# 应用部署文档

# 1.docker搭建mysql主从复制实现读写分离

**文档地址<https://blog.csdn.net/qq_40942490/article/details/109158285>**

## 2.部署nacos

### 2.1nacos配置持久化操作

|  |
| --- |
| cd /home/aisys/nacos/conf  vim application.properties |

### 2.2数据库mysql文件



上传到linux服务导入mysql数据库

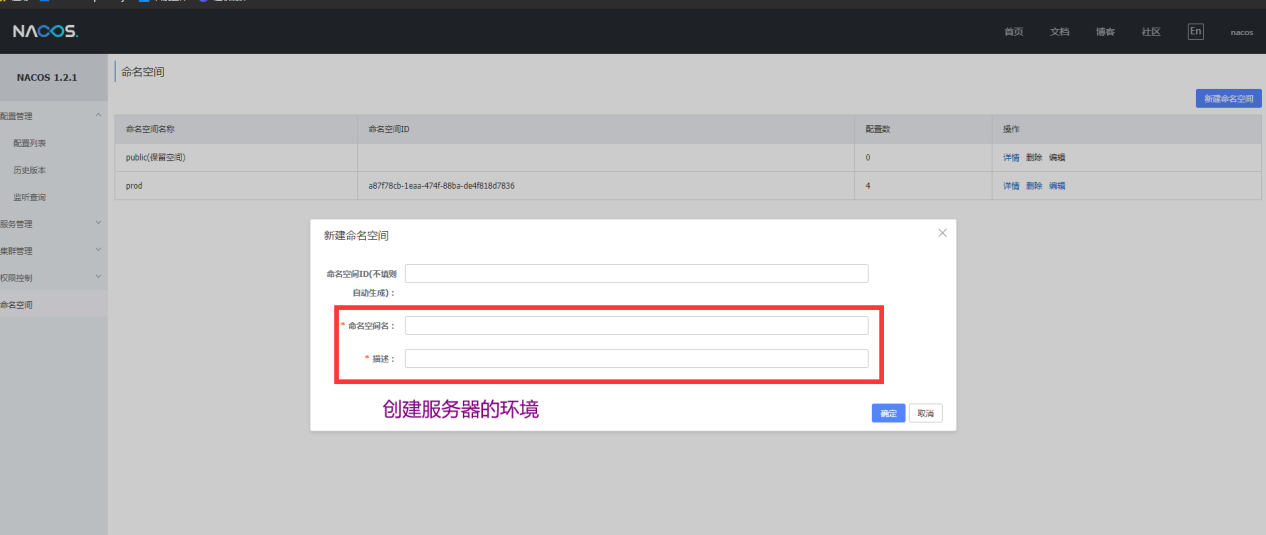
命令：<https://blog.csdn.net/H_Ystar/article/details/98470742>

