

## docker

笔记本: 我的第一个笔记本  
创建时间: 2020/11/13 9:47 更新时间: 2021/2/2 17:07  
作者: 986087957@qq.com  
URL: <https://www.cnblogs.com/duanxz/p/7905233.html>

---

## Docker国内镜像源

### 离线安装:

创建或修改 /etc/docker/daemon.json 文件

```
# vi /etc/docker/daemon.json
{
  "registry-mirrors": ["https://registry.docker-cn.com"]
}
systemctl restart docker.service
Docker中国区官方镜像
```

<https://registry.docker-cn.com>

网易

<http://hub-mirror.c.163.com>

ustc

<https://docker.mirrors.ustc.edu.cn>

中国科技大学

<https://docker.mirrors.ustc.edu.cn>

阿里云容器 服务

<https://cr.console.aliyun.com/>

```
SHGS-WXTS:~ # cd ../etc/docker/
SHGS-WXTS:/etc/docker # ll
total 8
-rw-r--r-- 1 root root 174 Nov 16 11:37 daemon.json
-rw----- 1 root root 244 Oct 15 2019 key.json
SHGS-WXTS:/etc/docker #
```

安装的docker在这里, 如果需要 修改镜像下载的地址, 在dameon.json 改

**查看是否已安装docker列表: yum list installed | grep docker, 什么都不提示说明咱们还没有安装**

启动docker

systemctl start docker

设置开机启动

systemctl enable docker.service

关闭docker

systemctl stop docker

查看docker的运行状态

systemctl status docker

启动Docker: **service docker start**

查看是否启动: **ps -ef|grep docker**

下载镜像: **docker pull 镜像名称**

#pull 后面是镜像名称, 你可以先docker search java 在私服搜索java 是否有这个镜像再下载

### 首先: 下载镜像

官方镜像中心仓库地址: <https://hub.docker.com/search/?type=image>

使用命令: **docker pull \$imageName** (\$imageName为具体镜像名字, 如tomcat,nginx等)

**查看镜像是否安装成功: docker images 镜像名称**

## 第二步：根据镜像创建一个可运行的容器

以tomcat为例，使用

#下载最新的tomcat版本  
docker pull tomcat 或 docker image pull tomcat

### 启动tomcat镜像

查看docker所有镜像docker images

因为服务器本身有的tomcat服务端口是8080，为避免端口冲突，所以启动tomcat镜像命令用的docker run -d -p 8081:8080 tomcat。

(docker run -d -p 8081:8080 tomcat命令相当于docker里面是8080端口，对于服务器而言是8081端口)  
(docker相当于一个容器，docker里面的端口和服务器的端口是独立的。)  
(所以，直接服务器ip:8080是访问不到docker的tomcat镜像的，要用服务器ip:8081。)

(若是中转服务器上部署的服务，那么当用浏览器访问的时候，要看当前“服务器IP:端口”对应的暴露给外部的url是多少。)

查看容器日志docker logs -f 容器id

[root@localhost ~]# docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
[root@localhost ~]# docker images						
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE		
docker.io/tomcat	latest	df72227b40e1	4 days ago	647 MB		
[root@localhost ~]# docker run -d -p 8081:8080 tomcat						
5cfa9100f191032b577cc4d345ed5c0b656fbeb46a4f76a8953dd61e25d695c						
[root@localhost ~]# docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ES						
5cfa9100f191	tomcat	"catalina.sh run"	19 seconds ago	Up 17 seconds	0.0.0.0:8081->8080/tcp	prl
celess kowalevski						
[root@localhost ~]#						

关闭容器命令docker stop 容器id

可以查看指定容器的log：`docker logs -f 容器id`

docker logs -f -t --since="2017-05-31" --tail=10 edu\_web\_1

--since：此参数指定了输出日志开始日期，即只输出指定日期之后的日志。-f：查看实时日志 -t：查看日志产生的日期 -tail=10：查看最后的10条日志。 edu\_web\_1：容器名称

docker restart 容器ID或容器名：不管容器是否启动，直接重启容器

docker images #查看本地全部镜像

#以后台交互式窗口的方式运行tomcat镜像，并将容器命名为tomcat8080  
#其中-itd为-i -t -d 三个命令的所写  
#**docker run**：创建一个新的容器并运行一个命令  
docker run -itd --name tomcat8080 tomcat

