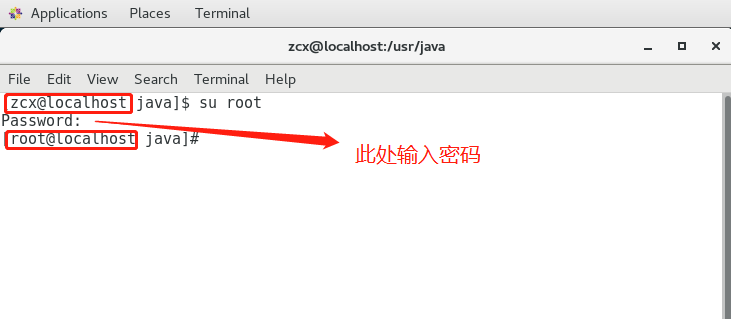
# CentOS安装程序部分说明

以下所有安装基于CentOS安装完毕，此处使用CentOS 7。

## JDK1.8安装

①切换用户权限

su root



①在/usr/目录下创建java目录

mkdir/usr/java --创建目录

cd /usr/java --进入目录

①手动下载JDK安装包后放到/usr/java目录下，并解压

①设置环境变量

vi /etc/profile

②在profile中添加如下内容：

#set java environment

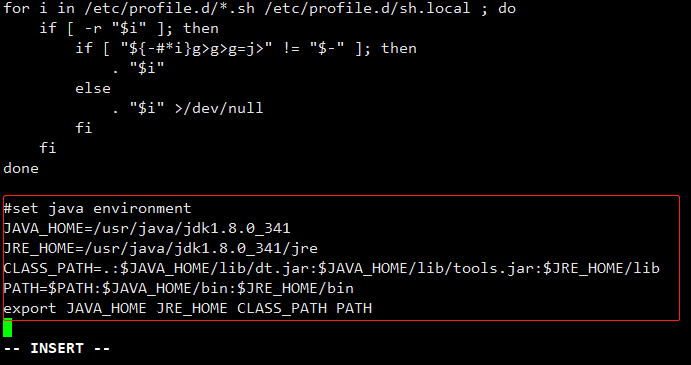
JAVA\_HOME=/usr/java/jdk1.8.0\_341

JRE\_HOME=/usr/java/jdk1.8.0\_341/jre

CLASS\_PATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

export JAVA\_HOME JRE\_HOME CLASS\_PATH PATH

注意：此处JDK名称为自己的

五，

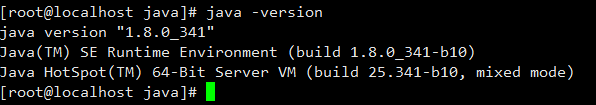
①让修改生效：

source /etc/profile

六，

①检查是否安装成功

java-version

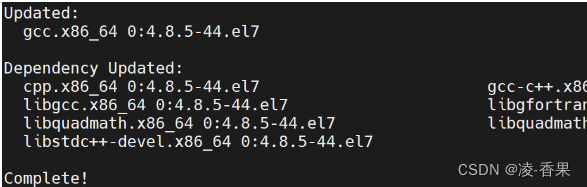


## Docker安装

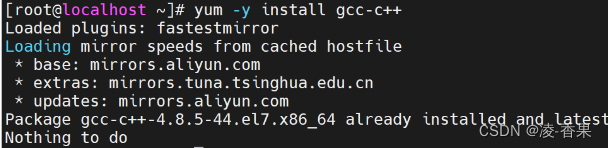
一，

①安装[gcc](https://so.csdn.net/so/search?q=gcc&spm=1001.2101.3001.7020" \o "gcc)相关

yum -y install gcc



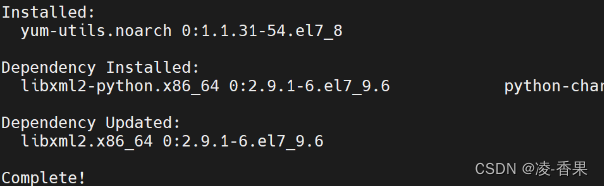
yum -y install gcc-c++



二，

①安装所需安装包

yum install -y yum-utils



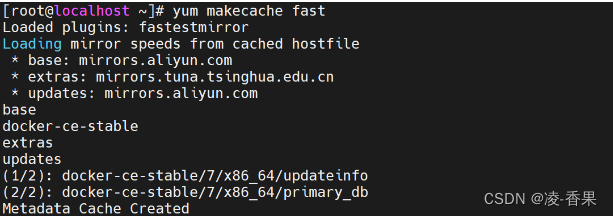
②设置stable镜像阿里云仓库

yum-config-manager --add-repo <http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>



