链滴

Docker 实验 pull_dockerfile_tomcat_ 简单 nginx

作者：patrickal

原文链接：https://ld246.com/article/1683913484224

来源网站：链滴

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

Docker实验 pull_dockerfile_ omcat_简单nginx

带“*”号的为省略，可选做

实验目的

1、掌握Docker安装方法。

2、掌握Docker pull 服务及软件并应用的方法。

3、了解通过Dockerfile和docker build 定制docker的方法

实验仪器设备/实验环境

1、Centos7操作系统 2、Docker仓库

实验原理

容器是一种轻量级、可移植、自包含的软件打包技术，使应用程序可以在几乎任何地方以相同的方式行。

开发人员在自己笔记本上创建并测试好的容器，无需任何修改就能够在生产系统的虚拟机、物理服务或公有云主机上运行。

容器核心技术是指能够让 container 在 host 上运行起来的那些技术。

• 容器规范:为了保证容器生态的健康发展，保证不同容器之间能够兼容，包含 Docker、CoreOS、Google在内的若干公司共同成立了一个叫 Open Container Initiative（OCI） 的组织，其目是制定开放容器规范。

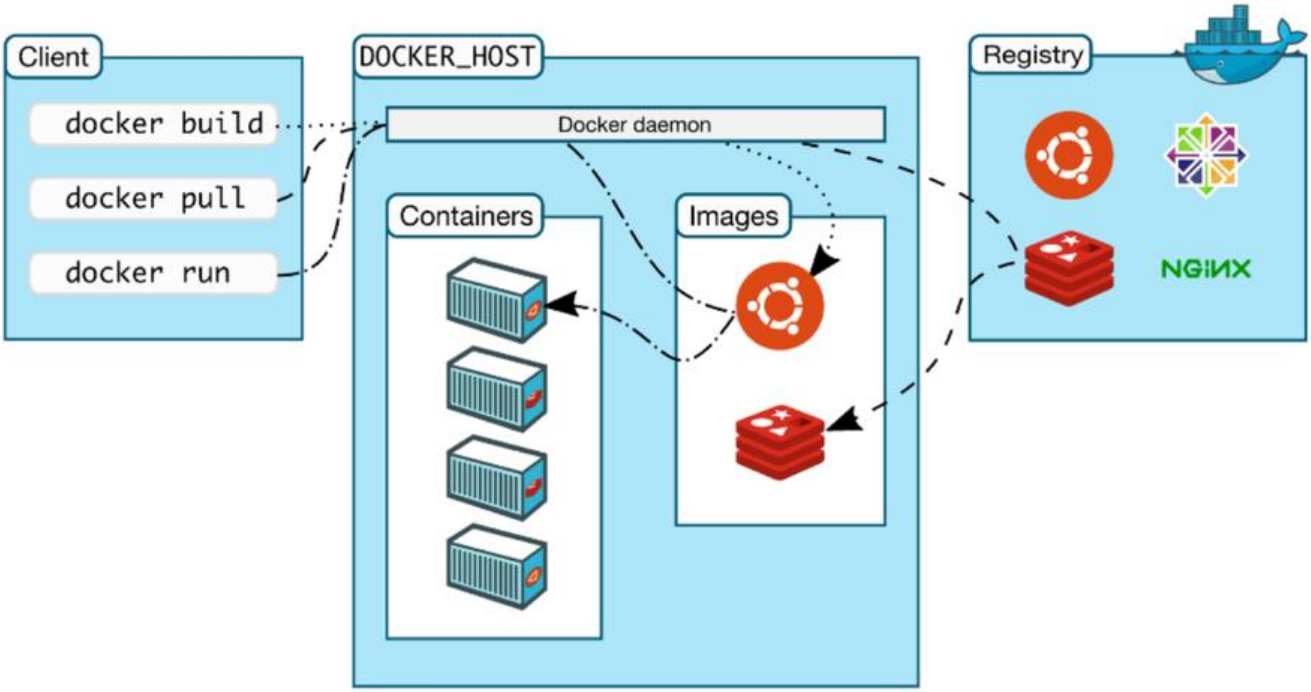
• 容器 runtime:runtime 是容器真正运行的地方。runtime 需要跟操作系统 kernel 紧密协作，为容提供运行环境。lxc、runc 和 rkt 是目前主流的三种容器 runtime。

• 容器管理工具:容器管理工具对内与 runtime 交互，对外为用户提供 interface，比如 CLI。

• 容器定义工具:容器定义工具允许用户定义容器的内容和属性，这样容器就能够被保存，共享和重建。

• Registry:容器是通过 image 创建的，需要有一个仓库来统一存放 image，这个仓库就叫做 Registr。

容器 OS:容器 OS 是专门运行容器的操作系统。CoreOS、atomic 和 ubuntu core 是其中的杰出代表



实验内容

1、Docker安装部署；

2、Docker pull 拉取镜像实现服务；

3、Dockerfile和Docker build定制；

阿里云镜像加速Docker（巨快）

直接把下列命令粘贴运行

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://knqne7y6.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Docker、镜像的安装

1、Docker安装部署；

请尽量配置yum源为阿里云，这样yum安装快速

```
yum install -y docker-ce  安装docker
systemctl start docker  开启docker*
```

systemctl enable docker
docker version 截图

修改防火墙设置

systemctl stop firewall
setenforce 0
getenforce

2、Docker pull 拉取镜像实现服务；

拉取一个nginx镜像

docker pull [镜像名]

docker images 查看已经拉取的镜像

docker run [...] [镜像名]

资料补充：

访问http测试：https://www.runoob.com/docker/docker-install-nginx.html

https://blog.csdn.net/Themoonlights_/article/details/12238597

腾讯云服务器在Centos中使用nginx
https://www.tencentcloud.com/zh/document/product/214/32390

docker pull nginx

```
➜  Downloads docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
9e3ea8720c6d: Pull complete
bf36b6466679: Pull complete
15a97cf85bb8: Pull complete
9c2d6be5a61d: Pull complete
6b7e4a5c7c7a: Pull complete
8db4caa19df8: Pull complete
Digest: sha256:480868e8c8c797794257e2abd88d0f9a8809b2fe956cbfbc05dcc0bca1f7cd43
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
➜  Downloads docker run -d -p 80:80 nginx
484d0708e69ecc58376da25374037c5f3f3e395ce5694b25de3b23c13e788db6
```

创建容器

docker run -d -p 8080:80 --name my-nginx nginx

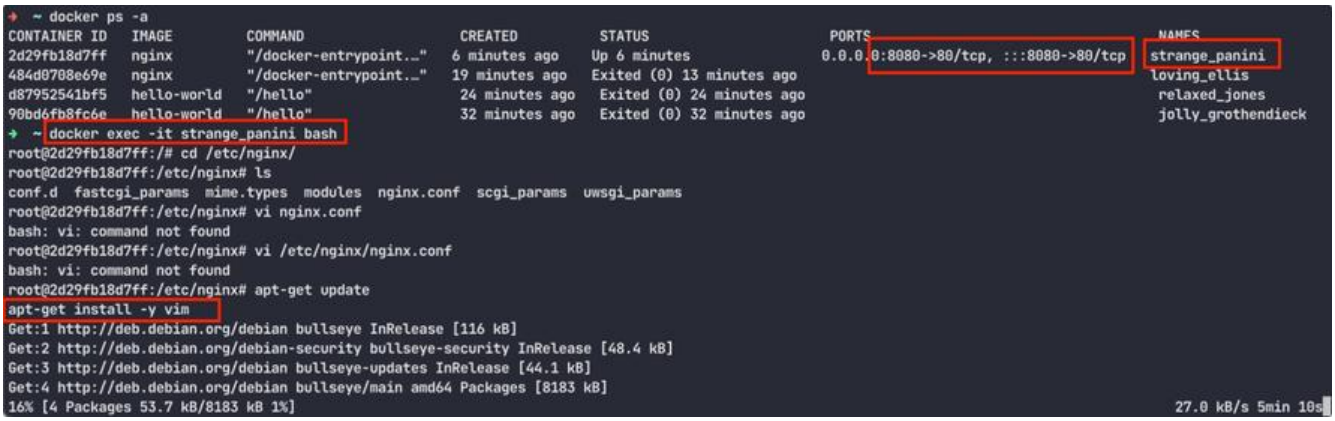
这个命令将拉取最新版本的 Nginx 镜像并在容器中运行它。-d 参数表示在后台运行容器，-p 8080:80