#查看当前运行的容器信息  
docker ps

通过如下两个都可以删除镜像：

docker rmi [image]

或者：

docker image rm [image]

# 移除容器  
docker rm 5421 #5421容器id

## 查看容器的信息container (ps)

docker ps命令可以查看容器的CONTAINER ID、NAME、IMAGE NAME、端口开启及绑定、容器启动后执行的COMMNAD。最常用的功能是通过ps来找到CONTAINER ID，以便对特定容器进行操作。

docker ps 默认显示当前正在运行中的container

docker ps -a 查看包括已经停止的所有容器

docker ps -l 显示最新启动的一个容器（包括已停止的）

### -- 列出机器上的镜像 (images)

# docker images REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE  
ubuntu 14.10 2185fd50e2ca 13 days ago 236.9 MB  
...

其中我们可以根据REPOSITORY来判断这个镜像是来自哪个服务器，如果没有 / 则表示官方镜像，类似于username/repos\_name表示Github的个人公共库，类似于[registory.example.com:5000/repos\\_name](#)则表示的是私服。

IMAGE ID列其实是缩写，要显示完整则带上--no-trunc选项

### 在docker index中搜索image (search)

Usage: docker search TERM

# docker search seanloNAME DESCRIPTION STARS OFFICIAL AUTOMATED

seanlook/centos6 sean's docker repos 0

```
1、docker -h 帮助
2、获取镜像
docker pull NAME[:TAG]
docker pull centos:latest
3、启动Container盒子
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
docker run -t -i centos /bin/bash
4、查看镜像列表，列出本地的所有images
docker images [OPTIONS] [NAME]
docker images centos
5、查看容器列表，可看到我们创建过的所有container
docker ps [OPTIONS]
docker ps -a
6、删除镜像，从本地删除一个已经下载的镜像
docker rmi IMAGE [IMAGE...]
docker rmi centos:latest
7、移除一个或多个容器实例
docker rm [OPTIONS] CONTAINER [CONTAINER...]
移除所有微运行的容器
docker rm sudo docker ps -aq
8、停止一个正在运行的容器
docker kill [OPTIONS] CONTAINER [CONTAINER...]
docker kill 026e
9、重启一个正在运行的容器
docker restart [OPTIONS] CONTAINER [CONTAINER...]
docker restart 026e
10、停止一个已经停止的容器
docker start [OPTIONS] CONTAINER [CONTAINER...]

docker start 026e
```

如果启动tomcat访问404，

使用命令: `docker exec -it 运行的tomcat容器ID /bin/bash` 进入到tomcat的目录

进入webapps文件夹,发现里面是空的(tomcat默认的欢迎页面实际上放在的路径应该是:webapps/ROOT/index.jsp或者index.html)

发现旁边还有个webapps.dist的文件,进入才发现原本应该在webapps文件中的文件都在webapps.dist文件中,现在也不知道为什么!!!

将webapps.dist重命名成webapps即可,原来的webapps(空文件)可以删除或者命名成其他的名字既:`mv webapps.dist`

webapp 注:如果是在服务器上记得将映射的宿主端口开放,不然也无法访问到。

docker部署web项目参考地址: [https://blog.csdn.net/qq\\_26709459/article/details/80475068](https://blog.csdn.net/qq_26709459/article/details/80475068)

停止tomcat后再次重启还会出现上次404的情况,下面是一劳永逸的办法:

[https://blog.csdn.net/qq\\_40891009/article/details/104012792](https://blog.csdn.net/qq_40891009/article/details/104012792)

测试成功案例:

`docker commit -a='WL' -m='将docker容器上pull下来的tomcat生成新的镜像' 26adffd0ba54 newimagetomcat`

