弹性 MapReduce

实践教程

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

实践教程
EMR on CVM 运维实践
HiveServer2 和 MetaStore 迁移到 Router
HDFS DataNode 维护状态切换实践教程
数据迁移实践
HDFS 通过对象存储数据迁移实践
HDFS 通过 DistCp 数据迁移实践
HBase 数据迁移实践
Hive 数据迁移实践
自定义伸缩实践教程
自动伸缩规则未执行原因排查实践
伸缩规则触发执行原则
伸缩规则设置实践教程
时间伸缩规则设置实践教程
负载伸缩规则设置实践教程
混合伸缩规则设置实践教程
在线容器集群部署 EMR

版权所有：腾讯云计算（北京）有限责任公司

实践教程
EMR on CVM 运维实践
HiveServer2 和 MetaStore 迁移到 Router

最近更新时间：2024-08-15 14:44:01

新增 Router 节点

登录 EMR 控制台，在**集群列表**中选择对应的集群单击 **ID/名称**进入集群详情页，在集群详情页中选择**集群资源 > 资源管理**，进入资源管理页面，单击**扩容**进入集群扩容页面。

在集群扩容页面中，选择扩容**节点类型**为 **Router**，**扩容服务**为 Hive，其他选项可根据需要自行选择。

迁移 HiveServer2 与 MetaStore

进入 EMR 控制台，通过**集群服务**使用 Hive 服务的**配置管理**功能，修改 Router 节点 `hive-site.xml` 配置文件以下参数：

参数	值	说明
hive.metastore.uris	thrift://${router-ip}:7004	修改 hive metastore 的地址信息，将 hive 元数据存储的服务地址指向新增的 Router 节点。其中 ${router-ip} 为 MetaStore 所在 Router 节点的内网 IP。

配置下发并保存后，在**集群服务**中选择 Hive 组件的**操作 > 角色管理**，将 Master 节点上的所有 hive 进程暂停，重启 Router 节点上的 hive 进程。

在 router 上测试 hiveserver2。如果能正确连接并查询到已有的表，则说明迁移成功。

```
beeline -u jdbc:hive2://${router-ip}:7001 -n hadoop -p hadoop
show tables
```

修改 Knox 代理地址

HiveServer2 迁移后，需要进入 Master 节点 修改 Knox 配置文件，将 Hive 及 HiveUI 代理到 Router 节点的 Hive 组件。

```
vim /usr/local/service/knox/conf/topologies/emr.xml  //修改 HIVE 和 HIVEUI。
<service>
 <role>HIVE</role>
 <url>http://Router-ip:7003</url>
 <param>
     <name>replayBufferSize</name>
     <value>8</value>
 </param>
</service>
<service>
 <role>HIVEUI</role>
 <url>http://Router-ip:7003</url>
</service>
```

命令行执行重启 Knox。

```
/usr/local/service/knox/bin/gateway.sh stop ; /usr/local/service/knox/bin/gateway.sh start
```

HDFS DataNode 维护状态切换实践教程

最近更新时间：2024-11-18 15:28:22

DataNode 维护状态（IN_MAINTENANCE）适用于 DataNode 短暂下线，但不需要迁移数据的场景，如服务快速维修，磁盘更换等。DataNode 维护模式操作入口控制台默认不开启，需要用户手动配置后支持。

> **说明：**
> 1. 该操作仅支持 Hadoop3.x 及以上版本。
> 2. 暂停状态的 DataNode 不支持维护操作。

开启控制台切换管理状态操作入口

1. 修改 hdfshosts 中的内容为 json 格式。

   使用控制台集群脚本功能在 Master 节点上执行脚本文件 hdfshosts_txt_to_json.sh 和 hdfshosts_txt_to_jso_rollback.sh，脚本文件内容如下：

```
#!/bin/bash

cd /usr/local/service/hadoop/etc/hadoop
file=hdfshosts
if [ ! -f "$file" ];then
  echo "$file not exists"
  exit -1
fi

bak_file="$file.txt.bak"
if [ ! -f "$bak_file" ];then
  cp -f $file $file.txt.bak
fi

output_file="$file.tmp"
## 生成json文件
echo '[' > "$output_file"
first_record=true
while IFS= read -r line
do
  if [ "$first_record" = false ]; then
    echo ',' >> "$output_file"
  fi

  echo "  {\"hostName\": \"$line\"}" >> "$output_file"

  first_record=false
done < "$file"
echo ']' >> "$output_file"
mv -f $output_file $file
chown hadoop:hadoop $file
chmod 755 $file
cat $file
```

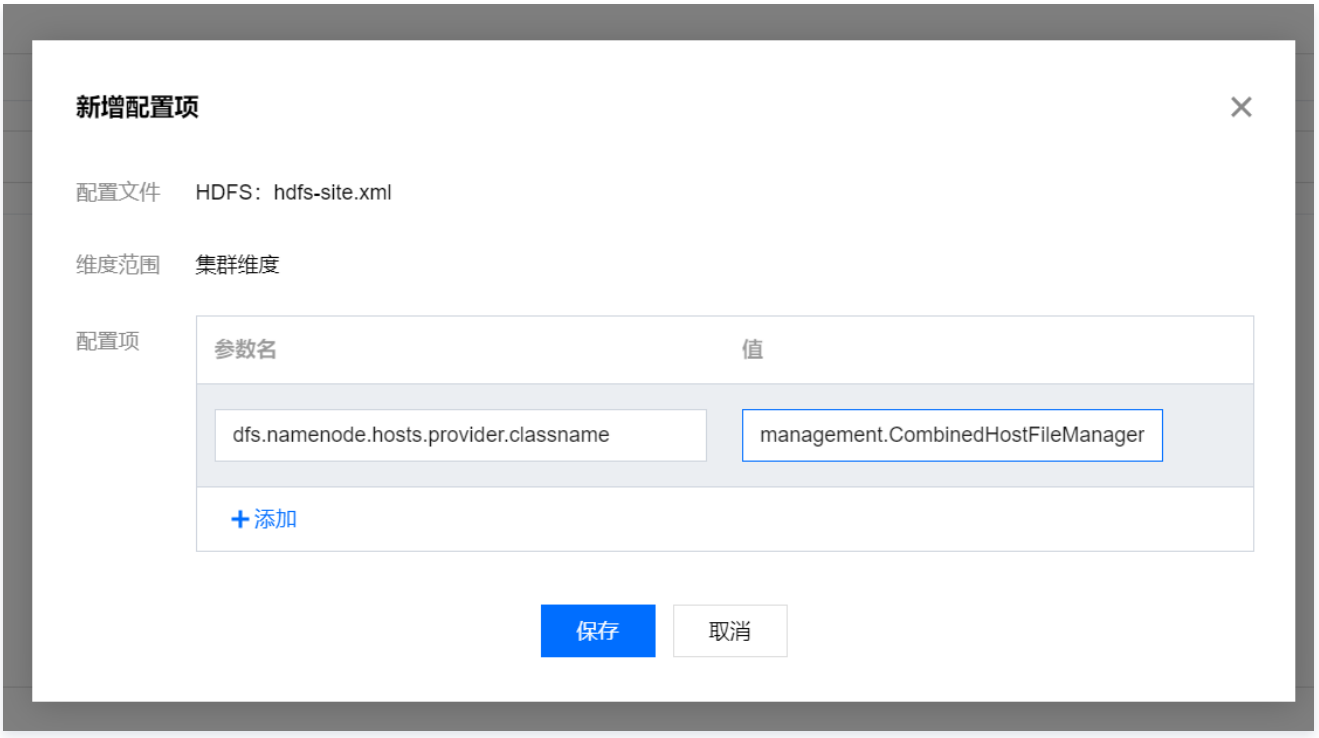
```
#!/bin/bash

cd /usr/local/service/hadoop/etc/hadoop
file=hdfshosts
bak_file="$file.txt.bak"
cp -f $bak_file $file
chown hadoop:hadoop $file
chmod 755 $file
```

```
cat $file
```