③更新yum软件包索引

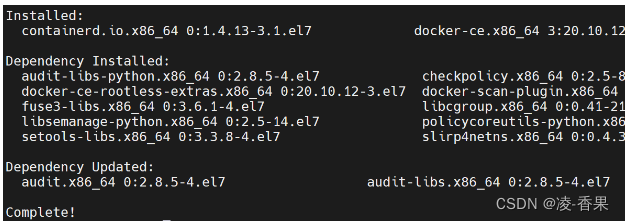
yum makecache fast



二，

①安装docker ce

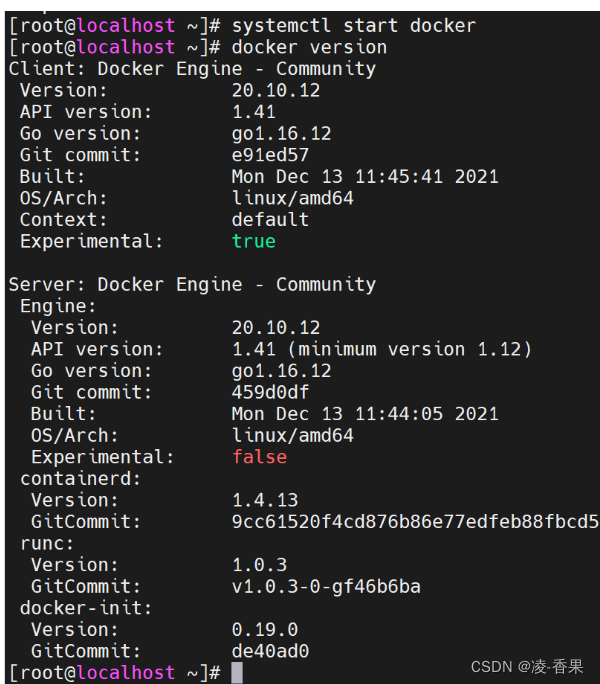
yum -y install docker-ce docker-ce-cli containerd.io



