

注册配置治理 迁移指南



腾讯云

【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】



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

迁移指南

- Zookeeper 迁移方案

 - Zookeeper 冷迁移方案

 - Zookeeper 热迁移方案

- Nacos 迁移方案

- Eureka 迁移方案

迁移指南

Zookeeper 迁移方案

Zookeeper 冷迁移方案

最近更新时间：2024-09-20 14:21:21

适用场景

将业务应用从自建 Zookeeper 冷迁到 TSE Zookeeper，在迁移过程中，新部署的服务暂不可用，存量服务不受影响。如果您希望线上运行的业务在迁移时不受影响，请参见 [Zookeeper 平滑热迁移方案](#)。

迁移步骤

步骤1：持久化数据迁移

创建包含事务日志和快照日志的压缩包

1. 首先，对自建 Zookeeper 集群的所有节点执行以下命令：，来获取自建 Zookeeper 集群 Leader 节点的 IP。

```
echo srvr | nc [Zookeeper 节点 IP] [Zookeeper 节点端口，默认为2181] | grep Mode
```

2. 然后，根据 Leader 节点的 zoo.cfg 配置获取其事务日志和快照日志所对应的存储路径。在 zoo.cfg 配置文件中， dataDir 表示快照日志对应的本地存储路径； dataLogDir 表示事务日志对应的本地存储路径。

请访问 [Github 地址](#)，下载工具 JAR 包，并上传至源集群所在的合适目录下。执行以下命令生成快照文件压缩包：

```
java -DconfigPath=*/conf/zoo.cfg -jar tar-zookeeper-1.0.0-Beta.jar -n 1
```

说明：

* 代表源集群 Leader 节点所在的路径。

3. 生成压缩包后，请将文件下载至本地。如果您的源集群部署在 CVM 中，可以通过以下命令下载文件至本地：

```
sz -e data.tar.gz
```

上传快照压缩包

事务日志和快照日志的组合包含了 Zookeeper 节点数据的全部信息。通过日志文件，可以快速的将自建 Zookeeper 的数据迁移到 TSE Zookeeper 中。

通过目标 TSE ZK 引擎详情 - 迁移方案 - 冷迁移，上传以上导出的压缩文件。

导入快照（可选）

data.tar.gz



删除

请上传 snapshot.* 的 tar.gz 格式压缩文件，大小800MB以内

导入至目标集群



导入完成

点击完成并重置集群



步骤2：新部署的业务应用使用 TSE Zookeeper

将新部署的业务应用使用 TSE Zookeeper，详情查看 [以 Spring cloud 应用为例](#)。

步骤3：新部署的业务应用上线

验证新部署的业务应用运行是否正常，若能够正常运行，则将请求切换到新部署的业务应用，并下线存量的业务应用。

Zookeeper 热迁移方案

最近更新时间：2023-11-01 17:40:41

操作场景

当您在生产环境上已经使用了自建的 Zookeeper 集群，并希望将其上已运行的服务迁移至腾讯云的 TSE Zookeeper 时，可以通过本指引进行生产环境的平滑热迁移，以保证在迁移过程中不影响现网服务的可用性和稳定性。