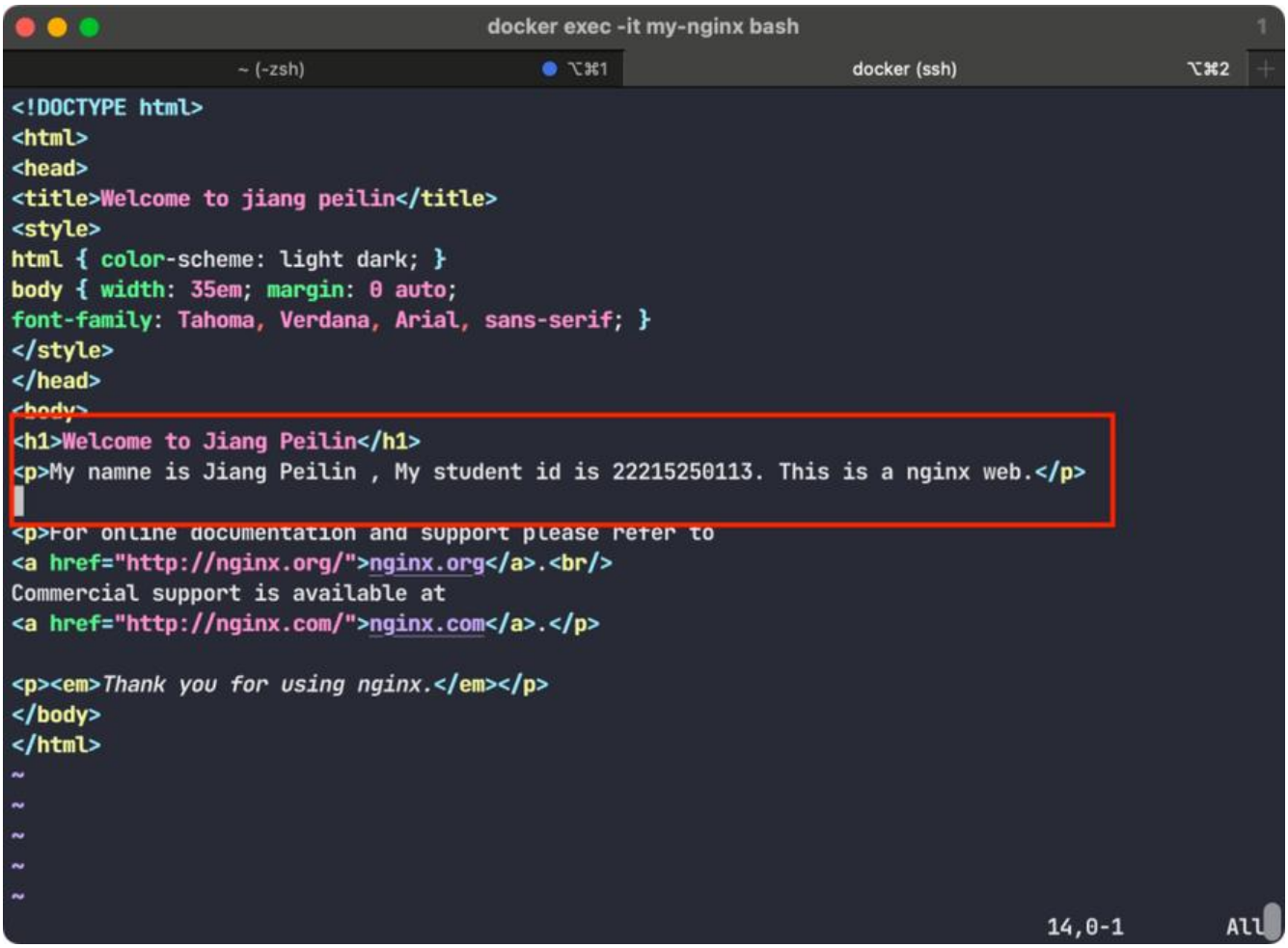
表示将主机的 8080 端口映射到容器的 80 端口，--name my-nginx 表示将容器命名为 my-nginx。