②启动docker，查看docker版本

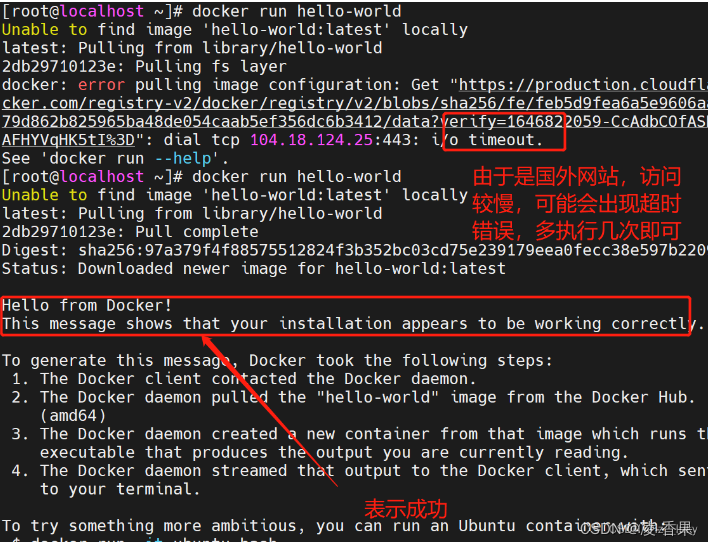
systemctl start docker

docker version



③测试hello world

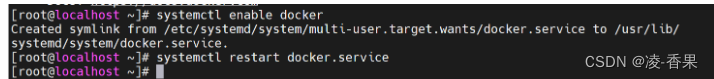
docker run hello-world



④设置docker开机自启

systemctl enable docker

systemctl restart docker.service



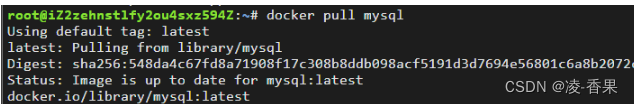
## MySQL安装

注意：没有安装docker直接安装mysql会导致服务无法启动（centos不支持mysql详情可自行了解）

一，

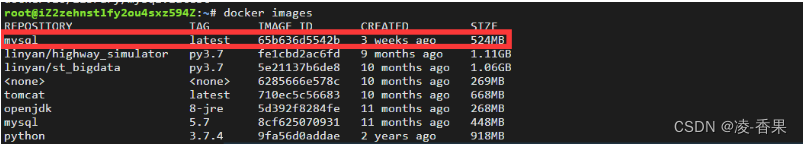
①安装命令

docker pull mysql（此命令为安装最新版本mysql）



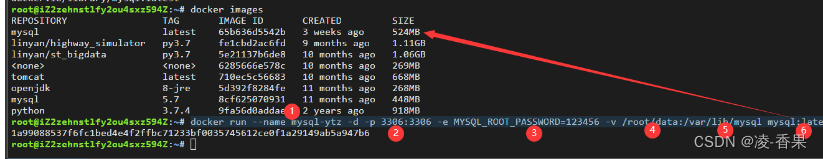
sudo docker pull mysql:5.7（安装指定版本）

②运行以下命令来检测mysql是否下载成功



③运行以下命令来启动mysql服务， 启动成功后就返回一个容器ID

docker run --name mysql-ytz -d -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 -v /root/data:/var/lib/mysql mysql:latest



解释：

1：--name mysql-ytz:该容器启动后的名字：（自定义命名）如果没有设置，系统会自动设置一个。

-d 代表后台启动该服务。

2：-p 3306(这是liunx的端口号，宿主机，如果我们想要远程服务的话，访问的端口就是这个端口):3306（docker容器的端口，每一个容器都是独立的，可理解成操作系统层面的系统），访问这个端口就是先通过远程访问宿主机的端口，再映射到docker容器的端口访问mysql。

3：-e MYSQL\_ROOT\_PASSWORD=123456 这是说mysql启动需要的开机密码，默认的账号是root ，密码就是上面设置的：123456

4: -v /root/data:/var/lib/mysql /root/data/:这是宿主机的数据存放路径（你也可以自定义），

5: /var/lib/mysql:这是mysql容器存放数据的地方。也是为了同步数据，防止，容器被删除以后，数据就不存在了。

6： 版本号

二，

①最后去连接数据库就可以了，我是在navicat中连接的



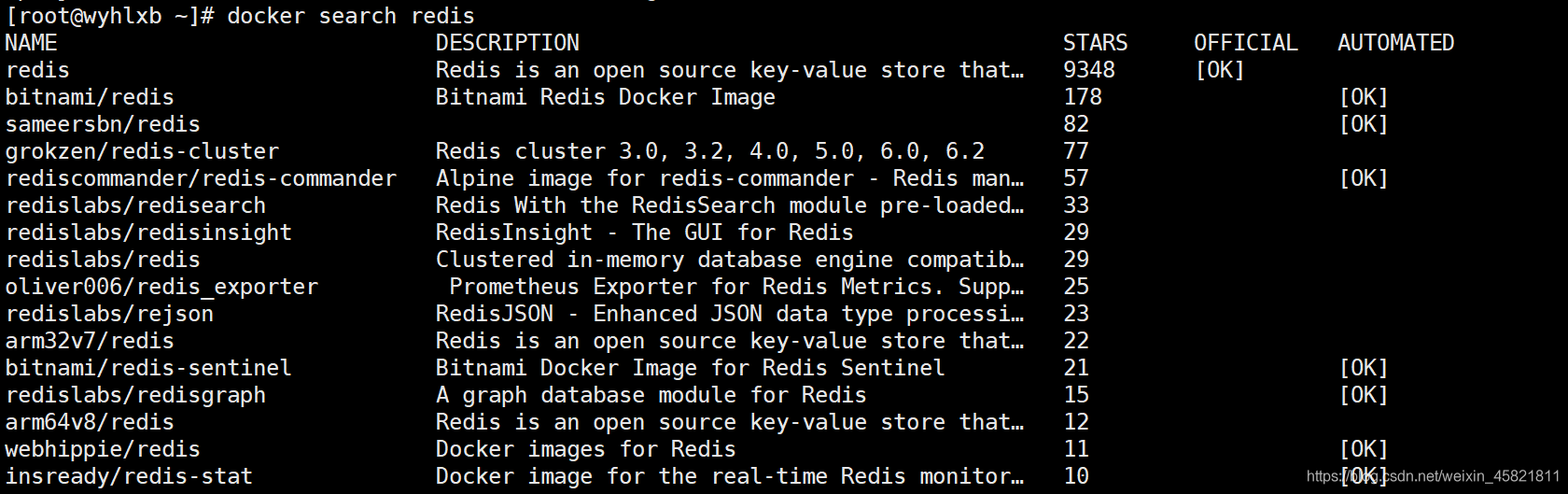
注意：此处主机为虚拟机IP 可用命令ip addr 或者 ifconfig(不是ipconfig)

