ICE 智能资源配置引擎

系统安装部署手册（集群）

INSTALLATION GUIDE

目录

1. 目的 4
2. 使用范围 4
3. 集群部署清单 4
4. 集群部署 5
4.1. 集群环境就绪 5
4.2. MongoDB 集群部署步骤 5
4.2.1. 系统内置任务部署 6
4.2.2. 手工部署 10
4.2.3. 鉴权设置 12
4.2.4. 修改 ICE 中的 MongoDB 配置 13
4.2.5. 修改 MongoDB 密码 15
4.2.6. 扩容 MongoDB 节点 18
4.1. Postgresql+PgPool-II+VIP 集群（高可用）部署步骤 19
4.1.1. 服务器基础配置 19
4.1.2. Postgresql13 安装 20
4.1.3. Postgresql13 配置 20
4.1.4. Pgpool-II 安装 22
4.1.5. Pgpool-II 配置 22
4.1.6. Pgpool-II 启动 25
4.1.7. 启用主从模式 25
4.1.8. 修改 ICE 中的 Postgresql 连接 26
4.1.9. pg+pgpool 启停 27
4.1.10. 修改 PostgreSQL 密码 28
4.2. Flink 集群部署步骤 31
4.2.1. 服务器互信免密登陆 31
4.2.2. 上传 Flink 安装包 32
4.2.3. 配置集群 32
4.2.4. 启动 Flink 32
4.2.5. 上传 ice-streaming 32
4.2.6. 启动 ice-streaming 33
4.3. Pulsar 集群部署步骤 33
4.3.1. Zookeeper 集群 33
4.3.2. Bookkeeper 集群 35
4.3.3. broker 集群 37
4.3.4. 初始化集群设置 38
4.3.5. 修改 ICE 中的 pulsar 连接配置 38
4.4. Nginx+Tomcat 集群部署步骤 39
4.4.1. Tomcat 多节点部署 mainServer 40
4.4.2. Nginx 部署 40
4.5. Redis 集群部署步骤 41
4.5.1. 上传 Redis 安装包 41

4.5.2. 修改 Redis 节点 IP 配置 41
4.5.3. 启动 Redis 节点 43
4.5.4. 启动 Redis Cluster 43
4.5.5. 修改 ICE 中的 Redis 连接 43
4.6. 集群自动部署程序 44
4.6.1. 集群自动部署准备事项 44
4.6.2. 设置主配置文件 45
4.6.3. 集群自动部署步骤 47
1.1. Subagent 和 sa 配置说明 50
1.1.1. 根据现场部署情况，规划 subagent 部署策略 50
1.1.2. subagent 具体配置 50
1.1.3. sa 具体配置 50

1. 目的

本文档详细描述了 ICE 智能资源配置引擎的安装步骤，以帮助用户全面了解和管理 ICE 系统的安装部署过程。

2. 使用范围

- 系统最终用户
- 系统管理用户

3. 集群部署清单

主机 IP	集群角色	安装服务
10.230.210.133	数据库	mongodb
10.230.211.22	数据库	mongodb
10.230.212.176	数据库	mongodb
10.230.201.168	后端服务	redis、pulsar、flink、mainserver、datacenter、sa、openapi
10.230.212.133	后端服务	redis、pulsar、flink、mainserver、datacenter、sa、openapi
10.230.211.61	后端服务	redis、pulsar、flink、mainserver、datacenter、sa、openapi
10.230.212.171	前端服务	nginx
10.230.212.114	前端服务	nginx
10.230.212.35	采集服务	subagent
10.230.212.126	采集服务	subagent

4. 集群部署

4.1. 集群环境就绪

- 禁用 SELinux：
  - vi /etc/sysconfig/selinux
  - 设置 SELINUX=disabled
  - 重启服务器
  - sestatus 查看是否禁用成功
- 安装 jdk：建议采用 rpm 方式直接安装
  - rpm -ivh jdk-8u231-linux-x64.rpm
  - java -vsrion 查看是否安装成功

4.2. MongoDB 集群部署步骤

ICE 的 MongoDB 采用副本集的方式组建集群。

MongoDB 副本集群模式下，集群节点分为三种角色：主节点（Primary）、从节点（Secondary）、监听节点（选举主节点）。

副本集群没有固定的“主节点”，由“监听节点”负责选出一个“主节点”。集群自动实现故障切换，当“主节点”挂掉后，“监听节点”在剩下的从节点中选择其他节点成为“主节点”。正常运行模式下，同一时刻，副本集总有一个活跃点（Primary）和一个或多个备份节点（Secondary）。

MongoDB 副本集群在工作过程中采用主从复制、读写分离的机制：

- 写入：只通过“主节点”写入数据；
- 读取：优先通过“从节点”读取数据。当“从节点”处理负载过高时，再通过“主节点”读取数据；
- 复制：“从节点”以拉取的方式自动从“主节点”同步最新数据变化。

ICE MongoDB 集群环境初次部署时，建议至少提供 3 台主机：

- 主节点：Linux7.x/2 核/8G 内存/存储空间按实际规划的资源纳管数量级估算，初始建议 500G
- 从节点：Linux7.x/2 核/8G 内存/存储空间按实际规划的资源纳管数量级估算，初始建议 500G

- 监听节点：Linux7.x/2 核/8G 内存/200G 磁盘

初始化部署 ICE MongoDB 集群分为 2 种方式：**系统内置任务部署**和**手工部署。**

4.2.1. 系统内置任务部署

用户可选择使用系统内置的“MongoDB 集群自动部署”任务，完成 MongoDB 集群的自动部署。此外，可使用系统内置的“MongoDB 集群自动添加节点”任务，在需要时添加新节点到 MongoDB 集群。

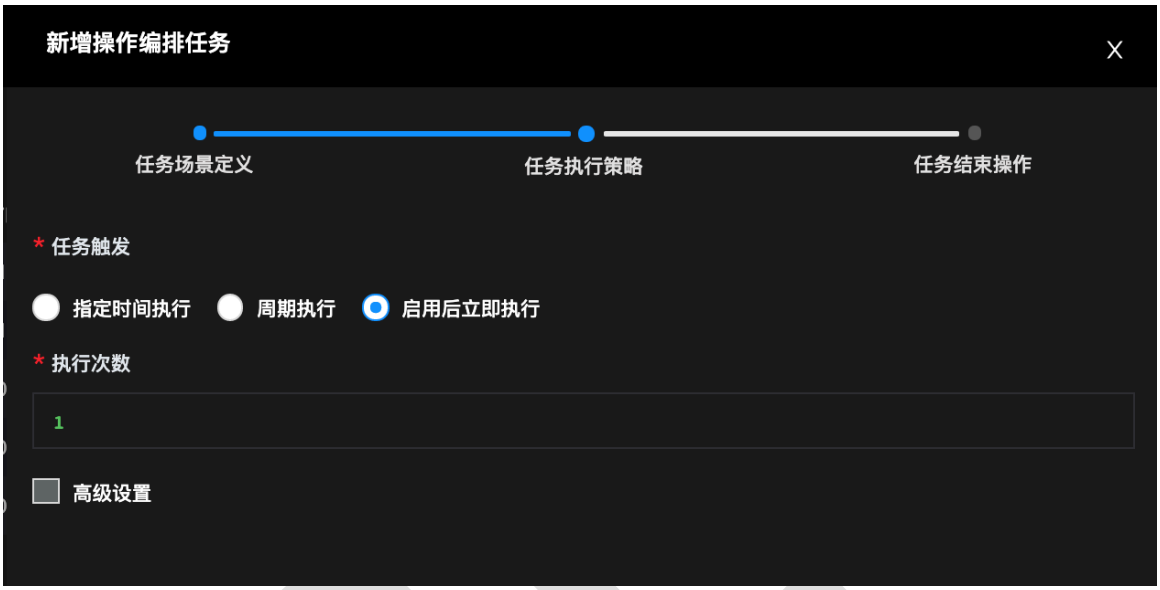
4.2.1.1. MongoDB 集群自动部署

用户以管理员身份登录系统，进入【系统】->【系统管理】模块；
点击“新增”按钮，打开“新增操作编排任务”的界面；
“任务描述”下拉框中选择“MongoDB 集群自动部署”；
设置任务场景参数，任务参数设置说明如下：

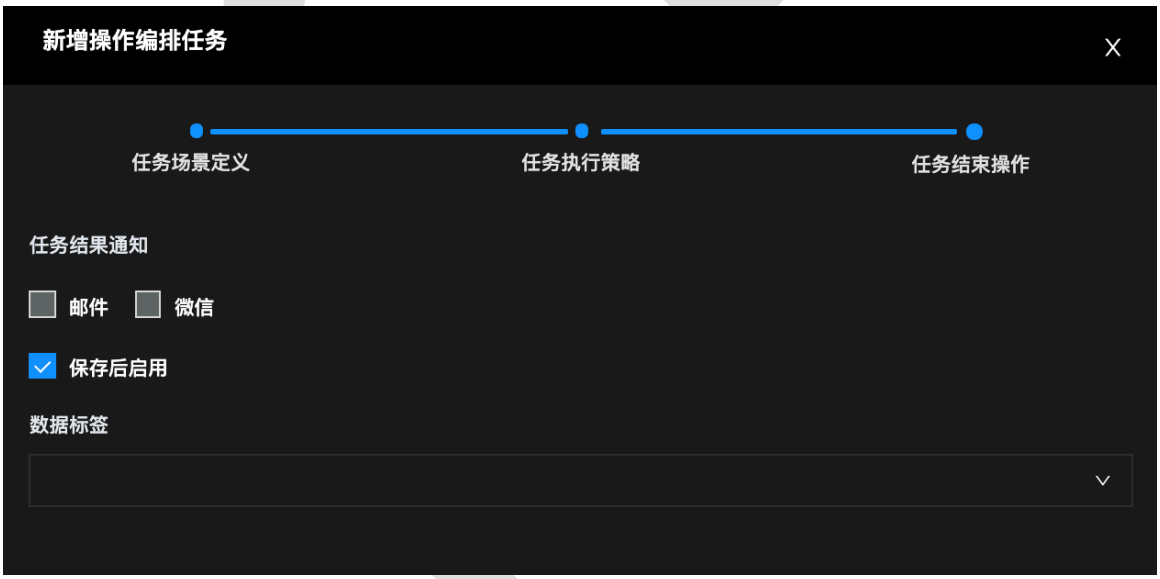
- 部署主机：需要初始化部署 MongoDB 节点的主机 IP 地址，多个节点用逗号分隔。
  NOTE:
  - **自动部署集群的节点 IP 地址中不能包含当前已部署的单机 MongoDB 的主机节点 IP；**
  - **部署主机中的 IP 地址，最后 1 个 IP 地址对应的集群节点主机担当集群中的“监听节点”角色。**
- SSH 端口：用于登录目标节点主机的 SSH 端口，默认 22
- 用户名：登录目标主机的 SSH 用户名
- 密码：登录目标主机的 SSH 密码
- 部署路径：目标主机上 MongoDB 服务包部署的指定路径
- MongoDB 端口：启用目标主机 MongoDB 服务所使用的端口，默认为 27017