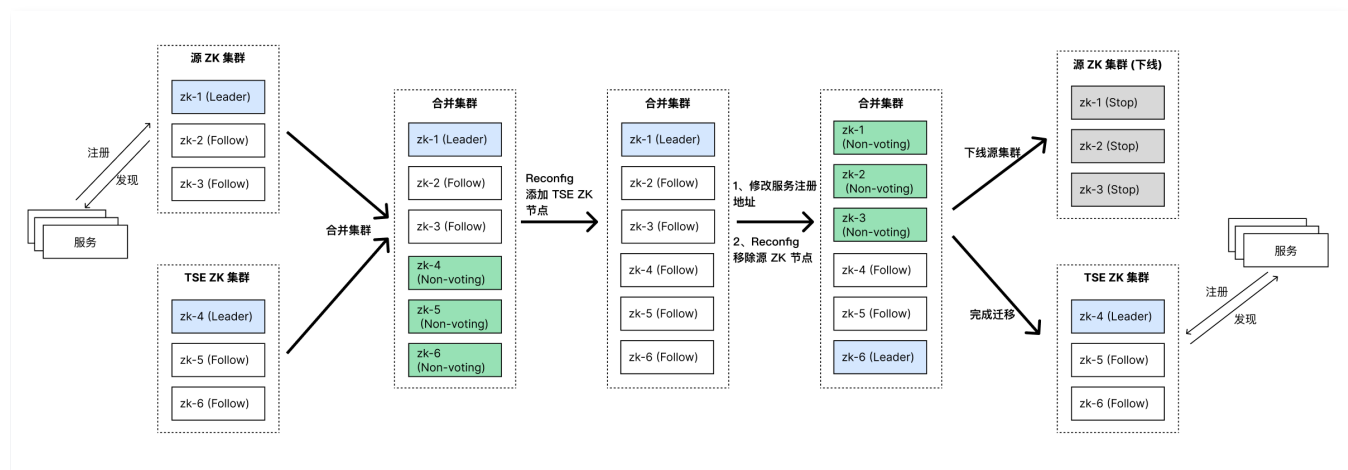
本文主要介绍 TSE Zookeeper 通过动态重新配置（Dynamic Reconfig）机制实现平滑迁移的方案和操作步骤，帮助您将自建 Zookeeper 集群迁移至 TSE Zookeeper 集群。

操作原理

Zookeeper 3.5.0 版本增加了动态重新配置（Dynamic Reconfig）功能，支持 Zookeeper 在运行过程中修改节点角色并立刻生效。

本迁移方案利用了动态重新配置机制：

1. 首先通过 Reconfig 操作实现源集群的扩容：将 TSE Zookeeper 的节点作为 Non-voting-follower 角色合并至您的源 Zookeeper 集群，并在数据同步完成后将其动态配置为 Participant。
2. 之后，在新组成的集群中，将源集群上运行的服务滚动修改注册至 TSE Zookeeper 节点。
3. 最后，再次通过 Reconfig 操作实现缩容：将源集群中的节点动态配置为 Non-voting-follower 角色，并逐渐将源集群下线，完成迁移至 TSE Zookeeper。



前提条件

1. 请确保您的源 Zookeeper 集群版本为 3.5.0 以上。如果您的源集群当前版本低，请进行版本升级后再进行迁移操作。
2. 请获取并记录您的源 Zookeeper 集群信息。您可以在 `zoo.cfg` 文件中查看到源集群信息，请获取源集群各节点的 IP、客户端连接端口、数据同步端口、选举端口。如以下代码中所展示的，数据同步端口为 3888，选举端口为 2888，客户端连接端口为 2181。

说明：

若 `server.id` 中客户端参数项没有配置，则以 `clientPort` 配置的客户端参数为准。

```
server.1=10.0.0.12:3888:2888;2181
server.2=10.0.0.5:3888:2888;2181
server.3=10.0.0.15:3888:2888;2181
```

3. 登录 [TSE 控制台](#)，在相同 VPC 或相通的网络环境下，新建一个 Zookeeper 实例，并获取客户端访问地址。详细操作请参见 [引擎管理](#)。请确保您的 TSE Zookeeper 大版本与源集群大版本一致，如 3.5.X。

说明：
如果您是云梯用户，无需区分 VPC，但如果自建集群端口非标准 2181、2888、3888 请联系TSE团队或 [快速提工单](#)，以解除 TSE Zookeeper 对非标端口的限制。

