**演示环境(uat)搭建步骤**

# 前言

uat环境用来为前端提供接口做演示使用

此搭建步骤与sit环境搭建一致,sit环境服务器中搭建了其他非业务功能的应用,才另安装了jdk

文件目录介绍:

/data 数据目录

/data/docker\_files 从docker挂载出来的目录或文件

# Docker

## 安装docker

yum -y list docker\*

yum install docker.x86\_64

## 配置docker

### 修改docker仓库:

vim /etc/docker/daemon.json

**内容修改为:**

{

"registry-mirrors":["https://registry.docker-cn.com"],

"live-restore":true,

"insecure-registries":["120.77.213.135:5000"]

}

注: insecure-registries: docker镜像私库地址

## 启动docker

启动        systemctl start docker

守护进程重启   sudo systemctl daemon-reload

重启docker服务   systemctl restart  docker

重启docker服务  sudo service docker restart

关闭docker service docker stop

关闭docker systemctl stop docker

## 验证是否启动成功

启动完后,执行docker images命令出现如下结果docker即启动成功,可以进一步docker pull镜像并docker run 镜像



# Redis

## 安装redis

docker search redis\*

docker pull docker.io/redis

## 配置redis

把redis配置文件redis.conf和/data目录挂载到宿主机/data/docker\_files/redis文件夹下.配置文件需要事先创建好,本次uat环境搭建从sit环境复制而来.

修改其中配置:daemonize:yes

## 启动redis

docker run -p 6379:6379 --name redis -v /data/docker\_files/redis:/etc/redis -v /data/docker\_files/redis/data:/data -d docker.io/redis redis-server /etc/redis/redis.conf --appendonly yes

## 验证是否启动成功

docker ps

docker exec -ti 容器ID /bin/sh

redis-cli

# Nginx

## 安装nginx

docker search nginx\*

docker pull docker.io/nginx

## 配置nginx

1)在启动命令中,把nginx配置文件及资源目录挂载到/data/docker\_files/nginx目录.

2)配置文件需要事先创建好,此处从sit复制来,配置按实际需要配置业务规则.

3)nginx需要支持多应用,所以在资源根目录创建多个文件夹来区分应用.

## 启动nginx

docker run -d -p 80:80 --name nginx -v /data/docker\_files/nginx/nginx.conf:/etc/nginx/nginx.conf -v /data/docker\_files/nginx/html/:/etc/nginx/html/ -v /data/docker\_files/nginx/logs:/etc/nginx/logs docker.io/nginx

# Rocketmq Nameserver

拉取NameServer并启动

docker run -d -p 9876:9876 --name rmqserver foxiswho/rocketmq:server

# Rocketmq Broker

1、新建 /data/docker\_files/rocketmq/conf/broker.conf文件，加入以下内容保存

brokerClusterName = DefaultCluster

brokerName = test-broker

brokerIP1 = Server IP

brokerId = 0

deleteWhen = 04

fileReservedTime = 48

autoCreateTopicEnable = true

brokerRole = ASYNC\_MASTER

flushDiskType = ASYNC\_FLUSH

2、启动broker

docker run -d -p 10911:10911 -p 10909:10909

--name rmqbroker

--link rmqserver:namesrv

-e "NAMESRV\_ADDR=namesrv:9876" -e "JAVA\_OPTS=-Duser.home=/opt"

-e "JAVA\_OPT\_EXT=-server -Xms128m -Xmx128m -Xmn128m"

-v /data/docker\_files/rocketmq/conf/broker.conf:/etc/rocketmq/broker.conf

foxiswho/rocketmq:broker

# Rocketmq Console

1、拉取console镜像

docker run -e "JAVA\_OPTS=-Drocketmq.namesrv.addr= [IP] :9876

-Dcom.rocketmq.sendMessageWithVIPChannel=false" -p 9511:8080

-t styletang/rocketmq-console-ng

2、浏览器输入 IP:9511 打开管理端查看

# mysql数据库

## 创建uat库

在原来的库名,用户名,密码后面拼一个\_uat.

