安装docker

宿主机或虚拟机上操作，我这里使用宿主机

准备:

需要 centos163 源

时间同步

需要能访问公网

配置 iptables,selinux,firewalld

更新Yum仓库

yum makecache fast

yum install docker -y

# vim /etc/sysconfig/docker

修改配置参数： "--selinux-enabled" 为 "--selinux-enabled=false" 把selinux 禁掉，把selinux disabled掉了，就可以不用修改这个参数。

systemctl start docker

systemctl status docker

systemctl enable docker

# docker version

# docker info

查看信息

## 镜像加速

鉴于国内网络问题，后续拉取 Docker 镜像十分缓慢，我们可以需要配置加速器来解决，我使用的是网易的镜像地址：**http://hub-mirror.c.163.com**。

新版的 Docker 使用 /etc/docker/daemon.json（Linux） 或者 %programdata%\docker\config\daemon.json（Windows） 来配置 Daemon。

请在该配置文件中加入（没有该文件的话，请先建一个）：

{

"registry-mirrors": ["http://hub-mirror.c.163.com"]}

重启docker

systemctl restart docker & docker info

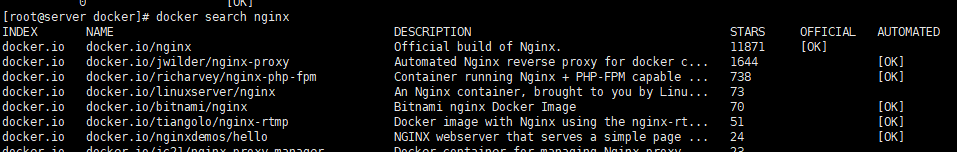


# Docker 安装 Nginx

### **docker pull nginx 命令安装**

查找 [Docker Hub](https://hub.docker.com/r/library/nginx/" \t "https://www.runoob.com/docker/_blank) 上的 nginx 镜像

[root@server1 /]# docker search nginx



选择官方的这个，

[root@server docker]# docker pull docker.io/nginx

docker images



运行镜像，产生新容器

docker run -d --name nginxtest -p 8081:80 docker.io/nginx

## 开始 nginx 部署

创建相关目录

[root@server docker]# mkdir /software /nginx/nginx/kis /nginx/nginx/conf.d /nginx/logs -p

创建 /nginx/nginx/conf 用来存域名证书文件

mkdir /nginx/nginx/conf

拷贝容器内 Nginx 默认配置文件到本地/nginx/目录下，容器 ID 可以查看 **docker ps** 命令输入中的第一列

[root@server nginx]# docker cp c57ccae23f23:/etc/nginx/ /nginx/

修改nginx.conf 文件，公司的标准模板

安装rsync

yum install -y rsync

启动

systemctl restart rsyncd

同步把远程的资料同步过来本地

rsync -av --delete 58.82.238.95::ziliao /software/

这过程比较长

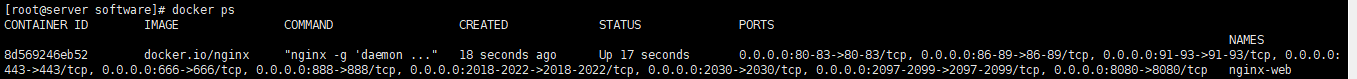
好了后创建软连接

ln -s /software/站点配置文件/ /nginx/nginx/kis/

ln -s /software/域名证书/ /nginx/nginx/conf

**准备启动镜像，运行容器**

docker run -d -p 2018:2018 -p 2019:2019 -p 2020:2020 -p 2021:2021 -p 2022:2022 -p 2030:2030 -p 8080:8080 -p 80:80 -p 81:81 -p 2097:2097 -p 2098:2098 -p 2099:2099 -p 82:82 -p 83:83 -p 86:86 -p 87:87 -p 888:888 -p 88:88 -p 89:89 -p 666:666 -p 91:91 -p 443:443 -p 92:92 -p 93:93 --name nginx-web -v /software:/software -v /nginx/nginx:/etc/nginx -v /nginx/logs:/var/log/nginx docker.io/nginx

可以了

把nginx 配置文件做了修改就重启容器

docker restart nginx-web

防火墙开放对外访问的端口，ports端口最多写15个，所以分两次，

iptables -A INPUT -p tcp -m multiport --dports 80,81,82,83,86,87,88,89,91,92,93,443,666,888,2018 -j ACCEPT

iptables -A INPUT -p tcp -m multiport --dports 2019,2020,2021,2030,2097,2098,2099,8080 -j ACCEPT