● 连通性测试

请在下方提供源集群中所有节点的IP与端口，并确认通过连通性测试。迁移任务需通过内网VPC建立连接，请检查您的网络配置与IP地址。

源IP	数据同步端口	选举接口	客户端连接端口	操作
10.0.0.12	2888	3888	2181	×
10.0.0.5	2888	3888	2181	×
10.0.0.15	2888	3888	2181	×
添加节点				

连通性测试

✓ 连接成功

3. 设置超级管理员

执行 Reconfig 操作需要超级管理员权限，需要获得源集群超级管理员的用户名和密码。

如果您的源集群当前未配置超级管理员，可以根据本步骤生成的 token，在下一步骤中进行配置；如果您的源集群当前已配置超级管理员，请检查是否与本步骤生产的 token 一致，如不一致请更改为一致，源集群超级管理员配置需要与 TSE 集群中的保持一致。

● 源集群管理员账号

管理员账号

管理员密码

Token admin:x1nq8J5GOJVPY6zgztTtA9izLc=

生成Token

4. 开启源集群 Reconfig 功能

如果您的源集群未开启 Reconfig 功能，请添加以下配置进行开启，后续的迁移步骤依赖于使用 Reconfig 机制；如果您的源集群已开启 Reconfig 功能，请参照以下配置进行检查。

JVM 相关参数配置

在原生 Zookeeper 集群中，可以通过 `zoo.cfg` 同级目录下创建 `java.env` 文件来引入 JVM 参数，Zookeeper 集群在启动时将会自动加载此目录下的内容。在 `java.env` 中我们需要填入如下内容：

● 开启源集群Reconfig功能

为了保证迁移的顺利进行，请确认您的源ZK集群已开启Reconfig功能，如未开启，可以参照以下指引进行配置：

1. 添加 JVM 启动参数

在原生zk集群中，您可以通过在 `zoo.cfg` 同级目录下创建 `java.env` 文件来引入 JVM 参数。在 `java.env` 中填入如下内容：

```
export JVMFLAGS="-Dzookeeper.DigestAuthenticationProvider.superDigest=admin:x1nq8J5GOJVPY6zgztTtA9izLc=$JVMFLAGS" #
```

或者，您也可以选择在 JVM 启动参数中添加以下内容：

```
-Dzookeeper.DigestAuthenticationProvider.superDigest=[digest占位符]
```

! 说明：

[digest 占位符] 为设置超级管理员时生成的 token。

zoo.cfg 相关参数配置

```
4lw.commands.whitelist=*
```

```
quorumListenOnAllIPs=true
reconfigEnabled=true
```

在 `zoo.cfg` 中添加以上参数。其中：

- `4lw.commands.whitelist=*` 表示开启四字命令。TSE Zookeeper 的迁移任务会使用四字命令来获取源集群各节点的状态。
- `quorumListenOnAllIPs=true` 表示 ZooKeeper 服务器将在所有可用 IP 地址上监听连接，而不仅是在配置文件的服务器列表中配置的地址。TSE Zookeeper 的实例会通过非配置的地址连接上源集群。
- `reconfigEnabled=true` 表示开启 Reconfig 功能。

❗ 说明：

如果您计划在迁移中依赖 ZK 原生的日志同步，可以增加 `zoo.cfg` 中 `syncLimit` 和 `initLimit` 时间，推荐为 30。防止数据量大导致 TSE 集群加入之后的同步过程超过预设时间。

步骤2：导入快照（可选）

如果您的源集群中的持久化数据量较大，建议通过导入包含事务日志和快照日志的压缩包的方式，将源集群中的数据提前迁移至 TSE 集群中。

1. 确认源集群 Leader 节点

由于导入的数据最好为最新版本，建议您选取源集群 Leader 节点中的数据进行导出。您可以在 TSE Zookeeper 迁移任务界面查看到源集群 Leader 节点的 IP。

节点ID	节点IP	客户端端口	角色	事务ID	事务ID差值
1	10.0.0.12	2181	follower	0xb00000007	-
2	10.0.0.5	2181	follower	0xb00000007	-
3	10.0.0.15	2181	leader	0xb00000007	-