## Redis安装

一，

①Docker搜索redis镜像

docker search redis

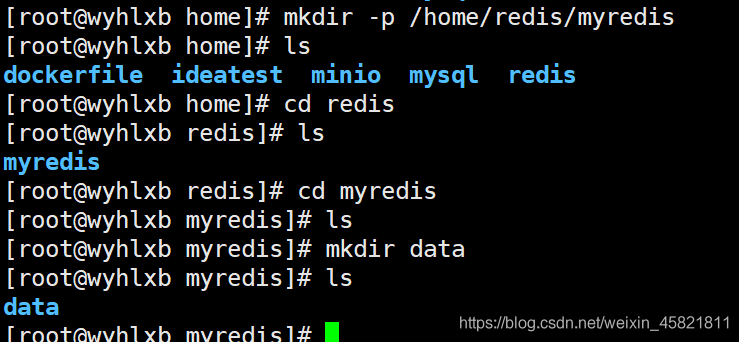


②Docker 拉取redis镜像

docker pull redis（不加版本号就是最新版本）

③创建挂载文件夹

mkdir -p /home/redis/myredis



④看下图还需创建一个data文件夹 和myredis.conf文件



⑤myredis.conf 文件内容如下

protected-mode no

port 6379

tcp-backlog 511

requirepass 123456

timeout 0

tcp-keepalive 300

daemonize no

supervised no

pidfile /var/run/redis\_6379.pid

loglevel notice

logfile ""

databases 30

always-show-logo yes

save 900 1

save 300 10

save 60 10000

stop-writes-on-bgsave-error yes

rdbcompression yes

rdbchecksum yes

dbfilename dump.rdb

dir ./

replica-serve-stale-data yes

replica-read-only yes

repl-diskless-sync no

repl-disable-tcp-nodelay no

replica-priority 100

lazyfree-lazy-eviction no

lazyfree-lazy-expire no

lazyfree-lazy-server-del no

replica-lazy-flush no

appendonly yes

appendfilename "appendonly.aof"

no-appendfsync-on-rewrite no

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

aof-load-truncated yes

aof-use-rdb-preamble yes

lua-time-limit 5000

slowlog-max-len 128

notify-keyspace-events ""

hash-max-ziplist-entries 512

hash-max-ziplist-value 64

list-max-ziplist-size -2

list-compress-depth 0

set-max-intset-entries 512

zset-max-ziplist-entries 128

zset-max-ziplist-value 64

hll-sparse-max-bytes 3000

stream-node-max-bytes 4096

stream-node-max-entries 100

activerehashing yes

hz 10

dynamic-hz yes

aof-rewrite-incremental-fsync yes

rdb-save-incremental-fsync yes

⑥[启动redis](https://so.csdn.net/so/search?q=%E5%90%AF%E5%8A%A8redis&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_45821811/article/details/_blank) 容器

docker run --restart=always --log-opt max-size=100m --log-opt max-file=2 -p 6379:6379 --name myredis -v /home/redis/myredis/myredis.conf:/etc/redis/redis.conf -v /home/redis/myredis/data:/data -d redis redis-server /etc/redis/redis.conf --appendonly yes --requirepass 123456