2. 在 hdfs-site.xml 中新增配置项。

   新增配置参数 dfs.namenode.hosts.provider.classname，参数值 org.apache.hadoop.hdfs.server.blockmanagement.CombinedHostFileManager。



3. 保存并下发配置后，重启 NameNode。
4. 查看 WebUI 或者执行 hdfs dfsadmin -report 进行检查。
5. （建议）缩容、扩容 core 节点一次，因为上述操作会修改 hdfshosts 文件，避免后续出现问题。
6. 在 HDFS **角色管理**中出现**切换管理状态**按钮。

DataNode 进入维护状态

1. 登录 EMR 控制台，在集群列表中单击对应的**集群 ID/名称**进入集群详情页。
2. 在集群详情页中单击**集群服务**，然后选择 HDFS 组件右上角**操作 > 角色管理**。
3. 在**角色管理**勾选操作状态为已启动的 DataNode 角色后，**切换状态**执行 DataNode 进入维护状态操作。
4. 选择进入维护状态时，可设置维护时间。维护时间内 DataNode 将不再对外提供服务，且不会进行数据迁移；超过配置的维护时间之后还没有恢复服务将开始数据迁移。

DataNode 退出维护状态

若维护时间内节点维修完成，超时后 DataNode 将重新对外提供服务，但是维护状态需要用户在控制台手动操作退出。

1. 登录 EMR 控制台，在集群列表中单击对应的**集群 ID/名称**进入集群详情页。
2. 在集群详情页中单击**集群服务**，然后选择 HDFS 组件右上角**操作 > 角色管理**。
3. 在**角色管理**勾选操作状态为已启动（维护中）的 DataNode 角色后，**切换管理状态**执行 DataNode 退出维护状态操作。

数据迁移实践
HDFS 通过对象存储数据迁移实践

最近更新时间：2024-10-31 11:01:02

如果您需要将自有 HDFS 的原始数据迁移至腾讯云 EMR，可以通过两种方式进行数据迁移，第一种是通过腾讯云对象存储（COS）进行数据中转迁移，第二种是通过 Hadoop 自带文件迁移工具 DistCp 进行数据迁移。本文主要介绍通过腾讯云对象存储（COS）进行数据中转迁移。

原始数据非 HDFS 数据

如果您的原始数据不是 HDFS 数据而是其他形式的文件数据，可以通过 COS 的 web 控制台或者 COS 提供的 API 来把数据传入到 COS，然后在 EMR 集群中进行分析，COS 传输数据请参见 本地数据迁移至 COS 。

原始数据在 HDFS 的数据迁移

1. 获取 COS 迁移工具。
   获取迁移工具，更多迁移工具请参考 工具概览 。
2. 工具配置。
   配置文件统一放在工具目录里的 conf 目录，将需要同步的 HDFS 集群的 core-site.xml 拷贝到 conf 中，其中包含了 NameNode 的配置信息，编辑配置文件 cos_info.conf，包括 appid、bucket、region 以及密钥信息。

   > ⚠ 注意
   > - 建议用户使用子账号密钥，遵循 最小权限原则说明，防止泄露目标存储桶或对象之外的资源。
   > - 如果您一定要使用永久密钥，建议遵循 最小权限原则说明 对永久密钥的权限范围进行限制。

   命令参数说明：

```
-ak <ak>                                the cos secret id //用户的 SecretId，建议使用子账号密钥，授权遵循最小
权限指引，降低使用风险。子账号密钥获取可参考：https://cloud.tencent.com/document/product/598/37140
-appid,--appid <appid>                  the cos appid
-bucket,--bucket <bucket_name>          the cos bucket name
-cos_info_file,--cos_info_file <arg>    the cos user info config default is ./conf/cos_info.conf
-cos_path,--cos_path <cos_path>         the absolute cos folder path
-h,--help                               print help message
-hdfs_conf_file,--hdfs_conf_file <arg>  the hdfs info config default is ./conf/core-site.xml
-hdfs_path,--hdfs_path <hdfs_path>      the hdfs path
-region,--region <region>               the cos region. legal value cn-south, cn-east, cn-north, sg
-sk <sk>                                the cos secret key //用户的 SecretKey，建议使用子账号密钥，授权遵循最
小权限指引，降低使用风险。子账号密钥获取可参考：https://cloud.tencent.com/document/product/598/37140
-skip_if_len_match,--skip_if_len_match  skip upload if hadoop file length match cos
```

3. 执行迁移：

```
# 所有操作都要在工具目录下。如果同时设置了配置文件和命令行参数，以命令行参数为准
./hdfs_to_cos_cmd -h
# 从 HDFS 拷贝到 COS（如果 COS 上已存在文件，则会覆盖）
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
# 从 HDFS 拷贝到 COS，同时要拷贝的文件和 COS 的长度一致，则忽略上传（适用于拷贝一次后，重新拷贝）
# 这里只做长度的判断，因为如果将 Hadoop 上的文件摘要算出，开销较大
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -skip_if_len_match
# 完全通过命令行设置参数
./hdfs_to_cos_cmd -appid 1********** -ak
   **************************************** -sk
   ********************************** -bucket test -cos_path /hdfs
   -hdfs_path /data/data -region cn-south -hdfs_conf_file
/home/hadoop/hadoop-2.8.1/etc/hadoop/core-site.xml
```

4. 验证运行命令后，输出如下日志::

```
[Folder Operation Result : [ 53(sum)/ 53(ok) / 0(fail)]]
[File Operation Result: [22(sum)/ 22(ok) / 0(fail) / 0(skip)]]
[Used Time: 3 s]
```

- sum 表示总共需要迁移的文件数。
- ok 表示成功迁移的文件数。
- fail 表示迁移失败的文件数。
- skip 表示在添加 skip_if_len_match 参数后，由于上传文件和同名文件具有相同长度的文件，则跳过的数量。

您也可以登录 COS 控制台查看数据是否已经正确迁移过来。对象存储使用指引请参见 快速入门 。

常见问题

- 请确保填写的配置信息，包括 appID、密钥信息、bucket 和 region 信息正确，以及机器的时间和北京时间一致（如相差1分钟左右是正常的），如果相差较大，请设置机器时间。
- 请保证对于 DateNode，拷贝程序所在的机器也可以连接。因 NameNode 有外网 IP 可以连接，但获取的 block 所在的 DateNode 机器是内网 IP，无法连接上，因此建议同步程序放在 Hadoop 的某个节点上执行，保证对 NameNode 和 DateNode 皆可访问。
- 权限问题，用当前账户使用 Hadoop 命令下载文件，看是否正常，再使用同步工具同步 Hadoop 上的数据。
- 对于 COS 上已存在的文件，默认进行重传覆盖，除非用户明确地指定 -skip_if_len_match，当文件长度一致时则跳过上传。
- cos path 都认为是目录，最终从 HDFS 上拷贝的文件都会存放在该目录下。

HDFS 通过 DistCp 数据迁移实践

最近更新时间：2023-12-20 17:06:32

如果您需要将自有 HDFS 的原始数据迁移至腾讯云 EMR，可以通过两种方式进行数据迁移，第一种是通过腾讯云对象存储（COS）进行数据中转迁移，第二种是通过 Hadoop 自带文件迁移工具 DistCp 进行数据迁移。本文主要介绍通过 DistCp 进行数据迁移。
DistCp（distributed copy）是 Hadoop 自带的文件迁移工具。它使用 MapReduce 来实现其分发、错误处理和恢复、报告的功能。它将文件和目录的列表扩展为映射任务的输入，每个任务将复制源列表中指定的文件的分区。使用 DistCp 需要实现自建集群和 EMR 集群的网络互通。
使用 DistCP 数据迁移步骤如下：

步骤1：网络打通

本地自建 HDFS 文件迁移到 EMR

本地自建 HDFS 文件迁移到 EMR 集群需要有专线打通网络，可以联系开发人员协助解决。

CVM 上的自建 HDFS 文件迁移到 EMR

- CVM 的所属网络和 EMR 集群的所属网络在同一 VPC 下，则可以自由传送文件。
- CVM 的所属网络和 EMR 集群的所属网络在不同 VPC 下，需要使用对等连接将网络打通。

使用对等连接

网段1：广州的 VPC1 中的子网 A 192.168.1.0/24。
网段2：北京的 VPC2 中的子网 B 10.0.1.0/24。
1. 登录 私有网络控制台-对等连接，在列表上方选择地域广州，选择私有网络 VPC1，然后单击**+新建**。

2. 进入建立对等连接页。
- 名称：对等连接的名称，例如 PeerConn。
- 本端地域：填写本地端地域，例如广州。
- 本端网络：填写本端网络，例如 VPC1。
- 对端账户类型：填写对端网络所属账户，如果广州和北京两个网络在同一账户下，选择**我的账户**，如果不在同一账户，则要选择**其它账户**。

> **说明**
> 如果本端网络和对端网络都在同一地域，例如广州，通信是免费的，也不需要选择**带宽上线**；如果不在同一地域，就要进行收费，同时带宽上限可选。

- 对端地域：填写对端地域，例如北京。

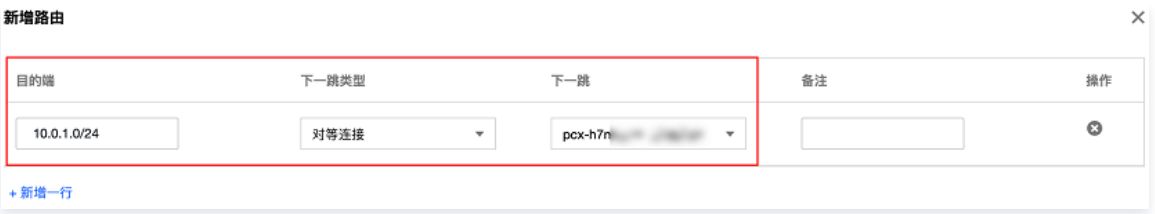
- 对端网络：填写对端网络，例如 VPC2。

3. 同账户内私有网络进行连接，新建后对等连接立即生效；与其它账户私有网络创建对等连接，需要对端接受此对等连接后生效。参见 同账号创建对等连接通信 和 跨账号创建对等连接通信 。
4. 为对等连接配置本端和对端路由表。

- 登录 私有网络控制台 ，单击左侧目录中的**子网**，进入管理页面。单击对等连接本端指定子网（例如广州的子网 VPC1）的关联路由表 ID，进入路由表详情页。

- 单击**新增路由策略**。

- 目的端中填入对端 CIDR（例如北京的 VPC2 的 CIDR 是10.0.1.0/24），下一跳类型选择**对等连接**，下一跳选择已建立的对等连接（PeerConn）。



- 以上步骤是配置广州 VPC1 到北京 VPC2 的路由表，还需要配置北京 VPC2 到广州 VPC1 的配置，配置过程同上。
- 路由表配置完成后，不同私有网络的网段之间即可进行通信。

步骤2：执行拷贝

```
# 集群间的拷贝，将一个文件夹拷贝到另一个集群
hadoop distcp hdfs://nn1:9820/foo/bar hdfs://nn2:9820/bar/foo

# 指定文件拷贝
hadoop distcp hdfs://nn1:9820/foo/a hdfs://nn1:9820/foo/b hdfs://nn2:9820/bar/foo

# 如果指定的文件太多，可使用 -f 参数。
```

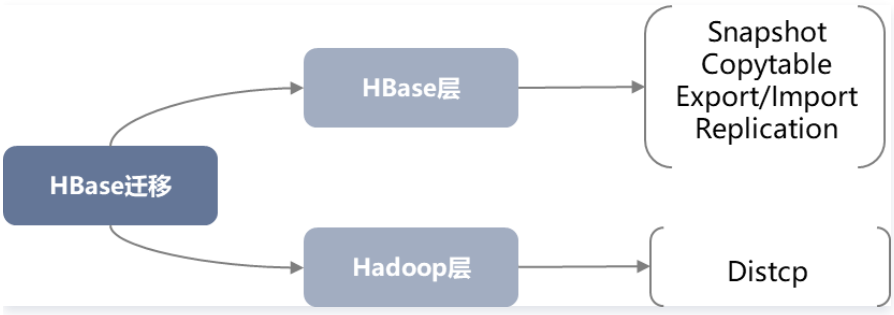
注意

- 对于上述命令，必须要求源和目的版本相同。
- 如果另一个客户端仍然在写入源文件，则该拷贝可能会失败；如果一个文件正在被拷贝到目的端，试图重写该文件的操作会失败；如果源文件在被复制之前被移动，那么拷贝将失败，报错信息为 FileNotFoundException。

HBase 数据迁移实践

最近更新时间：2025-05-27 14:40:42

Hbase 表是基于 hadoop HDFS 构建，所以 Hbase 的迁移可从两个维度来看，基于 hadoop HDFS 的 distcp 的迁移方式和基于 Hbase 表结构的 Hbase 层面提供的相关工具迁移。



如上图所示，HBase 迁移有多种方案，其中基于 Snapshot 的迁移方式是推荐的迁移方案。

HBase 基于 Snapshot 迁移

> **说明：**
> 以下步骤均以 hadoop 用户操作。

1. 在目标集群上建立表结构一样的表。

```
$ hbase shell
hbase> create 'myTable', 'cf1', 'cf2'
```

2. 在原集群上对表初始化数据。

```
$ hbase shell
hbase> put 'myTable', 'row1', 'cf1:a', 'value1'
hbase> put 'myTable', 'row2', 'cf2:b', 'value2'

hbase> scan 'myTable'
ROW                                    COLUMN+CELL
row1                                   column=cf1:a, timestamp=2023-08-09T16:43:10.024,
value=value1                           row2                                    column=cf2:b,
timestamp=2023-08-09T16:43:20.036, value=value2
```

