

[Docker是什么](#)

[Docker架构原理](#)

[Docker作用](#)

[Docker容器和虚拟机区别](#)

[Docker安装 \(CentOS7.3\)](#)

[Docker配置阿里云镜像仓库](#)

[Docker基本命令](#)

[Docker镜像](#)

[Docker容器](#)

[示例](#)

[示例](#)

[进阶操作](#)

[推送镜像到Hub服务器](#)

[推送镜像到阿里云服务器](#)

(1). 登录阿里云Docker Registry

(2). 将镜像推送到Registry

(3). 从Registry中拉取镜像

[Docker容器目录挂载](#)

[简介](#)

[作用](#)

[实现](#)

[挂载目录只读 \(ro-readonly\)](#)

[Docker常用软件安装](#)

[安装tomcat9和配置](#)

[安装MySQL5.7和配置](#)

[备份镜像](#)

[恢复镜像](#)

[DockerFile简介](#)

[DockerFile常用指令](#)

[DockerFile构建自定义centos \(相当于helloworld级别\)](#)

[DockerFile构建自定义tomcat](#)

[DockerFile通过VOLUME实现容器卷](#)

[CMD、ENTRYPOINT的区别和联系](#)

Docker是什么

Docker是一个开源的应用容器引擎，基于Go语言并遵从Apache2.0的协议

Docker可以让开发者打包他们的应用以及依赖包到一个轻量级、可移植的容器中，然后发布到任何流行的Linux机器上，也可以实现虚拟化。

容器是完全使用沙箱机制，相互之间不会有任何接口，最重要的是容器性能开销极低。

Docker17.03之后分为CE (Community Edition) 版和EE(Enterprise Edition)版本，我们使用CE版本就足够用了

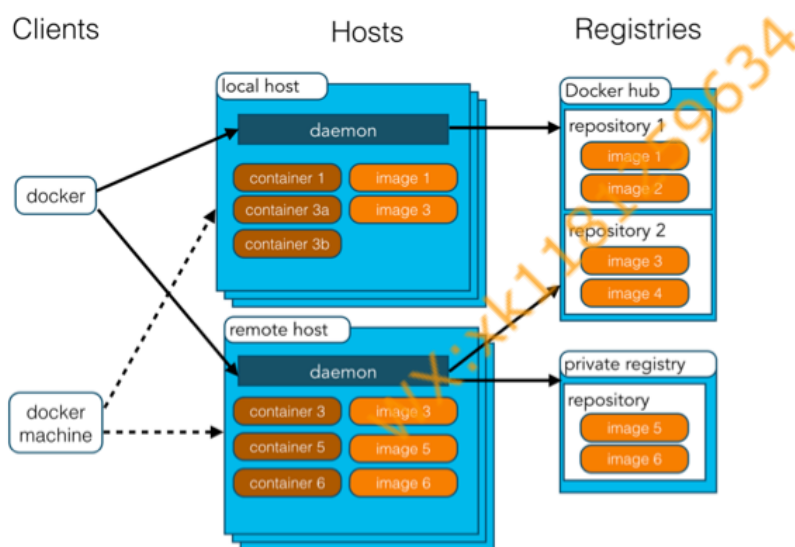
Docker架构原理

Docker 包括三个基本概念:

- **镜像 (Image)** : Docker 镜像 (Image) 就是一个只读的模板, 它可以是一个可运行软件 (tomcat、mysql) ,也可以是一个系统 (Ubuntu16.04、centos7) 。Docker 镜像用来创建 Docker 容器, 一个镜像可以创建多个容器, 每个容器都有各自的一个可读写层, 这些层相互独立共享下面的镜像。
- **容器 (Container)** : 镜像 (Image) 和容器 (Container) 的关系, 就像是面向对象程序设计中的类和实例一样, 镜像是静态的定义, 容器是镜像运行时的实体。容器可以被创建、启动、停止、删除、暂停等。可以把容器看成一个简易版的Linux环境 (包括root用户权限、进程空间、用户空间和网络空间等) 和运行在其中的应用程序。
- **仓库 (Repository)** : 仓库是集中存放镜像文件的场所, 类似于GitHub存放代码一样。每个镜像有不同的标签 (tag, 类似版本号) 。仓库分为公开仓库和私人仓库, 最大的公开仓库是Docker Hub(<https://hub.docker.com/>), 国内的公开仓库有阿里云、网易云等。

