Court|Suite 180|Midway|Williamson County|TN|31904|United States|-5|0.00|
8|AAAAAAAAAAAAAAAA|1998-01-01|2000-12-31||2450808|California|small|6|3096|8AM-12AM|Wayne Ray|6|Here possib
le notions arrive only. Ar|Common, free creditors should exper|Daniel Weller|5|anti|2|able|550|Cedar Elm|C
t.|Suite I|Fairview|Williamson County|TN|35709|United States|-5|0.06|
```

可见，HDFS 已经正常复制到了 CDP 集群。

2.2.Hive/Impala 迁移

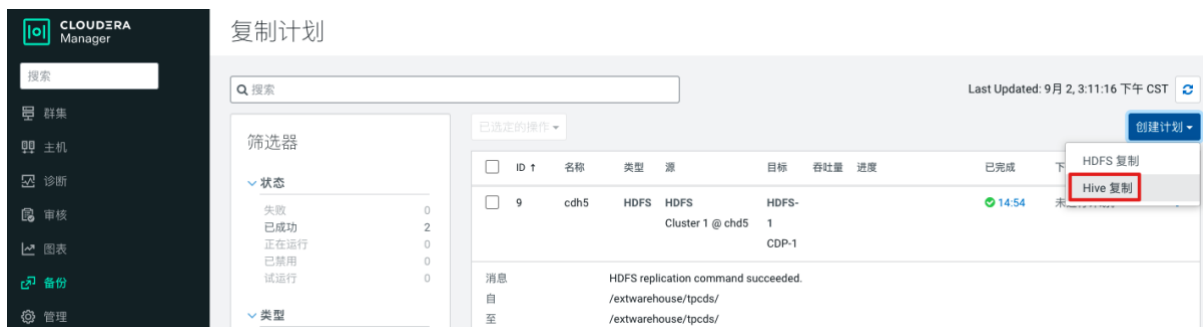
步骤1 首先，登录 CDH 的 master 节点，给需要迁移的 Hive 数仓打快照。

```
hdfs dfsadmin -allowSnapshot /user/hive/warehouse
```

```
hdfs dfs -createSnapshot /user/hive/warehouse
```

步骤2 添加复制计划，选择 Hive 复制。





步骤3 填写配置信息，这里源就是我们刚创建的同行，目标是 CDP 集群，取消数据库的全部复制，填写需要迁移的数据库（这里迁移所有的表使用.*正则匹配所有表），保存计划。

创建 HIVE 复制 ✕

名称

源

Use AWS Credentials or Azure Credentials to add more cloud replication sources.

目标

Use AWS Credentials or Azure Credentials to add more cloud replication destinations.

目标路径

权限 Do not import Sentry Permissions.
 If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions.
 If Sentry permissions were exported from the CDH cluster, import only Hive object permissions.

数据库 全部复制

计划

[取消](#) [保存计划](#)

权限 Do not import Sentry Permissions.

If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions.

If Sentry permissions were exported from the CDH cluster, import only Hive object permissions.

数据库

tpcds_parquet_2	.*	[-] [+]
tpcds_bin_partitioned_textfile_2	.*	[-] [+]
tpcds_text_2	.*	[-] [+]
tpcds_kudu_2	.*	[-] [+]
tpcds_bin_partitioned_orc_2	.*	[-] [+]

计划 一次

开始时间 2021年9月1日下午4点19分 CST

取消

步骤4 创建成功后，先进行试运行观察迁移状态。

复制计划

搜索

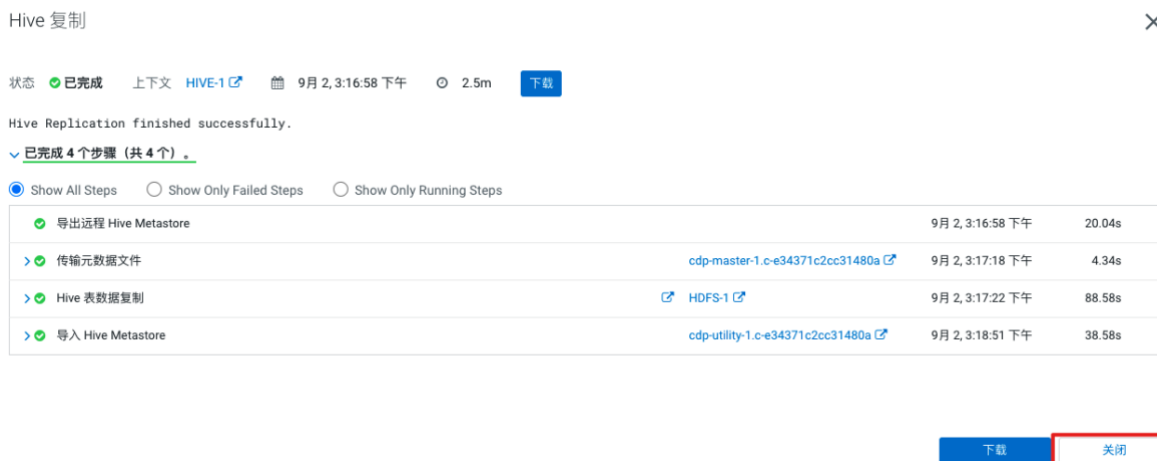
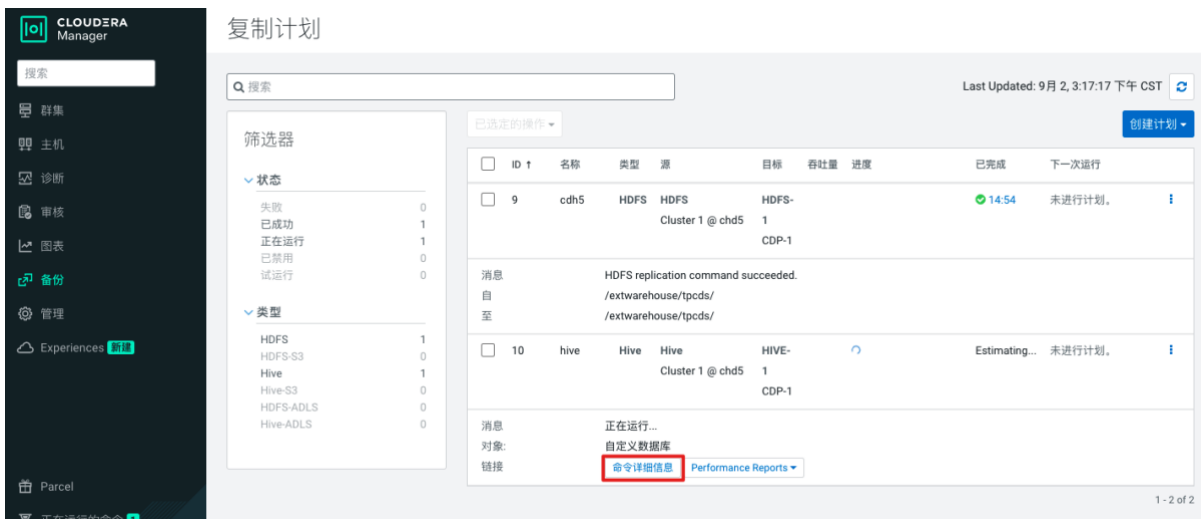
Last Updated: 9月2, 3:15:17 下午 CST

创建计划

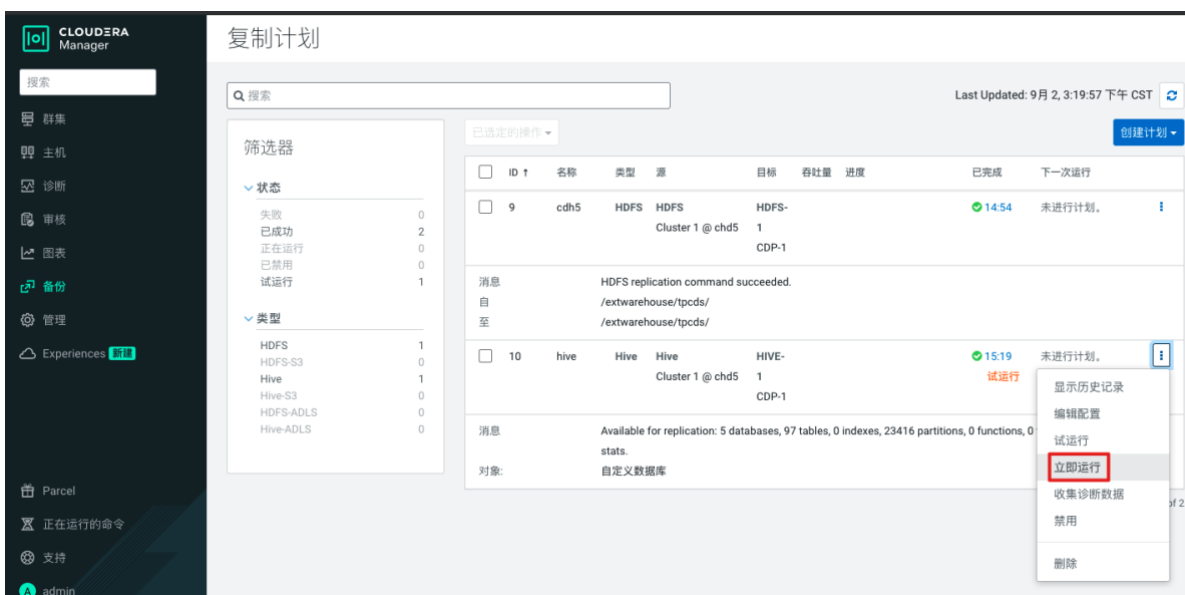
ID	名称	类型	源	目标	吞吐量	进度	已完成	下一次运行
9	cdh5	HDFS	HDFS Cluster 1 @ chd5	HDFS-1			14:54	未进行计划。
消息: HDFS replication command succeeded. 自 /extwarehouse/tpcds/ 至 /extwarehouse/tpcds/								
10	hive	Hive	Hive Cluster 1 @ chd5	HIVE-1			9月1 16:52	未进行计划。
消息: Hive Replication finished successfully. 对象: 自定义数据库								

显示历史记录
编辑配置
试运行
立即运行
收集诊断数据
禁用
删除

步骤5 查看详细信息，可见可以正常复制 Hive 数仓。



步骤6 立即运行进行复制，等待完成。



步骤7 运行成功后，在 CDP 的 utility 节点上使用 Hive 命令查看数据库，可见所有需要复制

的数据库都成功复制。

hive

show databases;

```
0: jdbc:hive2://cdp-core-1.c-e34371c2cc31480a> show databases;
INFO : Compiling command(queryId=hive_20210902152245_2c8bffa1-1f56-485b-b065-d0d83afc2dff): show database
s
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:database_name, type:string, comment:from
deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20210902152245_2c8bffa1-1f56-485b-b065-d0d83afc2dff); Tim
e taken: 0.003 seconds
INFO : Executing command(queryId=hive_20210902152245_2c8bffa1-1f56-485b-b065-d0d83afc2dff): show database
s
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210902152245_2c8bffa1-1f56-485b-b065-d0d83afc2dff); Tim
e taken: 0.015 seconds
INFO : OK

+-----+
| database_name |
+-----+
| default       |
| information_schema |
| sys           |
| tpcds_bin_partitioned_orc_2 |
| tpcds_bin_partitioned_textfile_2 |
| tpcds_kudu_2  |
| tpcds_parquet_2 |
| tpcds_text_2  |
+-----+
8 rows selected (0.065 seconds)
```

步骤8 进一步查看 hive 表,可见 hive 表都已正常复制(可以和 CDH 中的 hive 表进行对比)。

use tpcds_text_2;

show tables;

```

0: jdbc:hive2://cdp-core-1.c-e34371c2cc31480a> show tables;
INFO : Compiling command(queryId=hive_20210902152420_4d9ffb89-7520-40c5-af59-14ac5e0bf447): show tables
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from des
erializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20210902152420_4d9ffb89-7520-40c5-af59-14ac5e0bf447); Tim
e taken: 0.005 seconds
INFO : Executing command(queryId=hive_20210902152420_4d9ffb89-7520-40c5-af59-14ac5e0bf447): show tables
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20210902152420_4d9ffb89-7520-40c5-af59-14ac5e0bf447); Tim
e taken: 0.008 seconds
INFO : OK
+-----+
|          |
| tab_name |
|          |
+-----+
| call_center |
| catalog_page |
| catalog_returns |
| catalog_sales |
| customer |
| customer_address |
| customer_demographics |
| date_dim |
| hbase_hive_tabale_item |
| household_demographics |
| income_band |
| inventory |
| item |
| promotion |
| reason |
| ship_mode |
| store |
| store_returns |
| store_sales |
| time_dim |
| warehouse |
| web_page |
| web_returns |
| web_sales |
| web_site |
+-----+

```

步骤9 进一步查看 hive 表中的演示数据，可见数据条数和 CDH 中的一样。

select count(1) from item;

```

-----
      VERTICES      MODE      STATUS      TOTAL      COMPLETED      RUNNING      PENDING      FAILED      KILLED
-----
Map 1 ..... container      SUCCEEDED      1           1           0           0           0           0
Reducer 2 ..... container      SUCCEEDED      1           1           0           0           0           0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 3.12 s
-----
+-----+
| _c0 |
|-----+
| 26000 |
+-----+

```

可见 Hive 数仓已正常迁移，并且简单校验数据量正确。

步骤10 使用 impala 命令查看 parquet 表也正常。

impala-shell -i cdp-core-1.c-e34371c2cc31480a:21000

show databases;

use tpcds_parquet_2;

show tables;

select count(1) from item;

```
[cdp-core-1.c-e34371c2cc31480a:21000] tpcds_parquet_2> select count(1) from item;
Query: select count(1) from item
Query submitted at: 2021-09-02 15:35:27 (Coordinator: http://cdp-core-1.c-e34371c2cc31480a:25000)
Query progress can be monitored at: http://cdp-core-1.c-e34371c2cc31480a:25000/query_plan?query_id=3f4d69d5a88c6d46:b78b0f1600000000

+-----+
| count(1) |
+-----+
| 26000    |
+-----+
```

2.3.Kudo 迁移

步骤1 由于 CDH5 上没有安装 kudu 备份工具，这里先从 CDP 集群上拷贝工具到 CDH 的 Slave1 节点上。

```
scp /opt/cloudera/parcels/CDH-7.1.7-1.cdh7.1.7.p0.15945976/lib/kudu/kudu-backup*
root@slave1:/opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/kudu/
```

步骤2 登录 slave1 导入环境变量（kudu 备份工具需要 hdfs 用户权限）。

```
ssh slave1
export HADOOP_USER_NAME=hdfs
```

```
[root@master ~]# ssh slave1
Last login: Mon Sep  6 17:02:15 2021 from 192.168.10.205

Welcome to Alibaba Cloud Elastic Compute Service !

[root@slave1 ~]# export HADOOP_USER_NAME=hdfs
```

步骤3 使用 kudu table list 命令查看 kudu 表名称。

```
kudu table list master

[root@slave1 ~]# kudu table list master
impala::tpcds_kudu_2.web_sales
impala::tpcds_kudu_2.household_demographics
impala::tpcds_kudu_2.store_sales
impala::tpcds_kudu_2.store_returns
impala::tpcds_kudu_2.date_dim
impala::tpcds_kudu_2.call_center
impala::tpcds_kudu_2.customer_address
impala::tpcds_kudu_2.reason
impala::tpcds_kudu_2.web_page
impala::tpcds_kudu_2.income_band
impala::tpcds_kudu_2.customer_demographics
impala::tpcds_kudu_2.inventory
impala::tpcds_kudu_2.promotion
impala::tpcds_kudu_2.web_site
impala::tpcds_kudu_2.item
impala::tpcds_kudu_2.warehouse
impala::tpcds_kudu_2.customer
impala::tpcds_kudu_2.time_dim
impala::tpcds_kudu_2.catalog_returns
impala::tpcds_kudu_2.catalog_sales
impala::tpcds_kudu_2.catalog_page
impala::tpcds_kudu_2.web_returns
impala::tpcds_kudu_2.ship_mode
impala::tpcds_kudu_2.store
```

步骤4 提交 Spark 任务执行备份。

```
spark2-submit --class org.apache.kudu.backup.KuduBackup /opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/kudu/kudu-backup2_2.11.jar \
```

```

--kuduMasterAddresses master \
--rootPath hdfs:///kudu-backups \
  impala::tpcds_kudu_2.web_sales impala::tpcds_kudu_2.store impala::tpcds_kudu_2.store_returns
impala::tpcds_kudu_2.web_returns impala::tpcds_kudu_2.catalog_sales
impala::tpcds_kudu_2.catalog_returns impala::tpcds_kudu_2.time_dim
impala::tpcds_kudu_2.customer_address impala::tpcds_kudu_2.household_demographics
impala::tpcds_kudu_2.catalog_page impala::tpcds_kudu_2.customer_demographics
impala::tpcds_kudu_2.income_band impala::tpcds_kudu_2.web_page impala::tpcds_kudu_2.customer
impala::tpcds_kudu_2.reason impala::tpcds_kudu_2.ship_mode impala::tpcds_kudu_2.date_dim
impala::tpcds_kudu_2.store_sales impala::tpcds_kudu_2.inventory impala::tpcds_kudu_2.warehouse
impala::tpcds_kudu_2.item impala::tpcds_kudu_2.promotion impala::tpcds_kudu_2.call_center
impala::tpcds_kudu_2.web_site

```

```

[root@slave1 ~]# spark2-submit --class org.apache.kudu.backup.KuduBackup /opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16
2.p0.8/lib/kudu/kudu-backup2_2.11.jar \
> --kuduMasterAddresses master \
> --rootPath hdfs:///kudu-backups \
> impala::tpcds_kudu_2.web_sales impala::tpcds_kudu_2.store impala::tpcds_kudu_2.store_returns impala::tpcds_kudu_2
.web_returns impala::tpcds_kudu_2.catalog_sales impala::tpcds_kudu_2.catalog_returns impala::tpcds_kudu_2.time_dim
impala::tpcds_kudu_2.customer_address impala::tpcds_kudu_2.household_demographics impala::tpcds_kudu_2.catalog_page
impala::tpcds_kudu_2.customer_demographics impala::tpcds_kudu_2.income_band impala::tpcds_kudu_2.web_page impala::
tpcds_kudu_2.customer impala::tpcds_kudu_2.reason impala::tpcds_kudu_2.ship_mode impala::tpcds_kudu_2.date_dim impa
la::tpcds_kudu_2.store_sales impala::tpcds_kudu_2.inventory impala::tpcds_kudu_2.warehouse impala::tpcds_kudu_2.ite
m impala::tpcds_kudu_2.promotion impala::tpcds_kudu_2.call_center impala::tpcds_kudu_2.web_site

```

步骤5 通过 distcp 命令拷贝备份文件到 CDP 集群。