在上一篇[https://blog.csdn.net/qq\\_40891009/article/details/103898876](https://blog.csdn.net/qq_40891009/article/details/103898876)解决了容器报404错误后,如果还处于容器内既当前目录还是在tomcat镜像生成的容器目录下,可以使用:ctrl+p+q不退出容器的方式返回到宿主机目录下

使用docker commit命令将修改后的容器生成新的镜像

docker commit命令详解:

作用:将运行着的容器映射成新的镜像

格式: `docker commit -a='作者-这个随意写' -m='修改内容-随意写' 容器名称或者ID 新生成镜像的名称`

例子: `docker commit -a='谁修改了我的bug(既作者信息)' -m='将修改后的容器映射成新的镜像(既修改内容,随意写)' tomcat`  
idea/tomcat

运行新的镜像,访问tomcat首页,发现不会再出现404错误,以后每次创建tomcat容器时,使用我们自己生成的镜像即可(它跟阿里云拉下来的进行并没什么差别,只是保存了我们之前对容器做的修改)

```
root@003f010d0fb8:/usr/local/tomcat# ls -l
total 160
-rw-r--r-- 1 root root 19318 Dec 7 19:23 BUILDING.txt
-rw-r--r-- 1 root root 5408 Dec 7 19:23 CONTRIBUTING.md
-rw-r--r-- 1 root root 57011 Dec 7 19:23 LICENSE
-rw-r--r-- 1 root root 1726 Dec 7 19:23 NOTICE
-rw-r--r-- 1 root root 3255 Dec 7 19:23 README.md
-rw-r--r-- 1 root root 7136 Dec 7 19:23 RELEASE-NOTES
-rw-r--r-- 1 root root 16262 Dec 7 19:23 RUNNING.txt
drwxr-xr-x 2 root root 4096 Jan 4 05:07 bin
drwxr-sr-x 1 root root 4096 Jan 16 14:47 conf
drwxr-sr-x 2 root staff 4096 Jan 4 05:07 include
drwxr-xr-x 2 root root 4096 Jan 4 05:06 lib
drwxrwxrwx 1 root root 4096 Jan 16 14:47 logs
drwxr-sr-x 3 root staff 4096 Jan 4 05:07 native-jni-lib
drwxrwxrwx 2 root root 4096 Jan 4 05:06 temp
drwxr-sr-x 7 root root 4096 Dec 7 19:21 webapps
drwxr-sr-x 2 root staff 4096 Jan 4 05:06 webapps2
drwxrwxrwx 2 root root 4096 Dec 7 19:19 work
root@003f010d0fb8:/usr/local/tomcat#
```

1 将原来的webapps.list修改成webapps之后，我们访问tomcat首页不会再报404，但是因为我们修改的只是容器，但是容器停止后下一次再启动tomcat访问首页时还是会报404，因为此时启动的是镜像的另一个容器，跟上次修改的不一样，所以，解决方案就是我们需要将修改好的容器映射成新的镜像，下次启动映射成的镜像即可。不会在访问首页而出现404的错误