Docker 使用客户端-服务器 (C/S) 架构模式, 使用远程API来管理和创建Docker容器。

Docker	面向对象
镜像	类
容器	对象



Docker作用

1. 简化环境搭建, 提高开发生命周期效率
2. 大大简化运维工作量
3. 微服务利器

Docker容器和虚拟机区别

Docker是一种轻量级的虚拟化技术, 比传统的虚拟机性能更好

Docker的体系结构中, 不需要为虚拟机操作系统提供硬件模拟, 所有应用程序都作为Docker容器工作, 性能更好。

	Docker容器	虚拟机 (VMWare)
操作系统	与宿主机共享OS	宿主机上运行宿主OS
存储大小	镜像小，便于存储与传输	镜像庞大 (vmdk等)
运行性能	几乎无额外性能损失	操作系统额外的CPU、内存消耗
移植性	轻便、灵活、适用于Linux	笨重、与虚拟化技术耦合度高
硬件亲和性	面向软件开发者	面向硬件运维者

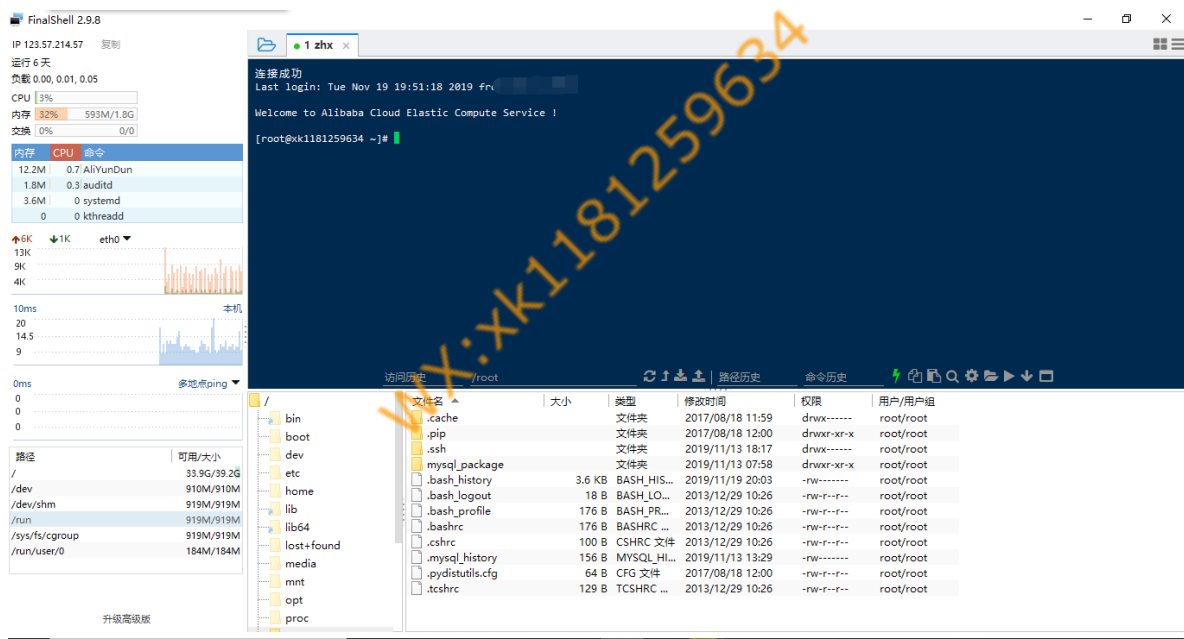
Docker优点：轻量级、速度快、运行应用隔离、方便维护

Docker安装 (CentOS7.3)