修改nginx web页面为带学号和姓名信息的页面，

找到进程名进入，安装vim编辑器，然后进入/etc/nginx编辑配置文件

docker exec -it my-nginx bash
cd /usr/share/nginx/html
vi index.html





这个命令将进入 my-nginx 容器并打开 /usr/share/nginx/html/index.html 文件。使用 vi 或其他编器修改网页内容并保存。

退出容器并重启 Nginx：

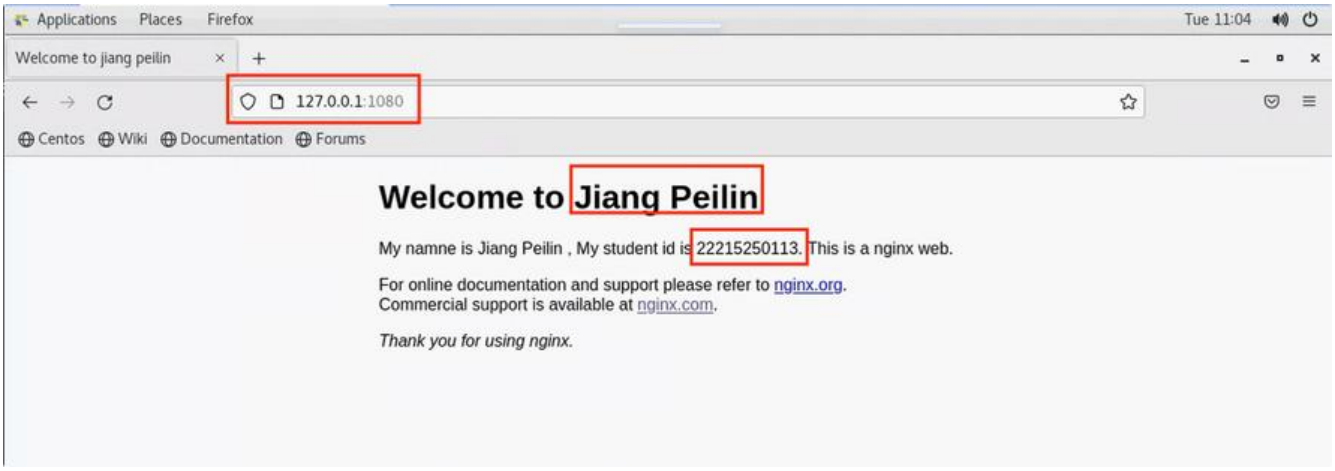
```
exit
docker restart my-nginx
```

这个命令将退出容器并在主机 shell 中重启 my-nginx 容器。Nginx 将重新加载配置文件和修改的网内容。

在浏览器中访问修改后的网页：

http://localhost:8080

在浏览器中输入上述地址，即可访问修改后的网页。



*Docker基础命令

启动

docker run 可加后面参数

--name="容器新名字" 为容器指定一个名称；

-d: 后台运行容器并返回容器ID，也即启动守护式容器(后台运行)；

-i：以交互模式运行容器，通常与 -t 同时使用；

-t：为容器重新分配一个伪输入终端，通常与 -i 同时使用；

也即启动交互式容器(前台有伪终端，等待交互)；

-P: 随机端口映射，大写P

-p: 指定端口映射，小写p

/bin/bash 为进入bash命令行终端

```
docker run -it  ubuntu:16.04 /bin/bash
```

docker run -d redis:6.0.8

退出

docker stop bc588d5fa956/ubuntu

重启

docker restart bc588d5fa956

删除

docker ps
docker rm bc588d5fa956
docker rm -f bc588d5fa956 //强制删除正在运行的容器

查看容器日志

docker logs [CONTAINER ID]

查看已拉取的镜像

docker images
docker images [IMAGE NAME]

查看容器内运行的进程

docker ps
top // linux查看进程
docker top // 查看docker运行的进程
docker inspect [CONTAINER ID]

对容器改名

docker ps
docker rename [CONTAINER ID] [NEW_NAME]

进入正在运行的容器并以命令行进行交互

docker exec -it [CONRAINER ID] /bin/bash
docker attach -it [CONTAINER ID] // 一次性进入，只要进去在退出了就永久停止了

退出正在运行的容器

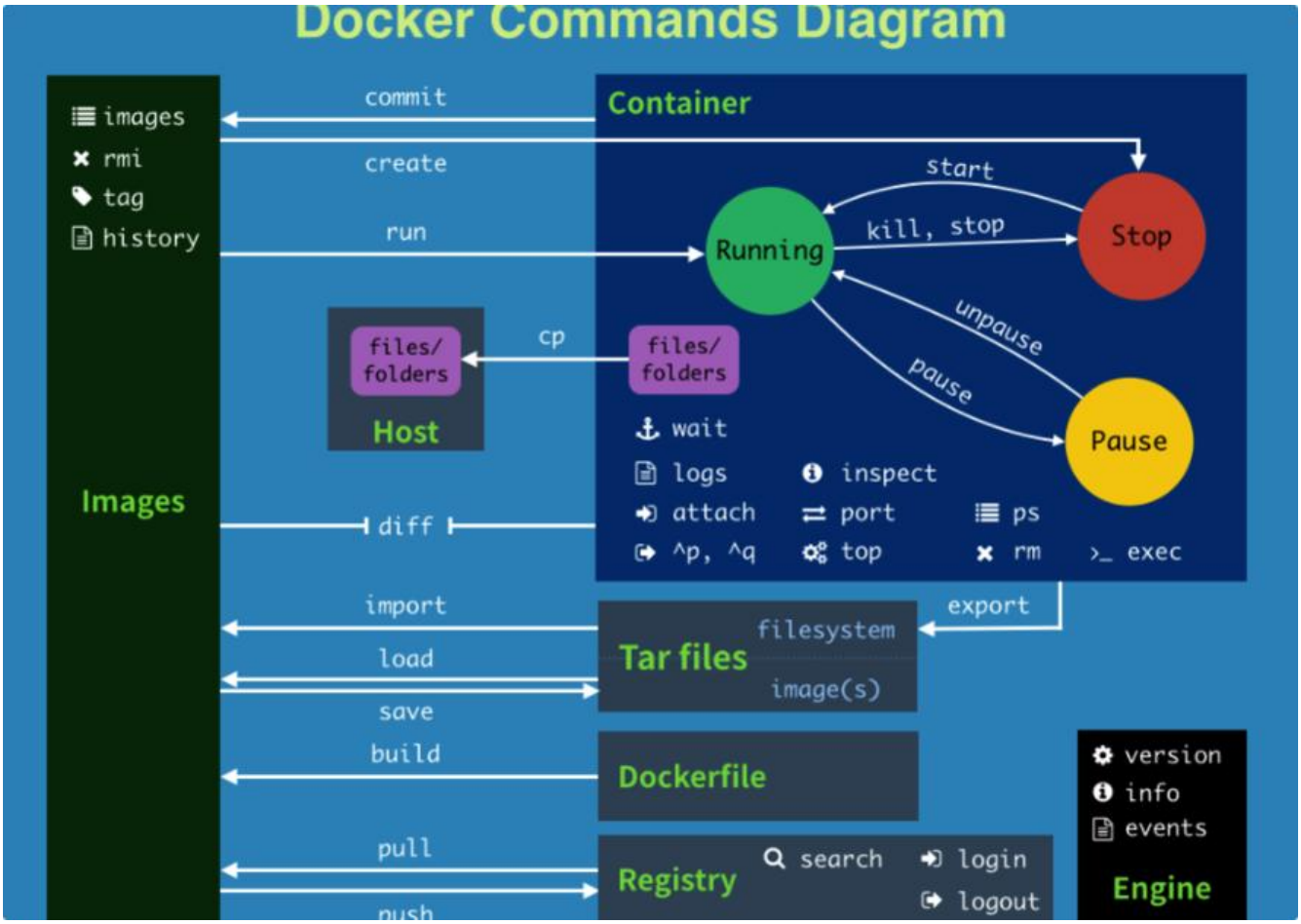
exit
或者
ctrl+p+q

从容器内拷贝文件到主机上

```
root@5ca4b4a8f971:/# cd /tmp
root@5ca4b4a8f971:/tmp# touch a.txt
root@5ca4b4a8f971:/tmp# ls
a.txt
exit

 /tmp docker cp 5ca4b4a8f971:/tmp/a.txt /tmp
Successfully copied 1.54kB to /tmp

 /tmp ls
a.txt                                                   tmpaddon
```

导入和导出容器

```
docker ps
docker export [CONTAINER_ID] > myubuntu.tar  // 把整个容器打包成tar镜像
docker rm-f [CONTAINER_ID] // 删除容器
cat myubuntu.tar | docker import - root/ubuntu:16.04
-> sha256:9303fe3ca04f946c0c5698bf7af8969cde77f0ae1aef6eff5f137755f2f63296
docker images //查看镜像

 /tmp docker run -it --name cloud 9303fe3ca04f /bin/bash
root@041a2243cb04:/#

// 刚才的文件还在 备份恢复成功
root@041a2243cb04:/# cd /tmp
root@041a2243cb04:/tmp# ls
a.txt
```



*Docker镜像

镜像分层，UnionFS联合文件系统，可以把容器看作简易版的Linux的系统

```
➜  /tmp docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
1bc677758ad7: Pull complete
0d0e0ecb256a: Pull complete
c24bf4c725c2: Downloading [============>                                      ]  46.2MB/192.6MB
4fb255c76461: Download complete
b388fec4cd21: Download complete
4800bac131aa: Downloading [===>                                               ]  917.5kB/12.64MB
fc4cc5ff9156: Waiting
```

docker commit 提交副本容器，使他成为新的镜像，或者可以用DOCKERFILE

安装vim

apt-get update
apt-get -y install vim

提交，定制版本

 ~ docker commit -m="vim cmd add ok" -a=="jpl" 6020ab0ffc8a jpl/ubuntu:1.3
sha256:06fbc784d58c75c71db55b958af55b921c36fb8eac27b6141bd6dfd81ad32035