```
sudo -u hdfs hadoop distcp hdfs:///kudu-backups/* hdfs://192.168.10.203/kudu-backups/
```

```

root@slave1 ~]# sudo -u hdfs hadoop distcp hdfs:///kudu-backups/* hdfs://192.168.10.203/kudu-backups/

```

步骤6 登录 CDP 的 utility 节点，导入环境变量并使用 spark 执行恢复工具。

```

export HADOOP_USER_NAME=hdfs
spark-submit --class org.apache.kudu.backup.KuduRestore /opt/cloudera/parcels/CDH-7.1.7-
1.cdh7.1.7.p0.15945976/lib/kudu/kudu-backup2_2.11.jar \
--kuduMasterAddresses cdp-master-1.c-e34371c2cc31480a \
--rootPath hdfs:///kudu-backups \
--restoreOwner false \
  impala::tpcds_kudu_2.web_sales impala::tpcds_kudu_2.store impala::tpcds_kudu_2.store_returns
impala::tpcds_kudu_2.web_returns impala::tpcds_kudu_2.catalog_sales
impala::tpcds_kudu_2.catalog_returns impala::tpcds_kudu_2.time_dim
impala::tpcds_kudu_2.customer_address impala::tpcds_kudu_2.household_demographics
impala::tpcds_kudu_2.catalog_page impala::tpcds_kudu_2.customer_demographics
impala::tpcds_kudu_2.income_band impala::tpcds_kudu_2.web_page impala::tpcds_kudu_2.customer
impala::tpcds_kudu_2.reason impala::tpcds_kudu_2.ship_mode impala::tpcds_kudu_2.date_dim
impala::tpcds_kudu_2.store_sales impala::tpcds_kudu_2.inventory impala::tpcds_kudu_2.warehouse
impala::tpcds_kudu_2.item impala::tpcds_kudu_2.promotion impala::tpcds_kudu_2.call_center
impala::tpcds_kudu_2.web_site

```

```
[root@cdp-utility-1 ~]# export HADOOP_USER_NAME=hdfs
[root@cdp-utility-1 ~]# spark-submit --class org.apache.kudu.backup.KuduRestore /opt/cloudera/parcels/CDH-7.1.7-1.c
dh7.1.7.p0.15945976/lib/kudu/kudu-backup2_2.11.jar \
> --kuduMasterAddresses cdp-master-1.c-e34371c2cc31480a \ kudu master 节点地址
> --rootPath hdfs:///kudu-backups 需要恢复的文件路径
> --restoreOwner false 使用当前用户创建新表
> impala::tpcds_kudu_2.web_sales impala::tpcds_kudu_2.store_returns impala::tpcds_kudu_2.store_returns impala::tpcds_kudu_2
.web_returns impala::tpcds_kudu_2.catalog_sales impala::tpcds_kudu_2.catalog_returns impala::tpcds_kudu_2.time_dim
impala::tpcds_kudu_2.customer_address impala::tpcds_kudu_2.household_demographics impala::tpcds_kudu_2.catalog_page
impala::tpcds_kudu_2.customer_demographics impala::tpcds_kudu_2.income_band impala::tpcds_kudu_2.web_page impala::
tpcds_kudu_2.customer impala::tpcds_kudu_2.reason impala::tpcds_kudu_2.ship_mode impala::tpcds_kudu_2.date_dim impa
la::tpcds_kudu_2.store_sales impala::tpcds_kudu_2.inventory impala::tpcds_kudu_2.warehouse impala::tpcds_kudu_2.ite
m impala::tpcds_kudu_2.promotion impala::tpcds_kudu_2.call_center impala::tpcds_kudu_2.web_site
```

步骤7 运行成功后，运行成功后可见 kudu 表都已经恢复。

kudu table list cdp-master-1.c-e34371c2cc31480a

```
[root@cdp-utility-1 ~]# kudu table list cdp-master-1.c-e34371c2cc31480a
impala::tpcds_kudu_2.catalog_sales
impala::tpcds_kudu_2.store_returns
impala::tpcds_kudu_2.store_sales
impala::tpcds_kudu_2.time_dim
impala::tpcds_kudu_2.web_site
impala::tpcds_kudu_2.promotion
impala::tpcds_kudu_2.reason
impala::tpcds_kudu_2.warehouse
impala::tpcds_kudu_2.item
impala::default.call_center
impala::tpcds_kudu_2.inventory
impala::tpcds_kudu_2.customer
impala::tpcds_kudu_2.ship_mode
impala::tpcds_kudu_2.store
impala::tpcds_kudu_2.customer_demographics
impala::tpcds_kudu_2.web_page
impala::tpcds_kudu_2.web_sales
impala::tpcds_kudu_2.household_demographics
impala::tpcds_kudu_2.call_center
impala::tpcds_kudu_2.customer_address
impala::tpcds_kudu_2.date_dim
impala::tpcds_kudu_2.income_band
impala::tpcds_kudu_2.catalog_page
impala::tpcds_kudu_2.web_returns
impala::tpcds_kudu_2.catalog_returns
```

步骤8 为了能使用 impala 查询 kudu 表，需进一步创建 impala 外表进行映射。首先查看 tpcds_kudu_2.db 文件授予 hdfs 用户的读写权限。

查看权限

hadoop fs -ls /user/hive/warehouse/tpcds_kudu_2.db

CDP 集群默认不允许切换用户到 hdfs，需先进行修改

vi /etc/passwd

把 hdfs:x:1008:1008:Hadoop HDFS:/var/lib/hadoop-hdfs:/sbin/nologin 编辑为

hdfs:x:1008:1008:Hadoop HDFS:/var/lib/hadoop-hdfs:/bin/bash

切换用户

su hdfs

给 DB 授权

hdfs dfs -chmod 777 /user/hive/warehouse/tpcds_kudu_2.db

退出

exit

```

apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
cdp:x:1000:10:/home/cdp:/bin/bash
accumulo:x:1001:1001:Accumulo:/var/lib/accumulo:/sbin/nologin
atlas:x:1002:1002:Atlas:/var/lib/atlas:/sbin/nologin
cloudera-scm:x:1003:1003:Cloudera Manager:/var/lib/cloudera-scm-server:/sbin/nologin
cruisecontrol:x:1004:1004:Cruise Control:/var/lib/cruise_control:/sbin/nologin
druid:x:1005:1005:Druid:/var/lib/druid:/sbin/nologin
flume:x:1006:1006:Flume:/var/lib/flume-ng:/sbin/nologin
hbase:x:1007:1007:HBase:/var/lib/hbase:/sbin/nologin
#hdfs:x:1008:1008:Hadoop HDFS:/var/lib/hadoop-hdfs:/sbin/nologin
hdfs:x:1008:1008:Hadoop HDFS:/var/lib/hadoop-hdfs:/bin/bash
hive:x:1009:1009:Hive:/var/lib/hive:/sbin/nologin
https:x:1010:1010:Hadoop HTTPS:/var/lib/hadoop-https:/sbin/nologin
hue:x:1011:1011:Hue:/usr/lib/hue:/sbin/nologin
impala:x:1012:1012:Impala:/var/lib/impala:/sbin/nologin
kafka:x:1013:1013:Kafka:/var/lib/kafka:/sbin/nologin
keytrustee:x:1014:1014:KeyTrustee KMS:/var/lib/keytrustee:/sbin/nologin
kms:x:1015:1015:Hadoop KMS:/var/lib/hadoop-kms:/sbin/nologin
knox:x:1016:1016:Knox:/var/lib/knox:/sbin/nologin
kudu:x:1017:1017:Kudu:/var/lib/kudu:/sbin/nologin
livy:x:1018:1018:Livy:/var/lib/livy:/sbin/nologin
mapred:x:1019:1019:Hadoop MapReduce:/var/lib/hadoop-mapreduce:/sbin/nologin

[root@cdp-utility-1 ~]# hdfs dfs -chmod 777 /user/hive/warehouse/tpcds_kudu_2.db

```

步骤9 把创建 impala 外表的 sql 脚本拷贝的 utility 节点。

```

# xianwang @ IT-FVFX52UDHV2H in ~ on git:master x [10:00:55]
$ scp /Users/xianwang/Desktop/弦望的最佳实践/CDH迁移CDP最佳实践/democode/tpcds_kudu_external.sql root@47.99.118.247:/
root@47.99.118.247's password:
tpcds_kudu_external.sql          100% 3608   364.0KB/s   00:00

```

步骤10 使用 impala-shell 命令执行脚本（这里注意替换 core 节点地址和 master 节点地址）。可见外表都已成功创建。

```
impala-shell -i cdp-core-1.c-e34371c2cc31480a:21000 --var=MASTER=cdp-master-1.c-e34371c2cc31480a -f /tpcds_kudu_external.sql
```

```
[root@cdp-utility-1 ~]# impala-shell -i cdp-core-1.c-e34371c2cc31480a:21000 --var=MASTER=cdp-master-1.c-e34371c2cc31480a -f /tpcds_kudu_external.sql
```



```
| summary |
+-----+
| Table has been created. |
+-----+
Fetched 1 row(s) in 0.03s
Query: create EXTERNAL TABLE tpcds_kudu_2.item
STORED AS KUDU
TBLPROPERTIES ('kudu.master_addresses' = 'cdp-master-1.c-e34371c2cc31480a:7051',
'kudu.table_name'='impala::tpcds_kudu_2.item')
+-----+
| summary |
+-----+
| Table has been created. |
+-----+
Fetched 1 row(s) in 0.02s
Query: create EXTERNAL TABLE tpcds_kudu_2.promotion
STORED AS KUDU
TBLPROPERTIES ('kudu.master_addresses' = 'cdp-master-1.c-e34371c2cc31480a:7051',
'kudu.table_name'='impala::tpcds_kudu_2.promotion')
+-----+
| summary |
+-----+
| Table has been created. |
+-----+
Fetched 1 row(s) in 0.02s
Query: create EXTERNAL TABLE tpcds_kudu_2.call_center
STORED AS KUDU
TBLPROPERTIES ('kudu.master_addresses' = 'cdp-master-1.c-e34371c2cc31480a:7051',
'kudu.table_name'='impala::tpcds_kudu_2.call_center')
+-----+
| summary |
+-----+
| Table has been created. |
+-----+
Fetched 1 row(s) in 0.03s
Query: create EXTERNAL TABLE tpcds_kudu_2.web_site
STORED AS KUDU
TBLPROPERTIES ('kudu.master_addresses' = 'cdp-master-1.c-e34371c2cc31480a:7051',
'kudu.table_name'='impala::tpcds_kudu_2.web_site')
+-----+
| summary |
+-----+
| Table has been created. |
+-----+
Fetched 1 row(s) in 0.03s
```

步骤11 进一步查看 kudu 表，可见 impala 已经可以查看到 kudu 表了。

```
impala-shell -i cdp-core-1.c-e34371c2cc31480a:21000
use tpcds_kudu_2;
show tables;
```

```

Fetches: 8 row(s) in 0.01s
[cdp-core-1.c-e34371c2cc31480a:21000] default> use tpcds_kudu_2;
Query: use tpcds_kudu_2
[cdp-core-1.c-e34371c2cc31480a:21000] tpcds_kudu_2> show tables;
Query: show tables
+-----+
| name |
+-----+
| call_center |
| customer |
| customer_address |
| date_dim |
| household_demographics |
| income_band |
| inventory |
| item |
| promotion |
| reason |
| ship_mode |
| store |
| store_returns |
| store_sales |
| time_dim |
| warehouse |
| web_page |
| web_returns |
| web_sales |
| web_site |
+-----+

```

步骤12 进一步查看 kudu 表中的演示数据，可见数据条数和 CDH 中的一样。

select count(1) from item;

```

[cdp-core-1.c-e34371c2cc31480a:21000] tpcds_kudu_2> select count(1) from item;
Query: select count(1) from item
Query submitted at: 2021-09-07 10:04:19 (Coordinator: http://cdp-core-1.c-e34371c2cc31480a:25000)
Query progress can be monitored at: http://cdp-core-1.c-e34371c2cc31480a:25000/query_plan?query_id=fc4b8094ba519169:e78b806900000000
+-----+
| count(1) |
+-----+
| 26000 |
+-----+
Fetches: 1 row(s) in 0.29s

```

2.4.Hbase 迁移

步骤1 首先登录 CDH 的 master 节点，查看当前集群 hbase 中快照。可见当前集群还没有快照。

hbase shell

list_snapshots

```
[root@master ~]# hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
21/09/07 15:22:37 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.16.2, rUnknown, Mon Jun  3 03:50:03 PDT 2019

hbase(main):001:0> list_snapshots
SNAPSHOT          TABLE + CREATION TIME
0 row(s) in 0.1510 seconds

=> []
```

步骤2 给 hbase_item 表打快照。可见已成功打了快照。

```
snapshot 'hbase_item','snapshot_item'
```

```
list_snapshots
```

```
quit
```

```
hbase(main):002:0> snapshot 'hbase_item','snapshot_item'
0 row(s) in 0.3390 seconds

hbase(main):003:0> list_snapshots
SNAPSHOT          TABLE + CREATION TIME
snapshot_item     hbase_item (Tue Sep 07 15:24:35 +0800 2021)
1 row(s) in 0.0130 seconds

=> ["snapshot_item"]
```

步骤3 查看快照所在的 HDFS 文件，可见存在.snapshotinfo 及.manifest 文件。

```
export HADOOP_USER_NAME=hdfs
```

```
hdfs dfs -ls /hbase/.hbase-snapshot
```

```
hdfs dfs -ls /hbase/.hbase-snapshot/snapshot_item
```

```
[root@master ~]# export HADOOP_USER_NAME=hdfs
[root@master ~]# hdfs dfs -ls /hbase/.hbase-snapshot
Found 2 items
drwxr-xr-x - hbase hbase      0 2021-09-07 15:24 /hbase/.hbase-snapshot/.tmp
drwxr-xr-x - hbase hbase      0 2021-09-07 15:24 /hbase/.hbase-snapshot/snapshot_item
[root@master ~]# hdfs dfs -ls /hbase/.hbase-snapshot/snapshot_item
Found 2 items
-rw-r--r--  3 hbase hbase      38 2021-09-07 15:24 /hbase/.hbase-snapshot/snapshot_item/.snapshotinfo
-rw-r--r--  3 hbase hbase     380 2021-09-07 15:24 /hbase/.hbase-snapshot/snapshot_item/data.manifest
```

步骤4 从 CDH 集群拷贝快照到 CDP 集群。

```
export HADOOP_USER_NAME=hbase
```

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot \
```

```
-snapshot snapshot_item \
```

```
-copy-from hdfs://master/hbase \
```

```
-copy-to hdfs://cdp-master-1.c-e34371c2cc31480a/hbase \
```

```
-mappers 20 \
```

```
-bandwidth
```

20

```
[root@cdp-utility-1 ~]# hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot \
> -snapshot snapshot_item \
> -copy-from hdfs://master/hbase \
> -copy-to hdfs://cdp-master-1.c-e34371c2cc31480a/hbase \
> -mappers 20 \
> -bandwidth 20
2021-09-10 14:08:57,234 INFO [main] snapshot.ExportSnapshot: Copy Snapshot Manifest from hdfs://master/hbase/.hbase-snapshot/snapshot_item to hdfs://cdp-master-1.c-e34371c2cc31480a/hbase/.hbase-snapshot/.tmp/snapshot_item
2021-09-10 14:08:57,621 INFO [main] client.RMPProxy: Connecting to ResourceManager at cdp-master-1.c-e34371c2cc31480a/192.168.10.203:8032
2021-09-10 14:08:57,868 INFO [main] mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/hbase/.staging/job_1631253786217_0002
2021-09-10 14:08:58,489 INFO [main] snapshot.ExportSnapshot: Loading Snapshot 'snapshot_item' hfile list
2021-09-10 14:08:58,668 INFO [main] mapreduce.JobSubmitter: number of splits:1
2021-09-10 14:08:58,807 INFO [main] mapreduce.JobSubmitter: Submitting tokens for job: job_1631253786217_0002
2021-09-10 14:08:58,808 INFO [main] mapreduce.JobSubmitter: Executing with tokens: []
2021-09-10 14:08:58,991 INFO [main] conf.Configuration: resource-types.xml not found
2021-09-10 14:08:58,991 INFO [main] resource.ResourceUtils: Unable to find 'resource-types.xml'.
```

步骤5 首先登录 CDP hbase 的 master 节点（这里安装在 core1 上），查看当前集群 hbase 中快照。可见快照已成功导入。

```
ssh root@cdp-core-1.c-e34371c2cc31480a
```

```
hbase shell
```

```
list_snapshots
```

```
quit
```

```
[root@cdp-utility-1 ~]# ssh root@cdp-core-1.c-e34371c2cc31480a
root@cdp-core-1.c-e34371c2cc31480a's password:
Permission denied, please try again.
root@cdp-core-1.c-e34371c2cc31480a's password:
Last failed login: Fri Sep 10 14:19:47 CST 2021 from 192.168.10.200 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Fri Sep 10 14:16:11 2021 from 192.168.10.200

Welcome to Alibaba Cloud Elastic Compute Service !

[root@cdp-core-1 ~]# hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.2.3.7.1.7.0-551, r0b5466e242cb6c55797291b2974957f06830ff09, Tue Aug 3 21:10:27 UTC 2021
Took 0.0017 seconds
hbase:001:0> list_snapshots
SNAPSHOT          TABLE + CREATION TIME
snapshot_item     hbase_item (2021-09-07 15:24:35 +0800)
1 row(s)
Took 0.3563 seconds
=> ["snapshot_item"]
```