任务配置示例参考：

以上场景任务参数设置完毕后，点击下一步，选择“启用后立即执行”，然后继续点击“下一步”。



勾选“保存后启用”，然后点击“保存”，任务开始执行。



任务成功执行完毕后，MongoDB 集群部署成功，且启动完毕。

4.2.1.2. MongoDB 集群自动添加节点

如需添加新的 MongoDB 集群节点，用户可以管理员身份登录系统，进入【系统】->【系统管理】模块，按以下步骤操作：

点击“新增”按钮，打开“新增操作编排任务”的界面；

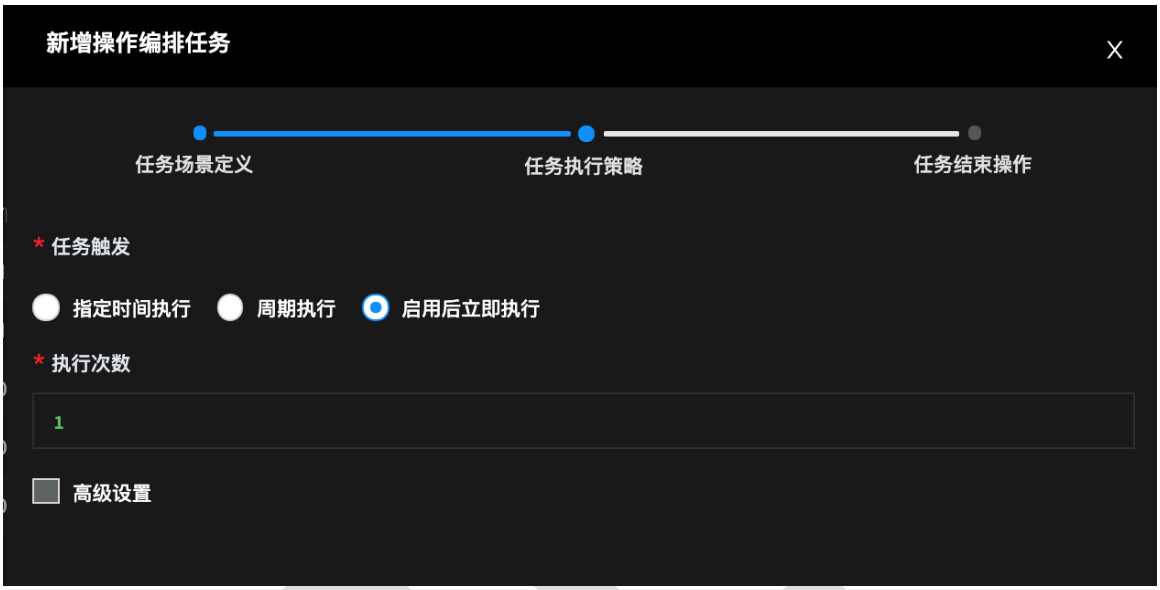
“任务描述”下拉框中选择“MongoDB 集群自动添加节点”；

设置任务场景参数，任务参数设置说明如下：

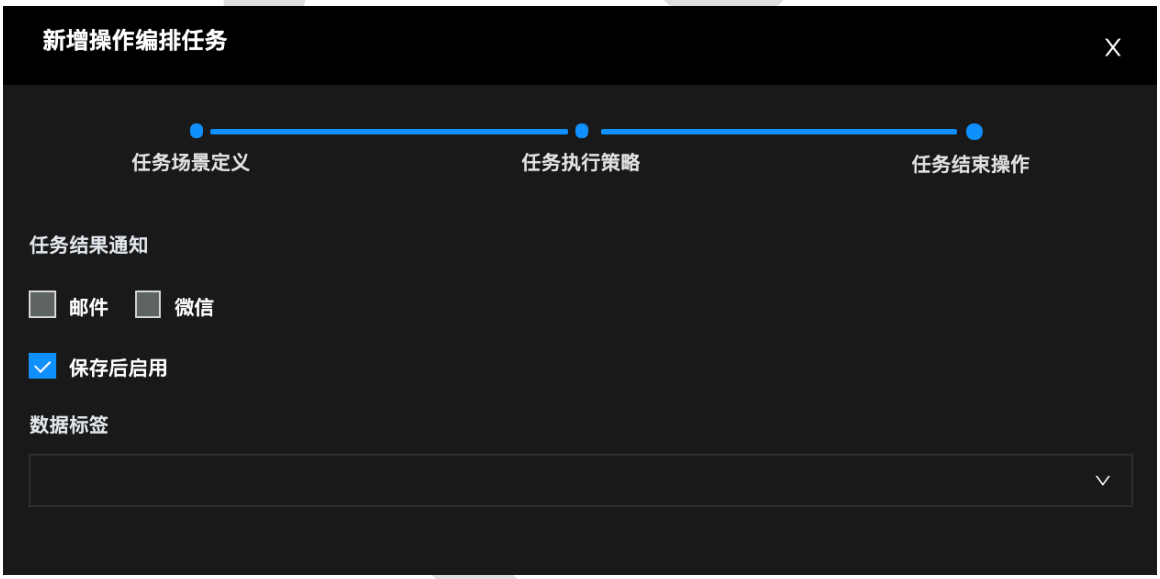
- 已部署 MongoDB 主机：配置现有 MongoDB 集群中的其中一台节点主机 IP
- 新部署 MongoDB 节点主机：新节点主机 IP；
  NOTE：请注意每次只能填写一个部署节点的 IP 地址
- 用户名：主机的 SSH 用户名
- 密码：主机的 SSH 密码
- 部署路径：MongoDB 部署的路径
- MongoDB 端口：MongoDB 的端口，默认 27017

任务配置示例参考：

以上场景任务参数设置完毕后，点击下一步，选择“启用后立即执行”，然后继续点击“下一步”。



勾选“保存后启用”，然后点击“保存”，任务开始执行。



任务成功执行完毕后，新的 MongoDB 节点就会自动添加到集群中。

4.2.2. 手工部署

现对手工部署 MongoDB 集群的步骤说明如下：

假定需要部署三个节点的 MongoDB 集群，IP 分别是：192.168.0.81、192.168.0.82、192.168.0.83，端口均为：27017。

4.2.2.1. 部署安装包

将 ICE 随包发布的 MongoDB 安装包 $ICE_HOME/app/subagent/route/pkgs/mongodb.cluster.tar.gz，分别拷至三台服务器上，并解压到相应安装目录：$MONGO_HOME，建议为：$ICE_HOME/app/

4.2.2.2. 初始化 MongoDB 节点

依次登录三台服务器，执行如下命令：

- cd $MONGO_HOME/mongodb/bin
- ./ice_install.sh 192.168.0.81 192.168.0.82 192.168.0.83 27017

单机的 MongoDB 完成配置和启动。

NOTE:

- **自动部署集群的节点 IP 地址中不能包含当前已部署的单机 MongoDB 的主机节点 IP；**
- **ice_install.sh 所使用的 IP 地址参数中，最后 1 个 IP 地址对应的集群节点主机担当集群中的“监听节点”角色。在上面的例子中，192.168.0.83 为“监听节点”。**

4.2.2.3. 初始化 MongoDB 集群

选择其中一台服务器，执行如下命令：

- cd $MONGO_HOME/mongodb/bin
- ./ice_initconfig.sh 192.168.0.81 27017

4.2.2.4. 导出数据

登录当前 ICE 运行的主服务器，导出 MongoDB 数据。执行如下命令：

- cd $ICE-HOME/app/subagent/route/script
- ./ice_mongodb_dump.sh

在$ICE-HOME/app/subagent/route/pkgs 下得到导出的数据文件：ice.data.tar.gz

4.2.2.5. 导入数据

将导出的 MongoDB 数据文件 ice.data.tar.gz，上传到其中一台 MongoDB 集群服务器的 $MONGO_HOME/mongodb/data 下。

- tar -xzvf ice.data.tar.gz
- ./ice_restore.sh 192.168.0.81 27017

4.2.3. 鉴权设置

4.2.3.1. 主库创建用户

登录 81，执行：

- cd $MONGO_HOME/mongodb/bin
- ./mongo
- use admin
- db.createUser(

```
{
  user:"admin",
  pwd:"9S18m/?&8Ek^n9MM52z",
  roles:[
    {role:"root",db:"admin"},
    {role:"read",db:"local"}
  ]
}
```

)

- use ice
- db.createUser(

```
{
  user:"admin",
  pwd:"9S18m/?&8Ek^n9MM52z",
  roles:[
    {role:"dbAdmin",db:"ice"}
  ]
}
```

)

- use config
- db.createUser(

```
{
  user:"admin",
  pwd:"9S18m/?&8Ek^n9MM52z",
  roles:[
    {role:"dbAdmin",db:"config"}
  ]
}
```

)

4.2.3.2. 停 MongoDB 集群服务

按照监听节点、副本节点、主节点的顺序，停止 mongo 库，通过执行：rs.status()，查看每个节点对应的类型。

为了保证数据正确性，每个节点上都按照如下命令停 mongo：

- cd $MONGO_HOME/mongodb/bin
- rs.stepDown()
- use admin
- db.shutdownServer()

4.2.3.3. 主库生成认证文件

登录 81，运行：

- cd $MONGO_HOME/mongodb/conf
- openssl rand -base64 90 -out ./mongo.keyfile
- chmod 400 ./mongo.keyfile

4.2.3.4. 复制主库认证文件到其他库

登录 81，运行：

- cd $MONGO_HOME/mongodb/conf
- scp ./mongo.keyfile root@192.168.0.82:/$MONGO_HOME/mongodb/conf
- scp ./mongo.keyfile root@192.168.0.83:/$MONGO_HOME/mongodb/conf

4.2.3.5. 启动 MongoDB 集群

登录 81，运行：

- cd $MONGO_HOME/mongodb/bin
- ./startMongo-Auth.sh

登录 82，运行：

- cd $MONGO_HOME/mongodb/bin
- ./startMongo-Auth.sh

登录 83，运行：

- cd $MONGO_HOME/mongodb/bin
- ./startMongo-Auth.sh

4.2.4. 修改 ICE 中的 MongoDB 配置

登录 ICE 服务器：

- vi $ICE_HOME/app/flink/myapps/conf/application_cmdboper.properties，修改:

- ice.mongodb.connection-string=
  mongodb://admin:9S18m%2F%3F%268Ek%5En9MM52z@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000

- vi $ICE_HOME/app/flink/myapps/conf/application_cmdbchain.properties，修改:
  - ice.mongodb.connection-string=
    mongodb://admin:9S18m%2F%3F%268Ek%5En9MM52z@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000

- vi $ICE_HOME/app/dc/conf/application.yml，修改:
  - ice.mongodb.connection-string=
    mongodb://admin:9S18m%2F%3F%268Ek%5En9MM52z@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000
  -

