**基础设施部署**

# 本机MAC

## 文件上传下载

配置文件备份目录：/Users/docker/基础设施备份

上传文件到服务器命令：scp -r /Users/docker/基础设施备份/kafka root@192.168.32.2:/usr/local/

下载服务器文件命令：scp -r root@192.168.32.2:/usr/local/redis-cluster/ /Users/docker/基础设施备份/

## jmeter压测运行

路径：/Users/soft/apache-jmeter-5.4.1/bin

执行：./jmeter.sh

# 192.168.32.2机器

## 容器挂载：

volumes配置

左边为服务器配置路径:右边为容器中配置路径

## 开放端口：

zookeeper

firewall-cmd --zone=public --add-port=2181/tcp --permanent

firewall-cmd --zone=public --add-port=2888/tcp --permanent

firewall-cmd --zone=public --add-port=3888/tcp --permanent

Kafka

firewall-cmd --zone=public --add-port=9092/tcp --permanent

firewall-cmd --zone=public --add-port=9093/tcp --permanent

firewall-cmd --zone=public --add-port=9094/tcp --permanent

Kafka-manager

firewall-cmd --zone=public --add-port=9000/tcp --permanent

firewall-cmd --zone=public --add-port=9999/tcp --permanent

firewall-cmd --zone=public --add-port=9998/tcp --permanent

firewall-cmd --zone=public --add-port=9997/tcp --permanent

mysql

firewall-cmd --zone=public --add-port=3306/tcp --permanent

redis

firewall-cmd --zone=public --add-port=6391/tcp --permanent

firewall-cmd --zone=public --add-port=6392/tcp --permanent

firewall-cmd --zone=public --add-port=6393/tcp --permanent

firewall-cmd --zone=public --add-port=6394/tcp --permanent

firewall-cmd --zone=public --add-port=6395/tcp --permanent

firewall-cmd --zone=public --add-port=6396/tcp --permanent

## 重启防火墙

firewall-cmd --reload

## zookeeper集群部署

自定义网络区间：docker network create --driver bridge --subnet 172.22.1.0/24 --gateway 172.22.1.1 zknet

路径：/usr/local/zookeeper-cluster/dockerFolder

部署命令：docker-compose -f docker-compose.ym up -d

## kafka集群部署

路径：/usr/local/kafka/

部署命令：docker-compose up -d

## 安装kafka\_exporter

wget <https://github.com/danielqsj/kafka_exporter/releases/download/v1.2.0/kafka_exporter-1.2.0.linux-amd64.tar.gz>

路径：/usr/local/kafka\_exporter

tar -xvf kafka\_exporter-1.2.0.linux-amd64.tar

cd kafka\_exporter-1.2.0.linux-amd64/

cd /usr/local/kafka\_exporter-1.2.0.linux-amd64

nohup ./kafka\_exporter --kafka.server=192.168.32.2:9092 &

## 安装prometheus

下载：

wget -c <https://github.com/prometheus/prometheus/releases/download/v2.15.0/prometheus-2.15.0.linux-amd64.tar.gz>

路径：/usr/local/prometheus

启动:

nohup ./prometheus --config.file=prometheus.yml &

## 安装grafana

路径：

docker pull grafana/grafana:7.3.6

docker run \

--user root \

-d \

-p 3000:3000 \

--name=grafana \

-v /home/grafana:/var/lib/grafana \

grafana/grafana:7.3.6

docker run \

--user root \

-d \

-p 3000:3000 \

--name=grafana \

-v /home/grafana:/var/lib/grafana \

grafana/grafana:5.3.2

## mysql部署（单节点）

路径：/usr/local/mysql

部署命令：docker-compose up -d

## Canal服务端部署--latest版本

1. 修改为mysql新增canal用户

mysql -uroot -proot

#创建账号(账号：canal;密码：canal）

CREATE USER canal IDENTIFIED BY 'canal';

#授予权限

GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON \*.\* TO 'canal'@'%';

#刷新并应用权限

FLUSH PRIVILEGES;

**(使用) 1.1.4版本**

路径：/usr/local/canal/jarrun/bin

部署命令：./startup.sh

**(弃用)**

路径：/usr/local/canal

部署命令：docker-compose up -d

**(弃用) 单机方式部署**

路径：/usr/local/canal

sh run.sh -e canal.admin.manager=192.168.32.2:8089 \

-e canal.admin.port=11110 \

-e canal.admin.user=admin \

-e canal.admin.passwd=6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9

## canal-admin部署--latest版本

**(使用) 1.1.4版本**

路径：/usr/local/canal-admin/jarrun/bin

部署命令：./startup.sh

**(舍弃)**

路径：/usr/local/canal-admin

部署命令：docker-compose up -d

**(舍弃)**

路径：/usr/local/canal-admin

sh run\_admin.sh \

-e server.port=8089 \

-e spring.datasource.address=192.168.32.2:3307 \

-e spring.datasource.database=canal\_manager \

-e spring.datasource.username=root \

-e canal.adminUser=admin \

-e canal.adminPasswd=123456 \

-e spring.datasource.password=root

## redis部署

路径:/usr/local/redis-cluster

执行: docker-compose up -d

启动Redis之后，三主三从，使用trib构建集群

docker run --rm -it zvelo/redis-trib create --replicas 1 192.168.32.2:6391 192.168.32.2:6392 192.168.32.2:6393 192.168.32.2:6394 192.168.32.2:6395 192.168.32.2:6396

## xxl-job部署（需下载源码，先执行sql）

下载镜像

docker pull xuxueli/xxl-job-admin:2.3.0

路径：/usr/local/xxl-job

部署命令：docker-compose up -d

## seata-server部署

路径：/usr/local/seata

执行命令：docker-compose up -d

将config.txt文件放到nacos-config.sh文件所在目录的上一级目录（nacos-config.sh不作任何修改）

sh nacos-config.sh -h {nacos服务的ip} -u {nacos用户名} -w {nacos密码}

sh nacos-config.sh -h 192.168.32.2 -p 8845 -g SEATA\_GROUP -t seata -u nacos -w nacos

## nacos2.0部署集成

9848

# 解决方案

## 分布式事务seata

seata使用测试，使用版本1.3.0

启动seata-server

启动compose-admin

启动compose-alert

启动compose-sync

调用接口alert服务中的接口：创建订单

POST

<http://127.0.0.1:8582/compose-alert/torder/create?userId=abc123&storageId=1&num=2>

库数据库

compose-admin 中account表修改money数值

compose-sync 中storage表修改num数值

compose-alert 中t\_order表新增订单数据

## 分布式定时任务xxl-job