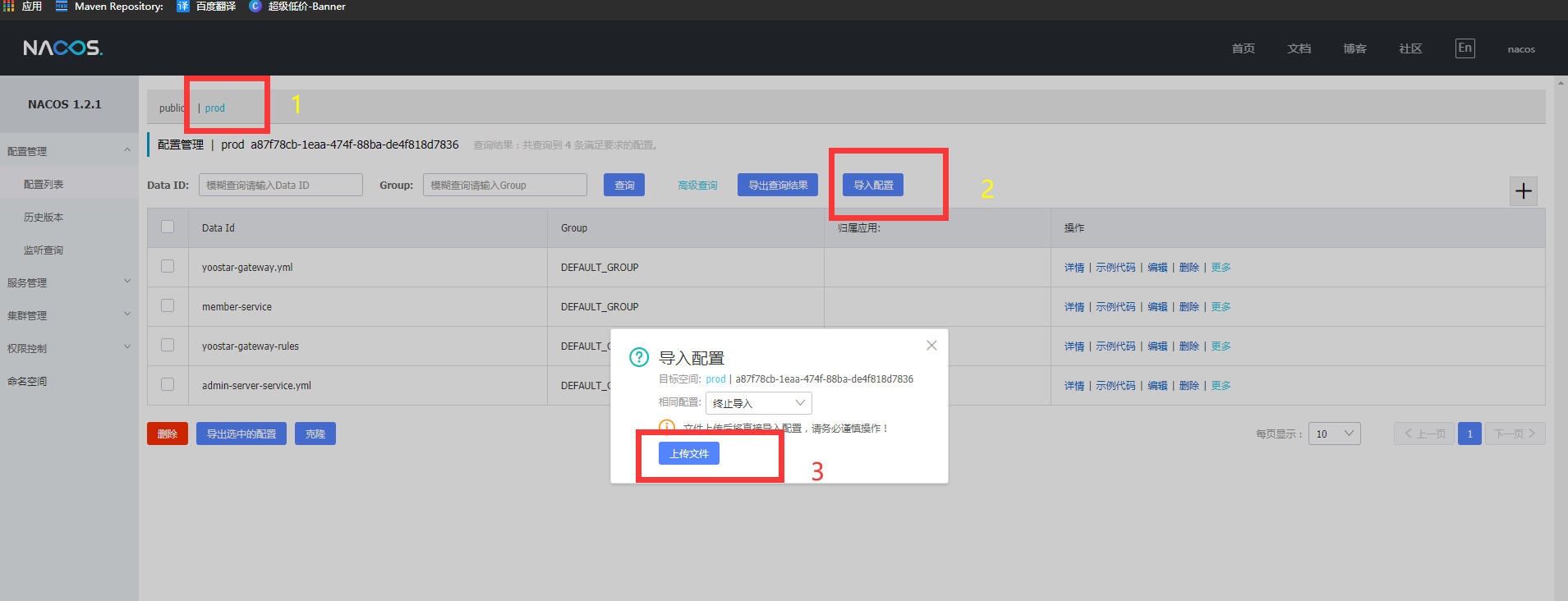
### 2.3运行

|  |
| --- |
| cd /home/aisys/nacos/bin  Centos：sh startup.sh -m standalone  Ubuntu：bash startup.sh -m standalone |

### 2.4在nacos中导入配置文件

<http://0.0.0.0:8848/nacos> 默认账号密码为nacos





# 3.部署sentinel

## 3.1运行sentinel

|  |
| --- |
| cd /home/aisys  nohup java -jar sentinel-dashboard-1.7.0.jar --server.port=8888 > sentinel.log  2>&1 & |

# 4.部署xxl-job-admin

## 4.1运行xxl-job-admin

|  |
| --- |
| cd /home/aisys/xxl-job  nohup java -jar xxl-job-admin-2.2.1-SNAPSHOT.jar > xxl-job.log  2>&1 & |

# 5.部署zipkin

## 5.1运行zipkin

### 5.1.1没有分布式日志收集运行方式

|  |
| --- |
| cd /home/aisys  nohup java -jar zipkin-server-2.12.9-exec.jar > zipkin.log  2>&1 & |

### 5.1.2有分布式日志运行方式（请更换对应的服务器ip）

|  |
| --- |
| nohup java -jar zipkin-server-2.19.0-exec.jar --KAFKA\_BOOTSTRAP\_SERVERS=IMG_25610.20.22.30:9092 --STORAGE\_TYPE=elasticsearch --ES\_HOSTS=IMG_257http://10.20.22.30:9200 >output 2>&1 & 访问 IMG_258http://10.20.22.30:9411/zipkin/ 即可查看zipkin |

# 6.修改nginx配置

|  |
| --- |
| cd /etc/nginx/conf.d  vim /member.conf |

|  |
| --- |
| server {  listen 80;  server\_name member.quicktvod.com;  location / {  root /home/aisys/quick/dist/;  server {  listen 80;  server\_name member.quicktvod.com;  location / {  root /home/aisys/quick/dist/;  index index.html;  try\_files $uri $uri/ /index.html;  }  location /quick/ {  proxy\_pass http://localhost:8080/;  proxy\_set\_header Equipment-Type 2;  # proxy\_cookie\_path / /quick;  }  location /google/ {  proxy\_pass https://translate.google.cn/;  add\_header 'Access-Control-Allow-Origin' "\*";  add\_header 'Access-Control-Allow-Credentials' 'true';  add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';  add\_header 'Access-Control-Allow-Headers' 'x-requested-with,Authorization';  }  }  server {  listen 8082;  server\_name member.superlifenjoy.com 34.203.11.129;  location / {  root /home/aisys/super/dist/;  index index.html;  try\_files $uri $uri/ /index.html;  }  location /super/ {  proxy\_pass http://localhost:8080/;  proxy\_set\_header Equipment-Type 1;  # proxy\_cookie\_path / /quick;  }  location /google/ {  proxy\_pass https://translate.google.cn/;  add\_header 'Access-Control-Allow-Origin' "\*";  add\_header 'Access-Control-Allow-Credentials' 'true';  add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';  add\_header 'Access-Control-Allow-Headers' 'x-requested-with,Authorization';  }  } |

|  |
| --- |
| cd /etc/nginx  vim /nginx.conf |

**开启前端静态资源加速在http{ }配置里面加入**

|  |
| --- |
| **# Gzip Settings**  **##**  **#启用或禁用gzipping响应。#**  **gzip on;**  **#设置用于压缩响应的缓冲区number和size。默认情况下，缓冲区大小等于一个内存页面。这是4K或8K，具体取决于平台。#**  **gzip\_static on;**  **#启用或禁用gzipping响应。#**  **gzip\_buffers 4 16k;**  **#设置level响应的gzip压缩。可接受的值范围为1到9。#**  **gzip\_comp\_level 5;**  **#设置将被gzip压缩的响应的最小长度。长度仅由“Content-Length”响应头字段确定。#**  **gzip\_min\_length 100;**  **#匹配MIME类型进行压缩，text/html默认被压缩。#**  **gzip\_types text/plain application/javascript text/css application/xml text/javascript application/x-httpd-php image/jpeg image/gif image/png;** |