 ~ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
jpl/ubuntu 1.3 06fbc784d58c 11 seconds ago 175MB

启动新镜像,vim成功部署

docker stop cloud
cloud
 ~ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
2d1cacd8b25d redis:6.0.8 "docker-entrypoint.s..." 16 hours ago Up 16 hours 6379/tcp
magical_hamilton
c1c1c0b7f40d nginx "/docker-entrypoint...." 3 days ago Up 3 days 0.0.0.0:1080->
0/tcp, :::1080->80/tcp my-nginx
2d29fb18d7ff nginx "/docker-entrypoint...." 3 days ago Up 3 days 0.0.0.0:8080->
0/tcp, :::8080->80/tcp strange_panini
 ~ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
jpl/ubuntu 1.3 06fbc784d58c About a minute ago 175MB
root/ubuntu 16.04 9303fe3ca04f 34 minutes ago 86.2MB
hello-world latest 9c7a54a9a43c 7 days ago 13.3kB
tomcat latest 311570738ca3 7 days ago 475MB
nginx latest 448a08f1d2f9 8 days ago 142MB
centos latest 5d0da3dc9764 20 months ago 231MB
ubuntu 16.04 b6f507652425 20 months ago 135MB
redis 6.0.8 16ecd2772934 2 years ago 104MB
 ~ docker run -it 06fbc784d58c /bin/bash
root@988f45401972:/# vim
root@988f45401972:/#

*Docker 容器卷

主机和容器实现共享 /宿主机决定路径:/容器目录

 image docker run -it --privileged=true -v /tmp/host_data:/tmp/docker_data --name=u1 ub
ntu:16.04
root@39f21ec174f1:/#

在共享的文件夹内 容器新建文件，宿主机也会同步

root@39f21ec174f1:/tmp/docker_data# cd /tmp/docker_data/
root@39f21ec174f1:/tmp/docker_data# touch build_form_ubuntu.txt
root@39f21ec174f1:/tmp/docker_data# ls
build_form_ubuntu.txt
root@39f21ec174f1:/tmp/docker_data# exit
exit

 image cd /tmp/host_data
 host_data ls
build_form_ubuntu.txt

查看容器

docker inspect 39f21ec174f1

容器关闭，宿主机新建文件，容器打开，文件夹依然同步共享

使容器可读可写:rw

image docker run -it --privileged=true -v /tmp/host_data:/tmp/docker_data:rw --name=u1 u
untu:16.04

使容器可读不能写:ro

image docker run -it --privileged=true -v /tmp/host_data:/tmp/docker_data --name=u1:ro u
untu:16.04

使容器2继承容器1的卷规则

docker run -it --privileged=true --volumes- from u1 --name u2 ubuntu:16.04

*Docker常规安装Tomcat

Tomcat

docker search tomcat
docker pull tomcat

使用tomcat镜像运行实例

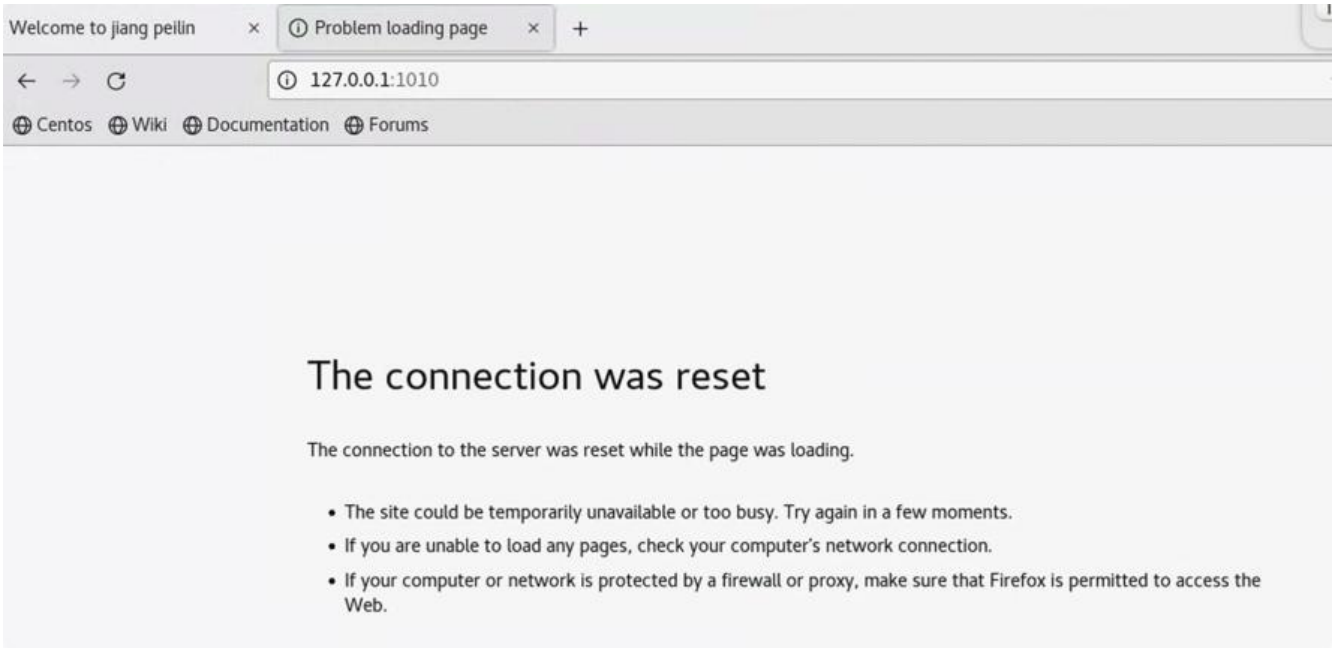
docker run -itd -p 8080:8080 --name=tomcat1 tomcat // -P随机分配端口

```
➜  ~ docker run -d -p 8080:8080 --name t1 tomcat
dae867a97bc1ef7874c81a1139e3b7f57bc11d18a732c1ae3d30882e65d88af7
➜  ~ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS             PORTS
dae867a97bc1   tomcat         "catalina.sh run"        4 seconds ago       Up 3 seconds       0.0.0.0:8080->8080/tcp,
39f21ec174f1   ubuntu:16.04   "/bin/bash"              About an hour ago   Up About an hour
2d1cacd8b25d   redis:6.0.8    "docker-entrypoint.s…"   18 hours ago        Up 18 hours        6379/tcp
milton
c1c1c0b7f40d   nginx          "/docker-entrypoint.…"   3 days ago          Up 3 days          0.0.0.0:1080->80/tcp, ::
➜  ~
```

登陆到Tomcat

 ~ docker exec -it dae867a97bc1 /bin/bash

root@dae867a97bc1:/usr/local/tomcat#



修改webapps文件权限

root@dae867a97bc1:/usr/local/tomcat# rm -r webapps
root@dae867a97bc1:/usr/local/tomcat# mv webapps.dist webapps