⑦以上命令注解

--restart=always 总是开机启动

--log是日志方面的

-p 6379:6379 将6379端口挂载出去

--name 给这个容器取一个名字

-v 数据卷挂载

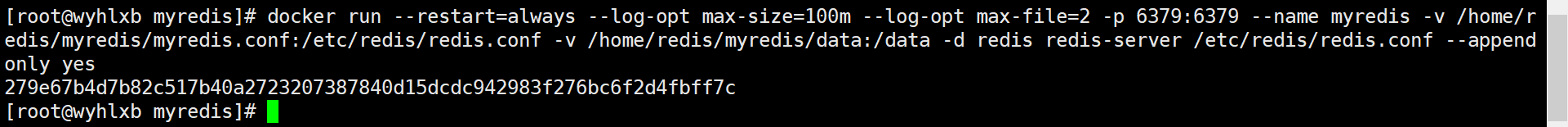
- /home/redis/myredis/myredis.conf:/etc/redis/redis.conf 这里是将 liunx 路径下的myredis.conf 和redis下的redis.conf 挂载在一起。

- /home/redis/myredis/data:/data 这个同上

-d redis 表示后台启动redis

redis-server /etc/redis/redis.conf 以配置文件启动redis，加载容器内的conf文件，最终找到的是挂载的目录 /etc/redis/redis.conf 也就是liunx下的/home/redis/myredis/myredis.conf