创建用户并授权:

CREATE USER 'pay\_transaction\_uat'@'%' IDENTIFIED BY 'pay\_transaction\_uat';

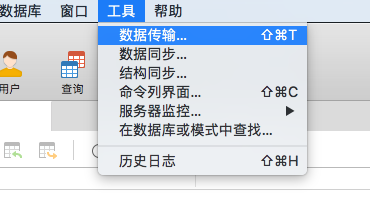
grant all privileges on pay\_transaction\_uat.\* to 'pay\_transaction\_uat'@'%';

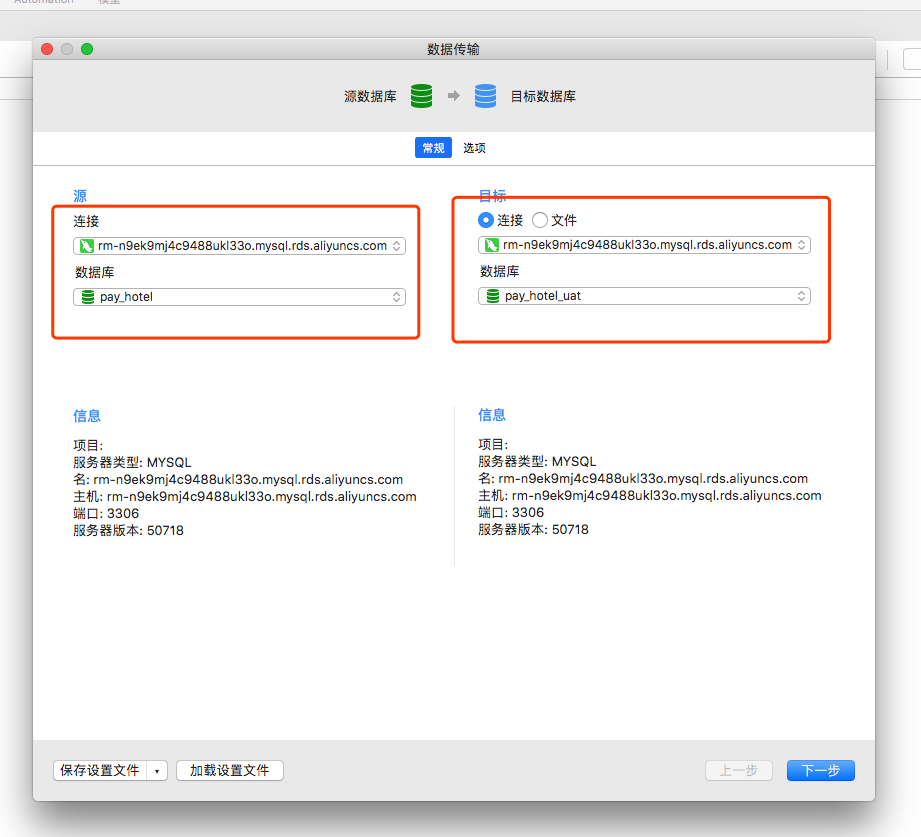
FLUSH PRIVILEGES;

## 复制表结构:

前提是先建好库,比如pay\_hotel\_uat

复制表此处以Navicat Premium为例.