步骤6 查看快照所在的 HDFS 文件，可见存在.snapshotinfo 及.manifest 文件。

```
hdfs dfs -ls /hbase/.hbase-snapshot
```

```
hdfs dfs -ls /hbase/.hbase-snapshot/snapshot_item
```

```
[root@cdp-core-1 ~]# hdfs dfs -ls /hbase/.hbase-snapshot
Found 2 items
drwxr-xr-x - hbase hbase          0 2021-09-10 14:09 /hbase/.hbase-snapshot/.tmp
drwxr-xr-x - hbase hbase          0 2021-09-10 14:08 /hbase/.hbase-snapshot/snapshot_item
[root@cdp-core-1 ~]# hdfs dfs -ls /hbase/.hbase-snapshot/snapshot_item
Found 2 items
-rw-r--r--  2 hbase hbase          38 2021-09-10 14:08 /hbase/.hbase-snapshot/snapshot_item/.snapshotinfo
-rw-r--r--  2 hbase hbase         380 2021-09-10 14:08 /hbase/.hbase-snapshot/snapshot_item/data.manifest
```

步骤7 查看 CDP 集群上的 Hbase 表，当前不存在任何表。

```
hbase shell
```

```
list
```

```
[root@cdp-core-1 ~]# hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.2.3.7.1.7.0-551, r0b5466e242cb6c55797291b2974957f06830ff09, Tue Aug 3 21:10:27 UTC 2021
Took 0.0012 seconds
hbase:001:0> list
TABLE
0 row(s)
Took 0.3259 seconds
=> []
```

步骤8 创建 Hbase 表并查看。

```
create 'hbase_item','cf1'
```

```
list
```

```
hbase:003:0> create 'hbase_item','cf1'
Created table hbase_item
Took 0.8263 seconds
=> Hbase::Table - hbase_item
hbase:004:0> list
TABLE
hbase_item
1 row(s)
Took 0.0055 seconds
=> ["hbase_item"]
hbase:005:0>
```

步骤9 为了防止其他操作影响'hbase_item'表，先禁用表，再恢复。然后查看表数据，可见从 CDH 迁移过来的数据可以正常查看了。

```
disable 'hbase_item'
```

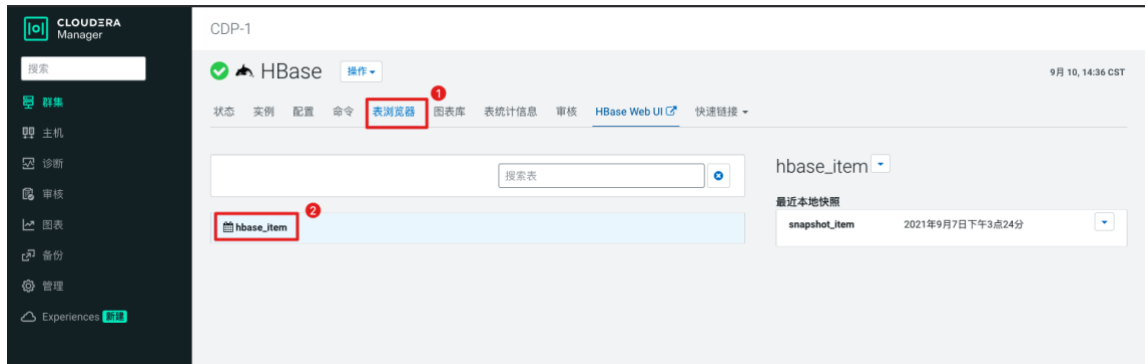
```
restore_snapshot 'snapshot_item'
```

```
enable 'hbase_item'
```

```
scan 'hbase_item'
```

```
14402 column=cf1:i_color, timestamp=1630401866693, value=plum
14402 column=cf1:i_container, timestamp=1630401866693, value=Unknown
14402 column=cf1:i_current_price, timestamp=1630401866693, value=10
14402 column=cf1:i_formulation, timestamp=1630401866693, value=593sky90185734959539
14402 column=cf1:i_item_desc, timestamp=1630401866693, value=Authorities retain with a auth
      orities. Warm, commercial things can bring. Eyes buy also for the minds. P
14402 column=cf1:i_item_id, timestamp=1630401866693, value=AAAAAAAAAPENCAAAA
14402 column=cf1:i_item_sk, timestamp=1630401866693, value=11599
14402 column=cf1:i_manager_id, timestamp=1630401866693, value=17
14402 column=cf1:i_manufact, timestamp=1630401866693, value=oughtcallyable
14402 column=cf1:i_manufact_id, timestamp=1630401866693, value=261
14402 column=cf1:i_product_name, timestamp=1630401866693, value=n stn stantioughtought
14402 column=cf1:i_rec_start_date, timestamp=1630401866693, value=1997-10-27
14402 column=cf1:i_size, timestamp=1630401866693, value=N/A
14402 column=cf1:i_units, timestamp=1630401866693, value=Dozen
14402 column=cf1:i_wholesale_cost, timestamp=1630401866693, value=6
14403 column=cf1:i_brand, timestamp=1630401866693, value=edu packmaxi #9
14403 column=cf1:i_brand_id, timestamp=1630401866693, value=8014009
14403 column=cf1:i_category, timestamp=1630401866693, value=Sports
14403 column=cf1:i_category_id, timestamp=1630401866693, value=8
14403 column=cf1:i_class, timestamp=1630401866693, value=tennis
```

步骤10 在 CDP 的控制台也可以正常查看表数据。登录 Hbase Web UI 可以查看更多详细信息。



3. 组件迁移

3.1. Sentry 迁移

3.1.1. 概述

与 Sentry 相比，Ranger 提供了更好的用户体验。Ranger 有更好的 UI，更灵活的访问控制模型，该模型包括基于属性的访问控制以及与 Atlas 集成。

为了简化迁移过程，Ranger 2.0 版本还进行了功能扩展，例如：支持基于角色的权限管理、提供 HDFS ACL Sync 功能，理论上可以实现 1 对 1 权限映射。

3.1.2. 迁移方法

Cloudera 提供两类 Sentry 迁移工具：

- Replication Manager：在定制 Hive 复制作业的时候，勾选” import both Hive object and URL permissions ” 或者” import only Hive object permissions”
 - Include Sentry Permissions with Metadata- 选择此选项可在复制作业期间迁移 Sentry 权限。
 - Exclude Sentry Permissions with Metadata- 如果您不想在复制作业期间迁移 Sentry 权限，请选择此选项。
 - Skip URI Privileges- 如果您不想在迁移 Sentry 权限时包含 URI 权限，请选择此选项。在迁移期间，URI 权限被转换为指向 S3 中的等效位置。如果资源在 Amazon S3 中具有不同的位置，请不要迁移 URI 权限，因为 URI 权限可能无效。
- authzmigrator：在导出 Sentry acl 的过程中，将其转换为 Ranger 可以读取的格式，然后通过 CM 界面执行” Import Sentry Policies” 命令。

推荐使用第二种 authzmigrator 进行 Sentry 迁移。

3.1.2.1. 复制策略的 Sentry 权限部分

下图显示了创建复制策略向导中的 Sentry 权限部分：

Sentry 策略向 Ranger 的迁移通过以下操作执行：

- **导出**- 导出操作在源集群中运行。在此操作期间，将获取 Sentry 权限并将其导出到 JSON 文件。根据配置，此文件可能位于本地文件系统或 HDFS 中。
- **翻译和摄取**- 这些操作发生在目标集群上。在转换操作期间，Sentry 权限被转换为 Ranger 可以读取的格式。然后将权限导入 Ranger。导入权限时，它们会被标记为源集群名称和摄取发生的时间。导入后，包含权限的文件将被删除。

为资源（例如数据库、表或列）创建 Ranger 策略时，策略名称源自资源名称。例如，如果资源是 Database:dinosaurs, table= theropods，则派生的策略名称是 database=dinosaurs->table=theropods。

迁移策略的优先级在 Ranger 中设置为 Normal。普通优先级允许您为相同的资源创建另一个策略来覆盖从 Sentry 导入的策略。

3.1.3. 转换规则

从 Sentry 到 Ranger 的转换规则如下：

- 授予角色的 Sentry 权限将授予 Ranger 中的 group。
- 授予父对象的 Sentry 权限也会被授予子对象。迁移过程保留了应用于子对象的权限。例如，在数据库级别应用的权限也将应用于该数据库内的表。
- Sentry OWNER 权限会被转换为 Ranger OWNER 权限。
- Sentry OWNER WITH GRANT OPTION 将转换为 Ranger OWNER 并带有 Delegated Admin。
- Sentry 不会区分表和视图。迁移视图权限时，它们将被视为表名。
- 如果您的集群包含 Kafka 服务，并且 Kafka sentry 策略具有“action”：“ALL”权限，那么迁移后的 Ranger 策略将缺少“alter”权限。升级后，您将需要手动添加该策略。请注意：缺少此权限不会产生任何功能影响。仅出于完整性考虑，才需要在升级后添加“alter”权限，因为拥有“configure”权限同样允许“alter”操作。

Sentry Action	Ranger Action
SELECT	SELECT
INSERT	UPDATE
CREATE	CREATE
REFRESH	REFRESH
ALL	ALL
SELECT with Grant	SELECT
INSERT with Grant	UPDATE
CREATE with Grant	CREATE
ALL with Grant	ALL with Delegated Admin Checked
ALTER	CONFIGURE

3.1.4. 风险提示

□ 需要手工调整

Solr 的 Sentry 权限目前无法自动转换，需要手动转换为 Ranger 策略。

与任何自动化工具一样，权限转换结果有可能会和预期不一致。因此，自动化迁移过程非常适合进行先期测试：将 Sentry acl 导入到测试环境的 Ranger 中，提前进行转换和验证，而不会影响生产集群。在升级之前，建议备份 Sentry 数据库。

□ Impala GRANT 语句改造

Sentry 支持通过 hive/impala SQL 中 grant 命令行赋权。目前在 Ranger 还不支持 Impala GRANT 语句赋权（计划在 CDP 7.1.6 中支持），因此相应的代码需要手工改成调用 Ranger API。

□ Replication Manager 的当前限制

如果启用 Replication Manager 的 import Sentry Permissions 功能，必须注意：

- 1) 不管源集群的 Sentry acl 是否发生变化，每次 RM 复制时都会重新创建 Ranger acl。
- 2) 即使 Hive 复制选择的是数据库的子集，则对于其中的每张表，数据库级 acl 都将转换为等效的表级 acl。（例如，对于每张表，数据库上的 ALL->每张表上的 ALL）。

- 3) 在 Ranger 中将不会引用源集群的 Sentry 定义的角色。Ranger acl 是基于资源而不是基于角色直接授予组和用户的。这个和通过 authzmigrator 工具迁移是完全不同的，authzmigrator 工具会基于角色。

3.1.5. 迁移步骤

3.1.5.1. 导出 Sentry 权限

1. 下载 authz_export.tar.gz 文件并解压。有关下载文件的信息，请联系阿里云或者 Cloudera 支持。
2. 该 authz_export.tar.gz 文件包含一个名为 jar 文件和目录的配置。它还有一个 authz_export.sh 文件。该配置目录包含可用于参考的默认配置。目录结构如下：
 - a) Jars
 - b) Config
 - c) Authz_export.sh

```
[root@master ~]# tar -zxvf authz_export.tar.gz
authzmigrator/
authzmigrator/jars/
authzmigrator/authz_export.sh
authzmigrator/config/
authzmigrator/config/authorization-migration-site.xml
authzmigrator/config/core-site.xml
authzmigrator/config/sentry-site.xml
authzmigrator/jars/authz-ingestor-1.2.0.jar
authzmigrator/jars/authz-common-1.2.0.jar
authzmigrator/jars/authz-main-1.2.0.jar
authzmigrator/jars/authz-translator-1.2.0.jar
authzmigrator/jars/authz-exporter-1.2.0.jar
[root@master ~]# ls
apache-maven-3.6.0-bin.tar.gz  authz_export.tar.gz  authzmigrator  cdh  extwarehouse  hive-testbench  ossutil64
[root@master ~]# cd authzmigrator/
[root@master authzmigrator]# ls -ltr
总用量 12
-rwxr--r-- 1 502 wheel  587 1月  9 2021 authz_export.sh
drwxr-xr-x 2 502 wheel 4096 1月  9 2021 config
drwxr-xr-x 2 502 wheel 4096 9月 10 18:45 jars
[root@master authzmigrator]#
```

3. 使用源集群中的 sentry 服务器的 sentry 运行目录中的配置文件替换上述解压的 sentry-site.xml 和 core-site.xml 文件。sentry 的运行目录为:/var/run/cloudera-scm-agent/process/<sentry-service>/

```
[root@master process]# cd /var/run/cloudera-scm-agent/process/335-sentry-SENTRY_SERVER
[root@master 335-sentry-SENTRY_SERVER]# ls
cloudera-monitor.properties  core-site.xml  navigator.client.properties  redaction-rules.json  sentry-site.xml
cloudera-stack-monitor.properties  creds.localjceks  process_timestamp  sentry.keytab  ssl-client.xml
config.zip  logs  proc.json  sentry-log4j.properties  supervisor.conf
[root@master 335-sentry-SENTRY_SERVER]#
```

4. 编辑 sentry-site.xml 文件以执行以下步骤：
 - a) 使用以下凭据更新 Sentry 数据库的数据库用户名和密码：

- a) sentry.store.jdbc.user
- b) sentry.store.jdbc.password

```
<property>
  <name>sentry.store.jdbc.user</name>
  <value>sentry</value>
</property>
<property>
  <name>sentry.store.jdbc.password</name>
  <value>*****</value>
</property>
```

- b) 删除文件中的 hadoop.security.credential.provider.path 属性。

```
<property>
  <name>hadoop.security.credential.provider.path</name>
  <value>localjceks://file//run/cloudera-scm-agent/process/335-sentry-SENTRY_SERVER/creds.localjceks</value>
</property>
```

5. 编辑 core-site.xml 文件以执行以下步骤:

- a) 将属性 fs.defaultFS 的值更新为 [file:///](#)。

```
<!--Autogenerated by Cloudera Manager-->
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>file:///</value>
  </property>
  <property>
    <name>fs.trash.interval</name>
    <value>1</value>
  </property>
```

6. 在 config 目录下的 authorization-migration-site.xml 文件中，执行以下步骤:

- a) 确保 authorization.migration.export.target_services 属性具有要为其导出权限的服务列表。

有效值包括: HIVE KAFKA

- b) 将 authorization.migration.export.output_file 属性中的信息更新为应导出权限的文件的绝对位置。

```

<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>authorization.migration.export.target_services</name>
    <value>HIVE,KAFKA</value>
  </property>
  <property>
    <name>authorization.migration.export.migration_objects</name>
    <value />
  </property>
  <property>
    <name>authorization.migration.export.cluster_name</name>
    <value>cluster1</value>
  </property>
  <property>
    <name>authorization.migration.export.output_file</name>
    <value>/tmp/permissions.json</value>
  </property>
  <property>
    <name>authorization.migration.ingest.is_dry_run</name>
    <value>>false</value>
  </property>
  <property>
    <name>authorization.migration.kerberos.authentication</name>
    <value>>false</value>
  </property>
</configuration>
~

```

7. 验证 Sentry 服务器的 Java 执行路径和 authz_export.sh 脚本中的 JAVA_HOME 属性是否匹配。要验证路径和属性，请执行以下任务：

- a) 要定位 Sentry 服务器使用的 Java 执行路径，请运行 `ps aux | grep org.apache.sentry.SentryMain` 命令。

```

[root@master config]# ps aux|grep org.apache.sentry.SentryMain
root      15066  0.0  0.0 112724 1020 pts/0    S+   21:09   0:00 grep --color=auto org.apache.sentry.SentryMain
sentry    27428  0.8  0.6 2172728 445948 ?        Sl   17:41   1:48 /usr/java/jdk1.8.0_301/bin/java -Xmx1000m -Dhadoop.log.dir=/opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8
/lib/hadoop/logs -Dhadoop.log.file=hadoop.log -Dhadoop.home.dir=/opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/hadoop -Dhadoop.id.str= -Dhadoop.root.logger=INFO,console
-Djava.library.path=/opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/hadoop/lib/native -Dhadoop.policy.file=hadoop-policy.xml -Djava.net.preferIPv4Stack=true -Djava.
net.preferIPv6Stack=true -Xms268435456 -Xmx268435456 -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp/sentry_sentry-SENTRY_SERVER-a564c56c3682e2be157de3ec22eb0d3f_pid27
428.hprof -XX:OnOutOfMemoryError=/usr/lib64/cmf/service/common/killparent.sh -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.util.RunJar /opt/cloudera/parcels/C
DH-5.16.2-1.cdh5.16.2.p0.8/lib/sentry/lib/sentry-core-common-1.5.1-cdh5.16.2.jar org.apache.sentry.SentryMain --command service --log4jConf /run/cloudera-scm-agent/process/3
53-sentry-SENTRY_SERVER-sentry-log4j.properties -confFile /run/cloudera-scm-agent/process/353-sentry-SENTRY_SERVER/sentry-site.xml
[root@master config]#

```

- b) 如果路径不是 `/user/java/default/bin/java`，请编辑 `authz_export.sh` 脚本，将 Sentry 服务器使用的路径添加到 `JAVA_HOME` 属性，然后保存文件。
- c) 例如，如果 Sentry 服务器使用 `/usr/java/jdk1.8.0_141-cloudera/bin/java` 路径，请将 `authz_export.sh` 脚本中的 `JAVA_HOME` 属性更改为 `/usr/java/jdk1.8.0_141-cloudera`。

```
JAVA_HOME=/usr/java/jdk1.8.0_301
SQL_CONNECTOR=/usr/share/java/mysql-connector-java.jar:/opt/cloudera/cm/lib/postgresql-42.1.4.jre7.jar:/usr/share/java/postgresql-connector-java.jar
SENTRY_HOME=/opt/cloudera/parcels/CDH/lib/sentry
HADOOP_CLASSPATH=`hadoop classpath`
AUTHZMIGRATOR_HOME=`dirname $0`
AUTHZMIGRATOR_HOME=`cd "$AUTHZMIGRATOR_HOME/jars"; pwd`
CLASSPATH=${AUTHZMIGRATOR_HOME}/*:${SENTRY_HOME}/lib/*:${SENTRY_HOME}/lib/server/*:${SQL_CONNECTOR}:${HADOOP_CLASSPATH}
exec ${JAVA_HOME}/bin/java -cp ${CLASSPATH} com.cloudera.enterprise.authzmigrator.Main -c export -dr ./config/
```