- vi $ICE_HOME/app/sa/conf/application.yml，修改:
  - ice.mongodb.connection-string=
    mongodb://admin:9S18m%2F%3F%268Ek%5En9MM52z@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000

-

- vi $ICE_HOME/app/openapi/conf/application.yml，修改:
  - ice.mongodb.connection-string=
    mongodb://admin:9S18m%2F%3F%268Ek%5En9MM52z@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000
  -
- vi $ICE_HOME/app/main/app/conf/ice-cmdb.properties，修改:
  - mongodb.uri=mongodb://admin:9S18m%2F%3F%268Ek%5En9MM52z@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000

4.2.5. 修改 MongoDB 密码

4.2.5.1. 备份 MongoDB 数据库:

1、使用 mongo 用户，分别登录 192.168.0.81、192.168.0.82、192.168.0.83
2、cd ice
3、./ice-ops-manager.sh 40

4.2.5.2. 备份 MongoDB 数据目录（包括数据节点和监听节点）:

1、使用 mongo 用户，分别登录 192.168.0.81、192.168.0.82、192.168.0.83
2、cd ice/app/mongodb && tar -cvzf mongodb.data.tar.gz data
3、cd ice/app/mongodb_arbiter && tar -cvzf mongodb-arbiter.data.tar.gz data

4.2.5.3. 查找 MongoDB 主节点:

1、使用 mongo 用户，分别登录 192.168.0.81、192.168.0.82、192.168.0.83
2、cd ice/app/mongodb/bin

3、 ./mongo --username admin --password '9S18m/?&8Ek^n9MM52z'

4、查看当前命令行是否显示 PRIMARY，即为主节点

4.2.5.4. 通过 MongoDB 主节点修改密码（!!!在命令行设置密码时，特殊字符不需要转义!!!）：

1、使用 mongo 用户登录主节点服务器

2、cd ice/app/mongodb/bin

3、 ./mongo --username admin --password '9S18m/?&8Ek^n9MM52z'

4、use admin

5、db.updateUser("admin", {pwd:"新密码"})

6、use ice

7、db.updateUser("admin", {pwd:"新密码"})

8、use config

9、db.updateUser("admin", {pwd:"新密码"})

10、 exit 退出

11、 ./mongo --username admin --password '新密码'

12、 可成功登录，即表示成功

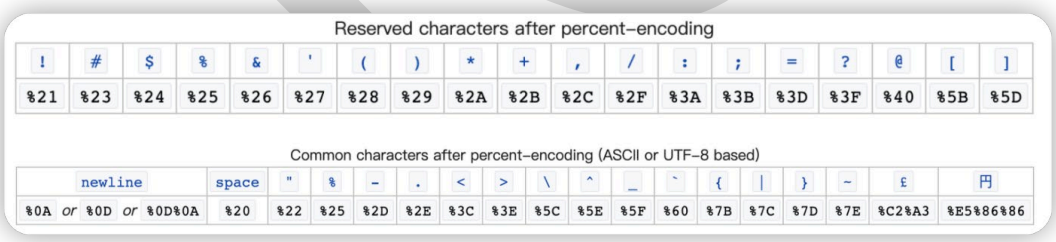
4.2.5.5. 修改所有的 mongodb 连接配置（!!!在修改 ICE 中的 mongo 连接时，特殊字符必须要先转义!!!）：

前提：参考

https://en.wikipedia.org/wiki/Percent-encoding

https://groups.google.com/g/mongodb-user/c/9o8ZNdYAPrY?pli=1

将新密码中的特殊字符进行转义处理



登录 ICE 服务器：192.168.0.81、192.168.0.82、192.168.0.83：

- vi $ICE_HOME/app/flink/myapps/conf/application_cmdboper.properties，修改:
  - ice.mongodb.connection-string= mongodb://admin:转义后的新密码@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-

1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000

- vi $ICE_HOME/app/dc/conf/application.yml，修改:
  - ice.mongodb.connection-string= mongodb://admin: 转义后的新密码@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000
- vi $ICE_HOME/app/sa/conf/application.yml，修改:
  - ice.mongodb.connection-string= mongodb://admin: 转义后的新密码@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000
- vi $ICE_HOME/app/openapi/conf/application.yml，修改:
  - ice.mongodb.connection-string= mongodb://admin: 转义后的新密码@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000
- vi $ICE_HOME/app/main/app/conf/ice-cmdb.properties，修改:
  - mongodb.uri=mongodb://admin: 转义后的新密码@192.168.0.81:27017,192.168.0.82:27017,192.168.0.83:27017/ice?authSource=admin&authMechanism=SCRAM-SHA-1&minPoolSize=10&maxPoolSize=100&waitQueueMultiple=4&waitQueueTimeoutMS=120000&connectTimeoutMS=120000&maxWaitTime=120000&maxIdleTimeMS=300000&maxLifeTimeMS=0&socketTimeoutMS=120000&ssl=false&heartbeatFrequencyMS=30000

4.2.6. 扩容 MongoDB 节点

1. 复制主节点或任意副本节点 mongodb 目录到新主机（如 192.168.1.11）或当前节点的新目录（新目录名）
2. 修改新节点上的配置文件（同节点下新目录需要修改 port 设置，如 27018）

cd /icesystem/ice/app/mongodb/bin

vi ./startMongo-Auth.sh 按需修改启动性能参数

cd /icesystem/ice/app/mongodb/conf

vi mongodb-auth.conf 按需修改 port、replSetName 参数

3. 启动新节点服务

cd /icesystem/ice/app/mongodb/bin

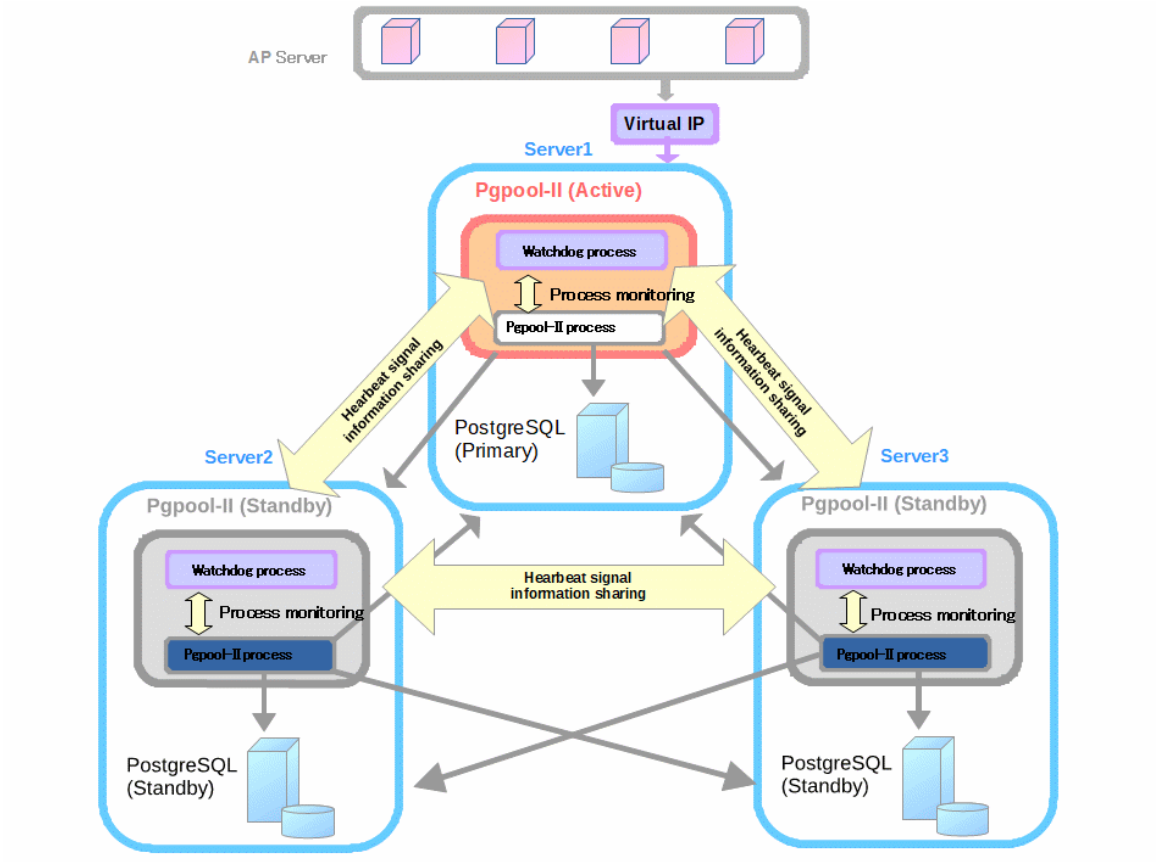
./startMongo-Auth.sh

4. 在主节点上执行

```
./mongo --username admin --password '9S18m/?&8Ek^n9MM52z' <<EOF
rs.add({host:"192.168.1.11:27018"});
EOF
```

5，添加新节点 ip 到集群 ice-deploy-config.conf 配置文件中的 mongo secondary 配置中。

4.1. Postgresql+PgPool-II+VIP 集群（高可用）部署步骤



Postgres 采用主备方式，实时备份；Pgpool-II 实时监控 pg 节点的状态，并对读写操作分流（主节点写、备节点读）；VIP（虚拟 IP）统一对外提供服务。

PG 版本为 13.3-1，pgpoo-II 版本为 4.2.2，安装包均包含在 ICE 发布包中，对应为：$ICE_HOME/app/toolkit/pg-13.3-1、$ICE_HOME/app/toolkit/pgpool-4.2.2。

采用 3 个 Postgresql+3 个 pgpool-II 方式部署，假定 IP 为：192.168.0.81、192.168.0.82、192.168.0.83，VIP 为 192.168.0.84，其中：192.168.0.81 作为 pg 的主库。

注：

下文中 **[all servers]** 表示所有的服务器上执行命令， **[81]** 或者 **[82]** 或者 **[83]** 表示在具体某一台服务器上执行命令。

4.1.1. 服务器基础配置

4.1.1.1. 关闭防火墙

[all servers]# systemctl disable firewalld

4.1.1.2. 关闭 selinux 重启生效

[all servers]# sed -i '/^SELINUX=.*/ s//SELINUX=disabled/' /etc/selinux/config
[all servers]# setenforce 0 && reboot（首次安装可以不用重启？）

4.1.1.3. 添加 hosts

[all servers]#
echo "192.168.0.81 cmdb_db01" >> /etc/hosts
echo "192.168.0.82 cmdb_db02" >> /etc/hosts
echo "192.168.0.83 cmdb_db03" >> /etc/hosts

对 81、82、83 分别设置 hostname 为：cmdb_01、cmdb_02、cmdb_03，命令参考如下：
[81]# hostnamectl set-hostname cmdb_db01
[82]# hostnamectl set-hostname cmdb_db02
[83]# hostnamectl set-hostname cmdb_db03