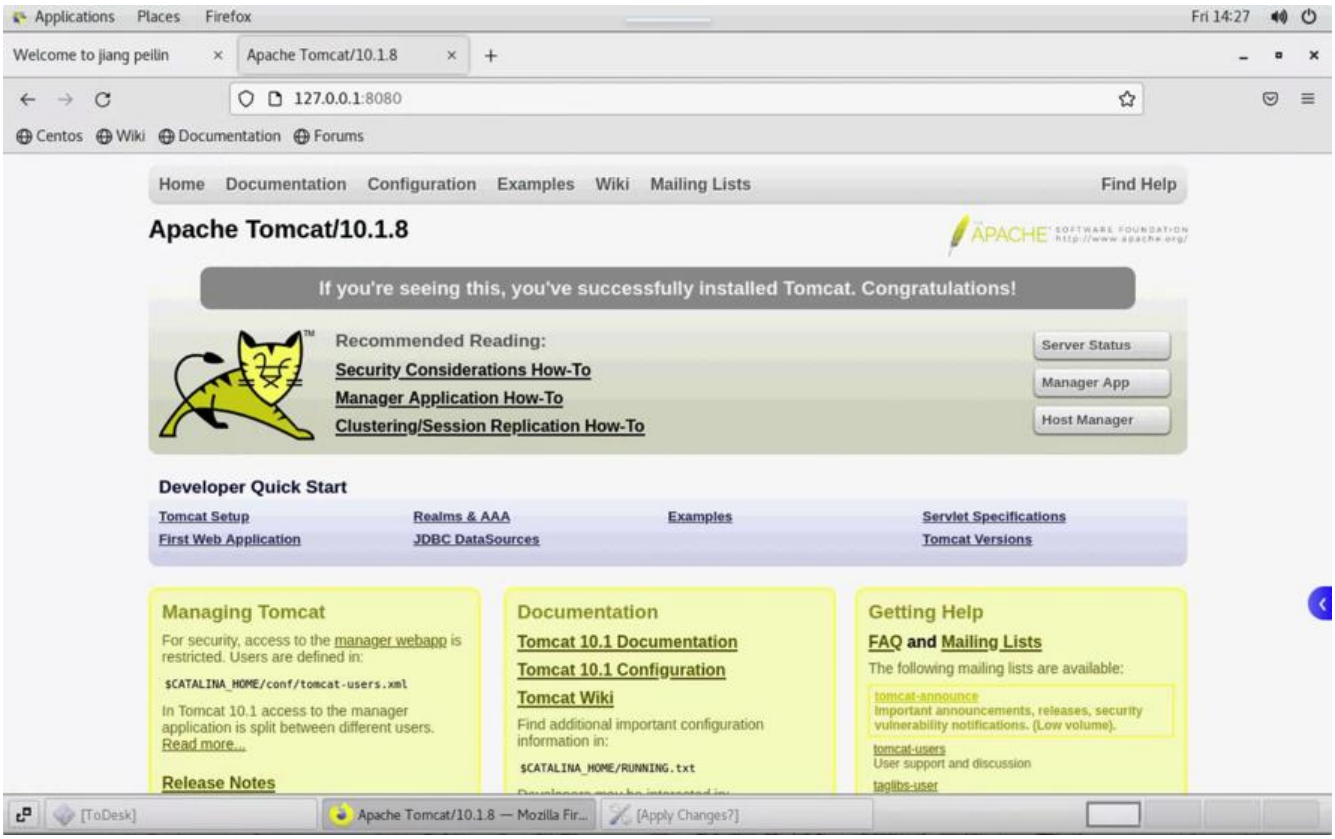
修改sysctl.conf的内核转发

通过vim /etc/sysctl.conf把里面的net.ipv4.ip_forward = 0修改为net.ipv4.ip_forward = 1后进行保存退出，通过sysctl -p命令使修改的内核转发文件生效

进入docker容器，启动相对应的tomcat服务

通过docker attach a85c8c323a30（正在启动的docker容器ID）

进入docker容器后，通过cd bin进入bin目录下，找到startup.sh文件，直接输入startup.sh这个命令动这个脚本后，这个tomcat服务就启动了



或者免修改版

```
docker pull billygoo/tomcat8-jdk8
docker run -d -p 8080:8080 --name mytomcat8 billygoo/tomcat8-jdk8
```

如果端口占用了-> 端口解绑

1.查看8080端口是否被占用

```
netstat -anp | grep 8080
```

输出结果：tcp 0 0 :::8080 :::* LISTEN 3000/java

由上可知8080端口已经被开启。

2.查看占用8080端口的进程：

```
fuser -v -n tcp 8080
```

输出结果：

```
USER        PID   ACCESS COMMAND   8080/tcp:
zhu         1154    F.... java
```

3.杀死占用8080端口的进程：

kill -s 9 1154(自己的进程号).