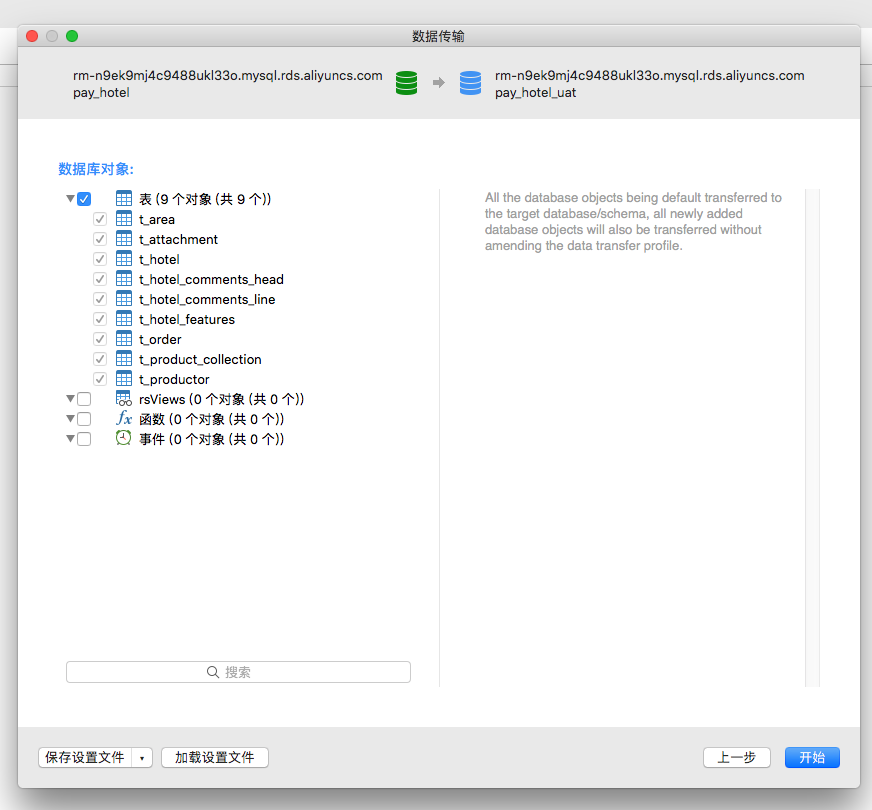




选项使用默认的就行了

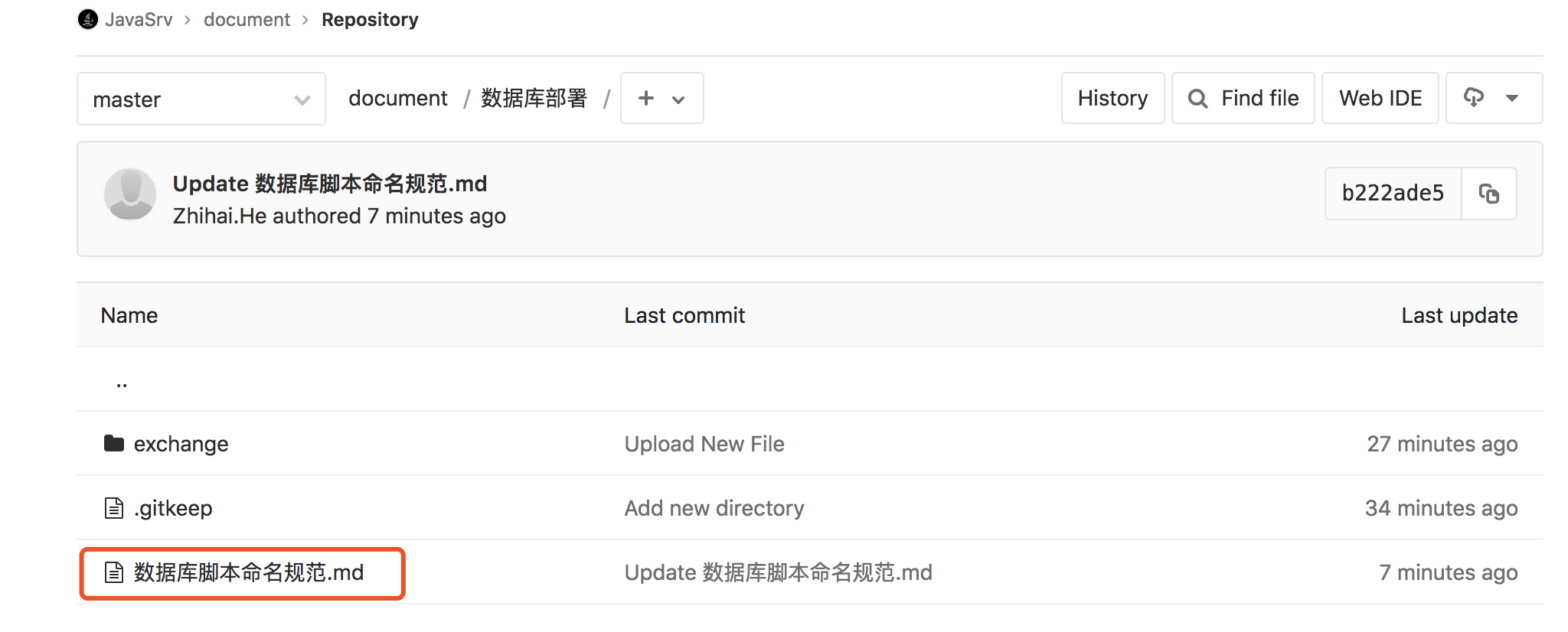


下一步,此处我只想复制表结构所以其他的都没选,也没有其他的对象需要复制的



注:初始化环境用这种方式,以后还是主要通过增量脚本来变更库

数据库部署，务必阅读此目录下数据库脚本命名规范，进行数据库创建初始化脚本上传，对不同项目进行文件夹分类



# mongoDB

在sit mongodb服务器(120.77.213.135)创建uat库

# 修改项目

增加uat配置文件

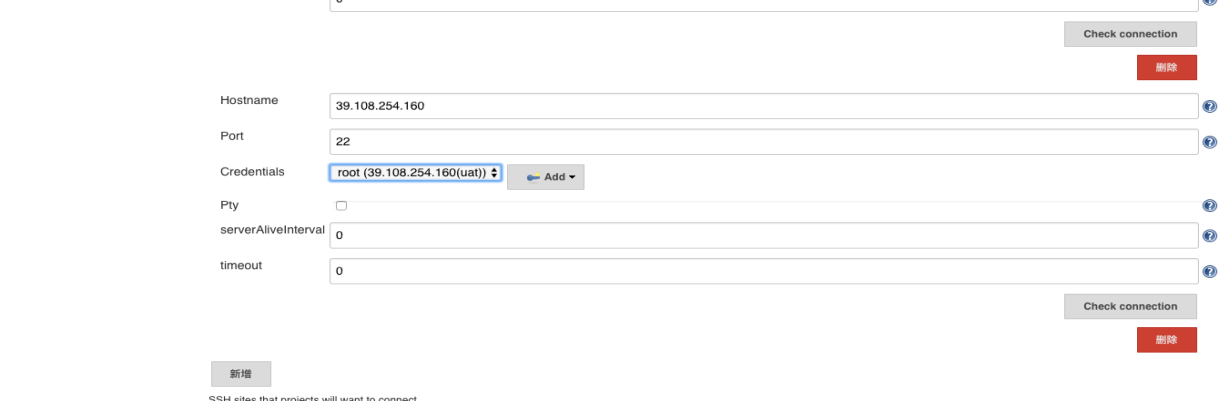