2. 制作快照压缩包

TSE Zookeeper 迁移任务支持导入的快照文件的格式为，名称为：`data.tar.gz` 的压缩包。

压缩包可以包含节点完整数据的 `snap` 和 `log` 日志文件，或者仅包含 `snap` 的文件。

⚠ 注意：

如果 `data.tar.gz` 导入的数据不全，可能会导致集群合并失败。因此，TSE 提供并且建议您使用 TSE ZK 快照导出工具来生成标准的压缩包。

请访问 [Github 地址](#)，下载工具 JAR 包，并上传至源集群所在的合适目录下。执行以下命令生成快照文件压缩包：

```
java -DconfigPath=*/conf/zoo.cfg -jar tar-zookeeper-1.0.0-Beta.jar -n 1
```

❗ 说明：

* 代表源集群 Leader 节点所在的路径。

生成压缩包后，请将文件下载至本地。如果您的源集群部署在 CVM 中，可以通过以下命令下载文件至本地：

```
sz -e data.tar.gz
```

3. 上传快照压缩包

在 TSE ZK 迁移任务中，上传导出的快照压缩包。

导入快照（可选）

data.tar.gz

删除

请上传 snapshot.* 的 tar.gz 格式压缩文件，大小800MB以内

导入至目标集群

导入完成

步骤3：实施迁移

进入迁移任务的第三阶段时，可以观察到源 ZK 集群与 TSE ZK 集群分别存在一个 Leader 节点，并且两个集群中的节点 ID（ZK 节点的 myid）分别为从 1 递增的数字。因为此时合并未开启，两个集群独立存在，如下图中所示：

源ZK集群

未连接

TSE ZK集群

节点ID	节点IP	客户端端口	角色	事务ID	事务ID基值
1	10.0.0.12	2181	follower	0xb00000017	-
2	10.0.0.5	2181	follower	0xb00000017	-
3	10.0.0.15	2181	leader	0xb00000017	-

节点ID	节点名称	节点IP	客户端端口	角色	事务ID	事务ID基值	运行状态
1	zookeeper-0	10.0.0.112	2181	follower	0xb0	-	Running
2	zookeeper-1	10.0.0.36	2181	follower	0xb100000000	-	Running
3	zookeeper-2	10.0.0.34	2181	leader	0xb100000000	-	Running

1. 合并集群

当您准备好开启迁移时，可以点击页面下方的合并集群按钮，TSE ZK 集群中的节点将会重启并以 Non-voting-follower 的角色加入至源 ZK 集群中。

说明：

合并集群过程将持续10~20分钟，如果合并过程中遇到问题，可以 [快速提工单](#) 以获取帮助。

源ZK集群

合并成功

TSE ZK集群

节点ID	节点IP	客户端端口	角色	事务ID	事务ID基值	节点ID	节点名称	节点IP	客户端端口	角色	事务ID	事务ID基值	运行状态
1	10.0.0.12	2181	follower	0xb0000001a	-	4	zookeeper-0	10.0.0.112	2181	non-voting follower	0xb0000001a	-	Running
2	10.0.0.5	2181	follower	0xb0000001a	-	5	zookeeper-1	10.0.0.36	2181	non-voting follower	0xb0000001a	-	Running
3	10.0.0.15	2181	leader	0xb0000001a	-	6	zookeeper-2	10.0.0.34	2181	non-voting follower	0xb0000001a	-	Running

当合并完成后，可以观察到合并后的集群中只存在一个 Leader 节点并且来自于源 ZK 集群。TSE ZK 集群中的节点 ID 变为了延续源集群中节点 ID 的递增数字，TSE ZK 集群中的节点当前全部处于 Non-voting-follower 角色，并且在数据同步完成后，TSE ZK 集群节点的事务 ID 与源 ZK 集群保持一致。