注：hostname **不能带中横线。**

4.1.2. Postgresql13 安装

将 ICE 发布包中的 pg13 安装文件（$ICE_HOME/app/toolkit/pg-13.3-1）上传至 81、82、83 服务器，执行 ice-psql-install-13.sh 进行安装。

4.1.3. Postgresql13 配置

4.1.3.1. 开启 WAL archiving

在主库 81 上执行：
在文件最后添加以下内容
[81]# su - postgres
[81]# vi /var/lib/pgsql/13/data/postgresql.conf
archive_mode = on
archive_command = 'cp "%p" "/var/lib/pgsql/archivedir/%f"'
max_wal_senders = 10
max_replication_slots = 10
wal_level = replica
hot_standby = on
wal_log_hints = on

所有服务器执行：
[all servers]# su - postgres
[all servers]$ mkdir /var/lib/pgsql/archivedir

4.1.3.2. 启动 postgres 服务

[all servers]# su - root
[all servers]# systemctl start postgresql-13

4.1.3.3. 创建数据库用户

用户 pgpool，用于 pgpool-II 使用；
用户 repl，用于流复制使用；
用户 postgres，用于 pg 常规操作使用。
密码统一为： **icedbadmin123#@!**
[all servers]# su - postgres
[all servers]# psql -U postgres -p 5432
postgres=# CREATE ROLE pgpool WITH LOGIN;
postgres=# CREATE ROLE repl WITH REPLICATION LOGIN;
postgres=# \password pgpool
postgres=# \password repl
postgres=# \password postgres
# pgpool_status 中显示
postgres=# grant pg_monitor to pgpool;
# 查看结果 replication_state
postgres=# \du
postgres=# exit

4.1.3.4. 设置 os 用户 postgres 的密码

密码统一为：**icedbadmin123#@!**
[all servers]# su - root
[all servers]# passwd postgres

4.1.3.5. 三台服务器设置 SSH 互信

[all servers]# su - root
[all servers]# cd ~/.ssh

[all servers]# ssh-keygen -t rsa -f id_rsa_pgpool
[all servers]# ssh-copy-id -i id_rsa_pgpool.pub postgres@cmdb_db01
[all servers]# ssh-copy-id -i id_rsa_pgpool.pub postgres@cmdb_db02
[all servers]# ssh-copy-id -i id_rsa_pgpool.pub postgres@cmdb_db03

[all servers]# su - postgres
[all servers]$ cd ~/.ssh
[all servers]$ ssh-keygen -t rsa -f id_rsa_pgpool
[all servers]$ ssh-copy-id -i id_rsa_pgpool.pub postgres@cmdb_db01
[all servers]$ ssh-copy-id -i id_rsa_pgpool.pub postgres@cmdb_db02
[all servers]$ ssh-copy-id -i id_rsa_pgpool.pub postgres@cmdb_db03

4.1.3.6. online-recover 配置免密

为数据库 repl online-recover 配置免密，最后一列为数据库中配置的密码
[all servers]# su - postgres
[all servers]$ vi /var/lib/pgsql/.pgpass
cmdb_db01:5432:replication:repl:icedbadmin123#@!
cmdb_db02:5432:replication:repl:icedbadmin123#@!
cmdb_db03:5432:replication:repl:icedbadmin123#@!

cmdb_db01:5432:postgres:postgres:icedbadmin123#@!
cmdb_db02:5432:postgres:postgres:icedbadmin123#@!
cmdb_db03:5432:postgres:postgres:icedbadmin123#@!
[all servers]$ chmod 600 /var/lib/pgsql/.pgpass

4.1.4. Pgpool-II 安装

将 ICE 发布包中的 pgpool-II 安装文件（$ICE_HOME/app/toolkit/pgpool-4.2.2）上传至 81、82、83 服务器，并执行：

[all servers]# su - root
[all servers]# rpm -ivh libevent-2.0.21-4.el7.x86_64.rpm
[all servers]# rpm -ivh libmemcached-1.0.16-5.el7.x86_64.rpm
[all servers]# rpm -ivh pgpool-II-pg13-*.rpm
[all servers]# systemctl enable pgpool

4.1.5. Pgpool-II 配置

将 ICE 发布包中的 pgpool-II 配置文件（$ICE_HOME/app/toolkit/pgpool-4.2.2/

pgpool.conf）上传至 81、82、83 服务器的/etc/pgpool-II/。调整虚拟 IP：

[all servers]# su - postgres

[all servers]# vi /etc/pgpool-II/pgpool.conf

delegate_IP = '192.168.0.84'

NOTE：需要手工将 pgpool.conf 中默认 vip 的网卡名称 ens192 改成实际 vip 所在主机使用的网卡名称

4.1.5.1. 设置 pool_passwd

统一密码为： icedbadmin123#@!

[all servers]# su – postgres

[all servers]# cd /etc/pgpool-II

[all servers]# pg_md5 -p -m -u postgres pool_passwd

[all servers]# pg_md5 -p -m -u pgpool pool_passwd

[all servers]# cat /etc/pgpool-II/pool_passwd

4.1.5.2. 设置 pgpool_node_id

[all servers]# su - postgres

[81]# echo 0 > /etc/pgpool-II/pgpool_node_id

[82]# echo 1 > /etc/pgpool-II/pgpool_node_id

[83]# echo 2 > /etc/pgpool-II/pgpool_node_id

4.1.5.3. 为 follow_primary.sh 中 PCP_USER=pgpool 配置免密登录

统一密码为：icedbadmin123#@!

[all servers]# su - postgres

[all servers]# echo 'pgpool:'$(pg_md5 'icedbadmin123#@!' `) >> /etc/pgpool-II/pcp.conf

[all servers]$ echo 'localhost:9898:pgpool:icedbadmin123#@!' > ~/.pcppass

[all servers]$ chmod 600 ~/.pcppass

修改 pool_hba.conf 的密码校验方式为 MD5:

[all servers]# su - postgres

[all servers]# echo 'host all pgpool 0.0.0.0/0 md5' >> /etc/pgpool-II/pool_hba.conf

[all servers]# echo 'host all postgres 0.0.0.0/0 md5' >> /etc/pgpool-II/pool_hba.conf

4.1.5.4. Failover 配置

[all servers]# su - postgres
[all servers]# cd /etc/pgpool-II
[all servers]# cp -p /etc/pgpool-II/failover.sh{.sample,}
[all servers]# cp -p /etc/pgpool-II/follow_primary.sh{.sample,}
[all servers]# chown postgres:postgres /etc/pgpool-II/{failover.sh,follow_primary.sh}

4.1.5.5. Online Recovery 配置

所有服务器执行：

[all servers]# su - postgres
[all servers]# cp -p /etc/pgpool-II/recovery_1st_stage.sample /var/lib/pgsql/13/data/recovery_1st_stage
[all servers]# cp -p /etc/pgpool-II/pgpool_remote_start.sample /var/lib/pgsql/13/data/pgpool_remote_start
[all servers]# chown postgres:postgres /var/lib/pgsql/13/data/{recovery_1st_stage,pgpool_remote_start}

主库 81 上执行：

[81]# su - postgres
[81]# psql template1 -c "CREATE EXTENSION pgpool_recovery"

4.1.5.6. VIP（虚拟 IP）配置

[all servers]# su - postgres
[all servers]# vi /etc/pgpool-II/escalation.sh
设置：
- VIP=192.168.0.84
- DEVICE=对应的网卡名称（如：eth0，即，服务器当前 ip 所对应的网卡名）

4.1.5.7. Logging 配置

[all servers]# su - root

[all servers]# mkdir /var/log/pgpool_log/
[all servers]# chown postgres:postgres /var/log/pgpool_log/
[all servers]# su - postgres
[all servers]# vi /etc/sysconfig/pgpool
设置：OPTS=" -D -n"

4.1.6. Pgpool-II 启动

su – root 切换到 root 用户
按照先主库、后备库的方式，启动 pgpool-II：
[81]# systemctl start pgpool
[82]# systemctl start pgpool
[83]# systemctl start pgpool

4.1.7. 启用主从模式

4.1.7.1. 停备库

[82]# systemctl stop postgresql-13
[83]# systemctl stop postgresql-13

4.1.7.2. 检查备库是否完全停止

查看有没有活动进程，有则 kill -9 杀掉，否则后面操作不成功
[82]# ps wax | grep `head -1 /var/lib/pgsql/13/data/postmaster.pid`
[83]# ps wax | grep `head -1 /var/lib/pgsql/13/data/postmaster.pid`

4.1.7.3. 主库同步到备库

主库操作(其中：192.168.0.84 为 VIP)
[81]# su - postgres
[81]# pcp_recovery_node -h 192.168.0.84 -p 9898 -U pgpool -n 1
[81]# pcp_recovery_node -h 192.168.0.84 -p 9898 -U pgpool -n 2
以上操作：
- 需要输入 pgpool 的密码 icedbadmin123#@!
- 当打印出 pcp_recovery_node -- Command Successful 即为成功

4.1.7.4. 启动备库

登录备库，执行 systemctl start postgresql-13（**NOTE：***该操作经实际环境验证会执行失败，因此不需要执行该操作**）

4.1.7.5. 检查主备状态

f-主 t-从

登录主库

[81]# su - postgres

[81]# psql -c "select pg_is_in_recovery()" （或 psql -h vip -p 9999 -U postgres postgres -c "show pool_nodes";命令 ）

如果备节点状态为 down，可尝试手动添加：

[81]# su - postgres

[81]# pcp_attach_node -h 192.168.0.84 -p 9898 -U pgpool -n 1

[81]# pcp_attach_node -h 192.168.0.84 -p 9898 -U pgpool -n 2

检查主节点复制信息：

[81]# su – postgres

[81]# psql -x -c "select * from pg_stat_replication" -d postgres

4.1.8. 修改 ICE 中的 Postgresql 连接

Postgresql+pgpool-II 集群搭建完毕后，通过 VIP（虚拟 IP）192.168.0.84 对外提供 JDBC 连接服务，默认服务端口为：9999。

登录 ICE 服务器：

- vi $ICE_HOME/app/dc/bin/run.sh: 修改
  - DC_ICEDB_URL=jdbc:postgresql://192.168.0.84:9999/icms（通过修改$PG_IP 和 $PG_PORT 两个变量）
- vi $ICE_HOME/app/openapi/bin/run.sh: 修改
  - OPENAPI_ICEDB_URL=jdbc:postgresql://192.168.0.84:9999/icms（通过修改$PG_IP 和$PG_PORT 两个变量）
- vi $ICE_HOME/app/sa/bin/run.sh: 修改
  - SA_ICEDB_URL=jdbc:postgresql://192.168.0.84:9999/icms（通过修改$PG_IP 和 $PG_PORT 两个变量）

- SA_OTHERDB1_URL=jdbc:postgresql://192.168.0.84:9999/icms（通过修改$PG_IP 和 $PG_PORT 两个变量）

- vi $ICE_HOME/app/main/app/conf/jdbc.properties:
  - jdbc.url=jdbc:postgresql://192.168.0.84:9999/icms
  - cur.jdbc.url=jdbc:postgresql://192.168.0.84:9999/icms