8. 使用 `sh authz_export.sh` 命令运行 `authz_export.sh` 脚本。

权限导出到 `/opt/backup/permissions.json` 文件。参考日志为：

```
[root@master authzmigrator]# sh authz_export.sh

log4j:WARN No such property [conversionPattern] in
org.apache.solr.util.SolrLogLayout.

277 T1 cceacv.CommonConfigurationValidator.validateServiceAndAuthzObjects No
migration objects specified. Migrate all privileges

293 T1 cceacv.CommonConfigurationValidator.validatePathScheme No matching
schemes with [[file, hdfs, s3a, s3, abfs]]. Use default scheme [file]

414 T1 odu.Log4JLogger.info Property datanucleus.schema.validateColumns
unknown - will be ignored

416 T1 odu.Log4JLogger.info Property datanucleus.schema.validateConstraints
unknown - will be ignored

416 T1 odu.Log4JLogger.info Property datanucleus.cache.level2 unknown - will
be ignored

416 T1 odu.Log4JLogger.info Property datanucleus.schema.validateTables unknown
- will be ignored

589 T1 cjb.BoneCPConfig.sanitize WARN Max Connections < 1. Setting to 20

1431 T1 odu.Log4JLogger.warn WARN Metadata has jdbc-type of CHAR(40) yet this
is not valid. Ignored

1612 T1 cjb.BoneCPConfig.sanitize WARN Max Connections < 1. Setting to 20
```

```
1885 T1 ccea.PolicyStore.lambda$fetchPermissionInfo$1 Fetching permissions for
all SQL objects for service type [HIVE]
1942 T1 ccea.PolicyStore.fetchDbPermissionInfo Total privileges retrieved [3]
1950 T1 ccea.PolicyStore.lambda$getRoleGroupMap$4 Fetched 3 roles
1951 T1 ccea.PolicyStore.lambda$getRoleGroupMap$4 Roles to group mapping
retrieved [3]
1977 T1 ccea.PolicyStore.fetchKafkaPermissionInfo No privileges retrieved for
export
1979 T1 ccea.SentryExportTask.constructPolicyList [Completed] Constructing
policy list
2013 T1 ccea.FSClient.write [Started] Writing policy information to
/tmp/permissions.json
2019 T1 oahu.NativeCodeLoader.<clinit> WARN Unable to load native-hadoop
library for your platform... using builtin-java classes where applicable
2026 T1 ccea.FSClient.write [Completed] Writing policy information to
/tmp/permissions.json
2026 T1 ccea.SentryExportTask.execute Exporting permission information to
location /tmp/permissions.json successful
2026 T1 ccea.Main.main Exporting the permissions is complete
[root@master authzmigrator]#
```

```
[root@master authzmigrator]# sh authz_export.sh
log4j:WARN No such property [conversionPattern] in org.apache.solr.util.SolrLogLayout.
277 T1 cceacv.CommonConfigurationValidator.validateServiceAndAuthzObjects No migration objects specified. Migrate all privileges
293 T1 cceacv.CommonConfigurationValidator.validatePathScheme No matching schemes with [[file, hdfs, s3a, s3, abfs]]. Use default scheme [file]
414 T1 odu.Log4JLogger.info Property datanucleus.schema.validateColumns unknown - will be ignored
416 T1 odu.Log4JLogger.info Property datanucleus.schema.validateConstraints unknown - will be ignored
416 T1 odu.Log4JLogger.info Property datanucleus.cache.level2 unknown - will be ignored
416 T1 odu.Log4JLogger.info Property datanucleus.schema.validateTables unknown - will be ignored
589 T1 cjb.BoneCPConfig.sanitize WARN Max Connections < 1. Setting to 20
1431 T1 odu.Log4JLogger.warn WARN Metadata has jdbc-type of CHAR(40) yet this is not valid. Ignored
1612 T1 cjb.BoneCPConfig.sanitize WARN Max Connections < 1. Setting to 20
1885 T1 ccea.PolicyStore.lambda$fetchPermissionInfo$1 Fetching permissions for all SQL objects for service type [HIVE]
1942 T1 ccea.PolicyStore.fetchDbPermissionInfo Total privileges retrieved [3]
1950 T1 ccea.PolicyStore.lambda$getRoleGroupMap$4 Fetched 3 roles
1951 T1 ccea.PolicyStore.lambda$getRoleGroupMap$4 Roles to group mapping retrieved [3]
1977 T1 ccea.PolicyStore.fetchKafkaPermissionInfo No privileges retrieved for export
1979 T1 ccea.SentryExportTask.constructPolicyList [Completed] Constructing policy list
2013 T1 ccea.FSClient.write [Started] Writing policy information to /tmp/permissions.json
2019 T1 oahu.NativeCodeLoader.<clinit> WARN Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2026 T1 ccea.FSClient.write [Completed] Writing policy information to /tmp/permissions.json
2026 T1 ccea.SentryExportTask.execute Exporting permission information to location /tmp/permissions.json successful
2026 T1 ccea.Main.main Exporting the permissions is complete
[root@master authzmigrator]#
```

```
[root@master tmp]# ls /tmp/permissions.json
/tmp/permissions.json
[root@master tmp]# cat /tmp/permissions.json
{
  "metaData": {
    "sentryVersion": "1.5.0-cdh5-2",
    "sourceClusterName": "cluster1",
    "serviceName": "[HIVE, KAFKA]",
    "exportVersion": 2.0,
    "roleBasedPermissions": false,
    "hdfsPrefixes": [
      ""
    ],
    "hiveServiceUser": "hive",
    "hiveGroupName": "hive"
  },
  "dbPolicies": [
    {
      "resource": {
        "authorizableType": "TABLE",
        "database": "default",
        "table": "test",
        "column": ""
      },
      "serviceType": "HIVE",
      "permissions": [
        {
          "principals": [
            {
              "principalType": "GROUP",
              "principalName": "fayson"
            }
          ]
        }
      ]
    }
  ]
}
```

3.1.5.2. 导入权限到 Ranger

1. 将导出的权限文件复制到目标集群中的以下 HDFS 位置。

```
/user/sentry/export-permissions/permissions.json
```

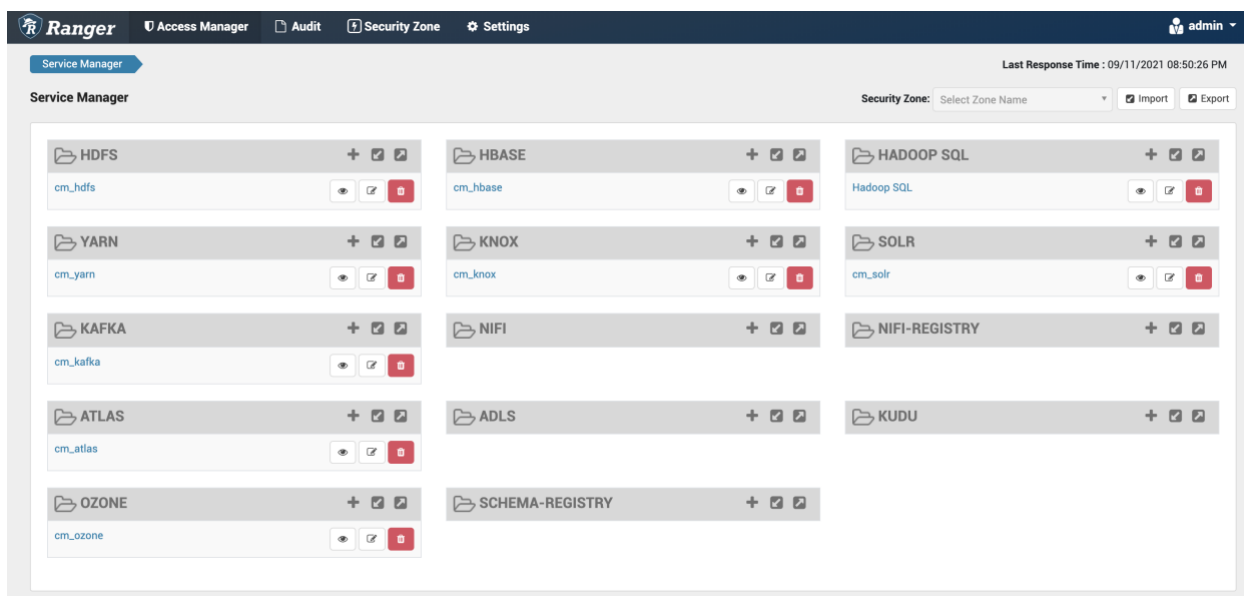
2. 使用该用户：“ranger” 对该文件具有读写权限。

```
hdfs dfs -setfacl -m -R user:ranger:rwx /user/sentry/export-
permissions/permissions.json
```

```
[root@cdp-utility-1 ~]# export HADOOP_USER_NAME=hdfs
[root@cdp-utility-1 ~]# hdfs dfs -mkdir -p /user/sentry/export-permissions
[root@cdp-utility-1 ~]# hdfs dfs -put permissions.json /user/sentry/export-permissions/
[root@cdp-utility-1 ~]# hdfs dfs -ls /user/sentry/export-permissions
Found 1 items
-rw-r--r--  2 hdfs supergroup      1368 2021-09-10 21:40 /user/sentry/export-permissions/permissions.json
[root@cdp-utility-1 ~]# hdfs dfs -setfacl -m -R user:ranger:rwx /user/sentry/export-permissions/permissions.json
[root@cdp-utility-1 ~]# hdfs dfs -ls /user/sentry/export-permissions
Found 1 items
-rw-rwxr--+  2 hdfs supergroup      1368 2021-09-10 21:40 /user/sentry/export-permissions/permissions.json
[root@cdp-utility-1 ~]#
```

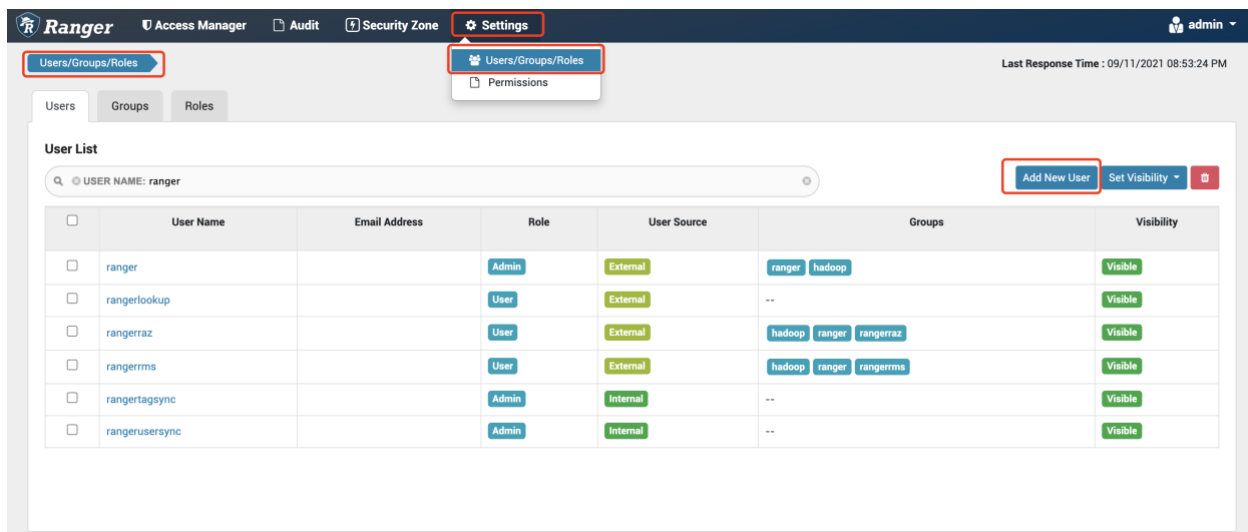
3. 登陆 CM，在 Ranger 服务中，单击操作>导入 Sentry 权限。

Ranger 同步权限前：

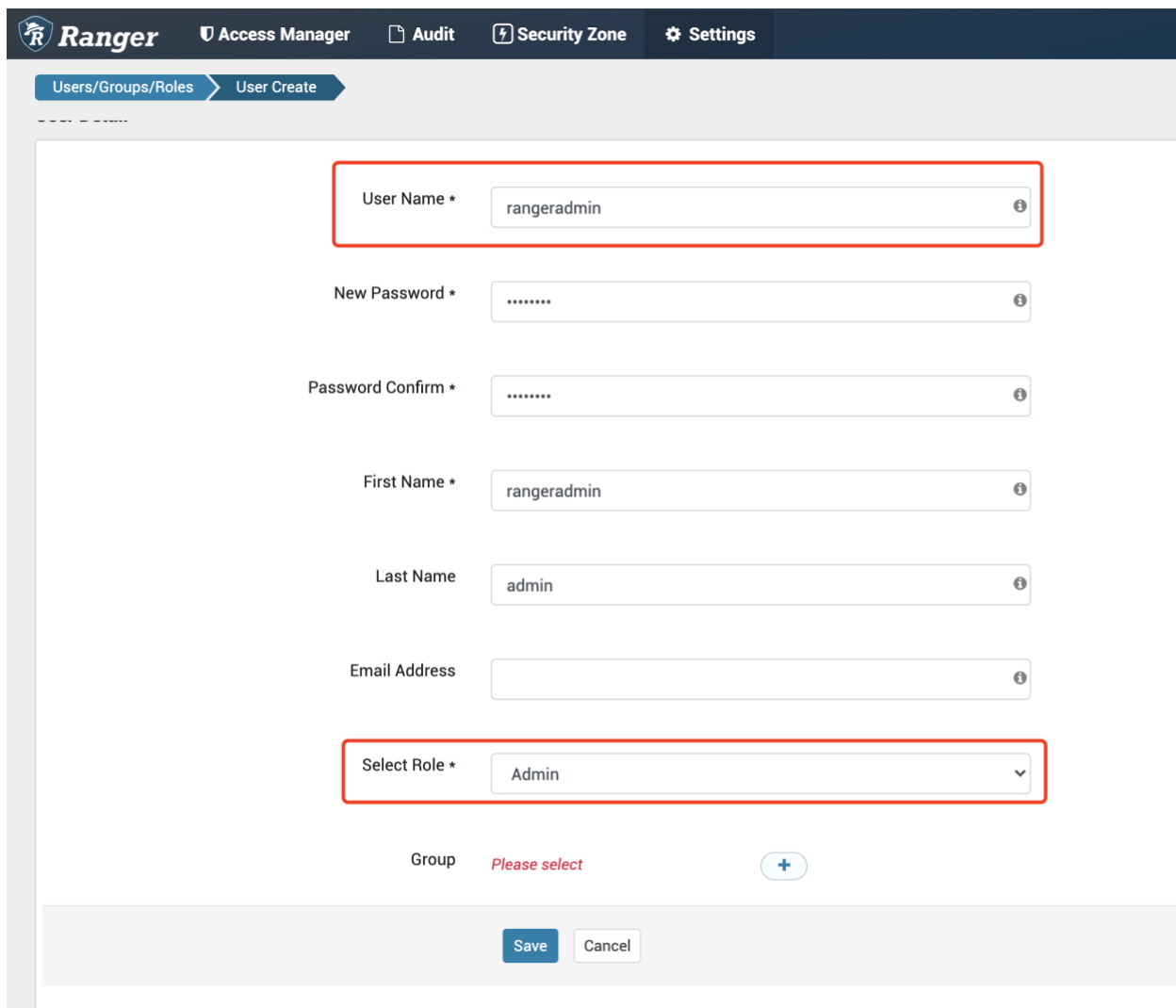


如果导入权限前没有配置启用对应的服务，则导入会报错。

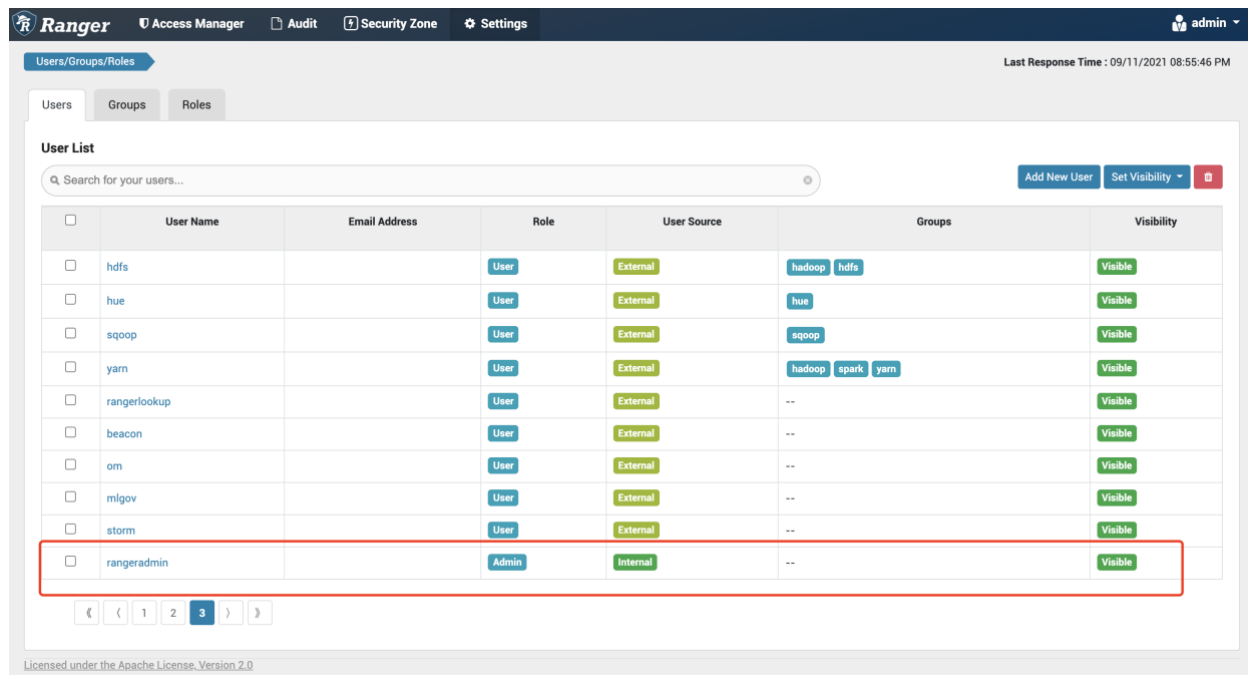
Step1. 当进程用户和 Kerberos principal 不匹配时, Ranger 的 Import Sentry Policies 命令不起作用。因此, 需要先将 rangeradmin 置为 Admin 角色, 因为 ranger.keytab 中的 Kerberos principal 为 rangeradmin。



添加 rangeradmin 用户



点击 save, 创建 rangeradmin 用户。



Ranger 服务获取 `/user/sentry/export-permissions/permissions.json` 文件中的权限。

CDP-1

The screenshot shows the Ranger management interface for CDP-1. The 'Actions' dropdown menu is open, listing various operations. The 'Import Sentry Policies' option is highlighted with a red box. The interface also displays 'Health Tests' (3 Good), 'Status Summary' (Ranger Admin, Tagsync, Usersync, Hosts all good), and 'Health History' (3 Became Good, 3 Became Disabled, 3 Became Unknown).