# 7.服务部署

## 7.1运行member-service

|  |
| --- |
| nohup java -jar member-service-1.0-SNAPSHOT.jar > member.log  2>&1 & |

## 7.1运行yoostar-gateway

|  |
| --- |
| nohup java -jar yoostar-gateway-1.0-SNAPSHOT.jar > gateway.log  2>&1 & |

# 8.部署canal

**canal实现mysql与redis数据同步，以及数据库实时备份**

### **8.1 将canal.deployer 上传到/home/aisys目录下**

ar -zxvf **[canal.deployer-1.1.5-SNAPSHOT.tar.gz](https://github.com/alibaba/canal/releases/download/canal-1.1.5-alpha-2/canal.deployer-1.1.5-SNAPSHOT.tar.gz)**

### **8.2 配置修改参数**

修改instance 配置文件 vi conf/example/instance.properties



**修改canal 配置文件vi /usr/local/canal/conf/canal.properties**



### mq相关参数说明

| **参数名** | **参数说明** | **默认值** |
| --- | --- | --- |
| canal.mq.servers | kafka为bootstrap.servers rocketMQ中为nameserver列表 | 127.0.0.1:6667 |
| canal.mq.retries | 发送失败重试次数 | 0 |
| canal.mq.batchSize | kafka为ProducerConfig.BATCH\_SIZE\_CONFIG rocketMQ无意义 | 16384 |
| canal.mq.maxRequestSize | kafka为ProducerConfig.MAX\_REQUEST\_SIZE\_CONFIG rocketMQ无意义 | 1048576 |
| canal.mq.lingerMs | kafka为ProducerConfig.LINGER\_MS\_CONFIG , 如果是flatMessage格式建议将该值调大, 如: 200 rocketMQ无意义 | 1 |
| canal.mq.bufferMemory | kafka为ProducerConfig.BUFFER\_MEMORY\_CONFIG rocketMQ无意义 | 33554432 |
| canal.mq.acks | kafka为ProducerConfig.ACKS\_CONFIG rocketMQ无意义 | all |
| canal.mq.kafka.kerberos.enable | kafka为ProducerConfig.ACKS\_CONFIG rocketMQ无意义 | false |
| canal.mq.kafka.kerberos.krb5FilePath | kafka kerberos认证 rocketMQ无意义 | ../conf/kerberos/krb5.conf |
| canal.mq.kafka.kerberos.jaasFilePath | kafka kerberos认证 rocketMQ无意义 | ../conf/kerberos/jaas.conf |
| canal.mq.producerGroup | kafka无意义 rocketMQ为ProducerGroup名 | Canal-Producer |
| canal.mq.accessChannel | kafka无意义 rocketMQ为channel模式，如果为aliyun则配置为cloud | local |
| --- | --- | --- |
| canal.mq.vhost= | rabbitMQ配置 | 无 |
| canal.mq.exchange= | rabbitMQ配置 | 无 |
| canal.mq.username= | rabbitMQ配置 | 无 |
| canal.mq.password= | rabbitMQ配置 | 无 |
| canal.mq.aliyunuid= | rabbitMQ配置 | 无 |
| --- | --- | --- |
| canal.mq.canalBatchSize | 获取canal数据的批次大小 | 50 |
| canal.mq.canalGetTimeout | 获取canal数据的超时时间 | 100 |
| canal.mq.parallelThreadSize | mq数据转换并行处理的并发度 | 8 |
| canal.mq.flatMessage | 是否为json格式 如果设置为false,对应MQ收到的消息为protobuf格式 需要通过CanalMessageDeserializer进行解码 | false |
| --- | --- | --- |
| canal.mq.topic | mq里的topic名 | 无 |
| canal.mq.dynamicTopic | mq里的动态topic规则, 1.1.3版本支持 | 无 |
| canal.mq.partition | 单队列模式的分区下标， | 1 |
| canal.mq.partitionsNum | 散列模式的分区数 | 无 |
| canal.mq.partitionHash | 散列规则定义 库名.表名 : 唯一主键，比如mytest.person: id 1.1.3版本支持新语法，见下文 | 无 |