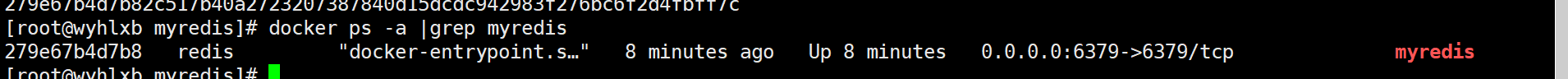
⑧成功界面



二，

①通过docker ps指令查看启动状态

docker ps -a |grep myredis

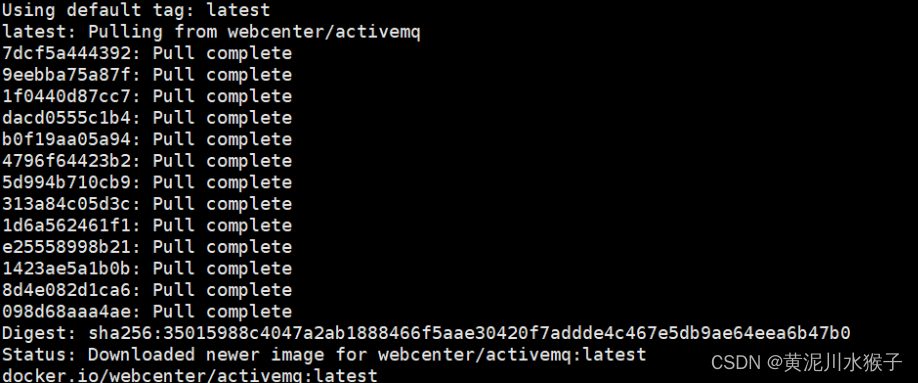


## ActiveMQ安装

一，

①获取 ActiveMQ 镜像

docker pull webcenter/activemq

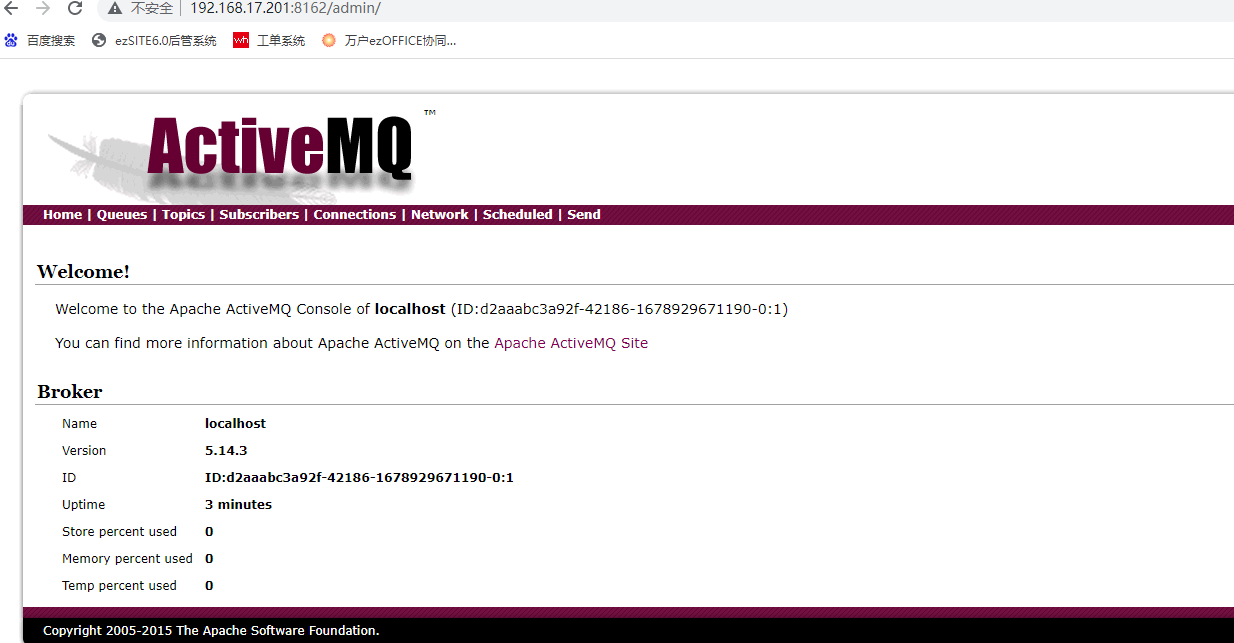


②docker 启动 ActiveMQ 命令

docker run -d --name activemq -p 61617:61616 -p 8162:8161 webcenter/activemq

* 61616是 activemq 的容器使用端口（映射为61617）
* 8161是 web 页面管理端口（对外映射为8162）

访问 <http://虚拟机地址:8161/admin（用户名和密码默认为admin），则启动成功。>



## Solr安装

一，

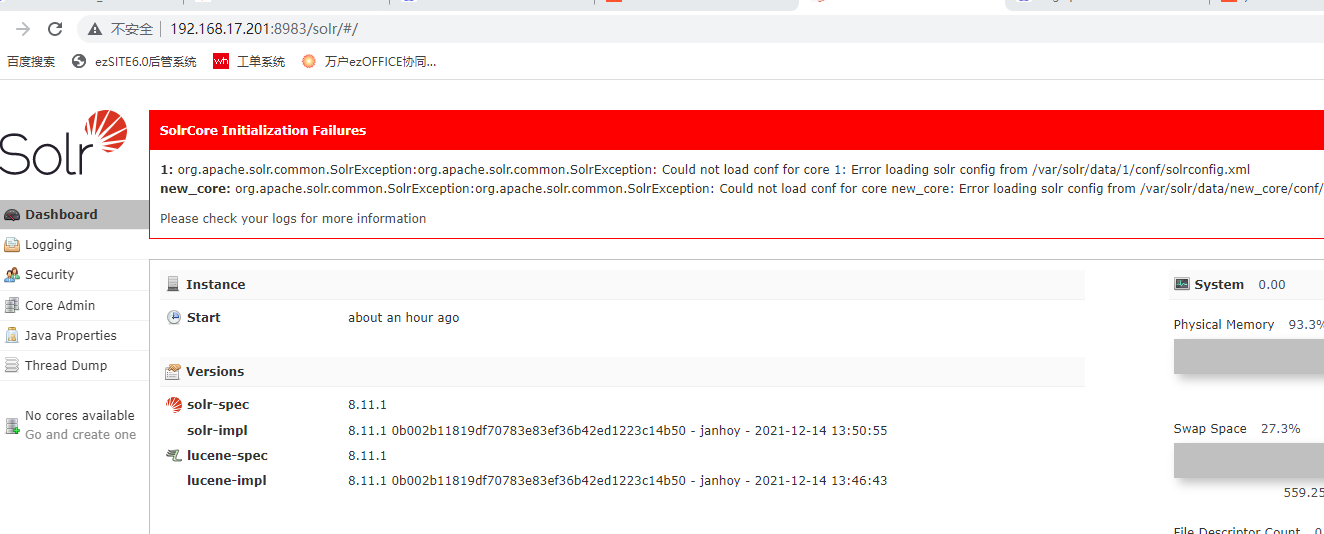
①拉取镜像

docker pull solr:7.5.0

②启动

docker run --name solr -d -p 8983:8983 -t solr:7.5.0

③查看可视化界面（ip+port），这样就是安装成功了



## Nginx安装

一，

①下载Nginx镜像

docker pull nginx

②创建挂载目录

mkdir -p /home/nginx/conf  
mkdir -p /home/nginx/log  
mkdir -p /home/nginx/html

③生成容器

docker run --name nginx -p 9001:80 -d nginx

④将容器nginx.conf文件复制到宿主机

docker cp nginx:/etc/nginx/nginx.conf /home/nginx/conf/nginx.conf

⑤将容器conf.d文件夹下内容复制到宿主机  
docker cp nginx:/etc/nginx/conf.d /home/nginx/conf/conf.d

⑥将容器中的html文件夹复制到宿主机  
docker cp nginx:/usr/share/nginx/html /home/nginx/

二，

①运行