Component	Status	Count
Ranger Admin	Good	1
Ranger Tagsync	Good	1
Ranger Usersync	Good	1
Hosts	Good	1

点击 Import Sentry Policies

Import Sentry Policies

✕

Are you sure you want to run the **Import Sentry Policies** command on the service **Ranger**?

Cancel
Import Sentry Policies

Import Sentry Policies

✕

Status Running Context [Ranger](#) Sep 11, 8:51:42 PM Abort

Completed 0 of 1 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

Execute command **Import Sentry Policies** on role **Ranger Admin (cdp-utility-1)** Ranger Admin (cdp-utility-1) Sep 11, 8:51:42 PM Abort

Waiting for command (Import Sentry Policies (1153)) to finish

Abort Close

Step3. 导入成功

Id	Name	Enabled	Enabled	Roles	Groups	Users	Actions
7	all - global	Enabled	Enabled	cdp_global_admin, admin			rangerlookup, rangerlookup, appprofiler, More...
8	all - database, table, column	Enabled	Enabled	cdp_global_admin, admin			rangerlookup, rangerlookup, appprofiler, More...
9	all - database, table	Enabled	Enabled				hive, beacon, appprofiler, hive, More...
10	all - database	Enabled	Enabled		public		hive, beacon, appprofiler, hive, More...
11	all - hiveservice	Enabled	Enabled	cdp_global_admin, admin			rangerlookup, rangerlookup, appprofiler, More...
12	all - database, udf	Enabled	Enabled	cdp_global_admin, admin			rangerlookup, rangerlookup, appprofiler, More...
13	all - uri	Enabled	Enabled	cdp_global_admin, admin			rangerlookup, rangerlookup, appprofiler, More...
14	default database tables columns	Enabled	Enabled	default_select, default_r, default_all, cdep_default_admin	public		
15	Information_schema database tables ...	Enabled	Enabled		public		
44	grant-1607913665950	Enabled	Enabled	lbb			etl_user
45	uri-hdfs://wdyson-cdh-4.wdyson-cdh...	2020-12-15-21-39-34 Cluster1	Enabled	Enabled	external_db		
46	db=default/ldf*	2020-12-15-21-39-34 Cluster1	Enabled	Enabled	default_select, default_r, default_all, cdep_default_admin		

参考: <https://cloud.tencent.com/developer/article/1841962>

3.2. Navigator 迁移

3.2.1. 概述

Navigator 用于从 CDH 集群中收集元数据和审计信息。

在 CDP 中，Atlas 会替代 Navigator 收集元数据，Ranger 会替代 Navigator 收集审计信息。因此，Navigator 的功能会被 Ranger 和 Atlas 组合替代。

为什么 CDP 使用 Atlas 替代 Navigator？因为以下功能 Navigator 都不具备：

- 1) 使用 Atlas 元数据控制 Ranger 中的数据访问：Atlas classifications 可以控制表，行或列级别的访问；Ranger 可以使用 Atlas classifications 作为触发器来启用数据屏蔽功能。
- 2) 高可用和高可扩展性：Atlas 使用 Kafka 传递元数据，并使用 HBase 和 Solr 进行存储。这些组件都提供高可用和高可扩展性。
- 3) 增加其他元数据源：Atlas 通过 REST API 添加其他数据源的元数据模型时非常灵活。除了 Navigator 能够集成的那些数据源之外，Atlas 还支持 NiFi 和 Kafka 元数据。
- 4) 业务术语表：Atlas 提供了一个 Web 界面，用于创建和管理业务术语表，这些术语可以帮助组织通过标准化来识别和使用数据。
- 5) Data Profiling：“Data Catalog”提供自动数据标记功能，用于列出常见的数据类型，也允许用户通过正则表达式标记其他数据类型。

3.2.2. 迁移方法

目前 Cloudera 仅提供元数据迁移工具 nav2atlas (Navigator -> Atlas)。

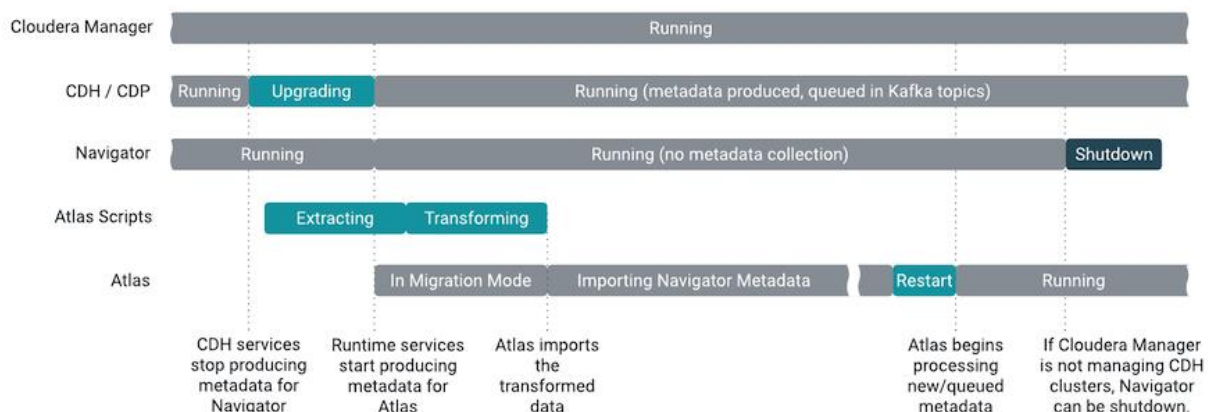
由于 Ranger Audit 和 Navigator Audit 在物理架构和数据格式方面的巨大差异，Cloudera **无法**提供审计数据迁移工具。

Navigator Audit 和 Ranger Audit 在物理架构和数据格式上有什么差异？

- Navigator: 首先由 HDFS 之类的服务在本地写入审计数据。然后，Cloudera Manager Agent 会读取这些文件，并将记录发送到 Navigator Server。最后，Navigator Server 将所有审计数据写入后台数据库。
- Ranger: 在 HDFS/HBase/Hive 等服务中嵌入 Ranger 插件，由 Ranger 插件直接将数据写入 HDFS 和 Solr。

相比而言，Ranger Audit 在架构上更具扩展性，审计数据不再需要流向一个中心点。

Navigator 中的审计数据将不会被迁移到 Ranger 中，而是变成并行使用的方式。在 CDP 中仍然可以安装 Navigator，升级之前的审计数据仍然在 Navigator 中可用，而升级后生成的审计数据将在 Ranger 中可用。在并行运行一段时间之后，可以根据需要删除 Navigator 组件和 Navigator 后台数据库。



3.2.3. 转换规则

可以被迁移的 Navigator 数据:

- Navigator 业务元数据, 包括: 标签, 自定义属性, 实体名称和描述, Navigator Managed 元数据
- Navigator 技术元数据, 包括: Hive, Impala, Spark, 引用到的 HDFS/S3

不能被迁移的 Navigator 数据:

- Navigator 审计数据, 原来的 Navigator 实例会变成 “只读” 模式, 以便查到历史审计信息
- 实体元数据, 包括:
 - hive/spark/impala 未引用到的 HDFS/S3
 - Sqoop, Pig, Map-Reduce v1&v2, Oozie, YARN
- Navigator Policy
- Navigator configuration settings

3.2.4. 风险提示

□ Navigator API 应用改造

有些应用程序可能会调用 Navigator API, 请按需改造为 Ranger API 或者 Atlas API。

□ Ranger 无法查到历史审计数据

如果要访问升级之前的历史审计记录, 只能在 CDP 上保留一个 “只读” 的 Navigator 实例, 直到不再需要它为止。

□ 必须确保升级后 CDP 上的 Navigator 处于 “只读” 状态

- 1) 删除用户的写权限。如果 Navigator 配置了 LDAP 或 Active Directory 身份验证, 建议修改用户或用户组, 删除其添加或更新元数据的权限。
- 2) 将 Navigator 实体标记为 stale, 标注该实体不再在 Navigator 中维护。有一种比较简单的实现方法是创建一个策略, 该策略将标签分配给 CDP 中的实体。

□ 元数据迁移非常耗时

请登录到 navigator 页面，然后查看 Cluster Group 有多少个 entity。



基于 Navigator entity 数量级，通过下表估算元数据迁移耗时：

Transition Phase	Transition Rate	Disk Space	Output File Size	Trial Data Points
Extraction	4 minutes / 1 million entities	100 MB / 1 million entities, less as volumes increase	65 MB / 1 million entities	10 million entities takes about 30 minutes; 256 million takes about 18 hours.
Transformation	1.5 minutes / 1 million entities	100 to 150 MB / 1 million entities, higher end of range with larger volumes	150 MB / 1 million entities	10 million entities takes about 20 minutes; 256 million takes about 6 hours.
Import	35 minutes / 1 million migrated entities	N/A	N/A	10 million entities takes about 4 hours; 256 million takes about 6 days.

□ Navigator Policy 无法迁移

Navigator Policy 提供了将元数据应用于 Navigator 实体的功能。但在 Atlas 中没有等价功能，因此不会被迁移。您可以通过“Data Catalog”中的 Data Profiler 来实现类似的功能。

您可以使用 Navigator API 查看当前所有的 Navigator Policy：

```
curl -X GET "<navigator host>:<port>/api/v14/policy" -u <username>:<password>
```

□ 迁移模式不支持 Atlas HA

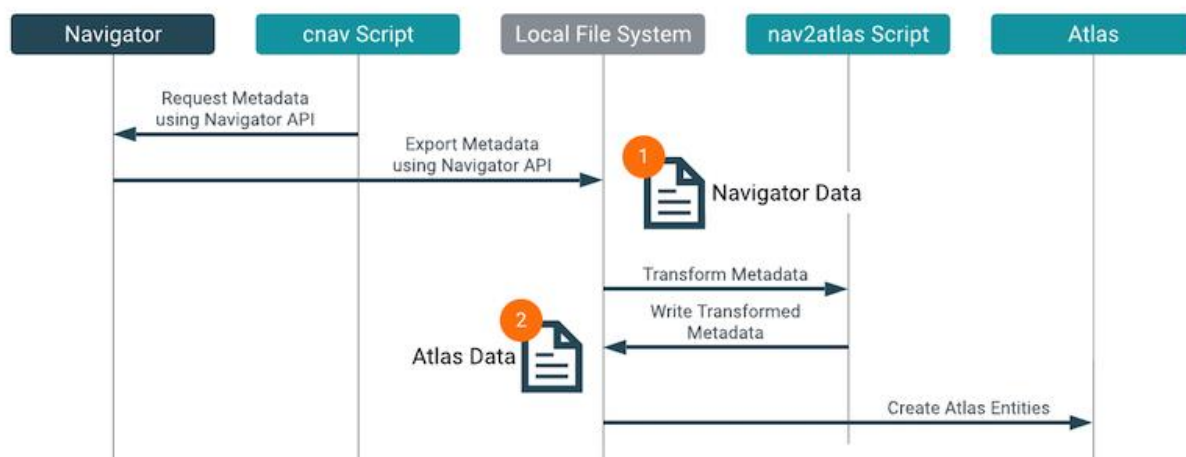
将 Navigator 转换后的元数据导入 Atlas 的时候，Atlas 必须改为“迁移模式”。

注意：Atlas 如果启用了 HA，是无法使用迁移模式的。

□ 需要验证和调整

与任何自动化工具一样，元数据转换结果需要后期验证和调整。因此，自动化迁移过程非常适合进行先期测试：将 Navigator metadata 导入到测试环境的 Atlas 中，提前进行转换和验证，而不会影响生产集群。在升级之前，建议备份 Navigator metadata 数据库。

3.2.5. 迁移步骤



迁移分成三个阶段（参见上图）：

1. 从 Navigator 中提取元数据。

利用脚本（`/opt/cloudera/cm-agent/service/navigator/service/cnav.sh`），该脚本调用 Navigator API，从 Navigator 中提取所有技术和业务元数据。此过程大约每 4 分钟提取 100 万个 Navigator 实体。该脚本会以压缩格式并写入 Linux 本地目录。每 100 万个 Navigator 实体文件大约占用空间 100 MB。

2. 将 Navigator 元数据转换为 Atlas 可以识别的格式。

利用脚本（`/opt/cloudera/parcels/CDH/lib/atlas/tools/nav2atlas/nav2atlas.sh`），转换上个步骤中提取的落地文件，然后再次以压缩格式写入 Linux 本地目录。此过程与提取步骤耗时大致相同。

3. 将转换后的元数据导入 Atlas。

Atlas 将以“迁移模式”运行，在该模式下，Atlas 暂停从 CDP 集群中收集元数据，而是查找第二步转换后的数据文件。Atlas 开始导入数据文件的内容，然后为每个 Navigator 实体创建等效的 Atlas 实体。

3.3. YARN Fair Scheduler 迁移

3.3.1. 概述

Fair Scheduler 最早由 Facebook 创建，目标是保证在用户之间公平共享资源。如果只有一个作业在运行，它将使用整个集群的资源。如果在这个时候，有其他的作业提交进来，资源将进行重新分配，以便使得每个队列里面的作业获得大致相同的 CPU 时间。Fair Scheduler 提出了 Weight（权重）的概念，根据权重来为各个队列分配资源。Weight 从本质上来说和 Capacity Scheduler 里面的 Capacity 是一样的。

Capacity Scheduler 最早由 Yahoo! 创建，目标是根据组织所需的 Capacity 来分配资源，为每个组织提供最小的容量保证。如果新提交了一个作业，但发现资源不够用了，就会根据 FIFO 规则排队，一直等到可用的资源为止。组织可以访问任何多余的容量而不会被他人使用，基于成本控制为组织提供弹性资源。

为什么 CDP 选择使用 Capacity Scheduler，替换了原有的 Fair Scheduler？

- Capacity Scheduler 易与 Ranger 集成，这意味着可以通过 Ranger 策略来控制哪些用户可以/不能提交作业到指定队列。
- Capacity Scheduler 支持节点分区和节点标签。
- Capacity Scheduler 更适合于云原生环境，例如：提供更好的封装服务，支持自动扩展等等。
- 性能优势：Capacity Scheduler 采用基于全局调度的技术框架，能够一次查找多个节点。

以下是 Capacity Scheduler 和 Fair Scheduler 的功能差异：

特性大类	特性小类	Capacity Scheduler	Fair Scheduler
放置规则	Node Labels	支持	不支持
放置规则	Node Attributes	支持	不支持
放置规则	Placement constraints	支持	不支持
放置规则	Queue Priority	支持	不支持
抢占规则	Queue Priority based preemption	支持	不支持

以下是旧版和新版放置规则之间最实质性的区别：

- 规则使用 JSON 中描述，但是，这对 CDP 中的用户是透明的。生成的 JSON 可以作为容量调度程序配置高级配置片段（安全阀）设置的一部分在 Cloudera Manager 中查看。
- 您可以配置当规则无法将应用程序放置到目标队列时应该发生的情况。有三个选项：继续下一个规则、拒绝提交、将应用程序放入默认队列。
- 新策略（映射操作）可供选择：specified, defaultQueue 和 reject。
- 将引入 create 标志：如果启用了不存在的队列只有动态创建。

3.3.2. 迁移方法

使用 YARN fs2cs 工具将 fair-scheduler.xml 转换为 capacity-scheduler.xml。

3.3.3. 转换规则

capacity-scheduler.xml 会从 fair-scheduler.xml 继承以下属性：

表 1. 队列资源配额相关特性

属性	描述
预先创建的分层队列。	转换后实现相同的队列层次结构。
<weight>	权重：队列的稳定公平份额。 该 queue.capacity 属性将设置为相同的比率。
<maxAMShare>	最大 AM 份额：限制可用于运行应用程序主机的队列公平份额的比例

属性	描述
<maxRunningApps>	最大正在运行的应用程序：限制队列中一次运行的应用程序数量
<maxContainerAllocation>	最大容器分配：队列可以为单个容器分配的最大资源量。
<schedulingPolicy>	队列的调度策略（例如，提交的应用程序如何随时间排序）。 它的转换有一些限制。有关更多信息，请参阅 <i>Fair Scheduler 功能和转换详细信息</i> 。
<aclSubmitApps> <aclAdministerApps>	ACL 设置 ：可以将应用程序提交到队列或可以管理队列的用户和/或组的列表。
maximum-applications	指定队列中任一时刻的并发活动应用程序的最大数量。
maximum-am-resource-percent	指定集群中可用于为队列运行应用程序主控的资源的最大百分比。
acl_submit_applications	指定控制谁可以向给定队列提交应用程序的 ACL 。
acl_administer_queue	指定控制谁可以管理给定队列中的应用程序的 ACL 。
ordering-policy	将队列排序策略为特定队列指定为 FIFO 或公平。

表 2.全局调度设置

属性	描述
yarn.scheduler.fair.allow-undeclared-pools	允许未声明的池。 设置是否可以在应用程序提交时创建新队列。
yarn.scheduler.fair.sizebasedweight	基于大小的权重。 是否根据单个应用程序的大小为它们分配份额，而不是为所有应用程序提供相等的份额，而不考虑大小。
<queueMaxAppsDefault>	队列最大应用程序默认值：设置所有队列的默认运行应用程序限制。
<queueMaxAMShareDefault>	默认最大 AM 共享：设置队列的默认 AM 资源限制。
yarn.scheduler.fair.locality.threshold.node	位置阈值节点：对于在特定节点上请求容器的应用程序，自上次容器分配以来在接受另一个节点上的放置之前要等待的调度机会数。
yarn.scheduler.fair.locality.threshold.rack	位置阈值机架：对于在特定机架上请求容器的应用程序，自上次容器分配以来在接受另一个机架上的

属性	描述
	放置之前等待的调度机会数。
yarn.scheduler.fair.max.assign	最大分配数：如果assignmultiple为true且dynamic.max.assign为false，则一次心跳可以分配的最大容器数。
yarn.scheduler.fair.assignmultiple	分配多个：是否允许在一次心跳中分配多个容器。
yarn.resourcemanager.scheduler.monitor.enable	允许较高优先级的应用程序抢占较低优先级的应用程序。
yarn.scheduler.capacity.maximum-am-resource-percent	指定集群中可用于运行应用程序主控的资源的最大百分比。
<userMaxAppsDefault>	默认最大运行应用程序。
<user name="..."> <maxRunningApps>...</maxRunningApps></user>	每个用户的最大运行应用程序。
yarn.scheduler.fair.user-as-default-queue	是否使用与分配关联的用户名作为默认队列名称。 权重模式： 这种行为是用一个放置规则模拟的（实际上，即使在Fair Scheduler中，这也被内部转化为放置规则）： <pre> { "type": "user", "matches": "*", "parentQueue": "root", "policy": "user", "create": true, "fallbackResult": "skip" } </pre> 有关百分比模式的信息，请参阅 Fair Scheduler 功能广告转换详细信息 。

表 3. 抢占

属性	描述
yarn.scheduler.fair.preemption	公平调度程序抢占已打开。 转换容量调度器抢占后默认开启使用默认值。
<allowPreemptionFrom>	按队列抢占已禁用。

属性	描述
	转换后相同队列抢占默认禁用。
<code>yarn.scheduler.fair.waitTimeBeforeKill</code>	在杀死容器之前的等待时间
<code>disable_preemption</code>	禁用提交到给定队列的应用程序容器的抢占。

表 4. 放置规则

公平调度程序放置规则	描述	转换详情
<code>create="false"</code> 或者 <code>"true"</code>	禁用或启用在 YARN 中动态创建队列。无法在以下放置规则策略中指定此选项： <code>reject</code> <code>setDefaultQueue</code> <code>defaultQueue</code>	权重模式： 完全支持此标志，除了嵌套规则，您只能在其中定义单个“创建”标志。因此，不能设置“真/假”和“假/真”。 相对模式： 部分支持。必须选择受管父队列作为父队列。该标志对常规父队列没有影响。
<code><rule name="specified"/></code>	如果用户通过指定队列名称（而不是“默认”队列）提交了申请，则此规则将成功。因此，不会执行剩余的规则集。	支持权重和百分比模式。
<code><rule name="primaryGroup"/></code>	如果提交的用户（ <code>userA</code> ）的主组名（ <code>groupA</code> ）存在，则提交到 <code>groupA</code> 。	匹配策略称为 <code>primaryGroup</code> 。
<code><rule name="secondaryGroupExistingQueue"/></code>	如果提交的用户（ <code>userA</code> ）的次要组名（ <code>groupB</code> ）存在，则提交到 <code>groupB</code> 。	匹配策略称为 <code>secondaryGroup</code> 。
<code><rule name="nestedUserQueue"></code>	根据嵌套规则，这会将作业放置到以下队列： <code>root.[primaryGroup].[userName]</code> <code>root.[secondaryGroup].[userName]</code> <code>root.[queuePath].[userName]</code>	由容量调度程序支持。三种可能的策略是（取决于外部规则）： <code>primaryGroupUser</code> <code>secondaryGroupUser</code> <code>userparentQueue</code> 明确地有一个集合。
<code><rule name="default"></code>	将应用程序放入名为“ <code>root.default</code> ”的默认队列或由	默认规则有一个名为 的匹配策略

公平调度程序放置规则	描述	转换详情
queue="qName" / >	"queue"属性表示的特定于用户的队列中。	<p>defaultQueue。</p> <p>如果"root.default"不是预期的默认队列，则有两种可能的方法：</p> <p>使用setDefaultQueue 策略更改"root.default"，然后应用defaultQueue。</p> <p>使用自定义策略并将策略字符串设置为目标队列。</p>

3.3.4. 调度程序迁移的限制

将 Fair Scheduler 配置转换为 Capacity Scheduler 配置有一些硬性限制，因为这两个调度程序并不等效。了解这些主要限制可以帮助您了解调度程序迁移后可能遇到的困难。

容量调度器的特性和配置与公平调度器的特性和配置不同，导致调度器迁移的限制。这些限制有时可以通过手动配置、微调或一些试错来克服，但在许多情况下没有解决方法。

笔记

这不是一个完整的列表。它仅包含最常导致问题的调度程序迁移限制。

3.3.4.1. 不能在同一级别上创建静态和动态叶队列

如果您在 capacity-scheduler.xml 文件中定义了一个父队列，其中至少有一个叶队列，则不可能在此特定父级下动态创建新叶。

已解决：Cloudera Runtime 7.1.6 支持权重模式和动态自动子创建。在权重模式下，可以在同一级别上存在静态和动态叶队列。基于阿里云部署的 CDP 版本已解决该问题。

3.3.4.2. 放置规则和映射规则不同

放置规则（在 Fair Scheduler 中使用）和映射规则（在 Capacity Scheduler 中使用）非常不同，因此无法自动转换。完成从 CDH 到 CDP 的升级后，您必须手动配置放置规则和映射规则。这有多种原因。以下是最显著的差异：

- 在 Fair Scheduler 中，您可以使用特殊的放置规则，例如“默认”或“指定”，而 Capability Scheduler 中完全没有这些规则。
- 在 Fair Scheduler 中，您可以为每个规则设置一个“创建”标志。映射规则不支持这一点。
- 在 Fair Scheduler 中，在嵌套规则的情况下，两个规则都解释了“创建”标志。这在容量调度程序中并非如此。
- 如果规则可以在 Fair Scheduler 中返回有效队列，则继续执行下一个规则。另一方面，Capacity Scheduler 返回“root.default”。

已解决：在 Cloudera Runtime 7.1.6 中引入了新的基于 JSON 的放置规则格式和新的放置规则评估引擎。它们解决了许多以前的放置规则迁移限制。基于阿里云部署的 CDP 版本已解决该问题。

3.3.4.3. 动态队列的容量值是固定的

在 Fair Scheduler 中，每次创建新队列时都会重新计算公平份额。相比之下，Capacity Scheduler 为动态创建的队列分配一个预定义的百分比值。

这个预定义的百分比可以更改，但在重新配置调度程序之前它是固定的。一旦该值达到 100，将创建下一个值为 0 的动态队列。例如，如果该值设置为 25.00，则同一父级下的第五个队列的容量将为 0。

以下是如何将 Fair Scheduler 队列权重转换为 Capacity Scheduler 队列容量（相对于其父级的百分比）的示例：

表 1. 权重转换示例

队列路径	权重	容量调度器等效（容量） yarn.scheduler.capacity.<queue-path>.capacity
根	1	100%
root.default	10	25%
根用户	30	75%
root.users.alice	1	33.333%
根用户.bob	1	33.333%
root.users.charlie	1	33.334%

在 Cloudera Runtime 7.1.5 和更低版本中，**fs2cs** 转换实用程序确保一个父队列下的所有直接子级的百分比加起来正好为 100.000%，如表中所示。例如，*root.users* 下的所有队列：
 $root.users.alice + root.users.bob + root.users.charlie = 100.000\%$ 。

权重通过以下方式转换为基于百分比的容量：在队列级别的根上，有 2 个队列：default 和 users。因为它被指定为 10 + 30 个权重（总共 40 个），所以 1 个“权重单位”是 2.5%。这就是 root.default 有 25% 和 root.users 有 75% 容量的原因。此计算可应用于所有队列级别。

已解决：Cloudera Runtime 7.1.6 支持权重模式和动态自动子创建。并且 **fs2cs** 转换实用程序默认转换为权重模式。基于阿里云部署的 CDP 版本已解决该问题。

3.3.4.4. 风险提示

□ 需要验证和调整

YARN Scheduler 是平台核心功能，用来确保租户获得适当级别的 YARN 资源。如果在升级之后 YARN Scheduler 规则出现不一致，那么租户在资源抢占中可能会表现异常，从而导致某些作业运行更长的时间。

建议在升级之前进行适当的压力测试和先期验证。如果条件允许，建议以生产规模为标准进行测试。此外，请密切监控和调整升级后的资源使用情况。

□ 部分属性无法转换

例如：FairScheduler 可以使用 minResource 和 maxResource 选项。但是，由于集成管理器 UI 不支持绝对资源，fs2cs 工具在转换过程中会忽略原有的 maxResources 配置。这意味着，在自动转换之后，Capacity Scheduler 队列都会具有 100% 的 maxResources。在升级过程完成之后，管理员需要手工调整此配置。

以下是无法转换的属性列表：

备注 1：是否有等价 CS 属性

备注 2：如何进行手工转换

备注 3：是否可以通过 Queue Manager UI 调整

fs2cs 转换实用程序无法自动转换某些 Fair Scheduler 属性。查看这些属性的列表以及容量调度程序和队列管理器 UI 是否支持它们，以了解如何配置它们。

表 1. 队列资源配额相关特性

属性	描述	转换信息
<minResources>	队列有权获得的最少资源。	容量调度程序部分支持。 被 fs2cs 转换实用程序忽略。 队列管理器 UI 不支持。
<maxResources>	将分配给队列的最大资源量。	容量调度程序中有一个等效的功能。 被 fs2cs 转换实用程序忽略。对于每个队列，最大容量将设置为 100%。 由队列管理器 UI 支持。
<maxChildResources>	可以分配给临时子队列的最大资源量。	容量调度程序中有一个等效的功能。 被 fs2cs 转换实用程序忽略。它的值可以是两个不同的百分比（vcore/memory）或绝对资源，但叶子队列模板只接受一个百分比。 由队列管理器 UI 支持。

表 2. 全局调度设置

属性	描述	转换信息
yarn.scheduler.fair.max.assign	动态最大分配	容量调度程序中有一个等效的功能。 需要对以下三个属性进行微调： <ul style="list-style-type: none"> yarn.scheduler.capacity.per-node-heartbeat.multiple-assignments-enable

属性	描述	转换信息
		<ul style="list-style-type: none"> <code>yarn.scheduler.capacity.per-node-heartbeat.maximum-container-assignments</code> <code>yarn.scheduler.capacity.per-node-heartbeat.maximum-offswitch-assignments</code> <p>队列管理器 UI 不支持。</p>
<code>yarn.scheduler.fair.user-as-default-queue</code>	用户作为默认队列	<p>相对模式：需要创建放置规则。</p> <ol style="list-style-type: none"> 1. 创建一个队列，例如“<code>root.users</code>”并为其启用动态自动子创建（使其成为托管父队列）。 2. 创建以下放置规则： <pre> { "type": "user", "matches": "*", "parentQueue": "root.users", "policy": "user", "create": true, "fallbackResult": "skip" } </pre> <p>以下限制适用：</p> <ul style="list-style-type: none"> • 动态创建的队列不可能有“<code>root</code>”作为父级。 • “<code>root.users</code>”队列不能有静态叶子。这些是始终存在并手动创建的队列。 <p>有关权重模式的信息，请参阅 <i>Auto-converted Fair Scheduler 属性</i>。</p> <p>由队列管理器 UI 支持。</p>

表 3. 抢占

属性	描述	转换信息
<code>yarn.scheduler.fair.preemption.cluster-utilization-threshold</code>	抢占开始后的利用率阈值。	容量调度程序中有一个等效的功能： <code>yarn.resourcemanager.monitor.capacity.preemption.max_ignored_over_capacity</code> 。它指定超过其配置容量的资源使用阈值，队列在有资格被抢占之前必须满足该阈值。 由队列管理器 UI 支持。
<code>minSharePreemptionTimeout</code>	队列在尝试抢占容器以从其他 <code>queue.s</code> 获取资源之前低于其最小份额的秒数	容量调度程序不支持。
<code>fairSharePreemptionTimeout</code>	队列在尝试抢占容器以从其他队列获取资源之前低于其公平共享阈值的秒数。	容量调度程序部分支持。 这可以通过结合使用以下配置来实现： <ul style="list-style-type: none"> <code>yarn.resourcemanager.monitor.capacity.preemption.natural_termination_factor</code> <code>yarn.resourcemanager.monitor.capacity.preemption.max_wait_before_kill</code> 由队列管理器 UI 支持。
<code>fairSharePreemptionThreshold</code>	队列的公平份额抢占阈值。	容量调度程序部分支持。 这可以通过结合使用以下配置来实现： <ul style="list-style-type: none"> <code>yarn.resourcemanager.monitor.capacity.preemption.natural_termination_factor</code> <code>yarn.resourcemanager.monitor.capacity.preemption.max_wait_before_kill</code> 由队列管理器 UI 支持。

表 4. 放置规则

公平调度程序放置规则	描述	转换信息
<code>create="false" or "true"</code>	禁用或启用在 YARN 中动态创建队列。可以在所有规则上指定此选项。	容量调度程序部分支持。 使用容量调度程序动态队列映射策略：

公平调度程序放置规则	描述	转换信息
		<ul style="list-style-type: none"> • <code>u:%user:[managedParentQueueName].[queueName]</code> • <code>u:%user:[managedParentQueueName].%user</code> • <code>u:%user:[managedParentQueueName].%primary_group</code> • <code>u:%user:[managedParentQueueName].%secondary_group</code> <p>由队列管理器 UI 支持。</p>
<code><rule name="specified"/></code>	如果用户通过指定队列名称（而不是“默认”队列）提交了申请，则此规则将成功。因此，不会执行剩余的规则集。	容量调度程序不支持。
<code><rule name="primaryGroupExistingQueue"/></code>	如果提交的用户（ userA ）的主组名（ groupA ）存在，则提交到 groupA 。	Capacity Scheduler 中有一个等效的放置规则： <code><value>u:%user:%primary_group</value></code>
		由队列管理器 UI 支持。
<code><rule name="secondaryGroupExistingQueue"/></code>	如果提交的用户（ userA ）的次要组名（ groupA ）存在，则提交到 groupA 。	Capacity Scheduler 中有一个等效的放置规则： <code><value>u:%user:%secondary_group</value></code>
		由队列管理器 UI 支持。
<code><rule name="nestedUserQueue"></code>	根据嵌套规则，这会将作业放置到以下队列：	容量调度程序不支持。
	<ul style="list-style-type: none"> • <code>root.[primaryGroup].[userName]</code> • <code>root.[secondaryGroup].[userName]</code> • <code>root.[queuePath].[userName]</code> 	
<code><rule name="default" queue="qName"/></code>	回退策略，如果没有指定队列属性（如果所有匹配失败），则规则将回退到属性“队列”中指定的队列或“默认”队列。	Capacity Scheduler 中有一个等效的放置规则： <code><value>u:%user:default</value></code>
		由队列管理器 UI 支持。

3.3.5. 迁移步骤

您可以使用 `fs2cs` 转换实用程序自动将某些 Fair Scheduler 配置转换为 Capacity Scheduler 配置。

从 CDP Private Cloud Base 7.1 版本开始, Cloudera 提供了一个转换工具, 称为 **fs2cs** 转换实用程序。此实用程序是一个 CLI 应用程序, 它是 `yarn CLI` 命令的一部分。它生成 `capacity-scheduler.xml` 和 `yarn-site.xml`, 并作为输出文件。

重要

Capacity Scheduler 的功能与 Fair Scheduler 的功能并不完全相同。因此, **fs2cs** 转换实用程序无法将每个 Fair Scheduler 配置转换为容量调度程序配置。在自动转换和迁移完成后, 您必须手动调整调度程序配置, 以确保生成的调度配置在转换后符合您组织的内部目标和 SLA。

- 请注意自动转换的公平调度程序属性、需要手动配置的属性以及容量调度程序中没有等效功能的属性。
 - 您必须已下载并分发了 CDP 目标版本的包。
 - 在 VPC 中, 要在升级后在新安装中使用当前的 Compute Cluster 队列配置, 您必须在开始更新过程之前手动保存它们, 然后将配置添加到新安装中。否则, 您的计算集群队列配置将丢失, 因为升级向导仅转换来自您的基本集群的队列。
 1. 在 Cloudera Manager 中, 导航到 Host > All Hosts。
 2. 找到具有 ResourceManager 角色的主机, 然后单击 YARN ResourceManager 角色。
 3. 单击进程选项卡。
 4. 查找和保存 `fair-scheduler.xml` 和 `yarn-site.xml` 配置文件以供将来参考。
1. 从 Cloudera Manager 数据存储下载 Fair Scheduler 配置文件
 - a. 在 Cloudera Manager 中, 导航到 Host > All Hosts。
 - b. 找到具有 ResourceManager 角色的主机, 然后单击 YARN ResourceManager 角色。

The screenshot shows the Cloudera Manager interface for a YARN cluster. The 'Instances' tab is active, displaying a table of roles and their configurations. The 'ResourceManager (Active)' role is highlighted with a red box.

Role Type	State	Host	Commission State	Role Group
JobHistory Server	Started	master	Commissioned	JobHistory Server Default Group
NodeManager	Started	slave1	Commissioned	NodeManager Default Group
NodeManager	Started	slave2	Commissioned	NodeManager Default Group
NodeManager	Started	slave3	Commissioned	NodeManager Default Group
ResourceManager (Active)	Started	master	Commissioned	ResourceManager Default Group

- c. 单击进程选项卡。
- d. 查找和保存 `fair-scheduler.xml` 和 `yarn-site.xml` 配置文件以供将来参考。

The screenshot shows the Cloudera Manager interface for the Resource Manager. The 'Processes' tab is active, and the 'yarn/yarn.sh [ResourceManager]' process is selected. The configuration files and environment variables are listed below the process details.