application-uat.properties:配置文件中的redis,mongodb,nginx,rocket等服务都换成新服务地址:39.108.254.160

mysql仍使用rm-n9ek9mj4c9488ukl33o.mysql.rds.aliyuncs.com,只是增加一套数据库,uat数据库名使用sit数据库名—"\_uat",例如pay\_hotel\_uat.用户名密码也一样在原来的用户名密码后面拼接“\_uat”以方便区分.

# Jenkins

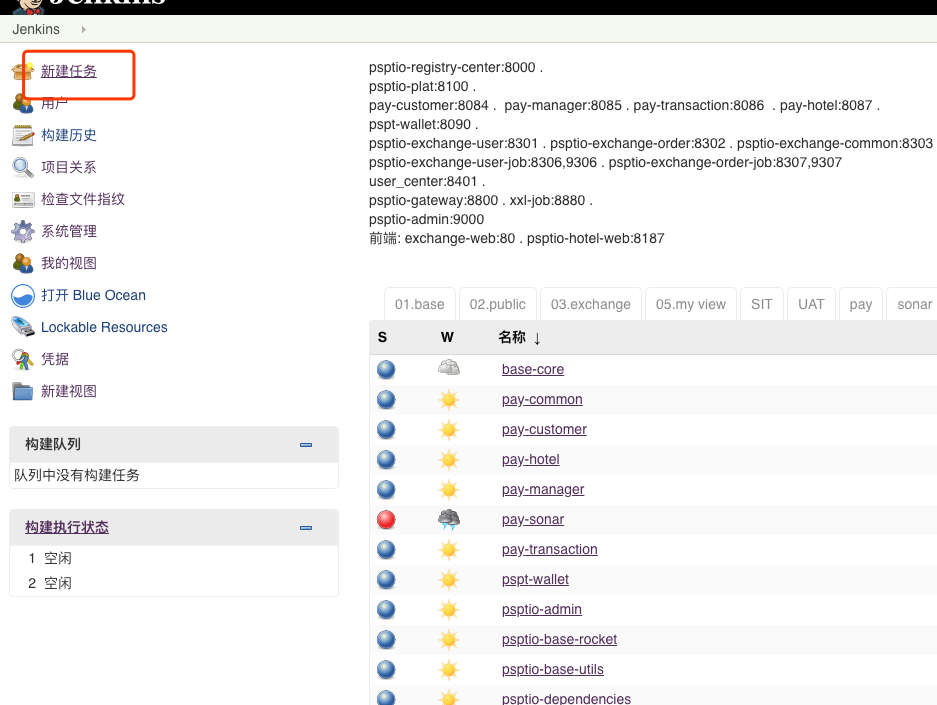
## 增加新服务器连接

系统管理》》系统设置》》SSH sites增加160服务连接.