参考官方文档进行安装：<https://docs.docker.com/install/linux/docker-ce/centos/>

工具准备：

- 阿里云ECS-CentOS7.3
- ssh连接工具FinalShell



安装步骤：

以下操作以root身份操作

1.Docker要求CentOS系统的内核版本高于3.10，首先使用命令uname -r，来查看你当前的内核版本

```
# uname -r
```

```
连接成功
Last login: Tue Nov 19 19:51:18 2019

Welcome to Alibaba Cloud Elastic Compute Service !

[root@xk1181259634 ~]# uname -r
3.10.0-514.26.2.el7.x86_64
[root@xk1181259634 ~]#
```

2.使用root权限登录到CentOS中，确保yum包更新到最新

```
# yum update
```

```
Last login: Tue Nov 19 19:51:18 2019 from 223.71.70.8
Welcome to Alibaba Cloud Elastic Compute Service !

[root@xk1181259634 ~]# uname -r
3.10.0-514.26.2.el7.x86_64
[root@xk1181259634 ~]# yum update
Loaded plugins: fastestmirror
base                                     | 3.6 kB  00:00:00
epel                                    | 5.3 kB  00:00:00
extras                                 | 2.9 kB  00:00:00
mysql-connectors-community             | 2.5 kB  00:00:00
mysql-tools-community                 | 2.5 kB  00:00:00
mysql57-community                     | 2.5 kB  00:00:00
updates                               | 2.9 kB  00:00:00
(1/2): epel/x86_64/updateinfo         | 1.0 MB  00:00:00
(2/2): epel/x86_64/primary_db         | 6.9 MB  00:00:00
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package GeoIP.x86_64 0:1.5.0-11.el7 will be updated
```

```
selinux-policy.noarch 0:3.13.1-252.el7.1
setup.noarch 0:2.8.71-10.el7
shared-mime-info.x86_64 0:1.8-4.el7
sudo.x86_64 0:1.8.23-4.el7_7.1
systemd.x86_64 0:219-67.el7_7.2
systemd-sysv.x86_64 0:219-67.el7_7.2
tcpdump.x86_64 1:4.9.2-4.el7
trousers.x86_64 0:0.3.14-2.el7
tzdata.noarch 0:2019c-1.el7
vim-common.x86_64 2:7.4.629-6.el7
vim-filesystem.x86_64 2:7.4.629-6.el7
virt-what.x86_64 0:1.18-4.el7
wpa_supplicant.x86_64 1:2.6-12.el7
yum.noarch 0:3.4.3-163.el7.centos
zlib.x86_64 0:1.2.7-18.el7

selinux-policy-targeted.noarch 0:3.13.1-252.el7.1
shadow-utils.x86_64 2:4.6-5.el7
strace.x86_64 0:4.12-9.el7
sysstat.x86_64 0:10.1.5-18.el7
systemd-libs.x86_64 0:219-67.el7_7.2
tar.x86_64 2:1.26-35.el7
teamd.x86_64 0:1.27-9.el7
tuned.noarch 0:2.11.0-5.el7_7.1
util-linux.x86_64 0:2.23.2-61.el7
vim-enhanced.x86_64 2:7.4.629-6.el7
vim-minimal.x86_64 2:7.4.629-6.el7
wget.x86_64 0:1.14-18.el7_6.1
xfsprogs.x86_64 0:4.5.0-20.el7
yum-plugin-fastestmirror.noarch 0:1.1.31-52.el7

Replaced:
NetworkManager.x86_64 1:1.4.0-20.el7_3
pygobject3-base.x86_64 0:3.14.0-3.el7
grub2.x86_64 1:2.02-0.44.el7.centos
rdma.noarch 0:7.3_4.7_rc2-6.el7_3
grub2-tools.x86_64 1:2.02-0.44.el7.centos

Complete!
[root@xk1181259634 ~]#
```