2. 查看集群信息

在集群成功合并后，TSE ZK 节点已经以 Non-voting-follower 的角色加入了源集群，现在需要通过 Reconfig 操作将 TSE ZK 节点的角色动态配置为 Follower 角色。

首先，通过 zkCli 连接至任意源集群中的节点：

```
zkCli.sh -server [源节点 IP]:[客户端连接端口]
```

此时可以在 zkCli 中执行 config 命令，查看当前集群的组成。通过以下结果可以观察到，TSE 集群节点虽然已经变成 Follower 加入到集群了，但是目前还只是 Non-voting-follower 节点，我们需要通过 Reconfig 功能，将其正式加入到集群中。

```
[zk: 10.0.0.12:2181(CONNECTED) 0] config
```

```
server.1=10.0.0.12:2888:3888:participant;0.0.0.0:2181
server.2=10.0.0.5:2888:3888:participant;0.0.0.0:2181
server.3=10.0.0.15:2888:3888:participant;0.0.0.0:2181
version=a00000000
```

3. 输入超级管理员权限

由于 Reconfig 功能需要管理员才能操作，这里我们需要通过 `addAuth` 命令完成管理员身份认证。管理员的用户名密码为步骤1中设置的用户名密码。

```
addauth digest [用户名]:[密码]
```

4. 滚动加入新节点

TSE 集群的选举端口、数据同步端口、客户端端口默认分别是：3888、2888、2181。因此，`reconfig add` 命令有以下两种可选方式将 TSE 节点加入到集群中，您可以根据自己的业务权衡：

警告：

请注意务必以节点 ID 从小到大的顺序添加新节点，并在实施迁移过程中保持集群中的 `server.ID` 与控制台中展示的节点 ID 一致。

方式一：以 participant 加入到源zk集群中

```
reconfig -add [节点id]=[ip]:2888:3888:participant;2181
```

方式二：将节点以 observer 加入到源zk集群中

```
reconfig -add [节点id]=[ip]:2888:3888:observer;2181
```

说明：

如果您选择以方式二将 TSE ZK 集群加入到源集群中，那么在移除源集群节点前，需要使用方式一将 TSE ZK 集群节点转为 follower 加入到集群中来。

这里直接以方式一作为示例：

```
[zk: 10.0.0.12:21811(CONNECTED) 5] reconfig -add 4=10.0.0.28:2888:3888:participant;2181
[zk: 10.0.0.12:21811(CONNECTED) 6] reconfig -add 5=10.0.0.49:2888:3888:participant;2181
[zk: 10.0.0.12:21811(CONNECTED) 7] reconfig -add 6=10.0.0.19:2888:3888:participant;2181
Committed new configuration:
server.1=10.0.0.12:2888:3888:participant;0.0.0.0:2181
server.2=10.0.0.5:2888:3888:participant;0.0.0.0:2181
server.3=10.0.0.15:2888:3888:participant;0.0.0.0:2181
server.4=10.0.0.112:2888:3888:participant;0.0.0.0:2181
server.5=10.0.0.36:2888:3888:participant;0.0.0.0:2181
server.6=10.0.0.34:2888:3888:participant;0.0.0.0:2181
version=a00000019
```

以上操作将 TSE ZK 集群中的三个节点通过 Reconfig 作为 follower 加入了合并集群，可以通过最新的 config 内容以及控制台上增加的事务 ID 观察到 Reconfig 已经生效。

源ZK集群

合并成功

TSE ZK集群

节点ID	节点IP	客户端端口	角色	事务ID	事务ID基值
1	10.0.0.12	2181	follower	0xb0000001e	-
2	10.0.0.5	2181	follower	0xb0000001e	-
3	10.0.0.15	2181	leader	0xb0000001e	-