[https://blog.csdn.net/qq\\_40891009](https://blog.csdn.net/qq_40891009)

```
drwxr-sr-x 3 root staff 4096 Jan 4 05:07 native-jni-lib
drwxrwxrwx 2 root root 4096 Jan 4 05:06 temp
drwxr-sr-x 7 root root 4096 Dec 7 19:21 webapps
drwxr-sr-x 2 root staff 4096 Jan 4 05:06 webapps2
drwxrwxrwx 2 root root 4096 Dec 7 19:19 work
root@003f010d0fb8:/usr/local/tomcat# read escape sequence
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker commit -a='谁修改了我的bug(既作者信息)' -m='将修改后的容器映射成新的镜像(既修改内容，随意写)' tomcat
Error response from daemon: No such container: idea/tomcat
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker commit -a='谁修改了我的bug(既作者信息)' -m='将修改后的容器映射成新的镜像(既修改内容，随意写)' tomcat
sha256:cb3db4d2fc7e234f423872e802d60e153cb90e1d84ffb45df0772b283c993c6
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
<none>              <none>             be2b5178c290       7 seconds ago     512MB
mycentos            latest             684bf6130f48       24 hours ago      254MB
mycentos            1.3               d7f448f7f417       2 days ago        316MB
test/centos         latest            6c91a4916c72       7 days ago        220MB
ticket              v1.2.004          c216e89bb4fe       11 days ago       761MB
<none>              <none>            8545916ecf99       11 days ago       761MB
tomcat              latest            ed94f55483b8       12 days ago       507MB
python              3.7-slim-buster   5b03cb8574a0       12 days ago       179MB
yinaoxiong/12306_code_server amd64             f3d4a84d93b1       3 weeks ago       378MB
centos              latest            0f3e07c0138f       3 months ago      220MB
hello-world         latest            fce289e99eb9       12 months ago     1.84kB
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker commit -a='谁修改了我的bug(既作者信息)' -m='将修改后的容器映射成新的镜像(既修改内容，随意写)' idea/tomcat tomcat
Error response from daemon: No such container: idea/tomcat
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker commit -a='谁修改了我的bug(既作者信息)' -m='将修改后的容器映射成新的镜像(既修改内容，随意写)' tomcat idea/tomcat
sha256:cb3db4d2fc7e234f423872e802d60e153cb90e1d84ffb45df0772b283c993c6
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
idea/tomcat         latest             cb3db4d2fc7e       11 seconds ago     512MB
<none>              <none>             be2b5178c290       About a minute ago 512MB
mycentos            latest             684bf6130f48       24 hours ago      254MB
mycentos            1.3               d7f448f7f417       2 days ago        316MB
test/centos         latest            6c91a4916c72       7 days ago        220MB
ticket              v1.2.004          c216e89bb4fe       11 days ago       761MB
<none>              <none>            8545916ecf99       11 days ago       761MB
tomcat              latest            ed94f55483b8       12 days ago       507MB
python              3.7-slim-buster   5b03cb8574a0       12 days ago       179MB
yinaoxiong/12306_code_server amd64             f3d4a84d93b1       3 weeks ago       378MB
centos              latest            0f3e07c0138f       3 months ago      220MB
hello-world         latest            fce289e99eb9       12 months ago     1.84kB
[root@iZj6chqfjhnsbsx9xfdsqZ /]#
```

1 按ctrl+p+q(不退出容器的方式)返回到宿主机)

2 使用docker commit命令将容器映射成新的镜像

3 下次直接启动新的镜像即可，访问首页不会出现404错误

[https://blog.csdn.net/qq\\_40891009](https://blog.csdn.net/qq_40891009)

```
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker stop tomcat
tomcat
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker run -d -p 8080:8080 --name='tomcat2' idea/tomcat
3fa02b37a2021760b568860d2c37c57c52b378c11cb4eb52d5aa7280a9e637da
[root@iZj6chqfjhnsbsx9xfdsqZ /]# docker ps
CONTAINER ID        IMAGE               COMMAND             NAMES
3fa02b37a202      idea/tomcat        "catalina.sh run"   tomcat2
b08a6a977ff9      mycentos:1.3       "/bin/sh -c /bin/bash" distracted_pi
ke                 yinaoxiong/12306_code_server:amd64 "unicorn app:app -c..." cantcha
1fbfbc76abe7      11 days ago        80/tcp
```

1 启动，然后访问即可

[https://blog.csdn.net/qq\\_40891009](https://blog.csdn.net/qq_40891009)

设置好之后，再启动就应该用更改好的镜像名称 如上图

## 二、运行容器：[https://blog.csdn.net/qq\\_40582459/article/details/103069964?utm\\_medium=distribute.pc\\_relevant.none-task-blog-searchFromBaidu-2.channel\\_param&depth\\_1-utm\\_source=distribute.pc\\_relevant.none-task-blog-searchFromBaidu-2.channel\\_param](https://blog.csdn.net/qq_40582459/article/details/103069964?utm_medium=distribute.pc_relevant.none-task-blog-searchFromBaidu-2.channel_param&depth_1-utm_source=distribute.pc_relevant.none-task-blog-searchFromBaidu-2.channel_param)