4.查看所有进程：

ps

输出结果：

PID TTY TIME CMD

2949 pts/1 00:00:00 bash

3037 pts/1 00:00:00 ps

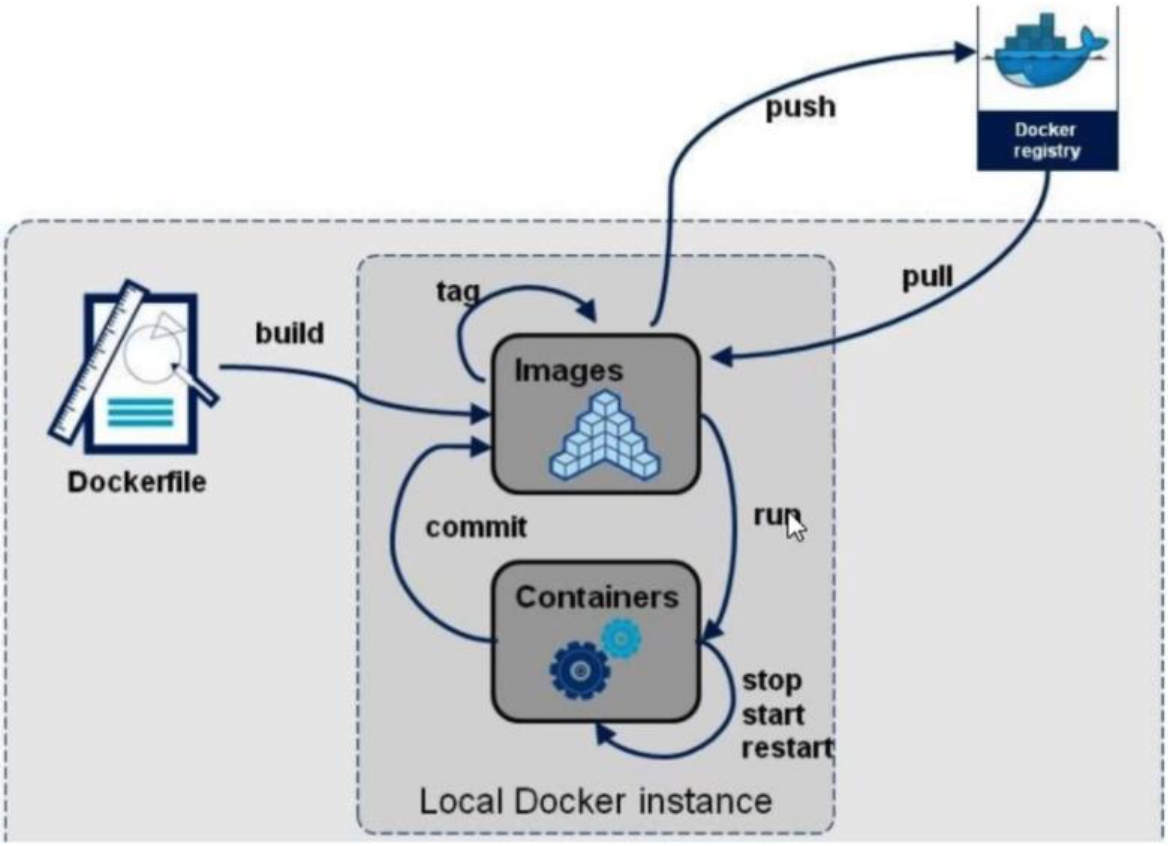
这是便可发现1154进程已经不存在了

编写Dockerfile文件-JDK8

用来构造Dokcer镜像的文本文件，随时变化，dockerfile独立于docker外部

Dockerfile面向开发，Docker镜像成为交付标准，Docker容器则涉及部署与运维，三者缺一不可，力充当Docker体系的基石。

关键保留字要大写



1：每条保留字指令都必须为大写字母且后面要跟随至少一个参数
2：指令按照从上到下，顺序执行
3：#表示注释
4：每条指令都会创建一个新的镜像层并对镜像进行提交

下载JDK8：https://mirrors.yangxingzhen.com/jdk/

jdk-8u171-linux-x64.tar.gz	182.0 MiB	2022-Sep-05 14:58

把下载好的jdk8移动到根目录的myfile文件夹

mv jdk-8u171-linux-x64.tar.gz /myfile

编写Dockerfile文件-jdk8

/myfile vim Dockerfile

vim命令： ggvG d全选删除

```
FROM ubuntu:16.04
MAINTAINER jpl<jiangpeilin22@s.nuit.edu.cn>

ENV MYPATH /usr/local
WORKDIR $MYPATH

#安装vim编辑器
#安装ifconfig命令查看网络IP
#安装java8及lib库
RUN apt-get update && \
```

```
    apt-get install -y wget vim net-tools locales build-essential && \
    rm -rf /var/lib/apt/lists/*

# 添加中文支持
ENV TZ=Asia/Shanghai
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
RUN locale-gen zh_CN.UTF-8 && \
    DEBIAN_FRONTEND=noninteractive dpkg-reconfigure locales
ENV LANG zh_CN.UTF-8
ENV LANGUAGE zh_CN:zh
ENV LC_ALL zh_CN.UTF-8
ENV LC_ALL="C.UTF-8" LANG="C.UTF-8"

#ADD 是相对路径jar,把jdk-8u171-linux-x64.tar.gz添加到容器中,安装包必须要和Dockerfile文件在一位置
ADD jdk-8u171-linux-x64.tar.gz /usr/local/java/
LABEL Description="This image is the base os images."  Version="1.0"
RUN echo "Asia/Shanghai" > /etc/timezone

#在构建镜像时，指定镜像的工作目录，之后的命令都是基于此工作目录，如果不存在，则会创建目录
WORKDIR /usr/local/java
ENV JAVA_HOME /usr/local/java/jdk1.8.0_171
ENV JRE_HOME $JAVA_HOME/jre
ENV CLASSPATH $JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib:$CLASSPTH
ENV PATH $JAVA_HOME/bin:$PATH

EXPOSE 80

CMD echo $MYPATH && echo "success--------------ok" && /bin/bash
```

docker build -t ubuntujava8:1.5 .

docker build -t [NAME]:[VERSION] .

空格加一点代表指定当前文件夹

```
➜  /myfile docker build -t ubuntujava8:1.5 .
[+] Building 9.4s (18/18) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 1.43kB                                                                              0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/ubuntu:16.04                                                     0.0s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 50B                                                                                    0.0s
 => [ 1/13] FROM docker.io/library/ubuntu:16.04                                                                     0.0s
 => CACHED [ 2/13] WORKDIR /usr/local                                                                               0.0s
 => CACHED [ 3/13] RUN  apt-get update &&     apt-get install -y wget                                               0.0s
 => CACHED [ 4/13] RUN  apt-get update &&     apt-get install -y vim                                                0.0s
 => CACHED [ 5/13] RUN  apt-get update &&     apt-get install -y net-tools                                          0.0s
 => CACHED [ 6/13] RUN apt update; exit 0                                                                           0.0s
 => CACHED [ 7/13] RUN apt install -y build-essential                                                               0.0s
 => CACHED [ 8/13] RUN ln -snf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime && echo Asia/Shanghai > /etc/timezone  0.0s
 => CACHED [ 9/13] RUN apt-get install -y locales                                                                   0.0s
 => CACHED [10/13] RUN locale-gen zh_CN.UTF-8 && DEBIAN_FRONTEND=noninteractive dpkg-reconfigure locales           0.0s
 => CACHED [11/13] RUN locale-gen zh_CN.UTF-8                                                                       0.0s
 => [12/13] ADD jdk-8u171-linux-x64.tar.gz /usr/local/java/                                                         4.3s
 => [13/13] RUN echo "Asia/Shanghai" > /etc/timezone ENV JAVA_HOME /usr/local/java/jdk1.8.0_171                    0.3s
 => exporting to image                                                                                              4.8s
 => => exporting layers                                                                                             4.8s
 => => writing image sha256:6eea5b9c39a2872550d2cb18c988f49f45d83215bbf702c67b78c76584e0e806                        0.0s
 => => naming to docker.io/library/ubuntujava8:1.5                                                                  0.0s
➜  /myfile
```

```
➜  /myfile docker images
REPOSITORY                                                TAG       IMAGE ID       CREATED          SIZE
jbuntujava8                                               1.5       8fb5d66ad6f3   7 minutes ago    780MB
registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu  1.3       06fbc784d58c   5 hours ago      175MB
root/ubuntu                                               16.04     9303fe3ca04f   6 hours ago      86.2MB
hello-world                                               latest    9c7a54a9a43c   7 days ago       13.3kB
tomcat                                                    latest    311570738ca3   7 days ago       475MB
nginx                                                     latest    448a08f1d2f9   8 days ago       142MB
centos                                                    latest    5d0da3dc9764   20 months ago    231MB
ubuntu                                                    16.04     b6f507652425   20 months ago    135MB
redis                                                     6.0.8     16ecd2772934   2 years ago      104MB
billygoo/tomcat8-jdk8                                     latest    30ef4019761d   4 years ago      523MB
```

验证

docker run -it --name jpl 8fb5d66ad6f3 /bin/bash

vim

```
root@8179a62d5927:/usr/local/java# vim a.txt
root@8179a62d5927:/usr/local/java# cat a.txt
name:蒋沛林
student_id:22215250113
```

ifconfig

```
root@8179a62d5927:/usr/local/java# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:656 (656.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

jdk

```
root@8179a62d5927:/usr/local/java# java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
root@8179a62d5927:/usr/local/java# javac -version
javac 1.8.0_171
```

编写Dockerfile文件-tomcat

下载Tomcat7

https://archive.apache.org/dist/tomcat/tomcat-7/v7.0.70/src/

新建一个文件夹，并命名为tomcat7-jre7

mkdir tomcat7-jre7