节点ID	节点名称	节点IP	客户端端口	角色	事务ID	事务ID基值	运行状态
4	zookeeper-0	10.0.0.112	2181	follower	0xb0000001e	-	Running
5	zookeeper-1	10.0.0.36	2181	follower	0xb0000001e	-	Running
6	zookeeper-2	10.0.0.34	2181	follower	0xb0000001e	-	Running

5. 迁移服务

在开启迁移服务前，请先确认数据同步已完成，这里可以通过两种方式进行观察确认。

- 方式一：TSE ZK 集群节点的事务 ID 在合并集群中是否已占据多数。
- 方式二：TSE ZK 控制台 - 数据管理页面中查看是否源集群中的数据已经完成迁移。

通过以上两种方式确认数据同步完成后，可以依次重启在源集群之上运行的服务，并将其注册地址改为 TSE ZK 的客户端访问地址。

6. 滚动移除旧节点

在完成以上全部步骤后，可以通过 `reconfig -remove` 命令将源集群节点从合并集群中移除：

```
reconfig -remove [节点id]
```

⚠ 注意：

在移除源集群节点的过程中，为了集群的稳定性，请最后一个移除 Leader 节点。

在本示例中，server.3 是 Leader 节点，因此将以1、2、3的顺序移除旧节点：

```
[zk: 10.0.0.12:2181(CONNECTED) 8] reconfig -remove 1
[zk: 10.0.0.12:2181(CONNECTED) 9] reconfig -remove 2
[zk: 10.0.0.12:2181(CONNECTED) 10] reconfig -remove 3
Committed new configuration:
server.4=10.0.0.112:2888:3888:participant;0.0.0.0:2181
server.5=10.0.0.36:2888:3888:participant;0.0.0.0:2181
server.6=10.0.0.34:2888:3888:participant;0.0.0.0:2181
version=a00000022
```

由于 Leader 节点的移除，集群会发生选举，并存在秒级的断连。

我们可以观察到源集群里面的节点都转为 Follower，TSE ZK 集群里面的节点出现了 Leader 节点。

源ZK集群						合并成功								TSE ZK集群			
节点ID	节点IP	客户端端口	角色	事务ID	事务ID基值	节点ID	节点名称	节点IP	客户端端口	角色	事务ID	事务ID基值	运行状态				
1	10.0.0.12	2181	non-voting follower	0xb00000021	4294967263	1	zookeeper-0	10.0.0.112	2181	leader	0xc00000000	-	Running				
2	10.0.0.5	2181	non-voting follower	0xb00000021	4294967263	5	zookeeper-1	10.0.0.36	2181	follower	0xb00000021	4294967263	Running				
3	10.0.0.15	2181	non-voting follower	0xb00000021	4294967263	6	zookeeper-2	10.0.0.34	2181	follower	0xb00000021	4294967263	Running				

7. 完成迁移任务

在观察业务正常后，可以逐渐退出源集群节点的进程，并在完成后点击完成迁移按钮。

ⓘ 说明：

引擎成功完成迁移后，将无法再次开启迁移任务。

Nacos 迁移方案

最近更新时间：2025-08-15 10:02:32

操作场景

当您在生产环境中已经使用了自建的 Nacos 集群，并希望将其上已运行的服务迁移至腾讯云的 TSE Nacos 时，可以根据您的业务需求，在本指引中选择适合您的迁移方案。

本文主要介绍 TSE Nacos 冷迁移与热迁移的方案和操作步骤，帮助您将自建 Nacos 集群迁移到 TSE Nacos 集群。

迁移原理

Nacos 的核心功能为服务注册发现与配置管理，故一次完整的迁移任务涉及以下两方面：

1. 配置数据迁移：Nacos 的配置数据、用户信息、命名空间数据、访问控制等信息在数据库中持久化存储，如果您的自建集群中已经存在配置数据，需要首先将自建集群中的持久化数据导入至 TSE Nacos。您可以选择以下方式中的一种进行配置数据迁移：
 - 配置数据冷迁移：可以通过 Nacos 的配置数据导入功能，在迁移之初导入自建集群的持久化数据。
 - 配置数据热迁移：可以通过 TSE Nacos 的数据同步功能实现源数据库到 TSE Nacos 数据库的全量迁移与实时增量同步。
2. 服务注册发现迁移：在服务注册发现迁移时，需要切换客户端的注册配置中心地址。您可以选择以下方式进行服务注册发现迁移：
 - 服务冷迁移：如果您的业务可以进行停机迁移，可以直接将服务注册配置中心地址修改为 TSE Nacos。