### 第一种方式

运行容器然后将项目复制到容器中

#### 1、启动容器：

运行命令docker run --name tomcat -d -p 8111:8080 tomcat

-p 主机端口(外网映射对应端口):tomcat端口 --name 容器名称 tomcat (镜像名称)

此时tomcat已在后台运行

#### 1. 进入容器

找到tomcat中webapps目录，进入命令： docker exec -it tomcat /bin/bash

```
[root@localhost loginweb]# docker exec -it tomcat /bin/bash
root@003f010d0fb8:/usr/local/tomcat# ls
BUILDING.txt  LICENSE  README.md  RUNNING.txt  conf  lib  native-jni-lib  webapps
CONTRIBUTING.md  NOTICE  RELEASE-NOTES  bin  include  logs  temp  work
```

### 3、将项目放入tomcat容器中

命令: docker cp AllLogin.war tomcat:/usr/local/tomcat/webapps

docker cp (项目名) 镜像名(或镜像id):/usr/local/tomcat/webapps(此目录路径为官方tomcat镜像目录)

### 第二种方式

将webapps挂载到主机目录, 今后升级更换项目只需要将项目包放入此路径:

命令: docker run -d -p 9001:8080 --name wzybtomcat -v /opt/apache-tomcat-9.0.14/webapps:/usr/local/tomcat/webapps newimagetomcat

wzybtomcat: 起得别名 /opt/apache-tomcat-9.0.14/webapps: war包存放路径 newimagetomcat: 容器名称 /usr/local/tomcat/webapps: 运行的tomcat容器

### 第三种方式

用Dockerfile创建:

1、创建Dockerfile:touch Dockerfile

2、编辑Dockerfile:vi Dockerfile

FROM tomcat:latest

COPY XXX.war webapps

#XXX.war(当前目录下你上传进来的项目包) (镜像中webapps的路径)

编辑完成按下esc, 然后输入冒号+wq, 然后按回车, 即:wq

3、构建项目镜像, 命令:docker build -t web .

-t 镜像名(自定义) 最后有**一点.**, 即当前目录

4、运行容器, 命令: docker run -d -p 8111:8080 --name web web

采用Dockerfile 发布镜像部署方式

1. 把项目的包放在和Dockerfile 同级目录下, 不不同级发布镜像会报错找不到, 如下:

首先进入到docker解压的文件夹下, 创建Dockerfile文件进行配置:

配好之后, 发布镜像 : docker build -t wx .

配置1和2, 我放在和Dockerfile同一级目录下, 正确的配置。3 配置的另一级目录, 是错误的, 导致发布镜像将失败, 如下图:

```
~
~
SHGS-WXTS:/opt/docker # docker build -t wx .
Sending build context to Docker daemon 264.6MB
Step 1/5 : from newtomcat:latest
----> 1dddac965eb1
Step 2/5 : MAINTAINER WL
----> Using cache
----> 89cec13d6a2a
Step 3/5 : COPY wzyb/wx.war /usr/local/tomcat/webapps
----> Using cache
----> 54a6c2010540
Step 4/5 : COPY ls/ls.war /usr/local/tomcat/webapps
----> Using cache
----> 5fe8cd54f7f4
Step 5/5 : COPY /opt/wzyb/wx.war /usr/local/tomcat/webapps
COPY failed: stat /var/lib/docker/tmp/docker-builder80711726/opt/wzyb/wx.war: no such file or directory
```

正确的配置:



```
10.1.20.102 (新的微信服务器 (吴, 朗, 上)) x 10.1.20.102 (新的微信服务器 (吴, 朗, 上)) (1)
#选择基础镜像来源 提前下载到本地的镜像
from newtomcat:latest
#作者
MAINTAINER WL
#拷贝war包tomcat目下
COPY wzyb/wx.war /usr/local/tomcat/webapps
COPY ls/ls.war /usr/local/tomcat/webapps
COPY sh/sh.war /usr/local/tomcat/webapps
~
~
~
~
~
~
~
~
~
~
```