3.卸载旧版本

```
# yum remove docker docker-common docker-selinux docker-engine
```

4.安装需要的软件包，yum-util提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

```
# yum install -y yum-utils device-mapper-persistent-data lvm2
```

```
yum.noarch 0:3.4.3-163.el7.centos                                yum-plugin-fastestmirror.noarch 0:1.1.31-52.el7
zlib.x86_64 0:1.2.7-18.el7

Replaced:
NetworkManager.x86_64 1:1.4.0-20.el7_3                        grub2.x86_64 1:2.02-0.44.el7.centos                grub2-tools.x86_64 1:2.02-0.44.el7.centos
pygobject3-base.x86_64 0:3.14.0-3.el7                        rdma.noarch 0:7.3.4.7_rc2-6.el7_3

Complete!
[root@xk1181259634 ~]# yum install -y yum-utils device-mapper-persistent-data lvm2
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
--> Package device-mapper-persistent-data.x86_64 0:0.8.5-1.el7 will be installed
--> Package lvm2.x86_64 7:2.02.185-2.el7_7.2 will be installed
--> Processing Dependency: lvm2-libs = 7:2.02.185-2.el7_7.2 for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: liblvm2app.so.2.2(Base)(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: libdevmapper-event.so.1.02(Base)(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: liblvm2app.so.2.2()(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Processing Dependency: libdevmapper-event.so.1.02()(64bit) for package: 7:lvm2-2.02.185-2.el7_7.2.x86_64
--> Package yum-utils.noarch 0:1.1.31-52.el7 will be installed
--> Processing Dependency: python-kitchen for package: yum-utils-1.1.31-52.el7.noarch

访问历史 /root
```

```
Installing : python-kitchen-1.1.1-5.el7.noarch                7/9
Installing : yum-utils-1.1.31-52.el7.noarch                  8/9
Installing : 7:lvm2-2.02.185-2.el7_7.2.x86_64                9/9
Verifying : python-chardet-2.2.1-3.el7.noarch                1/9
Verifying : yum-utils-1.1.31-52.el7.noarch                    2/9
Verifying : device-mapper-persistent-data-0.8.5-1.el7.x86_64  3/9
Verifying : 7:lvm2-2.02.185-2.el7_7.2.x86_64                4/9
Verifying : libxml2-python-2.9.1-6.el7_2.3.x86_64            5/9
Verifying : 7:lvm2-libs-2.02.185-2.el7_7.2.x86_64            6/9
Verifying : python-kitchen-1.1.1-5.el7.noarch                 7/9
Verifying : 7:device-mapper-event-1.02.158-2.el7_7.2.x86_64  8/9
Verifying : 7:device-mapper-event-libs-1.02.158-2.el7_7.2.x86_64 9/9

Installed:
device-mapper-persistent-data.x86_64 0:0.8.5-1.el7          lvm2.x86_64 7:2.02.185-2.el7_7.2          yum-utils.noarch 0:1.1.31-52.el7

Dependency Installed:
device-mapper-event.x86_64 7:1.02.158-2.el7_7.2          device-mapper-event-libs.x86_64 7:1.02.158-2.el7_7.2          libxml2-python.x86_64 0:2.9.1-6.el7_2.3
lvm2-libs.x86_64 7:2.02.185-2.el7_7.2                    python-chardet.noarch 0:2.2.1-3.el7          python-kitchen.noarch 0:1.1.1-5.el7

Complete!
[root@xk1181259634 ~]#

访问历史 /root
```