3. 使用 `hbase shell` 在原始集群中创建一个快照。

```
$ hbase shell
hbase>snapshot 'myTable', 'myTableSnapshot'
```

这里 `'myTable'` 是 hbase 的表名，`'myTableSnapshot'` 是快照的名称。创建完成后可使用 list_snapshots 确认是否成功，或使用 delete_snapshot 删除快照。

```
hbase> delete_snapshot 'myTableSnapshot'
```

4. 在源集群中导出快照到目标集群。

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot myTableSnapshot -copy-to
hdfs://10.0.0.38:4007/hbase
```

说明：

1. `10.0.0.38:4007` 是目标集群的 `$activeip:$rpcport`，导出快照时系统级别会启动一个 mapreduce 的任务，可以在后面增加 `-mappers 16 -bandwidth 200` 来指定 mapper 和带宽。其中200指的是200MB/sec。
2. 确保目标端集群 HDFS 上存在 /hbase 目录。

5. 在目标集群从 hdfs 恢复相应的 hbase 表及数据。

```
hbase> disable "myTable"
hbase> restore_snapshot 'myTableSnapshot'
hbase> enable 'myTable'
```

6. 最后可通过简单的 HBase 表操作进行测试。

Hive 数据迁移实践

最近更新时间：2024-10-18 17:00:22

Hive 迁移涉及两部分，数据迁移和元数据迁移。Hive 表数据主要存储在 HDFS 上，故数据的迁移主要在 HDFS 层。Hive 的元数据主要存储在关系型数据库，可平滑迁移到云上 TencentDB，并可保障高可用。

Hive 元数据迁移

1. Dump 源 Hive 元数据库。

```
mysqldump -hX.X.X.X -uroot -pXXXX --single-transaction --set-gtid-purged=OFF hivemetastore >
hivemetastore-src.sql
# 如果 mysql 数据没有开启 GTID，请删除命令行中的 --set-gtid-purged=OFF
# X.X.X.X为数据库服务器地址
# XXXX为数据库密码
# 如果数据库用户不是 root，请用正确的用户名
# hivemetastore 是 Hive 元数据库名
```

2. 确认目标集群 Hive 表数据在 HDFS 中的默认存储路径。
   Hive 表数据在 HDFS 中的默认存储路径由 `hive-site.xml` 中的 `hive.metastore.warehouse.dir` 配置项指定。如果目标集群 Hive 表在 HDFS 的存储路径需要与源集群 Hive 表路径一致，可以参考以下示例对配置文件进行修改。例如，源集群 `hive-site.xml` 中 `hive.metastore.warehouse.dir` 为下面的值。

```
<property>
 <name>hive.metastore.warehouse.dir</name>
 <value>/apps/hive/warehouse</value>
</property>
```

   目标集群 `hive-site.xml` 中 `hive.metastore.warehouse.dir` 为下面的值。

```
<property>
 <name>hive.metastore.warehouse.dir</name>
 <value>/usr/hive/warehouse</value>
</property>
```

   如果目标集群 Hive 表在 HDFS 的存储位置依然保持与源集群 Hive 一致，那么修改目标 `hive-site.xml` 中的 `hive.metastore.warehouse.dir`，即为：

```
<property>
 <name>hive.metastore.warehouse.dir</name>
 <value>/apps/hive/warehouse</value>
</property>
```

3. 确认目标 Hive 元数据 SDS.LOCATION 和 DBS.DB_LOCATION_URI 字段。
   通过下面的查询获取当前 SDS.LOCATION 和 DBS.DB_LOCATION_URI 字段。

```
SELECT DB_LOCATION_URI from DBS;
SELECT LOCATION from SDS;
```

   查询出的结果类似如下：

```
mysql> SELECT LOCATION from SDS;
+-------------------------------------------------+
| LOCATION |
+-------------------------------------------------+
| hdfs://HDFS2648/usr/hive/warehouse/hitest.db/t1 |
```

```
| hdfs://HDFS2648/usr/hive/warehouse/wyp |
+------------------------------------------------+
mysql> SELECT DB_LOCATION_URI from DBS;
+-----------------------------------------+
| DB_LOCATION_URI |
+-----------------------------------------+
| hdfs://HDFS2648/usr/hive/warehouse |
| hdfs://HDFS2648/usr/hive/warehouse/hitest.db |
+-----------------------------------------+
```

其中 `hdfs://HDFS2648` 是 HDFS 默认文件系统名，由 `core-site.xml` 中的 `fs.defaultFS` 指定。

```
<property>
 <name>fs.defaultFS</name>
 <value>hdfs://HDFS2648</value>
</property>
```

`/usr/hive/warehouse` 为 Hive 表在 HDFS 中的默认存储路径，也是 `hive-site.xml` 中 `hive.metastore.warehouse.dir` 指定的值。所以我们需要修改源 hive 元数据 sql 文件中的 SDS.LOCATION 和 DBS.DB_LOCATION_URI 两个字段。确保被导入的 Hive 元数据库中的这两个字段使用的是正确的路径。可使用如下 sed 命令批量修改 sql 文件。

```
替换ip：sed -i 's/oldcluster-ip:4007/newcluster-ip:4007/g' hivemetastore-src.sql
替换defaultFS：sed -i 's/old-defaultFS/new-defaultFS/g' hivemetastore-src.sql
```

其中 oldcluster-ip、newcluster-ip 分别为源集群、目标集群 namenode 的 ip 地址，old-defaultFS、new-defaultFS 分别为源集群、目标集群的 fs.defaultFS 配置项的值。

> **说明**
> 如果使用了 Kudu、Hbase 等部分组件，用 Metastore 作为元数据服务，也需更改目标 Hive 元数据中对应 location 字段。

4. 停止目标 Hive 服务 MetaStore、HiveServer2、WebHCataLog。
5. 备份目标 Hive 元数据库。

```
mysqldump -hX.X.X.X -uroot -pXXXX --single-transaction --set-gtid-purged=OFF hivemetastore >
hivemetastore-target.sql
# 如果 mysql 数据没有开启 GTID，请删除命令行中的 --set-gtid-purged=OFF
# X.X.X.X为数据库服务器地址
# XXXX为数据库密码
# 如果数据库用户不是 root，请用正确的用户名
# hivemetastore 是 Hive 元数据库名
```

6. Drop/Create 目标 Hive 元数据。

```
mysql> drop database hivemetastore;
mysql> create database hivemetastore;
```

7. 导入源 Hive 元数据库到目标数据库。

```
mysql -hX.X.X.X -uroot -pXXXX hivemetastore < hivemetastore-src.sql
# X.X.X.X为数据库服务器地址
# XXXX为数据库密码
# 如果数据库用户不是 root，请用正确的用户名
# hivemetastore 是 Hive 元数据库名
```

8. Hive 元数据升级。
   如果目标和源 Hive 版本一致，则可直接跳过该步骤；否则，分别在源集群和目标集群查询 Hive 版本。