4.1.9. pg+pgpool 启停

1、连接 vip，查看主备库情况：
[postgres]$ psql -h 22.9.10.207 -p 9999 -U postgres postgres -c "show pool_nodes"
其中：role 为 primary 的，是主库

2、依次停止 pgpool：
[postgres]$ pgpool -m fast stop

3、先主库后备库，停 pg：(205 主库)
[root]$ systemctl stop postgresql-13

启动>>>>>
1、先主库后备库，启 pg：(205 主库)
[root]$ systemctl start postgresql-13

2、检查 pg 是否启动
[root]$ systemctl status postgresql-13
打印状态：active (running)

3、依次启动 pgpool
[root]$ systemctl start pgpool

4、检查 pgpool
[root]$ systemctl status pgpool
打印状态：active (running)

5、连接 vip，查看主备库情况：
[postgres]$ psql -h 22.9.10.207 -p 9999 -U postgres postgres -c "show pool_nodes"
其中：role 为 primary 的，是主库

-- 查看 pg 各表使用情况

```
SELECT schemaname,relname,n_live_tup FROM pg_stat_user_tables
where schemaname='npas'
ORDER BY n_live_tup DESC;
```

4.1.10. 修改 PostgreSQL 密码

4.1.10.1. 备份 PostgreSQL

1、使用 mongo 用户，分别登录 192.168.1.81、192.168.1.82、192.168.1.83
2、cd ice
3、./ice-ops-manager.sh 43

4.1.10.2. 记录各节点的 nodeId

1、使用 mongo 用户，分别登录 192.168.1.81 主库
2、su – postgres
3、psql -h 192.168.1.84 -p 9999
4、show pool_nodes;
记录节点与 nodeId 的对应关系

4.1.10.3. 通过 VIP 登录 postgresql，修改数据库用户 postgres 密码

1、使用 mongo 用户，分别登录 192.168.1.81 主库
2、su – postgres
3、psql -h 192.168.1.84 -p 9999
4、alter role postgres with password '新密码'

4.1.10.4. 停备库 192.168.1.82、192.168.1.83

1、使用 mongo 用户，分别登录 192.168.1.82、192.168.1.83
2、systemctl stop postgresql-13
3、检查备库的 postgresql 是否停干净：
ps wax | grep `head -1 /var/lib/pgsql/13/data/postmaster.pid`
如果还有进程，使用 kill -9 PID 杀掉

4.1.10.5. 同步主库数据到备库(使用备库的 nodeId)

1、使用 mongo 用户，分别登录 192.168.1.81 主库

2、 su – postgres

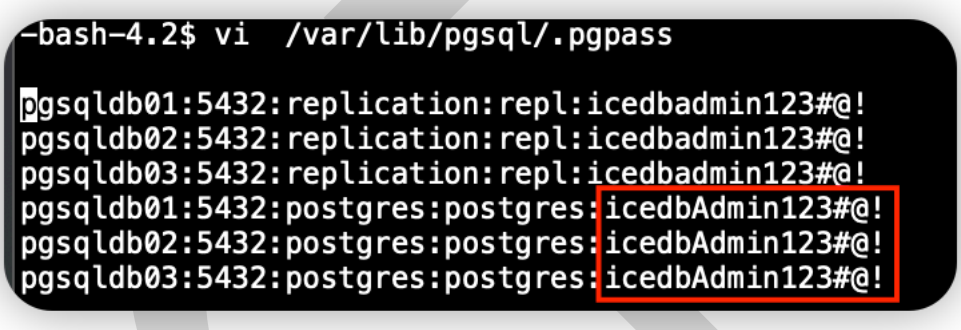
3、 pcp_recovery_node -h 192.168.1.84 -p 9898 -U pgpool -n 1
pcp_recovery_node -h 192.168.1.84 -p 9898 -U pgpool -n 2

4.1.10.6. 修改/var/lib/pgsql/.pgpass 中的 postgres 密码

1、使用 mongo 用户，分别登录 192.168.1.81、192.168.1.82、192.168.1.83

2、 su – postgres

3、 vi /var/lib/pgsql/.pgpass 将 postgres 的密码明文修改为新密码明文，如下图示例：



4、 chmod 600 /var/lib/pgsql/.pgpass

4.1.10.7. 修改/etc/pgpool-II/pgpool.conf 中的 recovery_password 为新密码

1、使用 mongo 用户，分别登录 192.168.1.81、192.168.1.82、192.168.1.83

2、 su – postgres

3、 vi /etc/pgpool-II/pgpool.conf 的 recovery_password 为新密码

4.1.10.8. 修改/etc/pgpool-II/pool_passwd 中的 postgres 密码

1、使用 mongo 用户，分别登录 192.168.1.81、192.168.1.82、192.168.1.83

2、 su – postgres

3、 cd /etc/pgpool-II/

4、 pg_md5 -p -m -u postgres pool_passwd

4.1.10.9. 停 pgpool

1、使用 mongo 用户，分别登录 192.168.1.81、192.168.1.82、192.168.1.83

2、 su – postgres

3、 pgpool -m fast stop

4.1.10.10. 停 postgresql（先主库后备库）

1、使用 mongo 用户，分别登录 192.168.1.81 主库
2、 systemctl stop postgresql-13
3、使用 mongo 用户，分别登录 192.168.1.82、192.168.1.83
4、 systemctl stop postgresql-13
5、检查备库的 postgresql 是否停干净：
ps wax | grep `head -1 /var/lib/pgsql/13/data/postmaster.pid`
如果还有进程，使用 kill -9 PID 杀掉

4.1.10.11. 启动 postgresql（先主库后备库）

1、使用 mongo 用户，分别登录 192.168.1.81 主库
2、 systemctl start postgresql-13
3、使用 mongo 用户，分别登录 192.168.1.82、192.168.1.83
4、 systemctl start postgresql-13

4.1.10.12. 启动 pgpool

1、使用 mongo 用户，分别登录 192.168.1.81、192.168.1.82、192.168.1.83
2、 su – postgres
3、 pgpool -C -D

4.1.10.13. 检查 postgresql 节点状态

1、使用 mongo 用户，分别登录 192.168.1.81 主库
2、 su – postgres
3、 psql -h 192.168.1.84 -p 9999
4、 show pool_nodes
各节点状态为 UP

4.1.10.14. 修改 ICE 中的 postgresql 数据库连接信息

登录 ICE 服务器：192.168.1.81、192.168.1.82、192.168.1.83

- vi $ICE_HOME/app/dc/bin/run.sh: 修改

  - PG_PWD=新密码
- vi $ICE_HOME/app/openapi/bin/run.sh: 修改
  - PG_PWD=新密码
- vi $ICE_HOME/app/sa/bin/run.sh: 修改
  - PG_PWD=新密码
- vi $ICE_HOME/app/main/app/conf/jdbc.properties:
  - jdbc.password=新密码
  - cur.jdbc.password=新密码

4.2. Flink 集群部署步骤

Flink 集群，采用 Master-Slave 方式部署，多节点组成 HA 高可用模式。
假定需要部署三个节点的 Flink 集群，IP 分别是：192.168.0.81、192.168.0.82、192.168.0.83。

NOTE:
需要先部署完 Pulsar 集群，并已启动 zookeeper。

4.2.1. 服务器互信免密登陆

登录 81：
- ssh-keygen
- 按三次回车，完成生成私钥和公钥
- cd /USER/.ssh，可查看到已经生成了公钥 id_rsa.pub 和私钥 id_rsa
- ssh-copy-id 192.168.0.82
- ssh-copy-id 192.168.0.83

登录 82：
- ssh-keygen
- 按三次回车，完成生成私钥和公钥
- cd /USER/.ssh，可查看到已经生成了公钥 id_rsa.pub 和私钥 id_rsa
- ssh-copy-id 192.168.0.81
- ssh-copy-id 192.168.0.83

登录 83：
- ssh-keygen
- 按三次回车，完成生成私钥和公钥
- cd /USER/.ssh，可查看到已经生成了公钥 id_rsa.pub 和私钥 id_rsa
- ssh-copy-id 192.168.0.81
- ssh-copy-id 192.168.0.82

至此，三台服务器已添加免密登录

4.2.2. 上传 Flink 安装包

将 Flink 的安装包 flink-1.9.0-bin-scala_2.11.tgz 分别上传至三台服务器的相应目录 $FLINK_HOME，并解压。

4.2.3. 配置集群

分别登录三台服务器：

- cd $FLINK_HOME/flink/conf
- vi flink-conf.yaml 文末添加：

high-availability: zookeeper
high-availability.zookeeper.quorum:
192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
high-availability.storageDir: file:///home/adminop/ice/app/flink/ha
high-availability.zookeeper.path.root: /flink

- vi masters 添加如下：

192.168.0.81:8081
192.168.0.82:8081
192.168.0.83:8081

- vi workers 添加如下：

192.168.0.81
192.168.0.82
192.168.0.83
每个 IP 单独一行

NOTE:
high-availability.storageDir：需要按照 flink 实际部署的路径来设置

4.2.4. 启动 Flink

只需要启动其中一个节点的 flink 即可，登录 81：

- cd $FLINK_HOME/flink/bin
- ./ice-start-flink.sh

4.2.5. 上传 ice-streaming

将 ice-streaming 包分别上传至三台 flink 服务器的$FLINK_HOME/flink/myapps

NOTE:

- **如果 MongoDB 为集群模式，需要修改**
  $FLINK_HOME/flink/myapps/conf/application_cmdboper.properties、
  application_cmdbchain.properties 中的 ice.mongodb.connection-string 连接内容

4.2.6. 启动 ice-streaming

登录其中一台服务器 81，执行：

- cd $FLINK_HOME/flink/bin
- ./ice-streaming-start.sh

至此，flink 集群搭建完成，并启动完毕。

4.3. Pulsar 集群部署步骤

Pulsar 集群，由：zookeeper 集群、bookkeeper 集群、broker 集群 三部分组成。
假定需要部署三个节点的 Pulsar 集群，IP 分别是：192.168.0.81、192.168.0.82、192.168.0.83。
将 Pulsar 的安装包 pulsar.tar.gz 分别上传至三台服务器的相应目录$PULSAR_HOME，并解压。

4.3.1. Zookeeper 集群

4.3.1.1. 创建 data 文件夹

分别登录 3 台服务器，执行：

- cd $PULSAR_HOME/data
- mkdir -p ./zookeeper/data
- mkdir -p ./zookeeper/log

4.3.1.2. zookeeper.conf 设置

分别登录 3 台服务器，执行：

- cd $PULSAR_HOME/conf
- vi zookeeper.conf，设置如下内容：
  - dataLogDir=logs/zookeeper
  - server.1=192.168.0.81:2888:3888
  - server.2=192.168.0.82:2888:3888
  - server.3=192.168.0.83:2888:3888

4.3.1.3. 创建 zookeeper 的 myid

登录 81，执行：

- echo 1 > $PULSAR_HOME/data/zookeeper/myid

登录 82，执行：

- echo 2 > $PULSAR_HOME/data/zookeeper/myid

登录 83，执行：

- echo 3 > $PULSAR_HOME/data/zookeeper/myid

4.3.1.4. 启动 zookeeper

登录 81，执行：

- cd $PULSAR_HOME
- ./ice-start-zookeeper.sh

登录 82，执行：

- cd $PULSAR_HOME
- ./ice-start-zookeeper.sh

登录 83，执行：

- cd $PULSAR_HOME
- ./ice-start-zookeeper.sh

4.3.1.5. 初始化元数据

初始化元数据，只需要选择其中一个 zookeeper 执行即可。

登录 81，执行：

- cd $PULSAR_HOME
- bin/pulsar initialize-cluster-metadata \
  --cluster ice-pulsar-cluster \
  --zookeeper 192.168.0.81:2181 \
  --configuration-store 192.168.0.81:2181 \
  --web-service-url
  http://192.168.0.81:8080,http://192.168.0.82:8080,http://192.168.0.83:8080 \
  --web-service-url-tls
  https://192.168.0.81:8443,https://192.168.0.82:8443,https://192.168.0.83:8443 \
  --broker-service-url
  pulsar://192.168.0.81:6650,pulsar://192.168.0.82:6650,pulsar://192.168.0.83:6650 \

--broker-service-url-tls
pulsar+ssl://192.168.0.81:6651,pulsar+ssl://192.168.0.82:6651,pulsar+ssl://192.168.0.83:6651

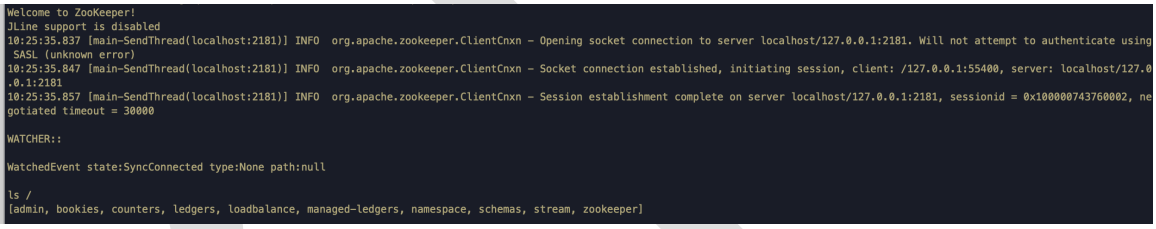
NOTE:

- **Cluster 名称在后续的集群配置中也会用到：**ice-pulsar-cluster

4.3.1.6. 查看 zookeeper 节点

登录其中一台服务器，执行：

- cd $PULSAR_HOME
- bin/pulsar zookeeper-shell
- ls / 查看所有 zookeeper 节点，如下：



4.3.2. Bookkeeper 集群

4.3.2.1. bookkeeper.conf 设置

登录 81，执行：

- cd $PULSAR_HOME/conf
- vi bookkeeper.conf，设置如下内容：
  - advertisedAddress=192.168.0.81
  - zkServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - prometheusStatsHttpPort=8100

登录 82，执行：

- cd $PULSAR_HOME/conf
- vi bookkeeper.conf，设置如下内容：
  - advertisedAddress=192.168.0.82
  - zkServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - prometheusStatsHttpPort=8100

登录 83，执行：

- cd $PULSAR_HOME/conf
- vi bookkeeper.conf，设置如下内容：

- advertisedAddress=192.168.0.83
- zkServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181

prometheusStatsHttpPort=8100

4.3.2.2. 初始化元数据

初始化元数据，只需要选择其中一个 bookkeeper 执行即可。

登录 81，执行：

- cd $PULSAR_HOME
- bin/bookkeeper shell metaformat

4.3.2.3. 启动 bookkeeper

登录 81，执行：

- cd $PULSAR_HOME
- ./ice-start-bookkeeper.sh

登录 82，执行：

- cd $PULSAR_HOME
- ./ice-start-bookkeeper.sh

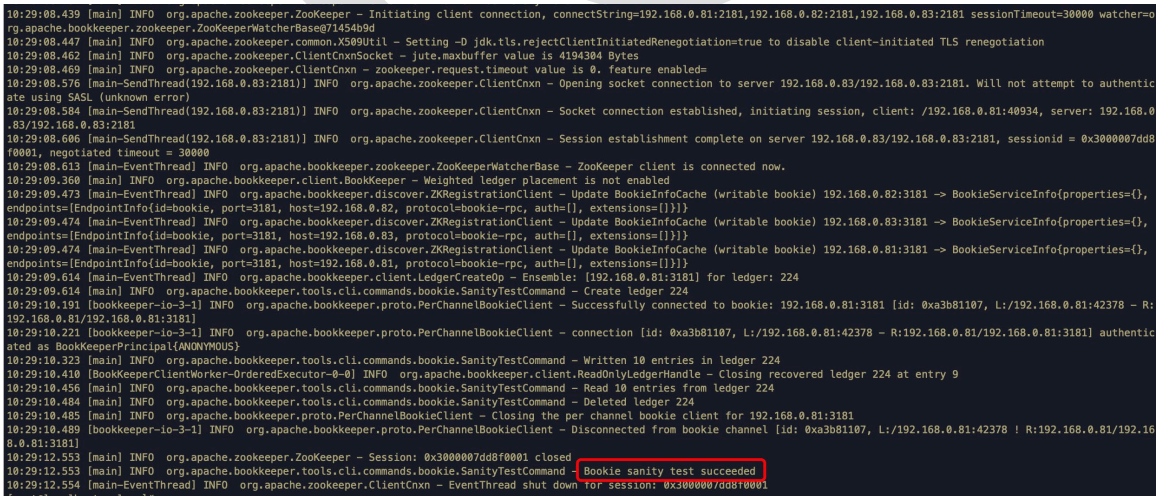
登录 83，执行：

- cd $PULSAR_HOME
- ./ice-start-bookkeeper.sh

4.3.2.4. 验证 bookkeeper

登录其中一台服务器，执行：

- cd $PULSAR_HOME
- bin/bookkeeper shell bookiesanity，显示如下内容：



4.3.3. broker 集群

4.3.3.1. broker.conf 配置

登录 81，执行：

- cd $PULSAR_HOME/conf
- vi broker.conf，设置如下内容：
  - zookeeperServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - configurationStoreServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - advertisedAddress=192.168.0.81
  - clusterName=ice-pulsar-cluster #与 zookeeper 元数据初始化时设置的 cluster 一致

登录 82，执行：

- cd $PULSAR_HOME/conf
- vi broker.conf，设置如下内容：
  - zookeeperServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - configurationStoreServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - advertisedAddress=192.168.0.82
  - clusterName=ice-pulsar-cluster #与 zookeeper 元数据初始化时设置的 cluster 一致

登录 83，执行：

- cd $PULSAR_HOME/conf
- vi broker.conf，设置如下内容：
  - zookeeperServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - configurationStoreServers=192.168.0.81:2181,192.168.0.82:2181,192.168.0.83:2181
  - advertisedAddress=192.168.0.83
  - clusterName=ice-pulsar-cluster #与 zookeeper 元数据初始化时设置的 cluster 一致

4.3.3.2. 启动 broker

登录 81，执行：

- cd $PULSAR_HOME
- ./ice-start-broker.sh

登录 82，执行：

- cd $PULSAR_HOME
- ./ice-start-broker.sh

登录 83，执行：

- cd $PULSAR_HOME
- ./ice-start-broker.sh

4.3.3.3. 查看 broker

登录其中一台服务器，执行：

- cd $PULSAR_HOME
- bin/pulsar-admin brokers list pulsar-cluster，显示如下内容：

```
[root@localhost pulsar]# bin/pulsar-admin brokers list pulsar-cluster
"192.168.0.81:8080"
"192.168.0.82:8080"
"192.168.0.83:8080"
```

4.3.4. 初始化集群设置

登录其中一台服务器，执行：

- cd $PULSAR_HOME/bin
- ./pulsar-admin namespaces set-persistence --bookkeeper-ack-quorum 1 --bookkeeper-ensemble 1 --bookkeeper-write-quorum 1 --ml-mark-delete-max-rate 0 public/default
- ./pulsar-admin namespaces set-retention --size 100M --time 5m public/default
- ./pulsar-admin namespaces set-message-ttl -ttl 600 public/default
- ./pulsar-admin namespaces set-backlog-quota --limit 20M --policy consumer_backlog_eviction public/default

以上设置完成后，通过如下命令查看设置的情况，登录其中一台服务器，执行：

- cd $PULSAR_HOME/bin
- ./pulsar-admin namespaces get-persistence public/default
- ./pulsar-admin namespaces get-retention public/default
- ./pulsar-admin namespaces get-message-ttl public/default
- ./pulsar-admin namespaces get-backlog-quotas public/default

4.3.5. 修改 ICE 中的 pulsar 连接配置

集群搭建完毕后，pulsar 的连接信息为：
pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650

登录 ICE 服务器：

- vi $ICE_HOME/app/dc/bin/run.sh: 修改
  - PULSAR_URL= pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650
  - ~~DC_PULSAR_MAIN=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650~~
- vi $ICE_HOME/app/openapi/bin/run.sh: 修改
  - PULSAR_URL= pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650
  - ~~OPENAPI_PULSAR_MAIN=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650~~
- vi $ICE_HOME/app/sa/bin/run.sh: 修改
  - PULSAR_URL= pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650
  - ~~SA_PULSAR_MAIN=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650~~
  - ~~SA_PULSAR_DISCOVER=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650~~
- vi $ICE_HOME/app/subagent/bin/ice-subagent-start.sh:
  - PULSAR_URL= pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650
  - ~~SUB_PULSAR_MAIN=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650~~
  - ~~SUB_PULSAR_DISCOVER=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650~~
- vi $ICE_HOME/app/flink/myapps/conf/application_cmdboper.properties：修改
  - pulsar.url.service=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650
  - pulsar.url.admin=http://192.168.0.81:8080,192.168.0.82:8080,192.168.0.83:8080
- vi $ICE_HOME/app/main/app/conf/iceportal.properties: 修改
  - pulsar.url.broker=pulsar://192.168.0.81:6650,192.168.0.82:6650,192.168.0.83:6650

4.4. Nginx+Tomcat 集群部署步骤

Nginx+Tomcat 集群，分成两部分内容：

- Tomcat：在多台服务器上部署 mainServer
- Nginx 两种部署方式：
  - 单 Nginx： 单机方式，在一台服务器上部署 Nginx
  - 双 Nginx：分别在两台服务器上部署 Nginx，通过 Keepalive 组件，实现双 Nginx 互备

本次采用单 Nginx 方式部署，假定集群涉及三个节点：

- 192.168.0.81：部署 mainServer
- 192.168.0.82：部署 mainServer
- 192.168.0.83：部署 Nginx

4.4.1. Tomcat 多节点部署 mainServer

4.4.1.1. 准备 mainServer 包

登录 ICE 服务器，执行：

- 创建临时目录：mkdir -p TMP_DIR/ice/app
- cp -r $ICE_HOME/app/main $TMP_DIR/ice/app
- cp -r $ICE_HOME/app/jre $TMP_DIR/ice/app
- cd $TMP_DIR
- tar -cvzf ice-mainServer.tar.gz ./ice

4.4.1.2. 部署 mainServer 包

登录 81、82，执行同样的操作：

- 将 ice-mainServer.tar.gz 上传至相应目录 ICE_HOME
- cd $ICE_HOME
- tar -xzvf ice-mainServer.tar.gz
- 参考 Mongodb 章节、Postgresql 章节、Puslar 章节的连接配置（如果 ICE 已经运行在集群环境，可跳过此步骤）
- 启动 mainServer：cd $ICE_HOME/app/main/app/bin/start.sh
- 验证 mainServer 启动成功：
  - 浏览器访问 http://192.168.0.81:48080/ice，并能成功登录
  - 浏览器访问 http://192.168.0.82:48080/ice，并能成功登录

4.4.2. Nginx 部署

Nginx 涉及的安装包和安装脚本，已包含在 ICE 的部署包内，路径为：$ICE_HOME/app/toolkit，具体为：

- ice-tengine.tar.gz
- ice-nginx-install.sh

将上述 2 个文件打包分别上传至 83 服务器的相应目录，然后执行：

- chmod 775 ice-nginx-install.sh
- ./ice-nginx-install.sh
- vi /home/tengine/tengine/conf/nginx.conf，修改：
  - upstream iceMainServer 中加入 81、82：
    - server 192.168.0.81:48080 weight=1 max_fails=5 fail_timeout=600s;
    - server 192.168.0.82:48080 weight=1 max_fails=5 fail_timeout=600s;
  - upstream iceDataCenter 中加入 81、82：
    - server 192.168.0.81:8601 weight=5 max_fails=5 fail_timeout=600s;
    - server 192.168.0.82:8601 weight=5 max_fails=5 fail_timeout=600s;

  - upstream iceOpenApi 中加入 81、82：
    - server 192.168.0.81:8602 weight=5 max_fails=5 fail_timeout=600s;
    - server 192.168.0.82:8602 weight=5 max_fails=5 fail_timeout=600s;
  - 将所有的 server_name 的 ip，设置为 Nginx 所在的服务器 IP
- cd /home/tengine/tengine/sbin
- 启动 Nginx： ./nginx
- 浏览器访问 http://192.168.0.83:48080/ice

4.5. Redis 集群部署步骤

采用官方推荐的 Cluster 方式搭建集群。为了保证高可用，redis-cluster 集群引入了主从复制模型，一个主节点对应一个或者多个从节点，当主节点宕机的时候，就会启用从节点。当其它主节点 ping 一个主节点 A 时，如果半数以上的主节点与 A 通信超时，那么认为主节点 A 宕机了。如果主节点 A 和它的从节点 A1 都宕机了，那么该集群就无法再提供服务了。

所有的 redis 节点彼此互联(PING-PONG 机制)，内部使用二进制协议优化传输速度和带宽。节点的 fail 是通过集群中超过半数的节点检测失效时才生效。客户端与 Redis 节点直连，不需要中间代理层.客户端不需要连接集群所有节点，连接集群中任何一个可用节点即可。

根据官方推荐，集群部署至少要 3 台以上的 master 节点，最好使用 3 主 3 从六个节点的模式。ICE 的 Redis 集群，也建议至少 3 台服务器部署的方式，每天服务器上部署 2 个 redis 节点，端口分别采用 7001、7002 对外服务。

Redis 集群部署所需软件包，随 ICE 发布包一起发布，路径为：$ICE_HOME/app/toolkit/redis-cluster-6.0.10。

假定需要部署三个节点的 Pulsar 集群，IP 分别是：192.168.0.81、192.168.0.82、192.168.0.83。

4.5.1. 上传 Redis 安装包

建议将 Redis 安装至 ICE 统一目录下。

在目标服务器路径$ICE_HOME/app/下创建 redis 文件夹，将 redis-7001.tar.gz、redis-7002.tar.gz、ice-redis_node-start.sh、ice-redis_node-stop.sh、ice-redis_cluster-start.sh，分别上传至$ICE_HOME/app/redis 下，并解压 redis-7001.tar.gz、redis-7002.tar.gz。

4.5.2. 修改 Redis 节点 IP 配置

登录 81，执行如下：

- cd $ICE_HOME/app/redis/
- vi ice-redis_node-start.sh，设置如下内容：
  - HOST_IP_1=192.168.0.81
- vi ice-redis_node-stop.sh，设置如下内容：
  - HOST_IP_1=192.168.0.81

- vi ice-redis_cluster-start.sh，设置如下内容：
  - HOST_IP_1=192.168.0.81
  - HOST_IP_2=192.168.0.82
  - HOST_IP_3=192.168.0.83
- vi /7001/conf/redis.conf、 vi /7002/conf/redis.conf，将：
  bind 122.51.197.74 修改为：
  bind 192.168.0.81

登录 82，执行如下：

- cd $ICE_HOME/app/redis/
- vi ice-redis_node-start.sh，设置如下内容：
  - HOST_IP_1=192.168.0.82
- vi ice-redis_node-stop.sh，设置如下内容：
  - HOST_IP_1=192.168.0.82
- vi ice-redis_cluster-start.sh，设置如下内容：
  - HOST_IP_1=192.168.0.81
  - HOST_IP_2=192.168.0.82
  - HOST_IP_3=192.168.0.83
- vi /7001/conf/redis.conf、 vi /7002/conf/redis.conf，将：
  bind 122.51.197.74 修改为：
  bind 192.168.0.82

登录 83，执行如下：

- cd $ICE_HOME/app/redis/
- vi ice-redis_node-start.sh，设置如下内容：
  - HOST_IP_1=192.168.0.83
- vi ice-redis_node-stop.sh，设置如下内容：
  - HOST_IP_1=192.168.0.83
- vi ice-redis_cluster-start.sh，设置如下内容：
  - HOST_IP_1=192.168.0.81
  - HOST_IP_2=192.168.0.82
  - HOST_IP_3=192.168.0.83
- vi /7001/conf/redis.conf、 vi /7002/conf/redis.conf，将：
  bind 122.51.197.74 修改为：
  bind 192.168.0.83

NOTE:在云环境中，bind 必须要用内网 ip，且要删除压缩包里的 2 个文件，方可安装成功。

4.5.3. 启动 Redis 节点

登录 81，执行如下：

- cd $ICE_HOME/app/redis/
- ./ice-redis_node-start.sh

登录 82，执行如下：

- cd $ICE_HOME/app/redis/
- ./ice-redis_node-start.sh

登录 83，执行如下：

- cd $ICE_HOME/app/redis/
- ./ice-redis_node-start.sh

4.5.4. 启动 Redis Cluster

选择其中一台服务器执行即可，如：81。

登录 81，执行如下：

- cd $ICE_HOME/app/redis/
- ./ice-redis_cluster-start.sh

4.5.5. 修改 ICE 中的 Redis 连接

Redis 集群搭建完毕后，其连接信息为：192.168.0.81:7001, 192.168.0.81:7002, 192.168.0.82:7001, 192.168.0.82:7002,192.168.0.83:7001, 192.168.0.83:7002

登录 ICE 服务器：

- vi $ICE_HOME/app/dc/bin/run.sh: 修改
  - REDIS_CLUSTER=192.168.0.81:7001, 192.168.0.81:7002, 192.168.0.82:7001, 192.168.0.82:7002,192.168.0.83:7001, 192.168.0.83:7002
- vi $ICE_HOME/app/openapi/bin/run.sh: 修改
  - REDIS_CLUSTER=192.168.0.81:7001, 192.168.0.81:7002, 192.168.0.82:7001, 192.168.0.82:7002,192.168.0.83:7001, 192.168.0.83:7002
- vi $ICE_HOME/app/sa/bin/run.sh: 修改
  - REDIS_CLUSTER=192.168.0.81:7001, 192.168.0.81:7002, 192.168.0.82:7001, 192.168.0.82:7002,192.168.0.83:7001, 192.168.0.83:7002
- vi $ICE_HOME/app/main/app/conf/iceportal.properties: 修改
  - ice.redis.isClusterMode=true

- ice.redis.cluster=192.168.0.81:7001, 192.168.0.81:7002, 192.168.0.82:7001, 192.168.0.82:7002,192.168.0.83:7001, 192.168.0.83:7002

4.6. 集群自动部署程序

4.6.1. 集群自动部署准备事项

- 各集群节点创建 adminop 账户，授予 sudo 权限
  - groupadd adminop
  - useradd -g adminop adminop
  - passwd adminop << eof
  - adminop
  - adminop
  - eof 创建 adminop 用户的密码，该命令无需手工输入密码，直接粘贴即可
  - usermod -s /bin/bash adminop
  - usermod -a -G wheel adminop
  - usermod -a -G adm adminop
  - sudo visudo 打开 sudo 配置文件， 手工添加 adminop ALL=(ALL) NOPASSWD: ALL
- 如果各集群节点挂载磁盘，请设置系统在启动时自动挂载该磁盘
  - lsblk 识别磁盘名称
  - mkfs.ext4 /dev/sdd 格式化磁盘（如果该磁盘尚未格式化。该命令使用 ext4 文件系统）
  - mkdir /icesystem 创建挂载点目录
  - mount /dev/sdb /mnt/sdb 挂载磁盘
  - df -Th 检查挂载情况
  - echo '/dev/sdd /icesystem ext4 defaults 0 0' | sudo tee -a /etc/fstab 挂载磁盘 automount（系统启动时自动挂载该磁盘）
- 上传集群安装包至主控节点
  - 选择 1 台服务器做主控节点，在/home/adminop 下创建 install-console 目录
  - 将集群安装包上传至主控节点/home/adminop/install-console 下，解压后生成 /home/adminop/install-console/install 目录结构：
    - install 集群管理主目录
    - install/ice-install-manager-nonroot.sh 安装部署入口主程序（non-root 用户权限安装，常用于生产环境）
    - install/ice-install-manager-root.sh 安装部署入口主程序（root 用户权限安装，常用于测试环境）
    - install/ice-deploy-config.conf 集群配置文件

- install/package 安装部署程序包及数据包所在目录
- install/patch 补丁程序包所在目录
- Install/log 自动获取主机节点运行日志存放目录
- install/standalone 单机安装部署子程序目录
- install/cluster-nonroot 集群安装部署子程序目录（non-root 用户权限安装，常用于生产环境），包含 pgsql, pgpool, redis, mongo, pulsar, flink, sa, dc,subagent,openapi,mainserver, nginx 等系统组件的集群安装子程序，以及集群状态检查及启停程序
- install/cluster-root 集群安装部署子程序目录（root 用户权限安装），包含 pgsql, pgpool, redis, mongo, pulsar, flink, sa, dc,subagent,openapi,mainserver, nginx 等系统组件的集群安装子程序，以及集群状态检查及启停程序