自己打的一个镜像，直接用下载的tomcat，启动下面webapps是空的，无法访问，

发布成功如下图，成功之后，在通过docker images查看现有的镜像就会看到自己发布的镜像，如下图

```
10.1.20.102 (新的微信服务器 (吴, 朗, 上)) x 10.1.20.102 (新的微信服务器 (吴, 朗, 上)) (1)
SHGS-WXTS:/opt/docker # docker build -t wx
"docker build" requires exactly 1 argument.
see 'docker build --help'.

Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
SHGS-WXTS:/opt/docker # docker build -t wx .
Sending build context to Docker daemon 284.6MB
Step 1/5 : from newtomcat:latest
--> 1dddac965eb1
Step 2/5 : MAINTAINER WL
--> Using cache
--> 89cec13d6a2a
Step 3/5 : COPY wzyb/wx.war /usr/local/tomcat/webapps
--> Using cache
--> 54a6c2010540
Step 4/5 : COPY ls/ls.war /usr/local/tomcat/webapps
--> Using cache
--> 5fe8cd54f7f4
Step 5/5 : COPY sh/sh.war /usr/local/tomcat/webapps
--> 9cd5303f3d97
Successfully built 9cd5303f3d97
Successfully tagged wx:latest
SHGS-WXTS:/opt/docker #

Successfully built 9cd5303f3d97
Successfully tagged wx:latest
SHGS-WXTS:/opt/docker # docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
wx                   latest       9cd5303f3d97      About a minute ago  727MB
ls                   latest       5fe8cd54f7f4      About an hour ago  702MB
wzyb                 latest       821f42e14175      About an hour ago  678MB
newtomcat            latest       1dddac965eb1      4 hours ago       653MB
tomcat               latest       2703bbe9e9d4      4 days ago        648MB
hub-mirror.c.163.com/library/tomcat latest       2703bbe9e9d4      4 days ago        648MB
docker.elastic.co/logstash/logstash 7.1.1       b0cb1543380d      18 months ago     847MB
docker.elastic.co/kibana/kibana       7.1.1       67f17df6ca3e      18 months ago     746MB
docker.elastic.co/elasticsearch/elasticsearch 7.1.1       b0e9f9f047e6      18 months ago     894MB
SHGS-WXTS:/opt/docker #
```

之后启动直接启动这个镜像就OK。

如果第二次发布的镜像名称和之前的一样，发布完成之后，查询正在运行的容器，如下图：

```
SHGS-WXTS:/opt/docker # docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS             PORTS
9e33ead3a1fe        5fe8cd54f7f4       "catalina.sh run"   About an hour ago   Up About an hour   0.0.0.0:9002->8080/tcp
78f7f93e6bad        wzyb               "catalina.sh run"   About an hour ago   Up About an hour   0.0.0.0:9001->8080/tcp
78f7f93e6bad        wzyb               "catalina.sh run"   About an hour ago   Up About an hour   0.0.0.0:9001->8080/tcp
euristic_shaw       7a8f6a99e09        docker.elastic.co/kibana/kibana:7.1.1 "/usr/local/bin/kiba..." 13 months ago       Up 4 hours         0.0.0.0:5601->5601/tcp
7a8f6a99e09         kibana:7.1.1
79d853deaaab        docker.elastic.co/logstash/logstash:7.1.1 "/usr/local/bin/dock..." 13 months ago       Up 4 hours         0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp
79d853deaaab        logstash:7.1.1
8bde9daa3a6         docker.elastic.co/elasticsearch/elasticsearch:7.1.1 "/usr/local/bin/dock..." 13 months ago       Up 4 hours         0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
8bde9daa3a6         elasticsearch:7.1.1
SHGS-WXTS:/opt/docker # docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS             PORTS
9e33ead3a1fe        5fe8cd54f7f4       "catalina.sh run"   About an hour ago   Exited (143) 35 seconds ago
78f7f93e6bad        wzyb               "catalina.sh run"   About an hour ago   Up About an hour   0.0.0.0:9001->8080/tcp
78f7f93e6bad        wzyb               "catalina.sh run"   About an hour ago   Up About an hour   0.0.0.0:9001->8080/tcp
euristic_shaw       7a8f6a99e09        docker.elastic.co/kibana/kibana:7.1.1 "/usr/local/bin/kiba..." 13 months ago       Up 4 hours         0.0.0.0:5601->5601/tcp
7a8f6a99e09         kibana:7.1.1
79d853deaaab        docker.elastic.co/logstash/logstash:7.1.1 "/usr/local/bin/dock..." 13 months ago       Up 4 hours         0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp
79d853deaaab        logstash:7.1.1
8bde9daa3a6         docker.elastic.co/elasticsearch/elasticsearch:7.1.1 "/usr/local/bin/dock..." 13 months ago       Up 4 hours         0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
8bde9daa3a6         elasticsearch:7.1.1
SHGS-WXTS:/opt/docker #
```