```
hive --service version
```

hive 的升级脚本存放在 `/usr/local/service/hive/scripts/metastore/upgrade/mysql/` 目录下。
hive 不支持跨版本升级，例如 hive 从1.2升级到2.3.0需要依次执行：

```
upgrade-1.2.0-to-2.0.0.mysql.sql -> upgrade-2.0.0-to-2.1.0.mysql.sql -> upgrade-2.1.0-to-
2.2.0.mysql.sql -> upgrade-2.2.0-to-2.3.0.mysql.sql
```

升级脚本主要操作为建表、加字段、改内容。如果表或字段已经存在，则升级过程中字段已存在的异常可以忽略。例如 hive 从2.3.3升级至3.1.1。

```
mysql> source upgrade-2.3.0-to-3.0.0.mysql.sql;
mysql> source upgrade-3.0.0-to-3.1.0.mysql.sql;
```

9. 如果源 Hive 中有 phoenix 表，修改目标 Hive 元数据中 phoenix 表的 zookeeper 地址。
通过下面的查询获取 phoenix 表的 `phoenix.zookeeper.quorum` 配置。

```
mysql> SELECT PARAM_VALUE from TABLE_PARAMS where PARAM_KEY = 'phoenix.zookeeper.quorum';
+-----------------------------------------------+
| PARAM_VALUE |
+-----------------------------------------------+
| 172.17.64.57,172.17.64.78,172.17.64.54 |
+-----------------------------------------------+
```

查看目标集群的 zookeeper 地址，即 hive-site.xml 配置文件中 `hbase.zookeeper.quorum` 指定的值。

```
<property>
 <name>hbase.zookeeper.quorum</name>
 <value>172.17.64.98:2181,172.17.64.112:2181,172.17.64.223:2181</value>
</property>
```

将目标 Hive 元数据中的 phoenix 表的 zookeeper 地址改为目标集群的 zookeeper 地址。

```
mysql> UPDATE TABLE_PARAMS set PARAM_VALUE  = '172.17.64.98,172.17.64.112,172.17.64.223' where
PARAM_KEY = 'phoenix.zookeeper.quorum';
```

10. 检查目标 Hive 元数据中表名的大小写格式，参考以下示例将所有小写表名改为大写：