```
➜  /tomcat7-jre7 pwd
/tomcat7-jre7
```

将下载的tomcat复制到该目录：

mv /root/Downloads/apache-tomcat-7.0.70-src.tar.gz /tomcat7-jre7

创建Dockerfile文件

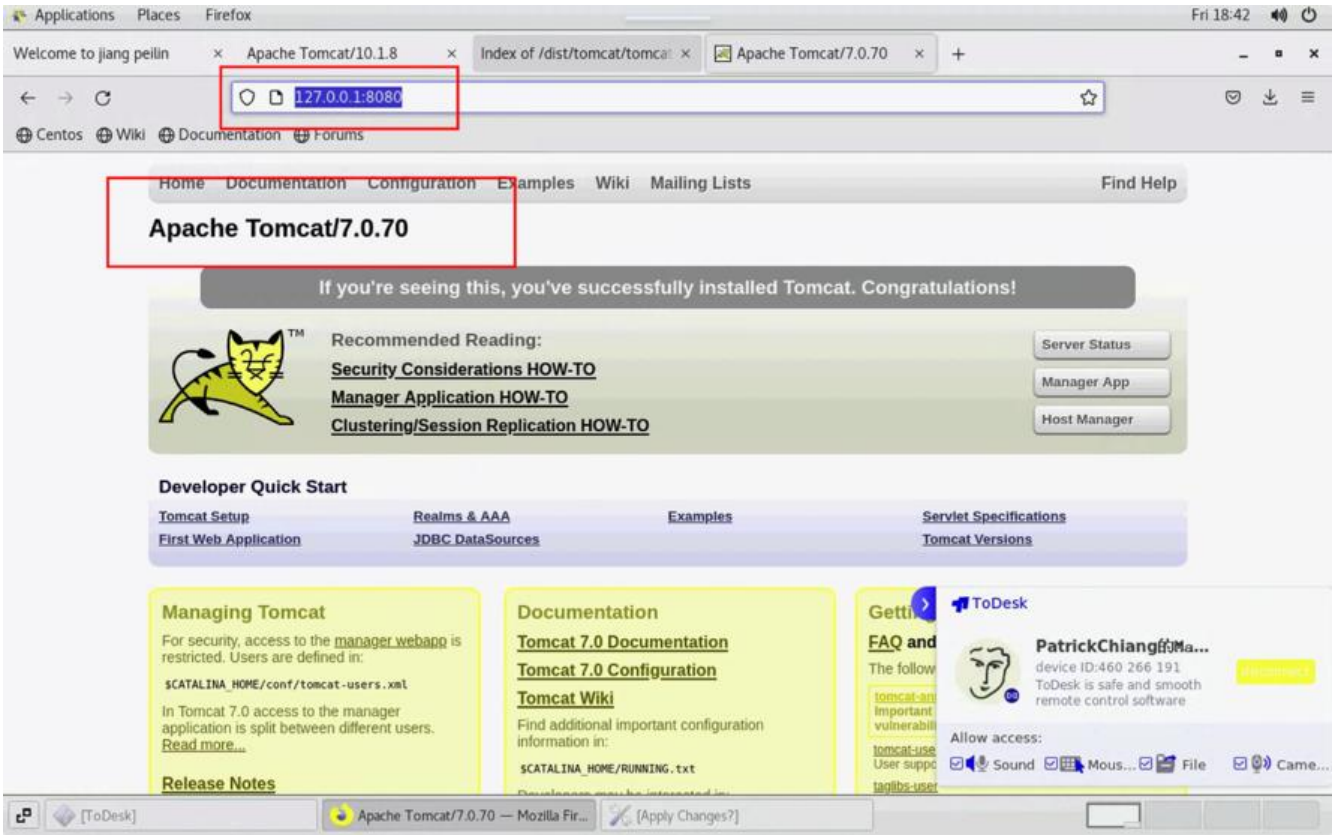
cd tomcat7-jre7

touch Dockerfile

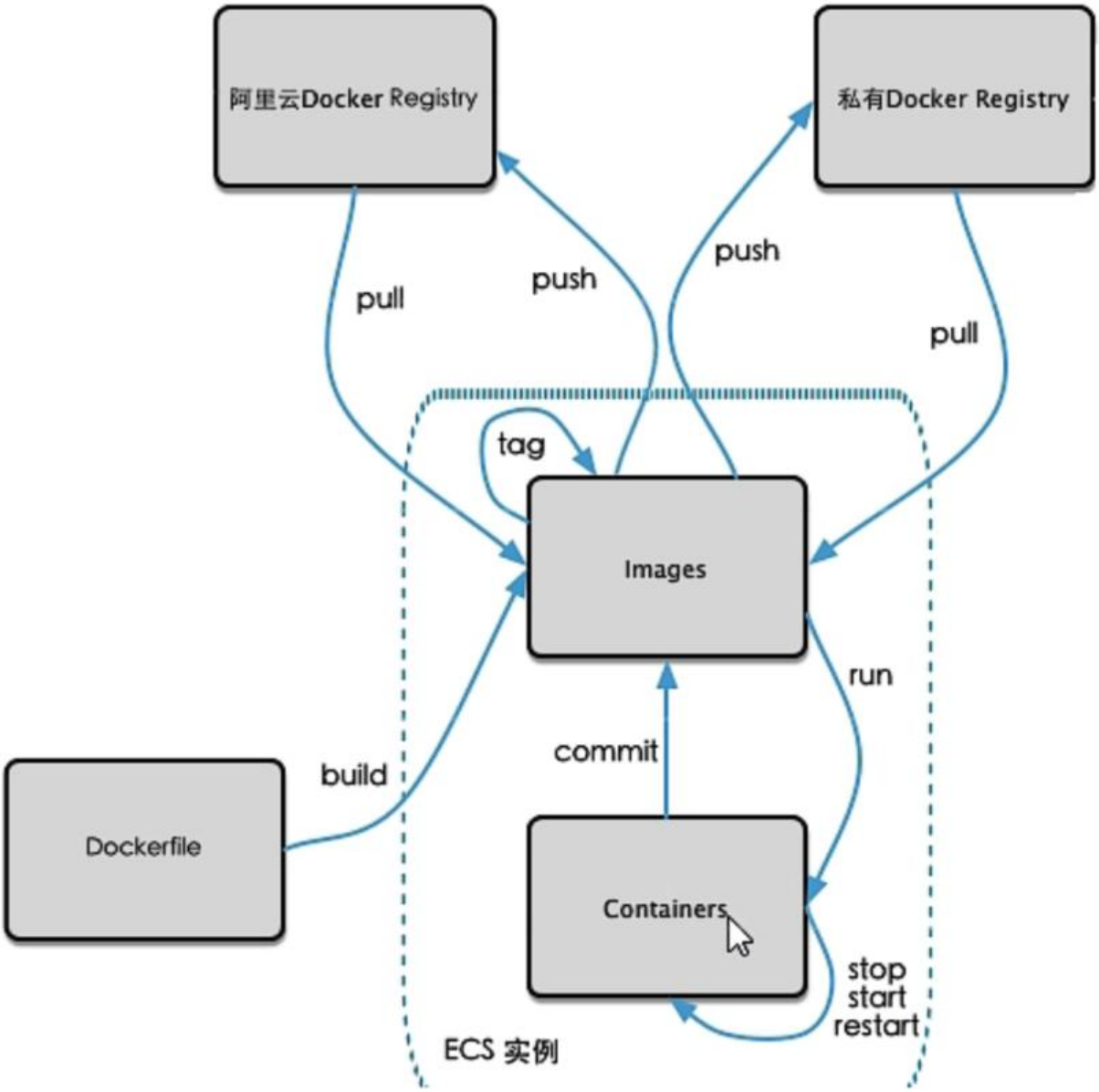
```
# Base image
FROM ubuntujava8:1.5

# Install dependencies
RUN apt-get update && \
    apt-get install -y openjdk-8-jdk wget && \
    apt-get clean

# Download and extract Tomcat
RUN mkdir -p /opt/tomcat && \
    wget -O /tmp/tomcat.tar.gz http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.70/bin/a
ache-tomcat-7.0.70.tar.gz && \
    tar xzvf /tmp/tomcat.tar.gz -C /opt/tomcat --strip-components=1 && \
    rm /tmp/tomcat.tar.gz