如果发现服务部署的有问题，这是可以再次根据这个容器的id启动上个版本。，如下图：



无法访问此网站

启动访问正常

## Apache Tomcat/9.0.39



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

配置好之后，发布镜像 查看运行的容器：docker ps，发现大家用的是同一个镜像。

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
c24859e31864	wx	"catalina.sh run"	13 seconds ago	up 12 seconds	0.0.0.0:9002->8080/tcp
xenodochial_saha	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9001->8080/tcp
40104011d0a4	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9001->8080/tcp
festive_tereshkova	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9001->8080/tcp
7a8f6a99ee09	docker.elastic.co/kibana/kibana:7.1.1	"/usr/local/bin/kiba..."	13 months ago	up 5 hours	0.0.0.0:5601->5601/tcp
kibana7.1.1	docker.elastic.co/kibana/kibana:7.1.1	"/usr/local/bin/kiba..."	13 months ago	up 5 hours	0.0.0.0:5601->5601/tcp
79d853d6aabd	docker.elastic.co/logstash/logstash:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp
logstash7.1.1	docker.elastic.co/logstash/logstash:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp
8bede9daa3a6	docker.elastic.co/elasticsearch/elasticsearch:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
elasticsearch7.1.1	docker.elastic.co/elasticsearch/elasticsearch:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
SHGS-WXTS:/opt/docker # docker ps					
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
565054496f4	wx	"catalina.sh run"	7 seconds ago	up 6 seconds	0.0.0.0:9003->8080/tcp
CONFIDENTIAL	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9002->8080/tcp
c24859e31864	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9002->8080/tcp
xenodochial_saha	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9001->8080/tcp
40104011d0a4	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9001->8080/tcp
festive_tereshkova	wx	"catalina.sh run"	About a minute ago	up About a minute	0.0.0.0:9001->8080/tcp
7a8f6a99ee09	docker.elastic.co/kibana/kibana:7.1.1	"/usr/local/bin/kiba..."	13 months ago	up 5 hours	0.0.0.0:5601->5601/tcp
kibana7.1.1	docker.elastic.co/kibana/kibana:7.1.1	"/usr/local/bin/kiba..."	13 months ago	up 5 hours	0.0.0.0:5601->5601/tcp
79d853d6aabd	docker.elastic.co/logstash/logstash:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp
logstash7.1.1	docker.elastic.co/logstash/logstash:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp
8bede9daa3a6	docker.elastic.co/elasticsearch/elasticsearch:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
elasticsearch7.1.1	docker.elastic.co/elasticsearch/elasticsearch:7.1.1	"/usr/local/bin/dock..."	13 months ago	up 5 hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
SHGS-WXTS:/opt/docker #					

设置docker 容器的时间和系统时间一致： **注意： 编辑Dockerfile 设置时区的时候，不能放在COPY服务war包的前面，否则第一个服务发布不了，必须放在后面。**