下面的规则也要开，要不然容器里对应的端口数据进不来宿主机

iptables -A INPUT -p tcp -m multiport --sports 80,81,82,83,86,87,88,89,91,92,93,443,666,888,2018 -j ACCEPT

iptables -A INPUT -p tcp -m multiport --sports 2019,2020,2021,2030,2097,2098,2099,8080 -j ACCEPT

**已经开启容器后也可以配置增加其他端口**，**但不建议手工添加，最好使用docker提供的功能**

假如需要增加600端口

首先开通宿主机防火墙进出规则

iptables -A INPUT -p tcp -m multiport --dports 600 -j ACCEPT

iptables -A INPUT -p tcp -m multiport --sports 600 -j ACCEPT

只要是访问docker IP 172.17.0.2 目标端口为600的都放行

iptables -A DOCKER -d 172.17.0.2/32 ! -i docker0 -o docker0 -p tcp -m tcp --dport 600 -j ACCEPT

设置nat表的路由规则

下面 这行就是端口映射了，只要是目标端口是本机 600 的就做DNAT转发到 容器600

iptables -t nat -A DOCKER ! -i docker0 -p tcp -m tcp --dport 600 -j DNAT --to-destination 172.17.0.2:600 -i 进来docker0网卡的流量入的数据

路由后 源ip 容器 容器目标ip tcp协议 模块 tcp

iptables -t nat -A POSTROUTING -s 172.17.0.2/32 -d 172.17.0.2/32 -p tcp -m tcp --dport 600 -j MASQUERADE 这条没有设置也可以

使用此方法有一个缺点，不能访问localhost:600，也就是说如果想对localhost也进行转发，需要进行额外的配置。

进入容器

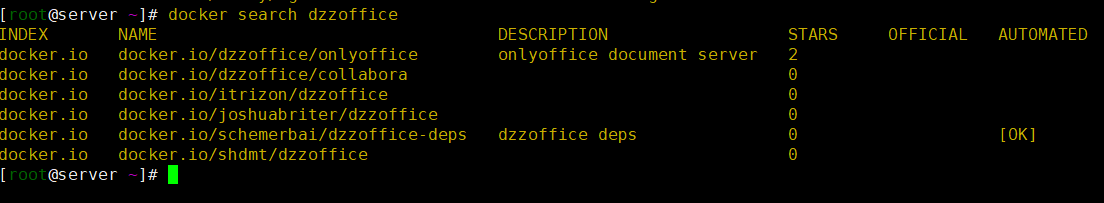
docker exec -it nginx-web /bin/bash

通过nginx -t 查看语法有没有配置错误；

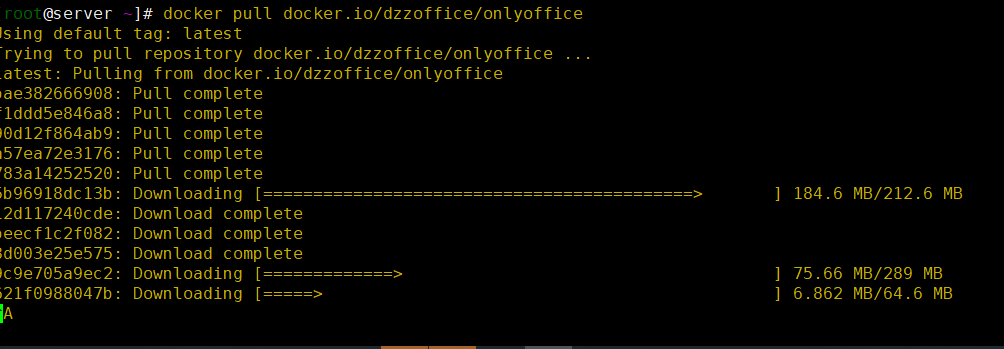
**Docker 安装dzzoffice**

先去搜索docker search dzzoffice

然后下载第一个



docker pull docker.io/dzzoffice/onlyoffice



查看镜像

docker images

然后 运行

注意外部的2018端口不能被其他服务占用，否则会报错，

docker run -d --name dzzoffice -p 2018:80 -p 2022:22 docker.io/dzzoffice/onlyoffice

把容器镜像上传，解压运行

docker run -d -p 2018:80 -p 2022:22 --name dzzoffice dzzoffice