# Set environment variables
ENV CATALINA_HOME /opt/tomcat
ENV PATH $CATALINA_HOME/bin:$PATH

# Remove unnecessary files
RUN rm -rf $CATALINA_HOME/webapps/examples $CATALINA_HOME/webapps/docs

# Expose Tomcat default port
EXPOSE 8080

# Start Tomcat
CMD ["catalina.sh", "run"]
```

docker build -t ubuntutomcat:1.0 .

docker run -d -p 8080:8080 234ab654772f


Applications Places Firefox
Fri 18:42
Welcome to jiang peilin
Apache Tomcat/10.1.8
Apache Tomcat/7.0.70
127.0.0.1:8080
Centos Wiki Documentation Forums
Home Documentation Configuration Examples Wiki Mailing Lists
Find Help
Apache Tomcat/7.0.70
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 7.0 access to the manager application is split between different users.
Read more...
Release Notes
Documentation
Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
$CATALINA_HOME/RUNNING.txt
ToDesk
PatrickChiang的Ma...
device ID:460 266 191
ToDesk is safe and smooth remote control software
Allow access:
Sound
Mous...
File
Came...
[ToDesk]
Apache Tomcat/7.0.70 — Mozilla Fir...
[Apply Changes?]

*本地镜像发布阿里云



阿里云

https://cr.console.aliyun.com/cn-hangzhou/instances

1. 登录阿里云Docker Registry

docker login --username=aliyun1451226290 registry.cn-hangzhou.aliyuncs.com

用于登录的用户名为阿里云账号全名，密码为开通服务时设置的密码。

您可以在访问凭证页面修改凭证密码。

2. 从Registry中拉取镜像

docker pull registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:[镜像版本号]

docker pull registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3

3. 将镜像推送到Registry

```
docker login --username=aliyun1451226290 registry.cn-hangzhou.aliyuncs.com
docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:[镜像版本号]
docker push registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:[镜像版本号]
```

请根据实际镜像信息替换示例中的[ImageId]和[镜像版本号]参数。

```
docker login --username=aliyun1451226290 registry.cn-hangzhou.aliyuncs.com
docker tag 06fbc784d58c registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
docker push registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
```

```
➜  ~ docker login --username=aliyun1451226290 registry.cn-hangzhou.aliyuncs.com
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
➜  ~ docker tag 06fbc784d58c registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
➜  ~ docker push registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
The push refers to repository [registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu]
4dc7cc5b4f19: Pushing [>                                                  ]  1.671MB/88.43MB
38a3ce3081a8: Pushing [===>                                               ]  5.913MB/86.21MB
```

4. 选择合适的镜像仓库地址

从ECS推送镜像时，可以选择使用镜像仓库内网地址。推送速度将得到提升并且将不会损耗您的公网量。

如果您使用的机器位于VPC网络，请使用 registry-vpc.cn-hangzhou.aliyuncs.com 作为Registry的名登录。

5. 示例

使用"docker tag"命令重命名镜像，并将它通过专有网络地址推送至Registry。

```
$ docker images
REPOSITORY                                         TAG                 IMAGE ID            CREATED
    VIRTUAL SIZE
registry.aliyuncs.com/acs/agent                    0.7-dfb6816         37bb9c63c8b2        7
ays ago         37.89 MB
$ docker tag 37bb9c63c8b2 registry-vpc.cn-hangzhou.aliyuncs.com/acs/agent:0.7-dfb6816
```

使用 "docker push" 命令将该镜像推送至远程。

docker push registry-vpc.cn-hangzhou.aliyuncs.com/acs/agent:0.7-dfb6816

```
➜  ~ docker images
REPOSITORY                                                TAG       IMAGE ID       CREATED          SIZE
jpl/ubuntu                                                1.3       06fbc784d58c   22 minutes ago   175MB
registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu  1.3       06fbc784d58c   22 minutes ago   175MB
root/ubuntu                                               16.04     9303fe3ca04f   56 minutes ago   86.2MB
hello-world                                               latest    9c7a54a9a43c   7 days ago       13.3kB
tomcat                                                    latest    311570738ca3   7 days ago       475MB
nginx                                                     latest    448a08f1d2f9   8 days ago       142MB
centos                                                    latest    5d0da3dc9764   20 months ago    231MB
ubuntu                                                    16.04     b6f507652425   20 months ago    135MB
redis                                                     6.0.8     16ecd2772934   2 years ago      104MB
➜  ~ docker push registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu
```

```
➜  ~ docker push registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
The push refers to repository [registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu]
4dc7cc5b4f19: Layer already exists
38a3ce3081a8: Layer already exists
1.3: digest: sha256:1411c2cbc75c54d2b0518ca4e38dac3bab1f11a50001771f9f8f821965f0cfe3 size: 741
```

验证：

删除本地镜像

```
 ~ docker rmi -f 06fbc784d58c
Untagged: jpl/ubuntu:1.3
Untagged: registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
Untagged: registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu@sha256:1411c2cbc7
c54d2b0518ca4e38dac3bab1f11a50001771f9f8f821965f0cfe3
Deleted: sha256:06fbc784d58c75c71db55b958af55b921c36fb8eac27b6141bd6dfd81ad32035
```

拉取

```
 ~ docker pull registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
1.3: Pulling from patrickchiang/myubuntu
03334ed17bce: Already exists
7fb45c3d1066: Already exists
Digest: sha256:1411c2cbc75c54d2b0518ca4e38dac3bab1f11a50001771f9f8f821965f0cfe3
Status: Downloaded newer image for registry.cn-hangzhou.aliyuncs.com/patrickchiang/myub
ntu:1.3
registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu:1.3
 ~ docker images
REPOSITORY                                                TAG       IMAGE ID       CREATED          SIZE
registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu  1.3       06fbc784d58c   26 minu
es ago      175MB
root/ubuntu                                               16.04     9303fe3ca04f   About an hour ago   86.2M

hello-world                                               latest    9c7a54a9a43c   7 days ago       13.3kB
tomcat                                                    latest    311570738ca3   7 days ago       475MB
nginx                                                     latest    448a08f1d2f9   8 days ago       142MB
centos                                                    latest    5d0da3dc9764   20 months ago    231MB
ubuntu                                                    16.04     b6f507652425   20 months ago    135MB
redis                                                     6.0.8     16ecd2772934   2 years ago      104MB
```

安装, 成功打开vim

```
➜  ~ docker images
REPOSITORY                                                TAG       IMAGE ID       CREATED             SIZE
registry.cn-hangzhou.aliyuncs.com/patrickchiang/myubuntu  1.3       06fbc784d58c   26 minutes ago      175MB
root/ubuntu                                               16.04     9303fe3ca04f   About an hour ago   86.2MB
hello-world                                               latest    9c7a54a9a43c   7 days ago          13.3kB
tomcat                                                    latest    311570738ca3   7 days ago          475MB
nginx                                                     latest    448a08f1d2f9   8 days ago          142MB
centos                                                    latest    5d0da3dc9764   20 months ago       231MB
ubuntu                                                    16.04     b6f507652425   20 months ago       135MB
redis                                                     6.0.8     16ecd2772934   2 years ago         104MB
➜  ~
➜  ~ docker run -it 06fbc784d58c /bin/bash
root@ebc20e1706cb:/# vim
root@ebc20e1706cb:/#
```

原文链接：Docker 实验 pull_dockerfile_tomcat_ 简单 nginx