秒杀

安装部署手册

|  |  |  |
| --- | --- | --- |
| **文件状态：**  **[ ] 草稿**  **[ ] 正在修改**  **[ √ ] 正式签收** | **文件标识：** | **MIS** |
| **当前版本：** | **1.0** |
| **作 者：** | **沈坤林** |
| **完成日期：** | **2020-11-06** |
| **签 收 人：** |  |
| **签收日期：** |  |

**二○二○年 十一 月 六 日**

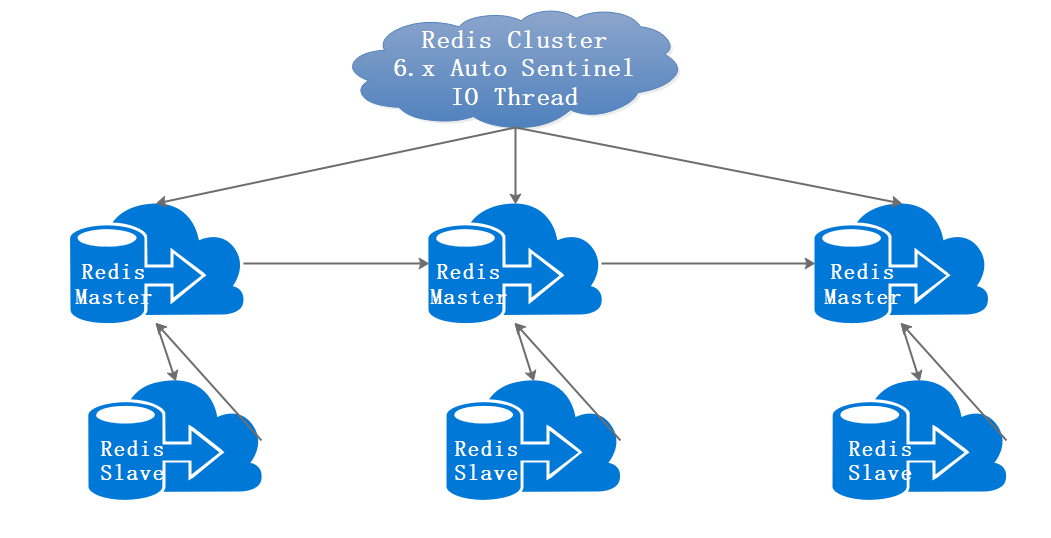
文档修改历史记录

|  |  |  |
| --- | --- | --- |
| 版本状态 | 修改日期 | 备注 |
| V1.0 | 2020-11-06 | 新建 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

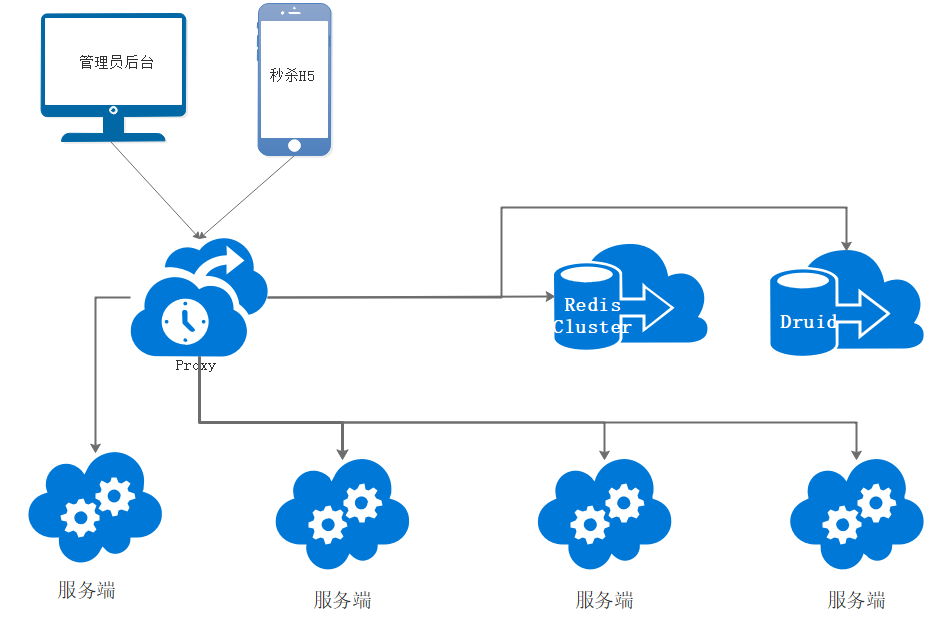
# 部署环境

## 系统部署结构

*Redis 部署结构*



*应用部署结构*



## 服务器配置

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | CPU | 内存 | 硬盘 | 系统 | 数量 | 备注 |
| 服务端 | 16 | 64G | 500G | Centos7 | 3 |  |
| API端 | 4 | 16G | 500G | Centos7 | 2 |  |
| 管理端 | 4 | 8G | 500G | Centos7 | 1 |  |
| 中间件 | 16 | 64G | 500G | Centos7 | 3 | 中间件集群部署（redis，mysql等） |

