# 1.3其他

## 一、创建Redis容器

docker load < Redis.tar.gz

在CentOS系统中创建/root/redis/conf/redis.conf文件，

bind 0.0.0.0  
protected-mode yes  
port 6379  
tcp-backlog 511  
timeout 0  
tcp-keepalive 0  
loglevel notice  
logfile ""  
databases 4  
save 900 1  
save 300 10  
save 60 10000  
stop-writes-on-bgsave-error yes  
rdbcompression yes  
rdbchecksum yes  
dbfilename dump.rdb  
dir ./  
requirepass abc123456

执行命令，创建Redis容器，做好端口映射。把Linux的6379端口，映射到Windows的6379端口。使用云主机的同学，要在安全组中开放6379端口。

docker run -it -d --name redis -p 6379:6379 \  
-v /root/redis/conf:/usr/local/etc/redis \  
-e TZ=Asia/Shanghai \  
redis:6.0.10 \  
redis-server /usr/local/etc/redis/redis.conf

修改Java项目的application.yml文件，把其中Redis连接配置一下。如果你是本地Redis，就写localhost；如果是云主机，就写云主机的IP地址。

## 二、创建RabbitMQ容器

在本项目中通知消息用到了RabbitMQ技术，所以我们要把RabbitMQ容器给创建出来。由于不管什么数据库，MySQL也好、MongoDB也好，甚至说HBase，我们都无法给保存的记录设置过期时间。在一个对外运营的商业项目中，总是会有很多僵尸用户：他们常年不上线。如果我们发送系统公告消息的时候，要不要给这些人发送消息呢？当然要，但是不必永久保存。我们把公告消息发送到RabbitMQ的消息队列中，给消息设置过期时间。假如在消息过期时间，用户上线了，那么他就可以收到公告消息，并且系统会把他收到的公告消息转存到数据库中，永久保存。那些常年不上线的僵尸用户，发给他们的系统公告消息过了过期时间，会自动从RabbitMQ中删除，也不会转存到数据库中，这样我们就避免了为大量僵尸用户保存系统消息而占用海量存储存空间。

执行命令，创建出RabbitMQ容器，做好端口映射。把4369、5671和5672这些端口，映射到Windows系统上面。使用云主机的同学，相应的开放这三个端口。

docker load < RabbitMq.tar.gz

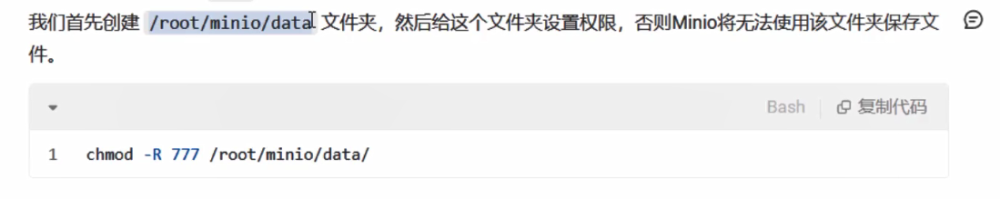
docker run -it -d --name mq \  
-p 4369:4369 -p 5671:5671 -p 5672:5672 \  
-e TZ=Asia/Shanghai \  
rabbitmq:3.8.9

## 三、创建Minio容器

作为医疗类的项目，我们把患者的诊断和处方等文件保存到公有云上面，不太尊重患者的隐私。所以接下来我们要在本地利用Minio搭建一套私有云存储，存放本项目用到医生照片、广告横幅、患者病志和处方等文件。

Minio本质上就是个存储文件的服务器，操作起来更腾讯云存储类似。都是先要创建存储桶，然后在存储桶里面创建文件夹放置文件。我们可以给存储桶设置读写的权限和访问过期时间等等，总之用起来很简单就是了。

/root/minio/data



chmod -R 777 /root/minio/data/

docker load <Minio.tar.gz

docker run -it -d --name minio \  
-p 9000:9000 -p 9001:9001 \  
-v /root/minio/data:/data \  
-e TZ=Asia/Shanghai --privileged=true \  
--env MINIO\_ROOT\_USER="root" \  
--env MINIO\_ROOT\_PASSWORD="abc123456" \  
bitnami/minio:latest

docker run \

-p 9000:9000 \

--name minio \

-d --restart=always \

-e "MINIO\_ACCESS\_KEY=root" \

-e "MINIO\_SECRET\_KEY=abc123456" \

-e TZ=Asia/Shanghai --privileged=true \

-v /root/minio/data:/data \

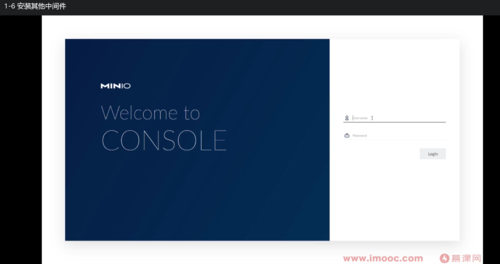
--console-address ":9000" --address ":9090"

宿主机端口 9010,9011(ip:9010--43.143.231.180:9011)

对应的容器端口9000,9001(docker端口)

docker run -d -p 9010:9000 -p 9011:9001 --name minio -v /data/minio/data:/data -e "MINIO\_ROOT\_USER=root" -e "MINIO\_ROOT\_PASSWORD=abc123456" minio/minio server /data --console-address ":9001" --address ":9000"  
  
  
43.143.231.180:9011  
82.157.40.209:9001

s



### 删除镜像image

要删除 Docker 中的镜像，可以使用 docker rmi 命令

### [删除容器](https://so.csdn.net/so/search?q=%E5%88%A0%E9%99%A4%E5%AE%B9%E5%99%A8&spm=1001.2101.3001.7020)

要删除 Docker 中的容器，可以使用 docker rm 命令。请注意，删除容器前请确保不再需要它，并且该容器已经停止运行。否则，需要首停止运行该容器。

2.开启指定端口

firewall-cmd --zone=public --add-port=9001/tcp --permanent

命令：**systemctl status firewalld**  —》查看firewalld状态

### 步骤二：开启防火墙。

命令：**systemctl start firewalld**