Program	User/Group	Links	Configuration Files/Environment	Process Status	PID	Uptime
yarn/yarn.sh [ResourceManager]	yarn/hadoop	ResourceManager Web UI	Hide	Running	17867	5.8h

Configuration Files:

- capacity-scheduler.xml
- core-site.xml
- fair-scheduler.xml
- hdfs-site.xml
- mapred-site.xml
- ssl-client.xml
- ssl-server.xml
- yarn-site.xml
- cloudera-monitor.properties
- topology.map
- topology.py

Environment Variables:

- yarn.keytab
- YARN_LOG_DIR
- YARN_ROOT_LOGGER
- YARN_LOGFILE
- YARN_RESOURCEMANAGER_OPTS
- YARN_RESOURCEMANAGER_AGENT_COMMON_DIR
- CDH_VERSION

2. 使用 *fs2cs* 转换实用程序

- 使用 *ssh* 登录到您下载 *fair-scheduler.xml* 和 *yarn-site.xml* 文件的主机。
- 创建一个新目录来保存 *fs2cs* 转换实用程序 *capacity-scheduler.xml* 生成的文件：

```
$ mkdir -p output
```

- 使用 *fs2cs* 转换实用程序自动转换资源池的结构。大括号之间列出的选项 [] 是可选的：

```
$ yarn fs2cs [--cluster-resource <vcores/memory>][--no-terminal-rule-check] --
yarnsiteconfig </path/to/yarn-site.xml> [--fsconfig </path/to/fair-
scheduler.xml>] --output-directory </output/path/> [--print] [--skip-
verification]
```

笔记

您必须提供 *fair-scheduler.xml* 配置文件的绝对路径。

3. 在 Cloudera Manager 中提供生成的容量调度程序配置。

- 在 Cloudera Manager 中，选择 YARN 服务。
- 单击配置选项卡。
- 搜索 *capacity-scheduler* 并找到 Capacity Scheduler Configuration Advanced Configuration Snippet (Safety Valves)
- 单击以 XML 格式查看并插入由转换器工具生成的 *capacity-scheduler.xml* 文件的完整内容。
- 单击保存更改。
- 搜索 *yarn-site* 并找到 YARN Service Advanced Configuration Snippet (Safety Valve) for *yarn-site.xml*
- 单击以 XML 格式查看并插入 *yarn-site.xml* 由转换工具生成的文件的完整内容。
- 手动删除以下无效标签：

- 标题
 - 配置标签
 - 最终标签
 - 源标签
- i. 单击保存更改。
4. 重新启动 YARN 和队列管理器服务。

继续迁移到 CDP。

迁移完成后，使用队列管理器 UI 和 Cloudera Manager Advanced 配置片段（安全阀）手动调整 **fs2cs** 转换实用程序生成的配置。

3.3.5.1. fs2cs 转换工具的 CLI 选项

fs2cs 转换工具的 CLI 选项列表。

选项	描述
<code>-d, --dry-run</code>	执行转换的试运行。输出是否可以转换。
<code>-f, --fsconfig <arg></code>	有效 <code>fair-scheduler.xml</code> 配置文件的绝对路径。 默认情况下， <code>yarn-site.xml</code> 包含定义路径的属性 <code>fair-scheduler.xml</code> 。因此，这些 <code>-f / --fsconfig</code> 设置是可选的。
<code>-h, --help</code>	显示选项列表。
<code>-o, --output-directory <arg></code>	<code>yarn-site.xml</code> 和 <code>capacity-scheduler.xml</code> 文件的输出目录。必须对运行此脚本的用户具有写入权限。 如果指定 <code>-p</code> 或 <code>--print</code> ，则将 <code>xml</code> 文件发送到标准输出，因此 <code>-o / --output-directory</code> 忽略。
<code>-p, --print</code>	如果定义，转换后的配置将只发送到控制台。 如果指定了 <code>-p</code> 或 <code>--print</code> ，则将 <code>xml</code> 文件发送到标准输出，因此忽略 <code>-o / --output-directory</code> 。
<code>-pc, -percentage</code>	CDP Private Cloud Base 7.1.6 及更高版本支持此选项。 默认情况下， fs2cs 转换实用程序会转换为权重模式。使用 <code>-pc</code> 您可以将其更改为相对（百分比）模式。请注意，有一些调度程序转换限制在权重模式下解决，但在相对（百分比）模式下没有解决。 相对模式可以被认为是容量调度程序的“传统”模式，其中容量以百分比表示。

选项	描述
<code>-r, --rulesconfig <arg></code>	可选参数。如果指定，则应指向转换规则文件（属性格式）的有效路径。
<code>-s, --skip-verification</code>	它不会验证转换后的容量调度程序配置。默认情况下，该实用程序会启动一个内部容量调度程序实例，以查看它是否可以正常启动。此开关禁用此行为。
<code>-t, --no-terminal-rule-check</code>	禁用检查放置规则是否是终端以保持与 YARN-8967 之前制作的配置的向后兼容性。 默认情况下，Fair Scheduler 会严格检查放置规则是否为终止规则。这意味着如果您使用 <code><reject></code> 规则后跟一个 <code><specified></code> 规则，那么这是不允许的，因为后者是不可访问的。然而，在 YARN-8967 之前，Fair Scheduler 更为宽松，并允许某些不再有效的规则序列。在该工具内部，实例化了一个 Fair Scheduler 实例以读取和解析分配文件。为了让 Fair Scheduler 接受这样的配置，必须提供 <code>-tor --no-terminal-rule-check</code> 参数以避免 Fair Scheduler 实例抛出异常。
<code>-y, --yarnsiteconfig <arg></code>	有效 <code>yarn-site.xml</code> 配置文件的路径。

3.3.6. 手动配置调度程序属性

迁移到 CDP 后，您必须使用 YARN 队列管理器 UI 手动微调调度程序配置，以确保生成的配置符合您的要求。您可以使用 Cloudera Manager Advanced 配置片段（安全阀）来配置 YARN 队列管理器 UI 中缺少的属性。

Capacity Scheduler 的功能与 Fair Scheduler 的功能并不完全相同。因此，转换实用程序无法将每个公平调度程序配置转换为容量调度程序配置。因此，您必须手动调整调度程序配置，以确保生成的调度配置在转换后符合您组织的内部目标和 SLA。如果需要，请进一步更改 **fs2cs** 转换实用程序生成的输出文件 `capacity-scheduler.xml` 和 `yarn-site.xml` 中的调度程序属性。有关由 **fs2cs** 转换实用程序自动转换的 Fair Scheduler 属性的信息，请参阅 *自动转换的 Fair Scheduler 属性*。

您可以使用 YARN 队列管理器 UI 手动配置属性，如果您看到队列管理器 UI 中不可用的属性，您可以使用 Cloudera Manager 配置（安全阀）来配置它们。

重要的

您不得同时使用队列管理器 UI 和 Cloudera Manager 安全阀，因为安全阀会使用队列管理器 UI 覆盖配置集。

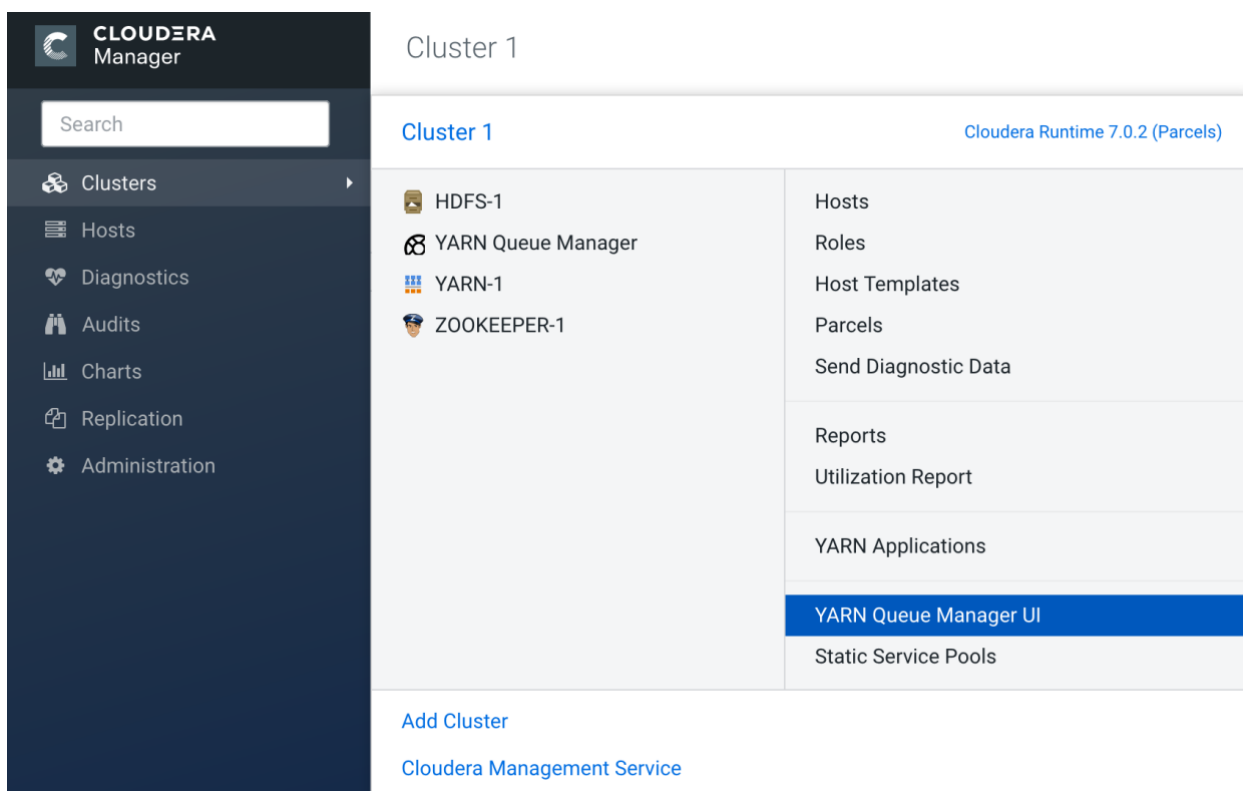
3.3.6.1. 使用 YARN 队列管理器 UI 配置调度程序属性

迁移到 CDP 后，您必须使用 *fs2cs* 转换实用程序生成的输出文件配置容量调度程序属性。您可以使用 YARN 队列管理器 UI 服务手动配置属性。

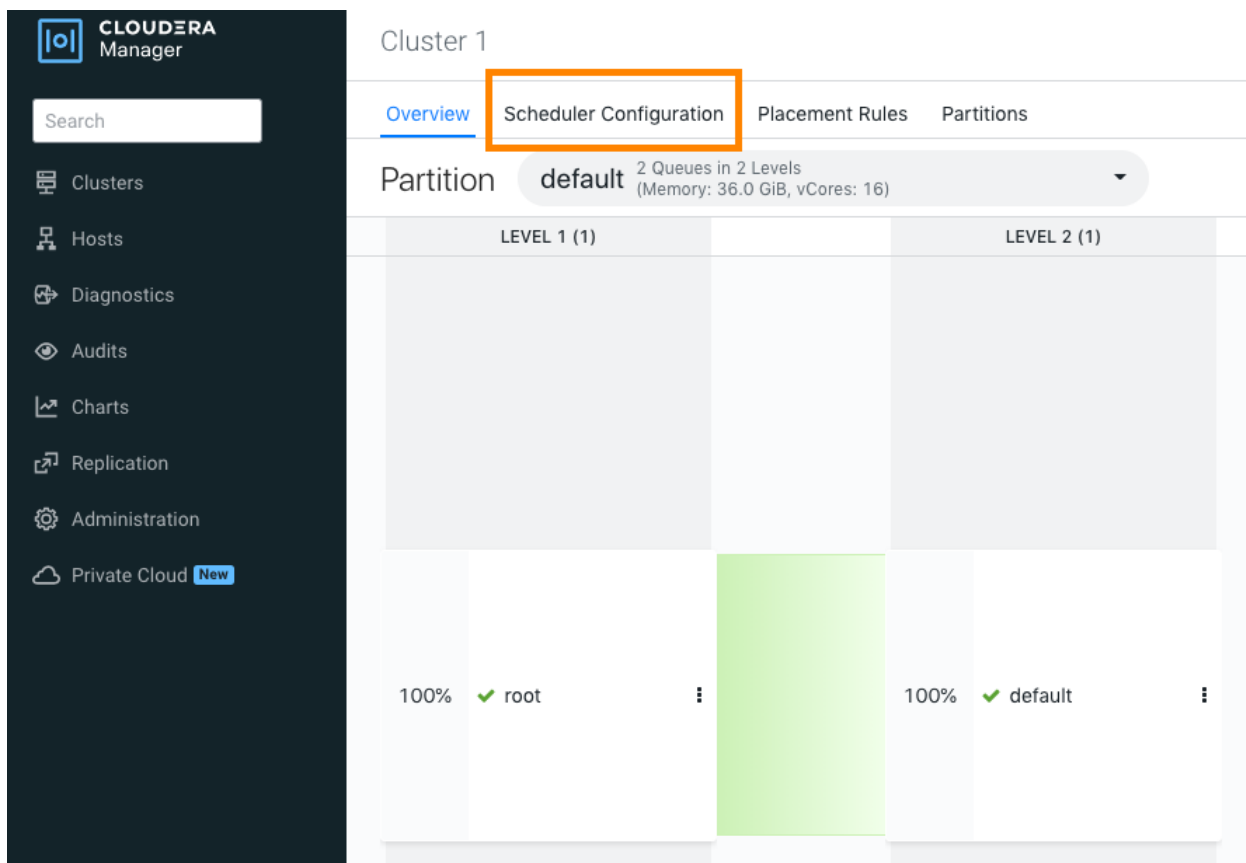
- 使用 *fs2cs* 转换实用程序生成 输出文件 `capacity-scheduler.xml` 和 `yarn-site.xml`。
- 完成迁移过程。
- 确定需要手动配置并可使用队列管理器 UI 配置的属性。

有关调度程序属性的更多信息，请参阅 [公平调度程序功能和转换详细信息](#)。

1. 在 Cloudera Manager 中，单击 Clusters 并选择 YARN Queue Manager UI 服务。



2. 在 YARN 队列管理器窗口中，单击调度程序配置选项卡。



3. 在 Scheduler Configuration 窗口中，输入属性的值并单击 Save。

3.3.7. 使用 Cloudera Manager 安全阀配置调度程序属性

某些调度程序属性既不能由 *fs2cs* 转换实用程序转换，也不能使用 YARN 队列管理器 UI 服务进行配置。迁移到 CDP 后，您必须使用 Cloudera Manager 高级配置片段（安全阀）手动配置这些属性。

- 使用 *fs2cs* 转换实用程序生成 输出文件 `capacity-scheduler.xml` 和 `yarn-site.xml`。
- 完成迁移过程。
- 确定需要手动配置且队列管理器 UI 不支持的调度程序属性。

1. 在 Cloudera Manager 中，选择 YARN 服务。
2. 单击配置选项卡。
3. 搜索 `capacity-scheduler` 并找到 容量调度程序配置高级配置片段（安全阀）。
4. 单击 View as XML，并插入 `capacity-scheduler.xml` 由转换器工具生成的完整文件。
5. 添加必要的配置属性。
6. 单击保存更改。
7. 搜索 `yarn-site` 并找到 `yarn-site.xml` 的 YARN 服务高级配置片段（安全阀）。
8. 单击以 XML 格式查看并以 XML 格式添加所需的配置。

（可选）使用 + 和 - 来添加和删除属性。




9. 单击保存更改。
10. 重新启动 YARN 服务。

原文链接：<https://docs.cloudera.com/cdp-private-cloud-upgrade/latest/data-migration/topics/cdp-data-migration-yarn-scheduler-migration.html>

4. 附录：代码改造指南

4.1.Hive 代码改造

参考如下：








Hive代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
1	Create table by hive	<ul style="list-style-type: none"> 生成non-ACID表 	<ul style="list-style-type: none"> 默认生成ORC格式的ACID表 	<ul style="list-style-type: none"> CDH的hive内部表需要改成external+purge=true CDP Spark访问ACID表只能通过Hive Warehouse Connector或者HWC Spark Direct Reader
2	Default Hive Location	<ul style="list-style-type: none"> /user/hive/warehouse/default.db/test_table 	<ul style="list-style-type: none"> /warehouse/tablespace/external/hive/default.db/test_table 	<ul style="list-style-type: none"> 建议设置hive.warehouse.external.dir 为 /user/hive/warehouse/，保持原C5行为
3	CREATE TABLE 指定内部表 LOCATION	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> 不要使用LOCATION子句创建内部表
4	CREATE INDEX	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> 删除该语句，Hive会自动在主表中的ORC或Parquet中构建和存储索引
5	Drop external table	<ul style="list-style-type: none"> HDFS上的文件不会被删除 	<ul style="list-style-type: none"> 取决于purge属性，默认purge=true会删除 	<ul style="list-style-type: none"> CDH的Hive外部表需要改成external+purge=false
6	Hive on Spark/MR	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 不支持，只支持Hive on Tez 	<ul style="list-style-type: none"> 需要手工剔除Hive on Spark/MR设置 Hive on Tez不支持Hive Index

CLUSTERA
© 2020 Cloudera, Inc. All rights reserved. 64

Hive代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
7	Hive CLI	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> 改用Beeline
8	cast函数： cast as timestamp	<ul style="list-style-type: none"> 得到本地时区时间 	<ul style="list-style-type: none"> 得到UTC时区时间 	<ul style="list-style-type: none"> 改用from_utc_timestamp和to_utc_timestamp函数
9	`db.table`	<ul style="list-style-type: none"> 支持`db.table` 	<ul style="list-style-type: none"> 不支持`db.table`，需要改成`db`.`table` 	<ul style="list-style-type: none"> 需要手工调整
10	alter table修改字段类型	<ul style="list-style-type: none"> 不兼容的字段类型也支持修改。string, char和varchar属于一个类型组中，int, bigint和decimal属于一个类型组中，支持将string类型改为int类型 	<ul style="list-style-type: none"> 只支持同类型组的类型转换。例如：不支持将string类型改为int类型 	<ul style="list-style-type: none"> 禁止修改不兼容的字段类型
11	保留关键字	<ul style="list-style-type: none"> N/A 	<ul style="list-style-type: none"> 额外增加几个关键字：TIME, NUMERIC, APPLICATION, SYNC 	<ul style="list-style-type: none"> 通过反引号，例如`application`来和C5保持一致
12	插入分区表时，对分区值进行类型检查	<ul style="list-style-type: none"> 不检查 	<ul style="list-style-type: none"> 检查，默认hive.typecheck.on.insert=true; 	<ul style="list-style-type: none"> 如果要保持原C5行为，手工set hive.typecheck.on.insert=false;