## 客户端配置

*无*

### 客户端机器配置要求

*无*

### 客户端机器系统软件环境

*管理端建议使用Chrome浏览器操作*

# 系统软硬件安装与配置参考

## 服务器操作系统和组件安装要点

Redis 安装

一：创建一个文件夹，用于适用docker-compose.yml文件

cd /usr/local #进入usr/local 目录

sudo mkdir -p docker/redis docker/sentinel #创建docker文件夹并在其下创建redis和sentinel文件夹

二：进入redis文件夹，并创建如下docker-compose.yml文件

cd docker/redis

sudo touch docker-compose.yml

三：编辑docker-compose.yml文件，并将如下内容复制进去，保存并退出

sudo vi docker-compose.yml

|  |
| --- |
| version: '3'  services:  master:  image: redis  container\_name: redis-master  ports:  - 6379:6379  slave1:  image: redis  container\_name: redis-slave-1  ports:  - 6380:6379  command: redis-server --slaveof redis-master 6379  slave2:  image: redis  container\_name: redis-slave-2  ports:  - 6381:6379  command: redis-server --slaveof redis-master 6379 |

四：进入刚刚创建的sentinel目录

cd /usr/local/docker/sentinel

五：创建docker-compose.yml文件，并将如下内容复制进去，保存并退出

sudo vi docker-compose.yml #编辑保存后自动创建这个文件

|  |
| --- |
| version: '3'  services:  sentinel1:  image: redis  container\_name: redis-sentinel-1  ports:  - 26379:26379  command: redis-sentinel /usr/local/etc/redis/sentinel.conf  volumes:  - ./sentinel1.conf:/usr/local/etc/redis/sentinel.conf  sentinel2:  image: redis  container\_name: redis-sentinel-2  ports:  - 26380:26379  command: redis-sentinel /usr/local/etc/redis/sentinel.conf  volumes:  - ./sentinel2.conf:/usr/local/etc/redis/sentinel.conf  sentinel3:  image: redis  container\_name: redis-sentinel-3  ports:  - 26381:26379  command: redis-sentinel /usr/local/etc/redis/sentinel.conf  volumes:  - ./sentinel3.conf:/usr/local/etc/redis/sentinel.conf |

六：创建哨兵文件，并将如下内容复制进去，保存并退出

sudo vi sentinel.conf

|  |
| --- |
| port 26379  dir /tmp  sentinel monitor mymaster 192.168.223.129 6379 2  sentinel down-after-milliseconds mymaster 30000  sentinel parallel-syncs mymaster 1  sentinel failover-timeout mymaster 180000  sentinel deny-scripts-reconfig yes |

七：将刚刚创建的sentinel.conf在创建3份，一模一样就可以了

sudo cp sentinel.conf sentinel1.conf

sudo cp sentinel.conf sentinel2.conf

sudo cp sentinel.conf sentinel3.conf

八：启动集群，先启动master-slave，在启动sentinel。

cd /usr/local/docker/redis

sudo docker-compose up -d #-d表示后台运行，就是守护态运行

cd /usr/local/docker/sentinel/

sudo docker-compose up -d #-d表示后台运行，就是守护态运行

Nacos 安装

一：下载镜像

docker pull nacos/nacos-server

二：启动

docker run -d -p 8848:8848 -e MODE=standalone --restart always --name nacos nacos/nacos-server

## 数据库服务器安装与配置

Mysql

一: 配置docker-compose.yml

|  |
| --- |
| version: '2'  services:  mysql-master:  build:  context: ./  dockerfile: master/Dockerfile  environment:  - "MYSQL\_ROOT\_PASSWORD=root"  - "MYSQL\_DATABASE=replicas\_db"  links:  - mysql-slave  ports:  - "33065:3306"  restart: always  hostname: mysql-master  mysql-slave:  build:  context: ./  dockerfile: slave/Dockerfile  environment:  - "MYSQL\_ROOT\_PASSWORD=root"  - "MYSQL\_DATABASE=replicas\_db"  ports:  - "33066:3306"  restart: always  hostname: mysql-slave |
|  |