- #设置时区

```
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime \
&& echo 'Asia/Shanghai' >/etc/timezone \
```

```
10.1.20.102 (新的微信服务器 (吴, 朗, 上)) | 10.1.20.102 (新的微信服务器 (吴, 朗, 上)) (1) | 10.1.20.102 (新的)
#选择基础镜像来源 提前下载到本地的镜像
from newtomcat:latest
#作者
#设置时区
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime \
&& echo 'Asia/Shanghai' >/etc/timezone \
COPY wzyb/wx.war /usr/local/tomcat/webapps
COPY ls/ls.war /usr/local/tomcat/webapps
COPY sh/sh.war /usr/local/tomcat/webapps
```

放在前面发布镜像如下图：

```
SHGS-WXTS:/opt/docker # docker build -t wx .
Sending build context to Docker daemon 264.6MB
Step 1/5 : FROM newtomcat:latest
--> 1ddda965eb1
Step 2/5 : MAINTAINER WL
--> Using cache
--> 89cec13d6a2a
Step 3/5 : RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime && echo 'Asia/Shanghai' >/etc/timezone COPY wzyb/wzyb.war /usr/local/tomcat/webapps
--> Running in 2078a0449ee3
Removing intermediate container 2078a0449ee3
--> a8d9680d5298
Step 4/5 : COPY ls/ls.war /usr/local/tomcat/webapps
--> 654deb34c116
Step 5/5 : COPY sh/sh.war /usr/local/tomcat/webapps
--> 770ac310d4db
Successfully built 770ac310d4db
Successfully tagged wx:latest
SHGS-WXTS:/opt/docker # clear
```

```

10.1.20.102 (新的微信服务器 (吴, 盼, 上) ) | 10.1.20.102 (新的微信服务器 (吴, 盼, 上) ) (1) | 10.1.20.102 |
#选择基础镜像来源 提前下载到本地的镜像
from newtomcat:latest
#作者
MAINTAINER WL
#拷贝war包tomcat目下
COPY wzyb/wzyb.war /usr/local/tomcat/webapps
COPY ls/ls.war /usr/local/tomcat/webapps
COPY sh/sh.war /usr/local/tomcat/webapps
#设置时区
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime \
    && echo 'Asia/Shanghai' >/etc/timezone
~
~
~
~
~
~
~
~
~
~

```

放在后面发布镜像如下图：

```

SHGS-WXTS:/opt/docker # docker build -t wx .
Sending build context to Docker daemon 264.6MB
Step 1/6 : from newtomcat:latest
--> 1dddc965eb1
Step 2/6 : MAINTAINER WL
--> using cache
--> 89cec13d6a2a
Step 3/6 : COPY wzyb/wzyb.war /usr/local/tomcat/webapps
--> using cache
--> 34be202e7269
Step 4/6 : COPY ls/ls.war /usr/local/tomcat/webapps
--> using cache
--> 1da58227608f
Step 5/6 : COPY sh/sh.war /usr/local/tomcat/webapps
--> using cache
--> e61bcdd07492
Step 6/6 : RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime && echo 'Asia/Shanghai' >/etc/timezone
--> Running in 2cd2135b7190
Removing intermediate container 2cd2135b7190
--> 9140fda38413
Successfully built 9140fda38413
Successfully tagged wx:latest
SHGS-WXTS:/opt/docker # vi Dockerfile
#选择基础镜像来源 提前下载到本地的镜像
from newtomcat:latest
#作者
MAINTAINER WL
#拷贝war包tomcat目下
COPY wzyb/wzyb.war /usr/local/tomcat/webapps
COPY ls/ls.war /usr/local/tomcat/webapps
COPY sh/sh.war /usr/local/tomcat/webapps
#设置时区

```

### docker容器中安装vi命令

在使用docker容器时，有时候里边没有安装vim，敲vim命令时提示说：vim: command not found，这个时候就需要安装vim，可是当你敲apt-get install vim命令时，提示：

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package vim

```

这时候需要敲：

```
apt-get update
```

，这个命令的作用是：同步 /etc/apt/sources.list 和 /etc/apt/sources.list.d 中列出的源的索引，这样才能获取到最新的软件包。

等更新完毕以后再敲命令：

```
apt-get install vim
```