5.设置yum源

```
# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

```
Verifying : yum-utils-1.1.31-52.el7.noarch                2/9
Verifying : device-mapper-persistent-data-0.8.5-1.el7.x86_64  3/9
Verifying : 7:lvm2-2.02.185-2.el7_7.2.x86_64                4/9
Verifying : libxml2-python-2.9.1-6.el7_2.3.x86_64            5/9
Verifying : 7:lvm2-libs-2.02.185-2.el7_7.2.x86_64            6/9
Verifying : python-kitchen-1.1.1-5.el7.noarch                 7/9
Verifying : 7:device-mapper-event-1.02.158-2.el7_7.2.x86_64  8/9
Verifying : 7:device-mapper-event-libs-1.02.158-2.el7_7.2.x86_64 9/9

Installed:
device-mapper-persistent-data.x86_64 0:0.8.5-1.el7          lvm2.x86_64 7:2.02.185-2.el7_7.2          yum-utils.noarch 0:1.1.31-52.el7

Dependency Installed:
device-mapper-event.x86_64 7:1.02.158-2.el7_7.2          device-mapper-event-libs.x86_64 7:1.02.158-2.el7_7.2          libxml2-python.x86_64 0:2.9.1-6.el7_2.3
lvm2-libs.x86_64 7:2.02.185-2.el7_7.2                    python-chardet.noarch 0:2.2.1-3.el7          python-kitchen.noarch 0:1.1.1-5.el7

Complete!
[root@xk1181259634 ~]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum/repos.d/docker-ce.repo

访问历史 /root
```

6.安装最新版本的Docker

```
# yum install docker-ce docker-ce-cli containerd.io
```

```
Verifying : 7:lvmlib-2.02.185-2.el7_7.2.x86_64 6/9
Verifying : python-kitchen-1.1.1-5.el7.noarch 7/9
Verifying : 7:device-mapper-event-1.02.158-2.el7_7.2.x86_64 8/9
Verifying : 7:device-mapper-event-libs-1.02.158-2.el7_7.2.x86_64 9/9

Installed:
device-mapper-persistent-data.x86_64 0:0.8.5-1.el7          lvmlib.x86_64 7:2.02.185-2.el7_7.2          yum-utils.noarch 0:1.1.31-52.el7

Dependency Installed:
device-mapper-event.x86_64 7:1.02.158-2.el7_7.2          device-mapper-event-libs.x86_64 7:1.02.158-2.el7_7.2          libxml2-python.x86_64 0:2.9.1-6.el7_2.3
lvmlib.x86_64 7:2.02.185-2.el7_7.2          python-chardet.noarch 0:2.2.1-3.el7          python-kitchen.noarch 0:1.1.1-5.el7

Complete!
[root@xk1181259634 ~]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
[root@xk1181259634 ~]# yum install docker-ce docker-ce-cli containerd.io
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
```

```
Verifying : checkpolicy-2.5-8.el7.x86_64 1/11
Verifying : policycoreutils-python-2.5-33.el7.x86_64 2/11
Verifying : 3:docker-ce-19.03.5-3.el7.x86_64 3/11
Verifying : libsemanage-python-2.5-14.el7.x86_64 4/11
Verifying : 2:container-selinux-2.107-3.el7.noarch 5/11
Verifying : python-IPy-0.75-6.el7.noarch 6/11
Verifying : audit-libs-python-2.8.5-4.el7.x86_64 7/11
Verifying : containerd.io-1.2.10-3.2.el7.x86_64 8/11
Verifying : 1:docker-ce-cli-19.03.5-3.el7.x86_64 9/11
Verifying : setools-libs-3.3.8-4.el7.x86_64 10/11
Verifying : libcgrouper-0.41-21.el7.x86_64 11/11

Installed:
containerd.io.x86_64 0:1.2.10-3.2.el7          docker-ce.x86_64 3:19.03.5-3.el7          docker-ce-cli.x86_64 1:19.03.5-3.el7