前提条件

1. 登录 [TSE 控制台](#)，新建一个 Nacos 实例，并获取客户端访问地址。详细操作请参见 [引擎管理](#)。

客户端访问地址

内网地址

私有网络

子网

内网地址

10.0.0.20:9848

公网地址 ① · 开启

操作步骤

步骤1：配置数据迁移

方式一：配置数据冷迁移

1. 创建命名空间与权限控制信息：在开始导入配置数据前，前往 TSE Nacos 原生控制台（访问方式请参见 [访问控制](#)），并根据您的业务需求，创建与自建集群相应的命名空间与权限控制信息。
2. 导出自建集群配置数据：前往自建 Nacos 集群控制台，在配置列表中依次导出各个命名空间的配置数据。
3. 配置数据导入 TSE Nacos：再次前往 TSE Nacos 原生控制台，在配置管理 - 配置列表中依次导入各个命名空间的配置数据。

方式二：配置数据热迁移

通过 TSE Nacos 的数据同步功能，将源 Nacos 数据库导入与同步至 TSE Nacos 集群。请参见 [Nacos 数据同步](#)。

步骤2：服务注册发现迁移

1. 将当前注册至自建 Nacos 集群中的服务的注册地址改为 TSE Nacos 的客户端访问地址，并依次重新部署。

❗ 说明：

如果您使用了域名访问，则无需重新部署服务，只需要更新域名对应的 Nacos 客户端访问 IP 即可。

2. 部署成功后，在 TSE Nacos 原生控制台的服务管理页面可以看到注册的服务。Nacos 原生控制台的访问方式请参见 [访问控制](#)。此时服务仅在 TSE Nacos 集群中进行了注册。
3. 持续观察一段时间，依次验证注册、发现、反注册，如果一切均正常，则可以下线自建的 Nacos 集群。如果您采用了配置数据热迁移，则可以终止数据同步任务。

Eureka 迁移方案

最近更新时间：2024-01-11 15:30:01

操作场景

您在生产环境上已经使用了自建的 Eureka 集群，希望将其上已运行的服务迁移至腾讯云的 TSE 注册中心。TSE 提供 PolarisMesh（北极星）完全兼容 Eureka 注册中心。在功能完全对齐的基础上，TSE 北极星在性能上相比开源 Eureka 有突破性提升。

本文将通过一个 Demo 应用，通过 `spring-cloud-eureka-client` 接入 TSE PolarisMesh 的全流程操作演示，帮助您快速了解如何实现 Eureka 迁移。

前提条件

- 已创建 PolarisMesh 北极星网格，请参见 [创建 PolarisMesh 治理中心](#)。
- 下载 Github 的 [demo 源码](#) 到本地并解压。
- 本地编译构建打包机器环境已安装了 Java JDK、Maven，并且能够访问 Maven 中央库。
- 根据您的业务，已准备好业务部署的资源，选择 [虚拟机部署](#) 或 [容器化部署](#) 其中一种方式即可：
 - [虚拟机部署](#)已创建 CVM 虚拟机，请参见 [创建 CVM 虚拟机](#)。
 - [容器化部署](#)已创建 TKE 容器集群，请参见 [创建 TKE 集群](#)。

