**Koi鞋城Online项目部署文档**

## 环境准备

选择云服务器且虚拟机系统为centos7

## 关闭防火墙

对于阿里云服务器，阿里云控制台配有出站与入站规则。所以可以直接关闭防火墙。

|  |
| --- |
| Bash systemctl status firewalld #查看防火墙状态 systemctl stop firewalld #关闭防火墙 systemctl disable firewalld #防止重启虚拟机后防火墙打开 |

## 配置虚拟内存

因为云服务器内存为4G，若部署微服务超10个服务，4G物理内存，所以配置虚拟内存。

|  |
| --- |
| Bash dd if=/dev/zero of=/root/swapfile bs=1M count=8192 #创建内存为8G的虚拟内存  Bash # 构建 mkswap /root/swapfile #激活 swapon /root/swapfile  chmod 600 /root/swapfile |

至此虚拟内存已经配置成功，接下来开始配置微服务所需要的服务。

## 搭建环境

首先清理旧文件：

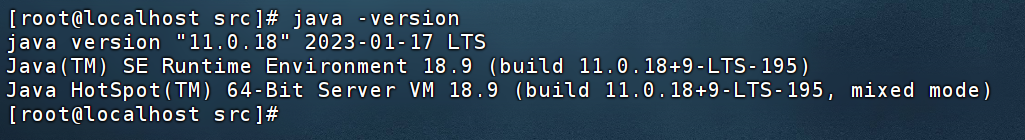
|  |
| --- |
| rpm -e --nodeps [name] |

安装docker：

|  |
| --- |
| yum install docker-ce docker-ce-cli containerd.io |

安装jdk：

|  |
| --- |
| 将jdk11的tar文件上传至虚拟机，然后通过tar -zxvf解压  配置环境遍历以及软链接 |



安装mysql：

|  |
| --- |
| 首先创建两个目录分别是数据目录和配置文件目录，用于docker部署后的挂载。  在配置文件目录下创建hmy.cnf配置文件，用于自定义mysql配置。  拉取镜像 docker pull mysql:8.0.23 #我们选用8.0.23的版本  最后通过docker命令启动镜像  docker run \  -d \  --name mysql \  --restart=always \  -e MYSQL\_ROOT\_PASSWORD=Root123321@ \  -p 3306:3306 \  -v /usr/local/src/mysql/conf:/etc/mysql/conf.d \  -v /usr/local/src/mysql/data:/var/lib/mysql \  mysql:8.0.23 |

安装nacos：

|  |
| --- |
| 拉取nacos镜像：docker pull nacos/nacos-server:v2.1.0  可以看到nacos镜像已经拉取成功，在数据库中创建nacos-config库用于nacos的数据持久化，创建nacos所需的表。不同的版本表的创建不同，可以在 github 上找到对应的表。  创建nacos挂载目录  最后启动nacos  docker run \  --restart=always \  --privileged=true \  --env MODE=standalone \  -v /usr/local/src/nacos/logs/:/home/nacos/logs/ \  -v /usr/local/src/nacos/conf/:/home/nacos/conf/ \  --restart=always \  -e PREFER\_HOST\_MODE=hostname \  --name nacos \  -d \  -p 8848:8848 \  -p 9848:9848 \  -p 9849:9849 \  nacos/nacos-server:v2.1.0 |

安装redis：

|  |
| --- |
| 拉群redis镜像：docker pull redis:7  创建redis配置文件以及数据文件，且配置redis.conf文件  最后启动redis  docker run \  -d \  --name redis \  -p 6379:6379 \  --restart=always \  -v /usr/local/src/redis/data:/data \  -v /usr/local/src/redis/conf:/etc/redis/redis.conf \  redis:7\  --appendonly yes \  --requirepass "123321" |

安装xxl-job：

|  |
| --- |
| 拉取镜像docker pull xuxueli/xxl-job-admin:2.3.0  创建配置文件夹以及在github上获取xxl-job配置文件  最后启动  docker run \  -e PARAMS="--spring.datasource.url=jdbc:mysql://192.168.150.102:3306/xxl\_job?Unicode=true&characterEncoding=UTF-8 \  --spring.datasource.username=root \  --spring.datasource.password=123" \  --restart=always \  -p 8880:8880 \  -v /usr/local/src/xxl-job/logs:/data/applogs \  -v /usr/local/src/xxl-job/application.properties:/application.properties \  --name xxljob \  -d \  xuxueli/xxl-job-admin:2.3.0 |

安装nginx

|  |
| --- |
| 下载nginx需要的插件  yum -y install gcc  yum install -y pcre pcre-devel  yum install -y zlib zlib-devel  yum install -y openssl openssl-devel  上传nginx文件到服务器上，最后启动nginx。  为nginx配置自启动。 |

## 部署后端服务

通过maven自带的package命令，打包所有的服务。每个服务都获得一个jar包。

上传至云服务器的指定目录下。且给每个文件夹配置Makefile与Dockerfile文件。并配置时区。

|  |
| --- |
| #Makefile：  default:  docker build -t xc-promotion .  docker run --name xc-promotion -e TZ="Asia/Shanghai" -p 8092:8092 -d xc-promotion  clean:  docker stop xc-promotion  docker rm xc-promotion  docker rmi xc-promotion  #Dockerfile：  FROM openjdk:11.0-jre-buster  ENV TZ=Asia/Shanghai  RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone  VOLUME ["/tmp","/log"]  COPY xc-promotion.jar xc-promotion.jar  EXPOSE 8092  ENTRYPOINT ["java","-jar","-Xmx512m","./xc-promotion.jar","&"] |

最后通过执行make命令启动后端微服务。

## 部署前端服务

对于用户端和管理端分别执行npm run build进行打包。将打包的文件上传到服务器指定文件夹下。修改nginx.conf文件。

|  |
| --- |
| server {  listen 80;  server\_name localhost  location / {  root /usr/local/src/xc-portal;  index index.html index.htm; }  }  server {  listen 18082;  server\_name localhost;  location / {  root /usr/local/src/xc-admin;  index index.html index.htm; }  } |

最后重启nginx即可访问。