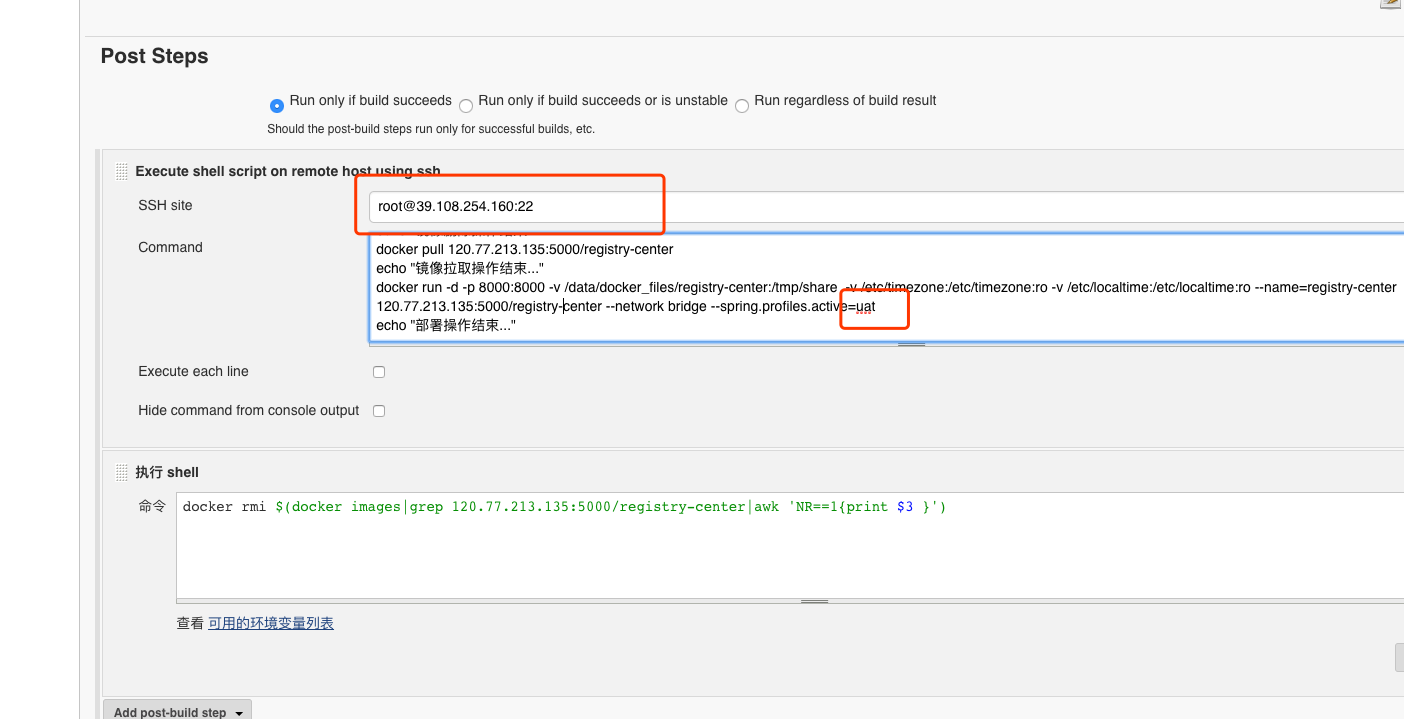
**check connection结果seccess**

## 新建构建任务

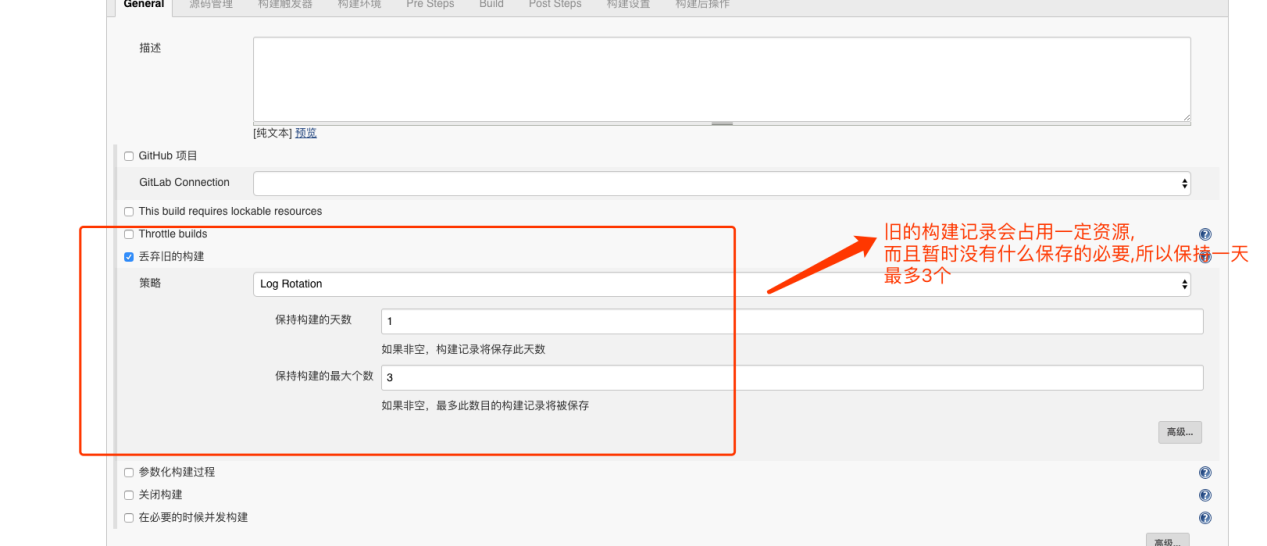
****

****

可以照搬sit环境的配置,只需要修改下图两个地方:修改要运行容器的服务器及要激活的环境配置(uat)即可.

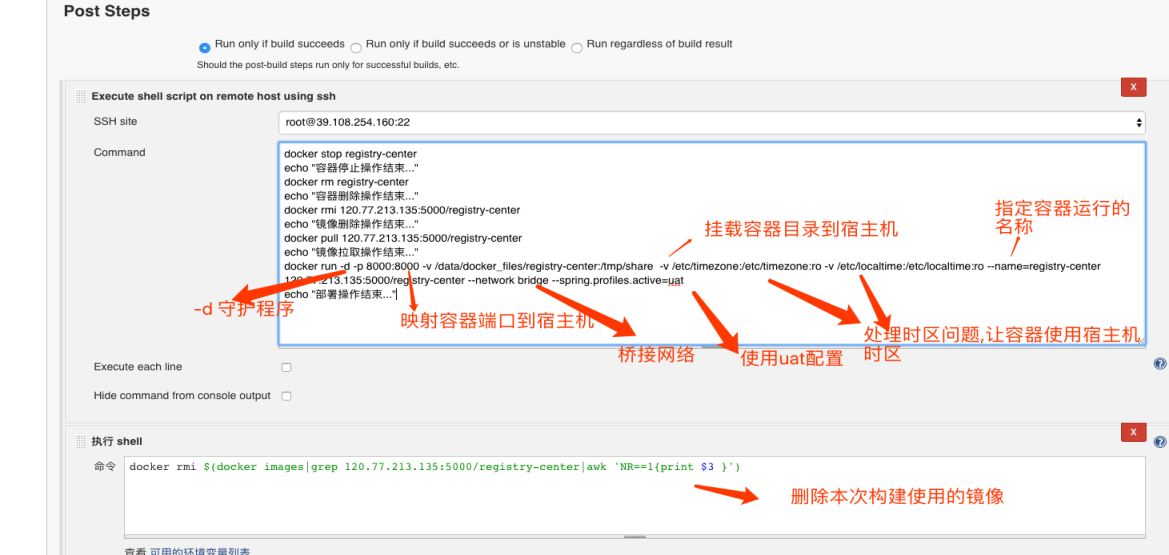
****

## 配置详解:

****

****

****

****

# 镜像与sit区分开

1:Pom.xml中给镜像名增加绿色部分:

<imageName>120.77.213.135:5000/${project.artifactId}/${spring.profiles.active}</imageName>

2:jenksin配置做如下修改,sit,uat环境最好都修改下:

