kubernetes 集群一键部署脚本使用说明

# kubernetes环境搭建

## 前提条件

* 请保证集群所有节点之间网络互通，并且Node与Master之间已经创建好互信（参考数据库的互信：https://opengauss.org/zh/docs/2.0.1/docs/installation/%E5%88%9D%E5%A7%8B%E5%8C%96%E5%AE%89%E8%A3%85%E7%8E%AF%E5%A2%83.html#%E6%89%8B%E5%B7%A5%E5%BB%BA%E7%AB%8B%E4%BA%92%E4%BF%A1）。
* 请保证所有机器能连接外网。
* 本脚本请使用root账户执行。
* 本脚本目前仅支持CentOS系统。
* 脚本下载路径https://gitee.com/ricardo\_cui/openGauss-distributed-solutions.git

## 软件版本

一键部署脚本将安装如下软件：

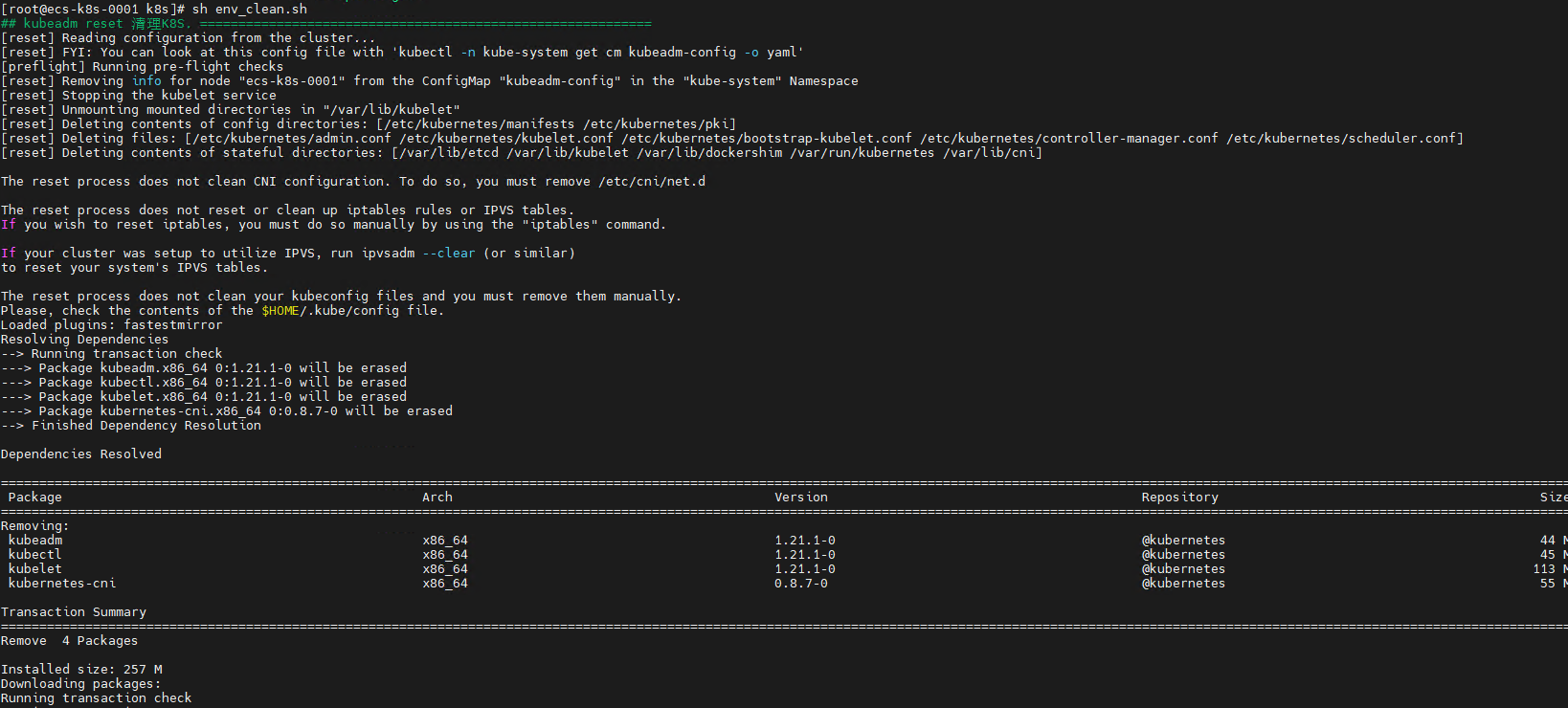
* kubeadm v1.21.1
* kubectl v1.21.1
* docker 18.06.1-ce

## 安装步骤

### 环境清理

请在脚本目录运行以下命令：本脚本将删除所有容器镜像，然后删除docker，删除kubeadm kubelet。如果为全新环境可以跳过此步骤。

sh env\_clean.sh



### 创建配置文件profile

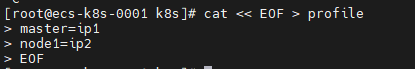
请用如下方式创建配置文件，并正确写入mater以及多个node的ip地址。

cat << EOF >> profile

master=ip1

node1=ip2

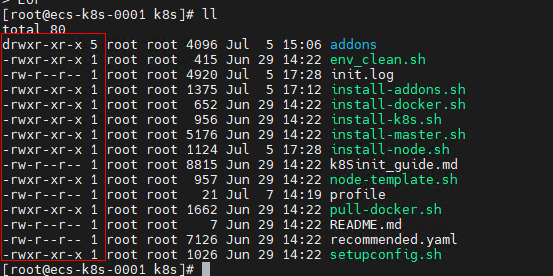
EOF



### 在master节点使用root账户执行如下命令

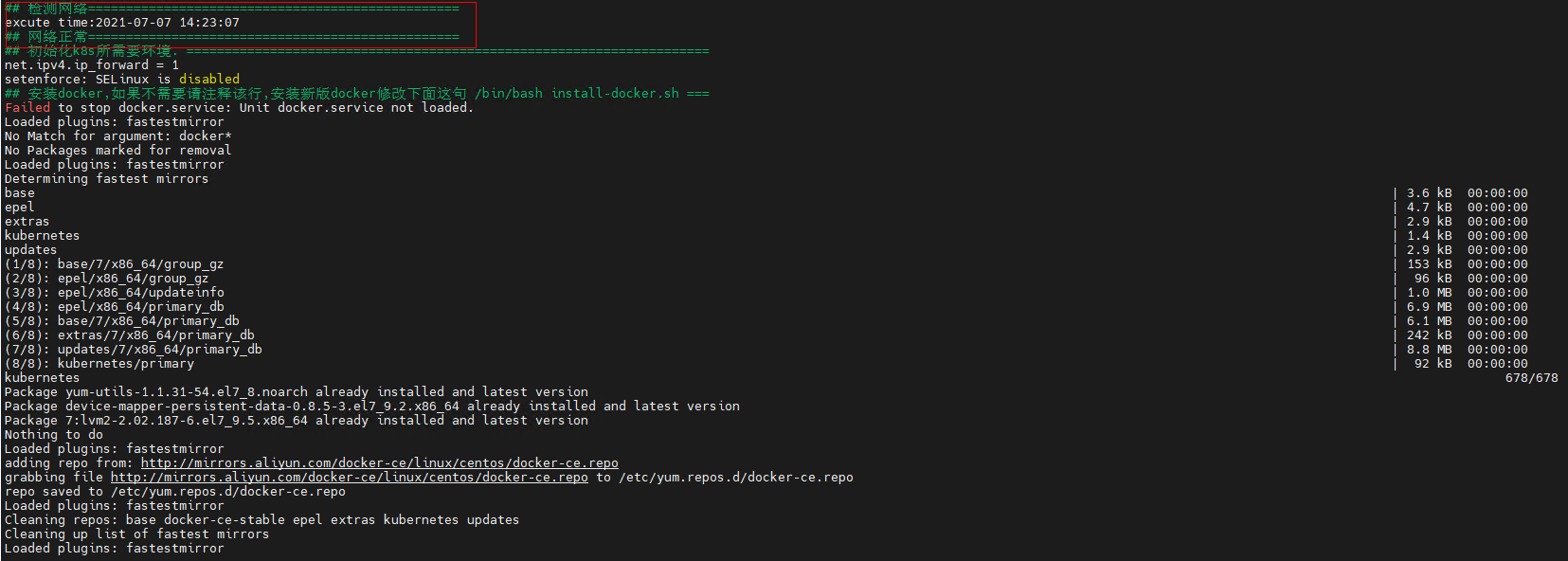
1. chmod +x \*.sh

保证脚本为用户为root且拥有可执行权限。

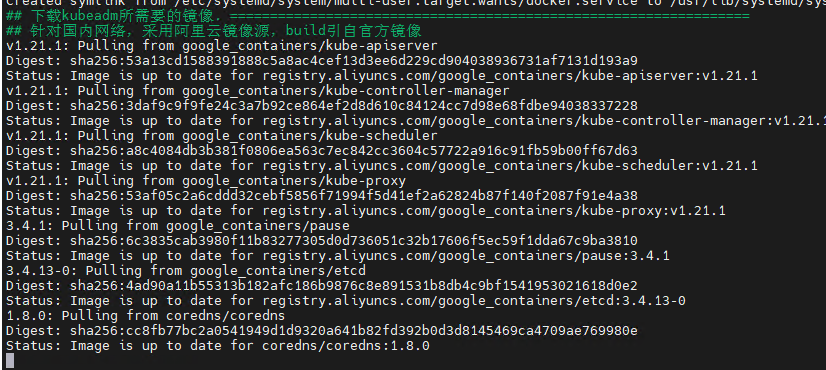


1. sh install-master.sh

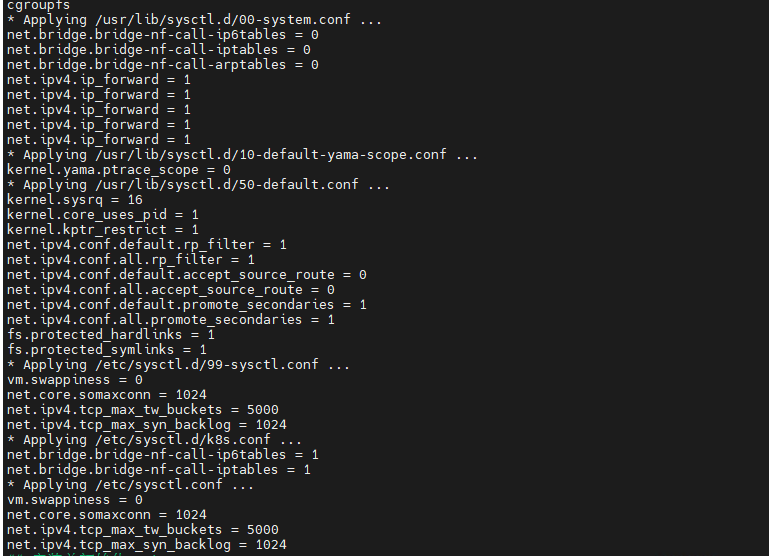
安装脚本首先检测各个节点能否连接外网，如果节点未建立互信会提示输入密码。接着yum安装docker如下图所示。



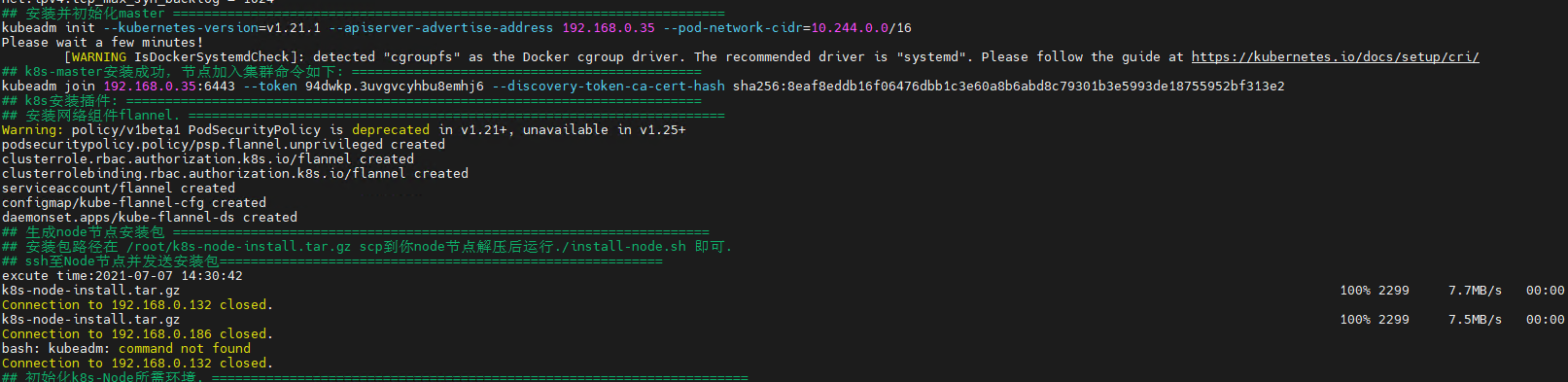
docker安装完成后下载kubernetes所需镜像。



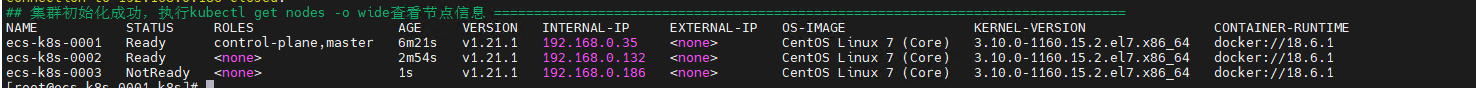
镜像下载完成后会安装kubeadm接着进行网络环境设置。



接着会将安装脚本发送至node节点。在node节点重新执行上述流程，并执行下图中mater生成的命令将node加入到集群中。集群基于kubeadm工具初始化，初始化日志在master脚本文件夹下的init.log中。



Node节点全部初始化并加入集群后会打印如下信息，STATUS为Ready则表示初始化集群成功。集群安装后会立即查询状态，根据机器条件不同可能Ready时间不一致。可以过几分钟再次在mater节点执行kubectl get nodes –o wide 命令查看node状态。



### 备注说明

本脚本会在所有节点部署kubernetes，同时重新安装docker。如果无需重新安装docker，请注释掉install-master.sh脚本中的`/bin/bash install-docker.sh new`，并取消注释`/bin/bash install-docker.sh`

# 2. 基于openGauss的kubernetes一键化部署

## 前提条件

1. k8s集群环境均安装python3
2. 下载镜像pengauss、shardingphere、haproxy镜像（版本1.0.0）并保证所有node节点均导入以上三个镜像等。**镜像载入方式docker load < 镜像包名**，镜像包名分别为**opengauss100.tar.gz**，**shardingsphere100.tar.gz**，**haproxy100.tar.gz。**
3. 已执行完第一章的环境搭建脚本 。