```
alter table metastore_db_properties rename to METASTORE_DB_PROPERTIES;
```

11. 启动目标 Hive 服务 MetaStore、HiveServer2、WebHCataLog。
12. 最后可通过简单的 Hive sql 查询进行验证。

自定义伸缩实践教程
自动伸缩规则未执行原因排查实践

最近更新时间：2023-03-24 15:05:00

1. 弹性资源达到最小实例数，如需继续缩容，可尝试调整最小实例数。
   出现原因：存在缩容规则被触发，但是当前弹性节点数小于最小节点数。
   解决方法：如需要继续缩容，重新设置最小节点数。
2. 弹性资源超过最大实例数，如需继续扩容，可尝试调整最大实例数。
   出现原因：存在扩容规则被触发，但是当前弹性节点数等于最大节点数。
   解决方法：如需要继续扩容，重新设置最大节点数。
3. 未设置伸缩规格，无法扩容，可尝试添加规格后重试。

   出现原因：如上图，存在自动伸缩规则被触发，但是在**控制台 > 自动伸缩 > 伸缩规格管理项**中，未添加节点规格。
   解决方案：单击右上角**添加规格**，选择期望的节点规格。
4. 资源不足，可尝试更换资源充足的规格或 提交工单 联系我们。
   出现原因：存在扩容规则被触发，但当前可用区下的所选机型资源不足。
   解决方案：重新设置资源充足的节点规格。
5. 当前设置重试时间过短，建议调长过期重试时间。
   出现原因：时间伸缩规则从触发时间到过期重试时间内，集群存在其它自动扩缩容流程，导致当前时间伸缩规则未被执行。
   解决方法：可以编辑规则，适当延长过期重试时间，确保规则能被执行。
6. 账号余额不足，无法进行扩容。
   出现原因：扩容规则被触发，在下单时出现余额不足。
   解决方案：进入 费用中心 对账户进行充值。
7. 当前无满足条件的弹性资源可进行缩容。
   出现原因：存在缩容规则被触发，但是当前无弹性节点资源或者节点均设置为定时销毁。
   解决方案：如需继续缩容定时销毁的节点，可以选择手动缩容。
8. 集群状态未处于可扩容状态，无法扩容。
   出现原因：扩容规则被触发，但是当前集群处于安装组件、扩容等非运行中状态，无法扩容。
   解决方案：可以手动扩容或者编辑规则，适当延长过期重试时间，确保规则能被执行。
9. 集群处于扩容冷却窗口期，暂时无法触发扩容，建议调整扩缩容规则的冷却时间。
   出现原因：扩容规则被触发，但集群当前处于其它扩缩容的冷却期中，无法被执行。
   解决方案：可以适当缩短其它规则的冷却时间，或者延长该扩容规则的过期重试时间。

版权所有：腾讯云计算（北京）有限责任公司

伸缩规则触发执行原则

最近更新时间：2025-04-16 16:20:32

扩容规则执行时预设资源添加原则

每个集群最多可配置10种伸缩规格，扩容规则触发时将根据规格优先级进行扩容，当高优先级规格数量不足时，由次优先级资源规格混合高优先级规格进行扩容补充计算资源（**按量计费和竞价实例执行顺序相同**）。

- **当资源充足时：1>2>3>4>5**

> **例如**
> 预设5种规格且资源充足，当扩容规则触发需要扩容10台节点时，按照顺序规格1扩容10台节点，其余预设规格不选择。

- **当资源不足时：1+2>1+2+3>1+2+3+4>1+2+3+4+5**

> **例如**
> 预设规格1有8台节点，规格2有4台节点，规格3有3台节点，当扩容规则触发需要扩容13台节点时，按照顺序规格1扩容8台节点，规格2扩容4台，规格3扩容1台节点。

- **当资源规格无货时，假设规格2无货：1+3>1+3+4>1+3+4+5**

> **例如**
> - 预设规格1有8台节点，规格2没货没有节点，规格3有3台节点，当扩容规则触发需要扩容10台节点时，按照顺序规格1扩容8台节点，规格2不选，规格3扩容2台节点。
> - 预设规格1有8台节点，其余预设规格均无货，当扩容规则触发，需要扩容10台节点时，扩容规则将会触发，并扩容规格1扩容8台节点，扩容部分成功。

扩容方式：支持选择：节点、内存、核数三种方式；三种方式仅支持整数非0值输入。当方式选择核数和内存时，扩容保证最大算力进行扩容节点数量换算。

> **例如**
> - 按核数扩容，设置扩容10核，但规格按优先顺序扩容规格为8核时，规则触发将扩容**2台8核节点**。
> - 按内存扩容，设置扩容20G，但规格按优先顺序扩容规格为16G时，规则触发将扩容**2台16G节点**。

开启资源补足重试时弹性节点扩容原则

自动扩容在高峰下单时可能由于资源争抢导致实际扩容机器数量达不到弹性目标数量，当您开启资源补足重试策略后，如果配置的伸缩规格资源充足，系统会自动重试申请资源，直到满足或接近目标数量。

> **注意：**
> - 重试次数上限为5次，达到次数上限后无论扩容数量是否达到目标数量，将不再重试。

- 预设资源规格优先级：1>2>3>4>5。假设预设规格1有8台节点，规格2有2台节点，规格3有6台节点，规格4有6台节点。

> **例如：**
> - **未开启资源补足重试：**当扩容规则触发需要扩容15台节点时，按照顺序规格1扩容8台节点，规格2扩容2台节点，规格3扩容5台节点。下单时由于资源争抢，实际发货：规格1扩容8台节点，规格2扩容0台节点，规格3扩容5台，共计扩容13台节点，扩容部分成功。
> - **开启资源补足重试：**当扩容规则触发需要扩容15台节点时，按照顺序规格1扩容8台节点，规格2扩容2台节点，规格3扩容5台节点。下单时由于资源争抢，初次扩容实际发货：规格1扩容8台节点，规格2扩容0台节点，规格3扩容5台，共计扩容13台节点。此时触发补足重试，轮询规格1、2、3、4、5，最终补充发货：规格2扩容1台，规格3扩容1台。2次发货总计：规格1扩容8台节点，规格2扩容1台节点，规格3扩容6台，共计扩容15台节点，扩容成功。

缩容规则执行时弹性节点缩容原则

自动伸缩功能扩容出的弹性节点，当缩容规则触发时：按时间缩容，将优先缩容空闲节点，根据“**先扩后缩，后扩先缩**”原则执行，不足缩容数量时，再选择缩容运行 container 的节点；按负载缩容，优先缩容部署了负载指标所属服务的节点，且优先缩容空闲节点，根据“**先扩后缩，后扩先缩**”原则执行，不足缩容数量

时，再选择缩容运行 container 的节点。非弹性节点将不受缩容规则触发而触发缩容动作，非弹性节点仅支持手动缩容。

> **注意**
> - 定时销毁节点将**不受“先扩后缩，后扩先缩”和集群“最小节点数”原则约束**；时间到达即可执行缩容，且默认优雅缩容30分钟范围。
> - 空闲节点的判断依据为5分钟内无正在运行的 container。

- 按负载缩容，假设节点创建时间从早到晚A>B>C>D>E。

> **例如：**
> - 设置 YARN 负载指标缩容，缩容5台节点，C、D、E部署了 YARN 组件，且D、E正在运行 container，当缩容规则触发时，缩容顺序为：C>E>D>B>A。
> - 设置 Trino 负载指标缩容，缩容5台节点，C、D、E部署了 Trino 组件，且D、B正在运行 container，当缩容规则触发时，缩容顺序为：E>C>D>A>B。

- 按时间缩容，假设节点创建时间从早到晚A>B>C>D>E。

> **例如：**
> - 按节点缩容，设置缩容5台节点，D、E正在运行 container，当缩容规则触发时，缩容顺序为：C>B>A>E>D。

缩容方式：支持选择：节点、内存、核数三种方式；三种方式仅支持整数非0值输入当方式选择核数和内存时，缩容保证业务正常按最小台数进行缩容节点数量换算，节点无任务运行时按时间倒序缩容且保证最少一台缩容。

> **例如**
> - 按核数缩容，设置缩容20核，缩容规则触发时，按时间倒序集群存在弹性节点分别为3台8核16G节点和2台4核8G节点，将成功缩容2台8核16G节点。
> - 按内存缩容，设置缩容30G，缩容规则触发时，按时间倒序集群存在弹性节点分别为3台8核16G节点和2台4核8G节点，将成功缩容1台8核16G节点。

伸缩规则触发顺序执行原则

支持时间伸缩和负载伸缩混合弹性规则设置，规则触发遵循“先触发先执行，同时触发根据规则优先顺序执行”；规则状态用于标记规则是否开启，默认为开启状态，当不需要规则运行但仍想保留规则配置时可将规则状态设置为关闭。