CLUSTERA
© 2020 Cloudera, Inc. All rights reserved. 65



Hive代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
13	NO_DROP不支持CASCADE	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> NO_DROP代码中删除CASCADE语句 改为Ranger管理,防止类似场景的drop table
14	最大值/最小值	<ul style="list-style-type: none"> 最大值函数返回值列表中的最大值,最小值函数返回值列表中的最小值,会自动过滤NULL值 	<ul style="list-style-type: none"> 当一个或多个参数为NULL时,会返回NULL 	<ul style="list-style-type: none"> 对最大或最小函数的参数列使用NULL过滤器或nvl函数。
15	重命名Hive内部表	<ul style="list-style-type: none"> 将移动其HDFS位置 	<ul style="list-style-type: none"> 仅在表在创建时不带LOCATION子句且位于其数据库目录下时才移动其位置 	<ul style="list-style-type: none"> 无需改造,了解变化即可
16	Load语法	<ul style="list-style-type: none"> Hive 3.0之前的加载操作是纯复制/移动操作,可将数据文件移动到与Hive表相对应的位置 	<ul style="list-style-type: none"> Hive在内部将加载重写为INSERT AS SELECT 	<ul style="list-style-type: none"> 注意加载的目标表是否为分区表,如果数据不符合分区表模式,可能会报错,根据情况修改Load语句



Hive代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
17	Hive.auto.convert.sortmerge.join	<ul style="list-style-type: none"> 默认为false 	<ul style="list-style-type: none"> 默认为true 	<ul style="list-style-type: none"> 注意默认值变化,酌情修改
18	Hive.cbo.enable	<ul style="list-style-type: none"> 默认为false 	<ul style="list-style-type: none"> 默认为true 	<ul style="list-style-type: none"> 注意默认值变化,酌情修改
19	Hive.exec.dynamic.partition.mode	<ul style="list-style-type: none"> 默认为strict 	<ul style="list-style-type: none"> 默认为nonstrict 	<ul style="list-style-type: none"> 注意默认值变化,酌情修改

4.2. Impala 代码改造

参考如下：



Impala 代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
1	默认建表行为	<ul style="list-style-type: none"> Text格式的Non-ACID表 	<ul style="list-style-type: none"> CREATE TABLE语句默认会生成Parquet格式的Insert-Only ACIDv2事务内表 	<ul style="list-style-type: none"> 显式添加STORED AS子句；或者，更改查询参数default_file_format
2	数据文件位置的更改	<ul style="list-style-type: none"> 位于/user/hive/warehouse 	<ul style="list-style-type: none"> 位于/warehouse/tablespace/managed/hive 	<ul style="list-style-type: none"> 直接操作数据文件的代码需要注意
3	默认内部表	<ul style="list-style-type: none"> 非事务表 	<ul style="list-style-type: none"> 具有insert_only属性的事务表 	<ul style="list-style-type: none"> 将DEFAULT_TRANSACTIONAL_TYPE查询选项设置为NONE，以便默认情况下所有新创建的内部表都不是事务性的
4	Hive和Impala表互操作	<ul style="list-style-type: none"> Impala可以操作Hive表 	<ul style="list-style-type: none"> Impala不能操作Hive内表 	<ul style="list-style-type: none"> 使用stored as parquet子句在 Hive 中创建表，如果他们想要Impala对这些表有操作能力 如果需要在系统级别更改此默认行为，则管理员应将Hive_on_Tez服务配置中的CM配置hive_default_fileformat_managed更改为Parquet

CLUSTER

© 2020 Cloudera, Inc. All rights reserved. 68



Impala 代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
5	ALTER TABLE命令	<ul style="list-style-type: none"> ALTER TABLE命令可以生效 	<ul style="list-style-type: none"> AnalysisException: ALTER TABLE not supported on transactional(ACID) table 	<ul style="list-style-type: none"> 通过Hive操作ALTER TABLE
6	RCFile Serde	<ul style="list-style-type: none"> Impala可以读取Hive的RC文件 	<ul style="list-style-type: none"> 默认的LazyBinaryColumnarSerDe在Hive中创建RC文件，Impala将无法读取RC文件。 	<ul style="list-style-type: none"> 将hive.default.rcfile.serde的配置选项设置为ColumnarSerDe，以保持Hive和Impala之间的互操作性
7	元数据刷新	<ul style="list-style-type: none"> 需手动INVALIDATE METADATA 	<ul style="list-style-type: none"> 无需手动INVALIDATE METADATA 	<ul style="list-style-type: none"> 从代码中删除INVALIDATE METADATA，以减少不必要的开销
8	保留关键字	<ul style="list-style-type: none"> free不是关键字 	<ul style="list-style-type: none"> free等成为了保留关键字 	<ul style="list-style-type: none"> 使用反引号包围关键字，例如`free`
9	Parquet格式增强	<ul style="list-style-type: none"> 二级编码 	<ul style="list-style-type: none"> 三级编码 	<ul style="list-style-type: none"> 如果注意到数组的奇怪结果，请尝试 SET PARQUET_ARRAY_RESOLUTION=TWO_LEVEL;

CLUSTER

© 2020 Cloudera, Inc. All rights reserved. 69



Impala 代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
10	禁止别名替换	<ul style="list-style-type: none"> GROUP BY, HAVING和ORDER BY的子表达式中支持别名替换 	<ul style="list-style-type: none"> GROUP BY, HAVING和ORDER BY的子表达式中不再支持别名替换 	<ul style="list-style-type: none"> 不使用别名, 改用原名
11	Decimal V2	<ul style="list-style-type: none"> Decimal V1 	<ul style="list-style-type: none"> 默认启用Decimal V2 	<ul style="list-style-type: none"> 要保持原C5行为, Set Decimal_V2=false
12	向分区表插入数据	<ul style="list-style-type: none"> 默认为NOCLUSTERED 模式 	<ul style="list-style-type: none"> 默认为CLUSTERED 模式, 在插入之前会按分区字段对数据进行排序, 耗时增长 	<ul style="list-style-type: none"> 要保持原C5行为, 添加/* + NOCLUSTERED */ / hint
13	Shutdown端口改变	<ul style="list-style-type: none"> 使用了Impala中的后端端口 	<ul style="list-style-type: none"> 使用KRPC端口, 例如: shutdown ('host100:27000') 	<ul style="list-style-type: none"> 修改对应脚本
14	客户端连接超时	<ul style="list-style-type: none"> 客户端无限期地等待打开新会话 	<ul style="list-style-type: none"> 新增参数 accepted client cxn timeout, 默认5分钟, 也即在5分钟超时之后拒绝新的连接请求 	<ul style="list-style-type: none"> 如果 -- accepted client cxn timeout=0, 客户端将无限期地等待连接到 Impala, 就像原C5行为

CLUSTERED

© 2020 Cloudera, Inc. All rights reserved. 70



Impala 代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
15	hms_event_polling_interval_s	<ul style="list-style-type: none"> 默认为0 	<ul style="list-style-type: none"> 默认为2 	<ul style="list-style-type: none"> 注意变化, 不建议修改
16	use_local_catalog	<ul style="list-style-type: none"> 默认为false 	<ul style="list-style-type: none"> 默认为true, 也即启用了On-demand metadata mode 	<ul style="list-style-type: none"> 注意变化, 不建议修改
17	HDFS cached table	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> 改为通过hdfs cacheadmin-addDirective添加HDFS caching
18	thread_reservation_limit	<ul style="list-style-type: none"> 无此参数 	<ul style="list-style-type: none"> 有此参数, 默认值为3000, 用于限制了每个节点上查询的保留线程数 	<ul style="list-style-type: none"> 对于复杂SQL, 建议set thread_reservation_limit=5000

CLUSTERED

© 2020 Cloudera, Inc. All rights reserved. 71

Impala:增强的十进制计算

使用 DECIMAL 表达式提高准确性

更接近 DECIMAL 计算的 SQL 标准.

	DECIMAL CDP PvC Base 7.1	DECIMAL CDH5
整体行为	返回结果或错误.	返回结果或返回 NULL 并发出警告.
溢出行为	错误中止.	发出警告并返回 NULL.
算术中的舍入	从后面截断并四舍五入数字	从后面截断数字, 不四舍五入.

DECIMAL V2是新的默认值, 恢复为以前的行为需要设置 SET DECIMAL V2 = FALSE;
 字符串强制转换 从后面截断并四舍五入. 从后面截断, 不四舍五入.

4.3. Spark 代码改造

参考如下:

Spark代码改造

主要变化

序号	项目	CDH5 (Spark1.6)	CDP7 (Spark2.4)	改造方案
1	编码入口	• HiveContext和SQLContext	• SparkSession	• HiveContext和SQLContext仍保留, 但建议改用SparkSession对象
2	spark-submit命令	• Spark1使用spark-submit, Spark2使用spark2-submit	• Spark2使用spark-submit, Spark3使用spark3-submit	• 保留spark2-submit命令: In -s spark-submit spark2-submit
3	默认Scala版本	• Scala2.10	• Scala2.11	• 原Scala 2.10代码需要重建
4	读取Hive内表	• Spark Native API	• 无法使用Spark Native API • 只能通过Hive Warehouse Connector或者HWC Spark Direct Reader	• Hive3 ACID打破Hive内表的Spark兼容性 • 但是Hive3 外表不受影响, 可以继续使用Spark Native API
5	create table ... location	• 生成Hive内表	• 生成Hive外表	• 在CDP Spark SQL中不允许为Hive内表指定位置, 创建用户指定位置的Hive表始终是Hive外部表



Spark代码改造

主要变化

序号	项目	CDH5 (Spark1.6)	CDP7 (Spark2.4)	改造方案
6	spark-assembly-*.jar	<ul style="list-style-type: none"> 可用 	<ul style="list-style-type: none"> spark-assembly-*.jar不复存在, 分散成多个小JAR包 	<ul style="list-style-type: none"> 将加载spark-assembly-*.jar 改为加载jars/*.jar
7	Dataset和DataFrame API	<ul style="list-style-type: none"> 属于两类不同的API unionAll 可用 explode 可用 registerTempTable 可用 	<ul style="list-style-type: none"> DataFrame成为Dataset [Row]类型别名 unionAll改为union explode改为function.explode()与select或flatMap一起使用 registerTempTable改为createOrReplaceTempView 	<ul style="list-style-type: none"> df.map -> df.rdd.map 使用新API替代废弃的API
8	spark.sql.parquet.cacheMetadata	<ul style="list-style-type: none"> 可用 	<ul style="list-style-type: none"> 不可用 	<ul style="list-style-type: none"> 不再使用spark.sql.parquet.cacheMetadata
9	Hive on Spark	<ul style="list-style-type: none"> 可用 	<ul style="list-style-type: none"> 不可用 	<ul style="list-style-type: none"> 改为Hive on Tez
10	Spark On HBase和Oozie Spark action	<ul style="list-style-type: none"> 基于Spark1 	<ul style="list-style-type: none"> 基于Spark2 	<ul style="list-style-type: none"> 改为使用Spark2

CLUSTERA

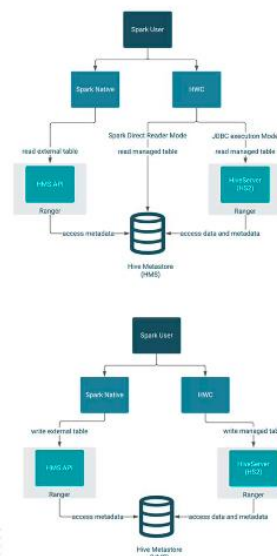
© 2020 Cloudera, Inc. All rights reserved. 74



HWC介绍

- HWC有两种模式
 - Spark Direct Reader 模式: 连接到Hive Metastore (HMS)
 - JDBC execution 模式: 连接到 HiveServer (HS2)

Capabilities	JDBC mode	Spark Direct Reader mode
Ranger integration (fine-grained access control)	✓	N/A
Hive ACID reads	✓	✓
Workloads handled	Small datasets	ETL without fine-grained access control



CLUSTERA

75

4.4.Hbase 代码改造

参考如下：



HBase代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
1	dfs.client.read.shortcircuit	<ul style="list-style-type: none"> 配置生效 	<ul style="list-style-type: none"> 配置未生效, 通过激活该配置, 可以提升30%-40%的性能 	<ul style="list-style-type: none"> 在CM中, 做如下配置: HBase -> Configurations -> HBase (Service-wide) -> HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml -> dfs.client.read.shortcircuit = true dfs.domain.socket.path = < Add same value which is configured in hdfs-site.xml >

CLUSTERA

© 2020 Cloudera, Inc. All rights reserved. 76



HBase代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
2	shaded artifacts工件	<ul style="list-style-type: none"> 客户端不依赖该工件 	<ul style="list-style-type: none"> 客户端需要切换依赖到该工件 	<ul style="list-style-type: none"> HBase客户端应用程序必须切换到hbase-shaded-client和hbase-shaded-mapreduce工件作为依赖项
3	HTable API变化	<ul style="list-style-type: none"> 支持HTable 	<ul style="list-style-type: none"> 不支持HTable, 改用Table table = connection.getTable(tableName); 	<ul style="list-style-type: none"> 参考 https://github.com/bithanjan/CDP-upgrade/blob/main/HBase.adoc
4	协处理器API变化	<ul style="list-style-type: none"> Observer API支持 extends 	<ul style="list-style-type: none"> 把 Foo extends BaseXXXObserver 改为 Foo implements XXXObserver 	<ul style="list-style-type: none"> 参考 https://github.com/bithanjan/CDP-upgrade/blob/main/HBase.adoc
5	master充当Region服务器	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> master默认不再携带Region 	

CLUSTERA

© 2020 Cloudera, Inc. All rights reserved. 77



HBase代码改造

主要变化

序号	项目	CDH5	CDP7	改造方案
6	日志的相关更改	<ul style="list-style-type: none"> 使用log4j 	<ul style="list-style-type: none"> 使用slf4j 	<ul style="list-style-type: none"> log4j.properties可能需要刷新
7	过滤器ReturnCode NEXT_ROW 行为的变化	<ul style="list-style-type: none"> 返回所有列族中的下一行 	<ul style="list-style-type: none"> 返回当前列族的下一行 	
8	HBase Shell命令依赖于捆绑的JRuby实例	<ul style="list-style-type: none"> Ruby 1.8 	<ul style="list-style-type: none"> Ruby 2.3.3 	<ul style="list-style-type: none"> 用户脚本可能需要修改来满足要求
9	指标 "totalRequestCount"	<ul style="list-style-type: none"> 增加了请求中执行的操作数 	<ul style="list-style-type: none"> 请求一次，加一 	
10	HBCK	<ul style="list-style-type: none"> HBCK v1 	<ul style="list-style-type: none"> HBCK v2, 且需要使用最新版本 	<ul style="list-style-type: none"> 必须使用HBCK v2, 使用v1可能会导致数据丢失, 请在每次需要使用该工具时, 向Cloudera售后索取该工具的最新版本

CLUDERA

© 2020 Cloudera, Inc. All rights reserved. 78

4.5.Kudo 代码改造

参考如下：



Kudu代码改造

主要变化



序号	项目	CDH5	CDP7	改造方案
1	版本变化	<ul style="list-style-type: none"> 1.7 	<ul style="list-style-type: none"> 1.13 	<ul style="list-style-type: none"> 使用新版本客户端库重新编译
2	Flume接收器	<ul style="list-style-type: none"> 包含 	<ul style="list-style-type: none"> 已删除 	<ul style="list-style-type: none"> 依赖于Flume集成的用户可以使用旧的Kudu-flume jar包, 或迁移到包含Kudu sink的Flume jar包。
3	精细授权	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> Ranger 	<ul style="list-style-type: none"> 配置Ranger相关权限
4	位置感知	<ul style="list-style-type: none"> 不支持 	<ul style="list-style-type: none"> 支持 	<ul style="list-style-type: none"> 如果分配了机架位置, 则必须运行kudu群集重新平衡工具以确保现有表符合机架感知放置策略。
5	Table 历史记录保留时间	<ul style="list-style-type: none"> 15分钟 	<ul style="list-style-type: none"> 7天 	

CLUDERA

© 2020 Cloudera, Inc. All rights reserved. 79


4.6. Kafka 代码改造

参考如下：





Kafka代码改造

主要变化





序号	项目	CDH5	CDP7	改造方案
1	版本变化	• 2.2.1	• 2.4.1	• 使用新版本客户端库重新编译
2	kafka-preferred-replica-election	• 支持	• 已弃用	
3	--zookeeper	• 支持	• 除kafka-configs和kafka-reassign-partitions之外，所有Kafka命令行工具均已弃用--zookeeper选项。	• 建议您改用--bootstrap-server选项
4	默认哈希算法	• MD5	• Murmur3	


© 2020 Cloudera, Inc. All rights reserved. 80


4.7. Solr 代码改造

参考如下：





Solr 代码改造

主要变化



序号	项目	CDH5	CDP7	改造方案
1	版本变化	• 4.10.3	• 8.4.1	• 根据API的变化重新实现应用
2	索引兼容	• N/A	• 不兼容	• 需重新针对数据生成索引


© 2020 Cloudera, Inc. All rights reserved. 81

样例代码参考：

Hive

- <https://github.com/bithanjun/CDP-upgrade/blob/main/Hive.adoc>
- Impala
- <https://github.com/bithanjun/CDP-upgrade/blob/main/Impala.adoc>
- Spark
- <https://github.com/bithanjun/CDP-upgrade/blob/main/Spark.adoc>
- HBase
- <https://github.com/bithanjun/CDP-upgrade/blob/main/HBase.adoc>