操作步骤

- 登录 [TSE 控制台](#)。
- 在北极星网格下的 `polarismesh` 页面，单击页面左下方下拉列表，选择目标地域。
- 单击目标引擎的“ID”，进入实例信息页面。
- 查看访问地址，eureka-client 应用访问使用 eureka 端口（8761）：

Polaris Server访问地址

客户端接入端口	注册发现	8091 (service-grpc)	8092 (service-trpc)
	配置管理	8093 (config-grpc)	
	监控上报	9091 (pushgateway)	
OpenAPI 端口	8090 (http)		
协议兼容端口	7779 (i5)	8761 (eureka)	15010 (xdsv3) 15020 (xdsv2)

内网负载均衡 ①

私有网络	子网	内网地址	访问控制	操作
		10.0.0.133		删除

添加内网负载均衡

5. 修改 demo 中的注册中心地址

- 在下载到本地的 [Demo 源码目录](#) 下，分别找到 `eureka/eureka-java/consumer/src/main/resources/application.yml` 和 `eureka/eureka-java/provider/src/main/resources/application.yml` 两个文件。
- 添加微服务引擎北极星网格地址到项目配置文件中（以 `eureka/eureka-java/consumer/src/main/resources/application.yml` 为例）。

```
eureka:
  client:
    serviceUrl:
      defaultZone: http://10.0.0.133:8761/eureka/
```

6. 打包 Demo 源码成 jar 包

6.1 在 `eureka-java` 源码根目录下，打开 cmd 命令，执行 `mvn clean package` 命令，对项目进行打包编译。

6.2 编译成功后，生成如下表所示的2个 Jar 包。

软件包所在目录	软件包名称	说明
<code>\eureka-java\provider\target</code>	<code>eureka-provider-\${version}-SNAPSHOT.jar</code>	服务生产者
<code>\eureka-java\consumer\target</code>	<code>eureka-consumer-\${version}-SNAPSHOT.jar</code>	服务消费者

7. 部署 provider 和 consumer 微服务应用，虚拟机部署方式、容器化部署或者本地运行，根据您的业务实际的部署方式选择一种即可。

7.1 虚拟机部署方式部署 provider 和 consumer 微服务应用。

- 上传 Jar 包至 CVM 实例。
- 执行启动命令进行启动：

```
nohup java -jar [jar包名称] &
```

7.2 容器化部署方式部署 provider 和 consumer 微服务应用。

- 编写 dockerfile 生成镜像，参考：

```
FROM java:8
ADD ./eureka-provider-${VERSION}.jar /root/app.jar
ENTRYPOINT ["java","-jar","/root/app.jar"]
```

- 通过 TKE 部署并运行镜像。

8. 确认服务注册结果。

8.1 进入前面提到的微北极星网格实例页面。

- 选择**服务管理 > 服务列表**，查看微服务 EUREKA-CONSUMER-SERVICE 和 EUREKA-PROVIDER-SERVICE 的实例数量：
- 若实例数量值不为0，则表示已经成功接入微服务引擎。
- 若实例数量为0，或者找不到具体服务的名称，则表示微服务应用接入微服务引擎失败。

服务列表							
新建 删除		请选择条件进行过滤 Q ☆ ↻					
☐ 服务名	命名空间	部门	业务	健康实例/总实例	创建时间	修改时间	操作
▶ ☐ EUREKA-CONSUMER-SERVICE	default	-	-	2/2	2022-01-10 23:04:02	2022-01-10 23:04:02	编辑 删除
▶ ☐ EUREKA-PROVIDER-SERVICE	default	-	-	2/2	2022-01-10 23:01:19	2022-01-10 23:01:19	编辑 删除

8.2 调用 consumer 的 HTTP 接口。执行 http 调用，其中 `${app.port}` 替换为 consumer 的监听端口（默认为20002），`${add.address}` 则替换为 consumer 暴露的地址。

```
curl -L -X GET 'http://${add.address}:${app.port}/echo?value=hello_world'
预期返回值：echo: hello_world
```