## 安装步骤

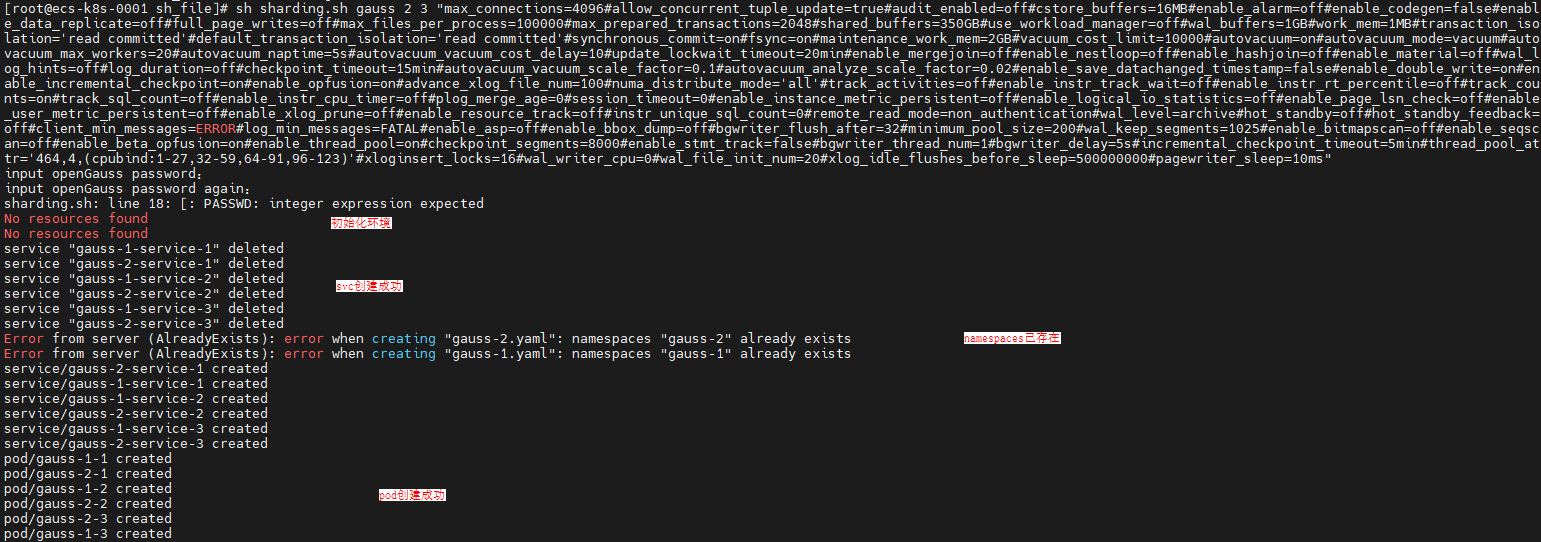
### 创建shardingphere集群分片

sh sharding.sh 自定义集群name 分片个数 单个分片数据库个数（不小于3，不大于9） 数据库配置参数（可选）

**说明：**执行后，**需要输入一次自定义的数据库密码（满足openGauss数据库密码要求）**，最后会生成一个user\_input.yaml文件。请自行补充完user\_input.yaml内容（请参考附录章节）。

**数据库配置参数：示例：xxx=xxx#xxx=xxx（分隔符：#），该参数需要使用英文“”包含。**

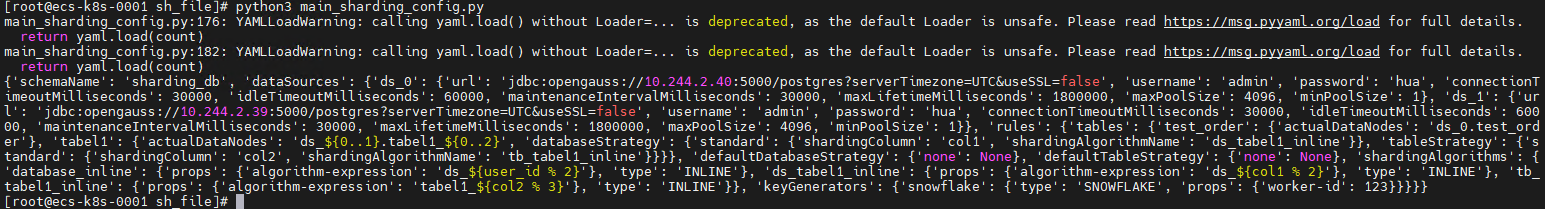
**sh sharding.sh opengauss 2 3 “max\_connections=4096#allow\_concurrent\_tuple\_update=true”**



### 生成shardingphere配置文件

python3 main\_sharding\_config.py

**说明：**执行完成后，会生成config-sharding\_tmp.yaml文件。用户也可自行修改该配置文件用以定制化配置。



### 部署shardingphere服务

sh delpoy\_sharding.sh 自定义项目name 配置文件存放路径（绝对路径） 服务部署的node节点hostname

**说明：**其他shardingphere配置文件（如server.yaml）如需修改，请提前将修改好的文件（名称不能改）放置到：**命令中【服务部署的node节点hostname】节点的【配置文件存放路径】。**

Shardingphere其他配置文件请参考：<https://github.com/apache/shardingsphere/tree/master/shardingsphere-proxy/shardingsphere-proxy-bootstrap/src/main/resources>