- 上传数据包
  - 准备 mongo 和 pgsql 数据包，上传到/home/adminop/install-console/install/package 目录；该目录中默认为 mongo 和 pgsql 发布基准数据包

4.6.2. 设置主配置文件

/home/adminop/install-console/install/ice-deploy-config.conf 为集群自动部署程序的主配置文件，包含了集群主机的登录账户、ip 地址、端口、默认目录等设置。**由于涉及账号密码等敏感信息，请在集群部署过程中妥善使用。**

打开并设置/home/adminop/install-console/install/ice-deploy-config.conf 集群配置文件：

- MongoDB 主、从和备节点配置：

  mongo_primary_node={10.230.210.133:22:adminop:adminop:/home/adminop:mongodb:27017};

  mongo_secondary_node={10.230.211.22:22:adminop:adminop:/home/adminop:mongodb-datanode:27018},{10.230.211.22:22:adminop:adminop:/home/adminop:mongodb:27017},{10.230.210.133:22:adminop:adminop:/home/adminop:mongodb-datanode:27018},{10.230.212.176:22:adminop:adminop:/home/adminop:mongodb-datanode:27018};

  mongo_monitor_node={10.230.212.176:22:adminop:adminop:/home/adminop:mongodb:27017}

- Redis 节点：
  redis_cluster_node={10.230.201.168:22:adminop:adminop:/home/adminop},{10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop}
- Pulsar 集群设置：
  pulsar_cluster_node={10.230.201.168:22:adminop:adminop:/home/adminop},{10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop}
- Flink 集群主、从节点：
  flink_master_node={10.230.201.168:22:adminop:adminop:/home/adminop};
  flink_slave_node={10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop}
- Mainserver：
  mainserver_cluster_node={10.230.201.168:22:adminop:adminop:/home/adminop},{10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop}
- Nginx：
  nginx_cluster_node={10.230.212.171:22:adminop:adminop:/home/adminop},{10.230.212.114:22:adminop:adminop:/home/adminop}
- Dc/sa/subagent/openapi：
  dc_cluster_node={10.230.201.168:22:adminop:adminop:/home/adminop},{10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop};
  sa_cluster_node={10.230.201.168:22:adminop:adminop:/home/adminop},{10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop}
  subagent_cluster_node={10.230.212.35:22:adminop:adminop:/home/adminop},{10.230.212.126:22:adminop:adminop:/home/adminop}
  openapi_cluster_node={10.230.201.168:22:adminop:adminop:/home/adminop},{10.230.212.133:22:adminop:adminop:/home/adminop},{10.230.211.61:22:adminop:adminop:/home/adminop}

完成主配置文件设置后，执行 install/cluster/ice-check-config-cluster.sh 对配置文件内容及目标机器可达性、目标机器基础环境就绪情况进行自动核检。

4.6.3. 集群自动部署步骤

在 home/adminop/install-console/install 目录下，执行 ice-install-manager.sh，按菜单目录输入序号，回车后执行对应安装部署程序。

除 ice-install-manager.sh 程序入口，也可进入/home/adminop/install-console/install/cluster-nonroot 目录(以下简称 cluster-nonroot 目录)，手工运行具体的服务集群安装程序，依照以下步骤安装部署各组件服务集群：

4.6.3.1. 安装 mongo 组件服务集群

- 执行 ./ice-install-cluster-mongo.sh
- 是否清理 Need to cleanup previous redis installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? y （如 mongo 安装为首个安装组件时，选择 y，其他情况选 n）
- 是否开始安装 Need to install expect&tcl tools (y/n)? y （如 mongo 安装为首个安装组件时，选择 y，其他情况选 n）
- 请确认本机 IP 地址 Please confirm local host ip address [use xxx.xxx.xxx.xxx by default]:（确认 默认 xxx.xxx.xxx.xxx 地址是否为本机 IP 地址，正确可直接回车；不正确，则手工输入正确的本机 IP 地址）
- 输入数字序号，自动上传并导入 mongo 数据包文件

4.6.3.2. 安装 redis 组件服务集群

- 执行 ./ice-install-cluster-redis.sh
- 是否清理 Need to cleanup previous redis installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send redis cluster installation package to cluster node (y/n)? y

IP 检查项：

- redis/ice-redis_cluster-config-master.sh 中的 HOST_IP_1、HOST_IP_2、HOST_IP_3 为 redis 服务器的 IP
- redis/ice-redis_cluster-config-slave.sh 中的 HOST_IP_1、HOST_IP_2、HOST_IP_3 为 redis 服务器的 IP
- redis/700X/conf/redis.conf 中 bind X.X.X.X 为当前 redis 服务器的 IP

4.6.3.3. 安装 pulsar 组件服务集群

- 执行 ./ice-install-cluster-pulsar.sh

- 是否清理 Need to cleanup previous pulsar installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send pulsar cluster installation package to cluster node (y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node (y/n)? y

4.6.3.4. 安装 dc 组件服务集群

- 执行 ./ice-install-cluster-dc.sh
- 是否清理 Need to cleanup previous sub-components installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send dc cluster installation package to cluster node(y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node (y/n)? n

4.6.3.5. 安装 sa 组件服务集群

- 执行 ./ice-install-cluster-sa.sh
- 是否清理 Need to cleanup previous sub-components installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send sa cluster installation package to cluster node (y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node (y/n)? n

4.6.3.6. 安装 openapi 组件服务集群

- 执行 ./ice-install-cluster-openapi.sh
- 是否清理 Need to cleanup previous sub-components installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send openapi cluster installation package to cluster node(y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node (y/n)? n

4.6.3.7. 安装 subagent 组件服务集群

- 执行 ./ice-install-cluster-subagent.sh
- 是否清理 Need to cleanup previous sub-components installation (y/n)? n

- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send subagent cluster installation package to cluster node(y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node (y/n)? 按需填写 y / n
- Need to installation discover tool packages on cluster node(y/n)? y

4.6.3.8. 安装 flink 组件服务集群

- 执行 ./ice-install-cluster-flink.sh
- 是否清理 Need to cleanup previous flink installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send flink cluster installation package to cluster node (y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node?(y/n) n

4.6.3.9. 安装 mainserver 组件服务集群

- 执行 ./ice-install-cluster-mainserver.sh
- 是否清理 Need to cleanup previous mainserver installation (y/n)? n
- 是否清理 Need to cleanup previous nginx installation (y/n)? n
- 是否安装和解压主安装包 Need to install ice core package (y/n)? n
- 是否开始安装 Need to install expect&tcl tools (y/n)? n
- 发送安装包/数据包 Send mainserver cluster installation package to cluster node (y/n)? y
- 发送 JRE 安装包 Send jre installation package to cluster node (y/n)? n

注意：1、集群版安装单机 pg 时，要及时修改 dc、openapi 的 run.sh 脚本的 pg 设置（单机 pg，采用的是原生安装方式，对外就是原始的 ip 和原始的端口 5432；集群 pg，通常采用 pg+pgpool 方式部署，所以，对外暴露的是 vip 和 9999 端口）；
2、修改 main 的 conf 下的 jdbc.properties 的 pg 配置。
3、集群安装完毕后要按照客户的网络规划设计 subagent 和 sa 的采集设置。

1.1. Subagent 和 sa 配置说明

1.1.1. 根据现场部署情况，规划 subagent 部署策略

- 模式 A：如果采用分网段采集，且分网段内 subagent 为单节点，那么，subagent 建议按照所在采集服务器 ip 配置；
- 模式 B：如果采用分网段采集，且分网段内 subagent 为多节点，那么，subagent 建议按照约定命名配置；
- 模式 C：如果不区分网段采集，且 subagent 为单节点，那么，subagent 建议采用单机默认配置；
- 模式 D：如果不区分网段采集，且 subagent 为多节点，那么，subagent 采用统一命名方式配置。

1.1.2. subagent 具体配置

- 【必需】修改文件：$ICE_HOME/app/subagent/route/conf/application.yml
- SUB_NAME_SCAN_INDEX：
  - ✧ 模式 A：采集服务器 IP
  - ✧ 模式 B：NetA-1、NetA-2、NetB-1、NetB-2
  - ✧ 模式 C：采集服务器 IP
  - ✧ 模式 D：Sub-XXX （自定义名称）
- SUB_TYPE_SCAN_INDEX：（SHARED 共享模式、FAILOVER 灾备模式）
  - ✧ 模式 A：SHARED
  - ✧ 模式 B：统一为 SHARED，或者 FAILOVER，均可
  - ✧ 模式 C：SHARED
  - ✧ 模式 D：统一为 SHARED，或者 FAILOVER，均可

1.1.3. sa 具体配置

- 【必需】修改文件：$ICE_HOME/app/sa/conf/application.yml

NOTE：在 sa application.yml 中需配置 subagent 选项下的 sub-a、sub-b、sub-c ...等子选项。按 subagent application.yml 中所定义的 SUB_NAME_SCAN_INDEX 选项的不同情况，sa application.yml 中 subagent 选项配置方法分为以下几种：

1. 如各 subagent 节点上的 **subagent application.yml** 中按**模式 D** 设置为相同的 **SUB_NAME_SCAN_INDEX**，则这些 subagent 节点视为同组节点。在 **sa application.yml 中仅需将** sub-a 选项下的 **ip 参数**设置为 subagent 共同的 **SUB_NAME_SCAN_INDEX**；

2. 如各 subagent 节点上的 **subagent application.yml** 中按**模式 D 以外的其他模式**设置的独立的 **SUB_NAME_SCAN_INDEX**，则这些 subagent 节点视为不同组节点。在 **sa application.yml** 中 sub-a、sub-b、sub-c …选项下 ip 参数应设置为各 **subagent application.yml** 中定义的 **SUB_NAME_SCAN_INDEX**；

- 【按需】修改文件：app/sa/route/ice/SceneIndexSender.xml
  - ✧ 严格按照 subagent 集群节点的部署规划，添加相应的<when></when>节点
  - ✧ 在<simple></simple>中添加${exchangeProperty.scanTarget}判定逻辑（即：根据采集目标）
    - ■ 支持的判断逻辑：startsWith 以 xxx 字符开始、contains 包含 xxx 字符、== 等于 xxx 字符
    - ■ 判断逻辑支持逻辑与（&&）逻辑或（||）
    - ■ 示例：<simple>${exchangeProperty.scanTarget} startsWith '22.9.40.' || ${exchangeProperty.scanTarget} contains '9.104.47.76' </simple>
  - ✧ 在<toD uri="pulsarDiscover:persistent://public/default/IceScanIndex_{{ice.subagent.sub-a.ip}}?producerName={{ice.core.my-ip}}_SA"/>中修改{{ice.subagent.sub-a.ip}}为对应的 subagent 的 SUB_NAME_SCAN_INDEX