- 仅按负载进行伸缩设置

  1.1 遵循“先触发先执行，同时触发根据规则优先顺序执行”，如：1>2>3>4>5。

  1.2 单条负载伸缩规则支持设置多指标，当所有指标都符合条件时触发规则。

  1.3 支持指定时间段内监控集群负载变化，设置负载伸缩生效。

- 仅按时间进行伸缩设置

  1.1 遵循“先触发先执行，同时触发根据规则优先顺序执行”，如：1>2>3>4>5。

  1.2 重复执行规则，若规则到期后，规则状态将失效并处于关闭状态；到期前将有告警通知，详情请参见 告警配置 。

- 按负载和时间混合进行伸缩设置

  遵循“先触发先执行，同时触发根据规则优先顺序执行”，如：1>2>3>4>5。

队列负载指标对应关系

负载类型	类别	维度	EMR 自动伸缩指标	指标含义
YARN	AvailableVCores	root	AvailableVCores#root	Root 队列可用虚拟核数的数量
		root.default	AvailableVCores#root.default	root.default 队列可用虚拟核数的数量
		自定义子队列	如：AvailableVCores#root.test	root.test 队列可用虚拟核数的数量
	PendingVCores	root	PendingVCores#root	Root 队列等待可用的虚拟核数

	root.default	PendingVCores#root.default	root.default 队列等待可用的虚拟核数
	自定义子队列	如：PendingVCores#root.test	root.test 队列等待可用的虚拟核数
AvailableMB	root	AvailableMB#root	Root 队列可用内存数量（MB）
	root.default	AvailableMB#root.default	root.default 队列可用内存数量（MB）
	自定义子队列	如：AvailableMB#root.test	root.test 队列可用内存数量（MB）
PendingMB	root	PendingMB#root	Root 队列等待可用的内存数量（MB）
	root.default	PendingMB#root.default	root.default 队列等待可用的内存数量（MB）
	自定义子队列	如：PendingMB#root.test	root.test 队列等待可用的内存数量（MB）
AvailableMemPercentage	集群	AvailableMemPercentage	剩余内存的百分比
ContainerPendingRatio	集群	ContainerPendingRatio	待分配的容器数与已分配的容器数的比率
AppsRunning	root	AppsRunning#root	Root 队列运行中的任务数
	root.default	AppsRunning#root.default	root.default 队列运行中的任务数
	自定义子队列	如：AppsRunning#root.test	root.test 队列运行中的任务数
AppsPending	root	AppsPending#root	Root 队列挂起的任务数
	root.default	AppsPending#root.default	root.default 队列挂起的任务数
	自定义子队列	如：AppsPending#root.test	root.test 队列挂起的任务数
PendingContainers	root	PendingContainers#root	Root 队列待分配的容器数
	root.default	PendingContainers#root.default	root.default 队列待分配的容器数
	自定义子队列	如：PendingContainers#root.test	root.test 队列待分配的容器数
AllocatedMB	root	AllocatedMB#root	Root 队列已分配的内存量
	root.default	AllocatedMB#root.default	root.default 队列已分配的内存量
	自定义子队列	如：AllocatedMB#root.test	root.test 队列已分配的内存量
AllocatedVCores	root	AllocatedVCores#root	Root 队列已分配的虚拟核数
	root.default	AllocatedVCores#root.default	root.default 队列已分配的虚拟核数
	自定义子队列	如：AllocatedVCores#root.test	root.test 队列已分配的虚拟核数
ReservedVCores	root	ReservedVCores#root	Root 队列预留的虚拟核数
	root.default	ReservedVCores#root.default	root.default 队列预留的虚拟核数
	自定义子队列	如：ReservedVCores#root.test	root.test 队列预留的虚拟核数
AllocatedContainers	root	AllocatedContainers#root	Root 队列已分配的容器数
	root.default	AllocatedContainers#root.default	root.default 队列已分配的容器数
	自定义子队列	如：AllocatedContainers#root.test	root.test 队列已分配的容器数

	ReservedMB	root	ReservedMB#root	Root 队列预留的内存量
		root.default	ReservedMB#root.default	root.default 队列预留的内存量
		自定义子队列	如：ReservedMB#root.test	root.test 队列预留的内存量
	AppsKilled	root	AppsKilled#root	Root 队列终止的任务数
		root.default	AppsKilled#root.default	root.default 队列终止的任务数
		自定义子队列	如：AppsKilled#root.test	root.test 队列终止的任务数
	AppsFailed	root	AppsFailed#root	Root 队列失败的任务数
		root.default	AppsFailed#root.default	root.default 队列失败的任务数
		自定义子队列	如：AppsFailed#root.test	root.test 队列失败的任务数
	AppsCompleted	root	AppsCompleted#root	Root 队列完成的任务数
		root.default	AppsCompleted#root.default	root.default 队列完成的任务数
		自定义子队列	如：AppsCompleted#root.test	root.test 队列完成的任务数
	AppsSubmitted	root	AppsSubmitted#root	Root 队列提交的任务数
		root.default	AppsSubmitted#root.default	root.default 队列提交的任务数
		自定义子队列	如：AppsSubmitted#root.test	root.test 队列提交的任务数
	AvailableVCoresPercentage	集群	AvailableVCoresPercentage	集群内可用虚拟核数百分比
	MemPendingRatio	root	MemPendingRatio#root	Root 队列等待可用的内存百分比
		root.default	MemPendingRatio#root.default	root.default 队列等待可用的内存百分比
		自定义子队列	如：MemPendingRatio#root.test	root.test 队列等待可用的内存百分比