## 测试链接

使用gsql远程连接shardingphere对外暴露的接口，如下示例：

gsql -d sharding\_db -p 30400 -r -h nodeIP/masterIP -U root -W root

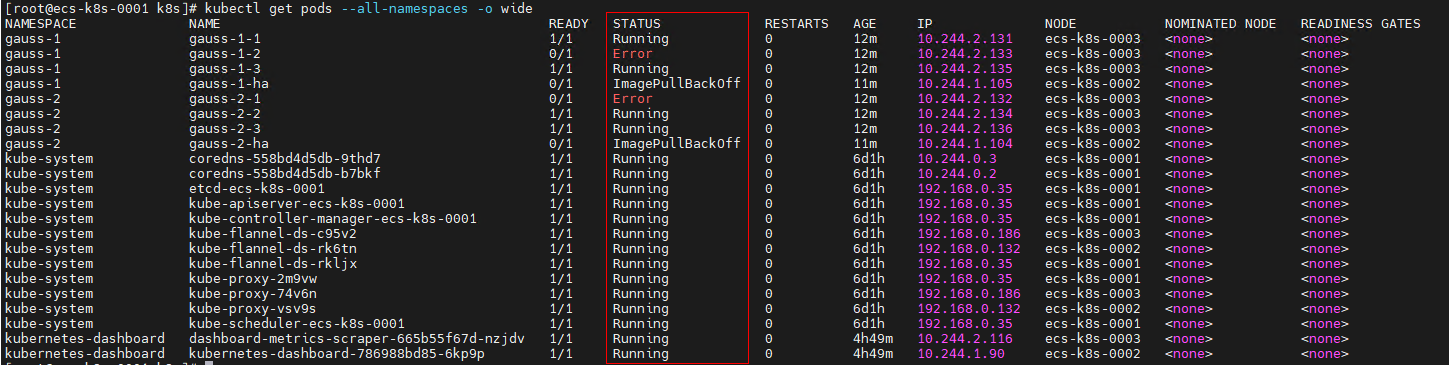
备注：以上脚本可重复执行，重复执行会先删除原有的项目。

# Kubernetes常用命令

## 查看pod/svc状态

kubectl get pod –all-namespaces –o wide

kubectl get svc –all-namespaces –o wide



状态为running的pod为正常，其他状态为异常。

## 查看pod日志

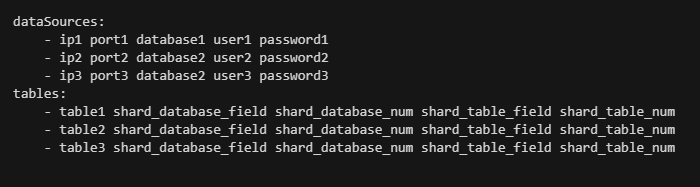
kubectl logs podname –n namespace

例：kubectl logs gauss-1-1 –n gauss-1

状态为Running的时候，镜像内部也可能报错，此时就需要查看pod日志进行错误排查。

# 附录

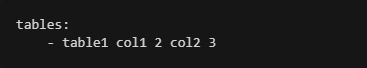
## user\_input内容说明



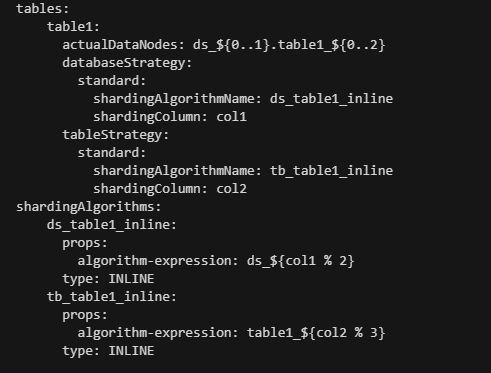
user\_input为用户输入文件，其中配置与shardingsphere相关的数据源和表的信息，配置项包括如下两类：

* dataSources指定数据源，以ip port database user password五个字段组成，其将对应连接数据库的url信息；
* tables指定表的分库分表信息，以表名 分库字段 分库数量 分表字段 分表数量五个字段组成，其中分库分表算法默认为取模算法；如果分库数量为1，则表示不分库，数据默认在ds\_0数据源；如果分表数量为1，则表示不分表。其他特殊的配置，如绑定表，其他分片算法等信息，用户需在config-sharding\_tmp.yaml文件中进行配置。

配置示例：



生成的分库分表信息如下：



## 容器默认进入为root用户，使用数据库需要切到omm用户，使用su omm（不要使用su - omm）切换 保留root的环境变量

## 持久化配置

默认的持久化地址在：/data/{数据库所在pod的名称}/；如需修改，请参考patroni.sh中创建pod相关的配置文件。