二: 主数据库配置

配置Dockerfile

|  |
| --- |
| FROM mysql:5.7.17  MAINTAINER harrison  ADD ./master/my.cnf /etc/mysql/my.cnf |

配置my.cnf文件

|  |
| --- |
| [mysqld]  ## 设置server\_id，一般设置为IP，注意要唯一  server\_id=100  ## 复制过滤：也就是指定哪个数据库不用同步（mysql库一般不同步）  binlog-ignore-db=mysql  ## 开启二进制日志功能，可以随便取，最好有含义（关键就是这里了）  log-bin=replicas-mysql-bin  ## 为每个session分配的内存，在事务过程中用来存储二进制日志的缓存  binlog\_cache\_size=1M  ## 主从复制的格式（mixed,statement,row，默认格式是statement）  binlog\_format=mixed  ## 二进制日志自动删除/过期的天数。默认值为0，表示不自动删除。  expire\_logs\_days=7  ## 跳过主从复制中遇到的所有错误或指定类型的错误，避免slave端复制中断。  ## 如：1062错误是指一些主键重复，1032错误是因为主从数据库数据不一致  slave\_skip\_errors=1062 |

三: 从数据库配置

配置Dockerfile

|  |
| --- |
| FROM mysql:5.7.17  MAINTAINER harrison  ADD ./slave/my.cnf /etc/mysql/my.cnf |

配置my.cnf文件

|  |
| --- |
| [mysqld]  ## 设置server\_id，一般设置为IP，注意要唯一  server\_id=101  ## 复制过滤：也就是指定哪个数据库不用同步（mysql库一般不同步）  binlog-ignore-db=mysql  ## 开启二进制日志功能，以备Slave作为其它Slave的Master时使用  log-bin=replicas-mysql-slave1-bin  ## 为每个session 分配的内存，在事务过程中用来存储二进制日志的缓存  binlog\_cache\_size=1M  ## 主从复制的格式（mixed,statement,row，默认格式是statement）  binlog\_format=mixed  ## 二进制日志自动删除/过期的天数。默认值为0，表示不自动删除。  expire\_logs\_days=7  ## 跳过主从复制中遇到的所有错误或指定类型的错误，避免slave端复制中断。  ## 如：1062错误是指一些主键重复，1032错误是因为主从数据库数据不一致  slave\_skip\_errors=1062  ## relay\_log配置中继日志  relay\_log=replicas-mysql-relay-bin  ## log\_slave\_updates表示slave将复制事件写进自己的二进制日志  log\_slave\_updates=1  ## 防止改变数据(除了特殊的线程)  read\_only=1 |

五: 创建容器

进入 docker 目录，运行 docker-compose 启动命令。

$ docker-compose up -d

六: 配置从数据库

检查从库的起始状态

$ show master status;

检查主库的状态

$ show master status;

## 客户端用户权限设置

*无*

# 系统安装与配置

## 总体说明

*第一步：安装mysql*

*第二部：安装redis*

*第三步：导入sql脚本*

*第四步：安装nacos*

*第五步：导入nacos配置，并修改*

*第六步：上传应用*

*第七部：启动应用*

## 数据库数据初始化

*上传seckill\_general\_order.sql、seckill\_goods.sql、seckill\_manager.sql、seckill\_user.sql，并执行脚本初始化数据。*

## 系统安装部署

上传应用包



编写启动脚本startall.sh：

cd /usr/local/server/jar

nohup java -jar ./seckill-gateway-0.0.1-SNAPSHOT.jar &

nohup java -jar ./seckill-user-0.0.1-SNAPSHOT.jar &

nohup java -jar ./seckill-manager-0.0.1-SNAPSHOT.jar &

nohup java -jar ./seckill-goods-0.0.1-SNAPSHOT.jar &

nohup java -jar ./seckill-order-0.0.1-SNAPSHOT.jar &

nohup java -jar ./seckill-search-0.0.1-SNAPSHOT.jar &

nohup java -jar ./seckill-page-0.0.1-SNAPSHOT.jar &

启动脚本授权：

Chmod +x startall.sh

执行脚本启动程序

./startall.sh

## 自动更新服务部署

*无*

## 自动服务部署

*无*