docker run -p 9002:80 --name nginx -v /home/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v /home/nginx/conf/conf.d:/etc/nginx/conf.d -v /home/nginx/log:/var/log/nginx -v /home/nginx/html:/usr/share/nginx/html -d nginx:latest

或者

docker run \

-p 9002:80 \

--name nginx \

-v /home/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \

-v /home/nginx/conf/conf.d:/etc/nginx/conf.d \

-v /home/nginx/log:/var/log/nginx \

-v /home/nginx/html:/usr/share/nginx/html \

-d nginx:latest

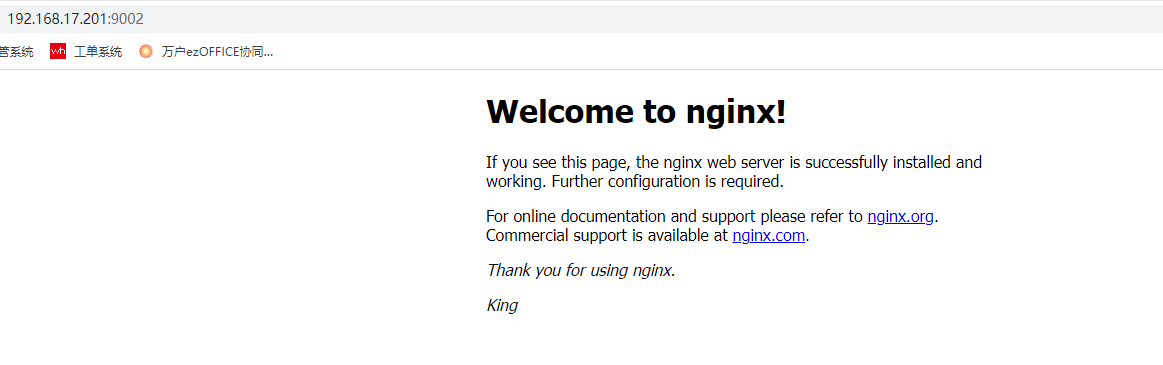
注意：以上两个命令仅有换行的区别都可使用。

描述：

|  |  |
| --- | --- |
| **命令** | **描述** |
| –name nginx | 启动容器的名字 |
| -d | 后台运行 |
| -p 9002:80 | 将容器的 9002(后面那个) 端口映射到主机的 80(前面那个) 端口 |
| -v /home/nginx/conf/nginx.conf:/etc/nginx/nginx.conf | 挂载nginx.conf配置文件 |
| -v /home/nginx/conf/conf.d:/etc/nginx/conf.d | 挂载nginx配置文件 |
| -v /home/nginx/log:/var/log/nginx | 挂载nginx日志文件 |
| -v /home/nginx/html:/usr/share/nginx/html | 挂载nginx内容 |
| nginx:latest | 本地运行的版本 |
| \ | shell 命令换行 |

②测试是否启动

Ip+端口



③部分可能使用到的命令  
docker restart nginx --重启容器  
docker rm nginx --删除该容器  
docker stop nginx --关闭该容器