	usageRatioVCores	root	usageRatioVCores#root	Root 队列 CPU 资源的使用率
		root.default	usageRatioVCores#root.default	root.default 队列 CPU 资源的使用率
		自定义子队列	如：usageRatioVCores#root.test	root.test 队列 CPU 资源的使用率
	usageRatioMem	root	usageRatioMem#root	Root 队列内存资源的使用率
		root.default	usageRatioMem#root.default	root.default 队列内存资源的使用率
		自定义子队列	如：usageRatioMem#root.test	root.test 队列内存资源的使用率
Trino	FreeDistributed	集群	FreeDistributed	集群可用 Distributed 内存
	QueuedQueries	集群	QueuedQueries	队列中等待执行的查询总数

伸缩规则设置实践教程
时间伸缩规则设置实践教程

最近更新时间：2024-08-23 14:52:21

根据业务可能在一定周期内的明显波峰和波谷，选择重复执行或者只执行一次的执行频率，配置扩容和缩容规则，选择重复执行时，通过配置规则有效期，可以设置规则生效的截止时间，超出有效期后不再触发伸缩规则。

> **例如：**
> 您的业务在每天的22点开始增加，凌晨6点开始减少，且预计会持续1个月，您可以配置按时间策略类型，设置两条伸缩规则（一扩一缩）或一条扩容规则（定时销毁）。

- **扩容规则：**按天重复执行，每天的22点配置扩容规则，持续一月。

- **缩容规则：**按天重复执行，每天的凌晨6点配置缩容规则，持续一月。

- **扩容规则+定时销毁：**按天重复执行，每天的22点配置扩容规则，该批次资源使用8小时（换算到每天的凌晨6点），持续一月重复执行支持：按“每天”、“每周”、“每月”请根据实际情况设置，其他规则配置项及使用介绍请参见 设置时间伸缩 。

注意：

1. 以上时间点补充资源到集群为理想状态，实际扩容资源耗时跟单次数量有关，建议结合需求情况时间规则设置提前5分钟以上。
2. 扩容在高峰下单时可能由于资源争抢，导致实际扩容机器数量达不到弹性目标数量，建议您扩容规则“开启资源补足重试策略”。
3. 缩容动作触发时可能节点正在执行任务，为避免节点不会被立即释放，建议您开启优雅缩容，详见优雅缩容。

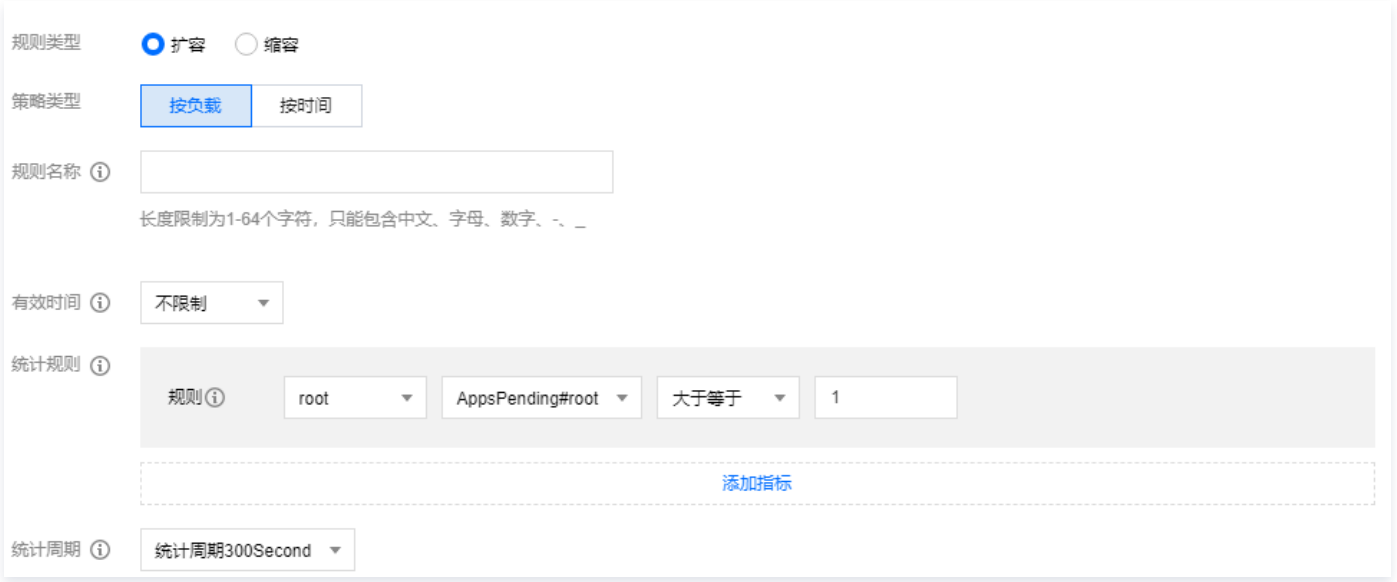
负载伸缩规则设置实践教程

最近更新时间：2024-08-23 14:52:21

根据集群 YARN的指标变化的情况，选择过去时间符合业务变化的指标，配置具体的阈值，然后保存并应用，在业务发生变化后，即会触发对应规则；指标的选择要与容量变化成反比，在伸缩活动发生后，实例数量的变化可以降低对应的指标。

> **例如：**
> 配置扩容规则，如果在300秒内 AppsPending#root 的平均值 >= 1，重复连续出现2次，则触发扩容动作，可以有效的减少队列中挂起的任务数。

- **扩容规则：**



- **缩容规则同理：**请根据实际情况设置，其他规则配置项及使用介绍详见：设置负载伸缩 。

1.1 每一条规则内，可以配置多条指标条件，当同时满足指标条件时，触发伸缩。

1.2 为了避免频繁的扩缩容导致资源浪费，可以为规则配置一定的冷却时间。在冷却时间内，即使满足伸缩条件也不会发生伸缩活动。

1.3 配置有效时间（当前规则在自定义时间范围内生效），可以组合不同的伸缩规则，在不同时间段配置不同内容的伸缩条件。

> **注意：**
> 1. 扩容在高峰下单时可能由于资源争抢，导致实际扩容机器数量达不到弹性目标数量，建议您扩容规则“开启资源补足重试策略”。
> 2. 缩容动作触发时可能节点正在执行任务，为避免节点不会被立即释放，建议您开启优雅缩容，详见优雅缩容 。

混合伸缩规则设置实践教程

最近更新时间：2024-08-23 14:52:21

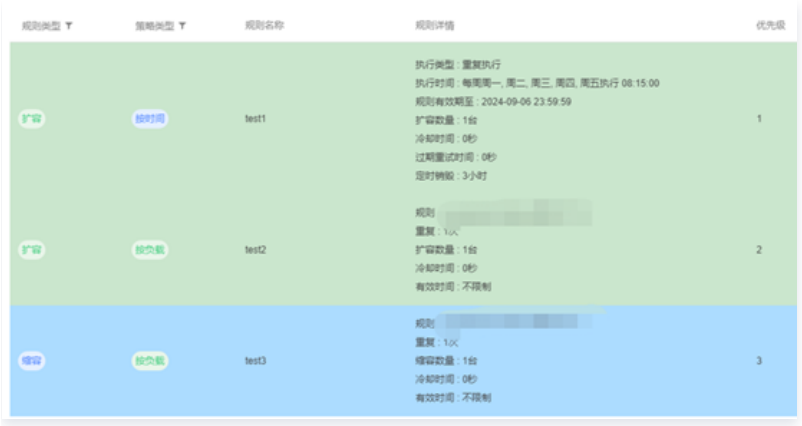
混合场景1：

业务存在一定周期内明显的波峰波谷，也存在非周期性的突然业务短暂高峰。

> **例如：**
> 每周工作日早上8:30需要进行固定业务量分析统计持续2小时，需要1台节点补充算力；其他时间存在突发业务情况所需算力无法确定；此时可配置三条伸缩规则达到保证业务算力需要和成本节约。

规则1：设置时间扩容规则，选择重复执行，按每周1/2/3/4/5，明天早上8:15进行扩容1台节点，定时销毁时长设置3小时。
规则2：设置负载扩容规则，根据所需监控指标进行选择，有效期建议不设置，默认全天，扩容1台。
规则3：设置负载缩容规则，根据所需监控指标进行选择，有效期建议不设置，默认全天，缩容1台。

> **注意：**
> 1. 扩容资源存在耗时，耗时与扩容数量成正比，建议提前15分钟准备资源；常规耗时较短。
> 2. 三条扩容规则优先级顺序为：规则1＞规则2＞规则3；扩缩数量可根据实际业务需要而定。



混合场景2：

业务存在明显的昼夜高低峰变化。

> **例如：**
> 每天早6:30和晚5:30为业务高峰，需要10台节点补充算力，业务高峰持续时长不定，其他时间范围所需算力小，需要1台节点；此时可配置三条伸缩规则达到保证业务算力需要和成本节约。

规则1：设置时间扩容规则，选择重复执行，按每天，明天早上6:15进行扩容10台节点。
规则2：设置时间扩容规则，选择重复执行，按每天，明天晚上5:15进行扩容10台节点。
规则3：设置负载缩容规则，根据所需监控指标进行选择，有效期建议不设置，默认全天，缩容9台。

版权所有：腾讯云计算（北京）有限责任公司

注意：

1. 基础设置最大节点数为10和最小节点数为1，避免资源浪费和无弹性资源从而导致算力不足；三条扩容规则优先级顺序为：规则1＞规则2＞规则3。
2. 扩容资源存在耗时，耗时与扩容数量成正比，建议提前15分钟准备资源；常规耗时较短。

在线容器集群部署 EMR

最近更新时间：2025-06-20 16:25:42

EMR 集群支持在离线运算需求高峰期将 EMR 服务部署到闲置或利用率低的在线容器集群，实现在线离线负载混合部署。这项能力目前为白名单功能，对象为具备在离线管控经验的客户，要求有非常成熟的在离线资源管理实践。
选择运行有在线业务的 TKE 集群对 EMR 集群进行扩容时，需要提前进行一系列隔离配置，以实现资源管控，按照业务优先级调度分配资源。腾讯云 EMR 支持对选中的 TKE 集群进行自动化前置配置，代替原有的手工配置过程，主要包括插件安装和权限配置。

前提条件

1. 已经创建 EMR 集群，集群类型为 EMR on CVM。
2. 已经创建 TKE 集群，且需要包含原生节点。

操作流程

1. 登录 腾讯云 EMR 控制台，在集群详情页 > 集群资源 > 资源管理，单击**扩容**。
2. 在扩容弹窗中，选择扩容的资源类型为“POD”。
3. 容器类型选择“TKE 集群”，在下拉列表中选择要运行离线任务的 TKE 集群。
4. 默认情况下的 TKE 集群都需要先进行前置配置，否则系统无法进行后续步骤。
5. 单击 TKE 集群下拉菜单下方的提示信息中的前往配置打开配置弹窗。
6. 用户选择 TKE 集群后，系统会对 TKE 集群进行预检查，并对检查项给出是否就绪的提示。
7. 用户需要进行以下两项操作：
   7.1 按照系统提示内容的对 CVM 和 TKE 的服务角色绑定预设策略。
   7.2 单击**开始安装** 后，启动将安装和配置任务，具体执行情况请前往 EMR > 任务中心查看。
8. 待任务中心中相应任务执行完成后，重复以上步骤2，则可以正常选择 TKE 集群，进行后续的扩容配置。

预设策略说明

需要对 TKE_QcsRole 和 CVM_QcsRole 授予访问 EMR 状态同步接口 syncstate 的权限，可以通过绑定预设策略和绑定自定义策略的方式实现。
系统建议通过对 TKE_QcsRole 和 CVM_QcsRole 授予系统预设策略的方式实现：
预设策略名称：QcloudAccessForEMRLinkedRoleInHybridDeploy