Dependency Installed:
audit-libs-python.x86_64 0:2.8.5-4.el7          checkpolicy.x86_64 0:2.5-8.el7          container-selinux.noarch 2:2.107-3.el7
libcgrouper.x86_64 0:0.41-21.el7          libsemanage-python.x86_64 0:2.5-14.el7          policycoreutils-python.x86_64 0:2.5-33.el7
python-IPy.noarch 0:0.75-6.el7          setools-libs.x86_64 0:3.3.8-4.el7

Complete!
[root@xk1181259634 ~]#
```

7.启动Docker并设置开机启动

```
# systemctl start docker
# systemctl enable docker
```

```
连接成功
Last login: Tue Nov 19 20:03:28 2019 fr
Welcome to Alibaba Cloud Elastic Compute Service !

[root@xk1181259634 ~]# systemctl start docker
[root@xk1181259634 ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@xk1181259634 ~]#
```

8.验证Docker

```
# docker version
```

出现以下图示说明安装成功

```
[root@xk1181259634 ~]# docker version
Client: Docker Engine - Community
Version: 19.03.5
API version: 1.40
Go version: go1.12.12
Git commit: 633a0ea
Built: Wed Nov 13 07:25:41 2019
OS/Arch: linux/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version: 19.03.5
API version: 1.40 (minimum version 1.12)
Go version: go1.12.12
Git commit: 633a0ea
Built: Wed Nov 13 07:24:18 2019
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.2.10
GitCommit: b34a5c8af56e510852c35414db4c1f4fa6172339
```

9. Docker HelloWorld测试

```
# docker run hello-world
```

第一次从远程从远程仓库拉取

```
[root@xk1181259634 ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:c3b4ada4687b170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
```

第二次就从本地仓库直接读取

```
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

[root@xk1181259634 ~]# docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
```

Docker配置阿里云镜像仓库

1. 登录阿里云镜像服务中心，获取镜像地址

阿里云

搜索文档、控制台、API、解决方案和资源

费用 工单 备案 企业 支持与服务 官网 购物车 帮助 简体中文

容器镜像服务

默认实例

镜像仓库

命名空间

授权管理

代码源

访问凭证

企业版实例

镜像中心

镜像搜索

我的收藏

镜像加速器

使用加速器可以提升获取Docker官方镜像的速度

加速器地址

https://xxxxxxx.aliyuncs.com 复制

操作文档

Ubuntu CentOS Mac Windows

1. 安装 / 升级Docker客户端

推荐安装 1.10.0 以上版本的Docker客户端，参考文档 [docker-ce](#)

2. 配置镜像加速器

针对Docker客户端版本大于 1.10.0 的用户

您可以通过修改daemon配置文件/etc/docker/daemon.json 来使用加速器

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

2.在/etc/docker目录下找到daemon.json（没有就新建），将以下内容写入

```
{
  "registry-mirrors": ["https://xxxxxxx.mirror.aliyuncs.com"]
}
```

访问历史 /etc/docker

文件名	大小	类型	修改时间	权限	用户/用户组
daemon.json	66 B	JSON 文件	2019/11/20 08:09	-rw-r--r--	root/root
key.json	244 B	JSON 文件	2019/11/20 07:35	-rw-----	root/root

3.重启daemon

```
# systemctl daemon-reload
```

4.重启Docker服务

```
# systemctl restart docker
```

```
unknown shorthand flag: 'i' in -i
See 'docker images --help'.
[root@xk1181259634 ~]# docker images -h
Flag shorthand -h has been deprecated, please use --help

Usage: docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:
  -a, --all           Show all images (default hides intermediate images)
  --digests           Show digests
  -f, --filter filter  Filter output based on conditions provided
  --format string      Pretty-print images using a Go template
  --no-trunc          Don't truncate output
  -q, --quiet         Only show numeric IDs

[root@xk1181259634 ~]# docker images -a
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    fce289e99eb9   10 months ago  1.84kB

[root@xk1181259634 ~]# systemctl daemon-reload
[root@xk1181259634 ~]# systemctl restart docker
[root@xk1181259634 ~]#
```

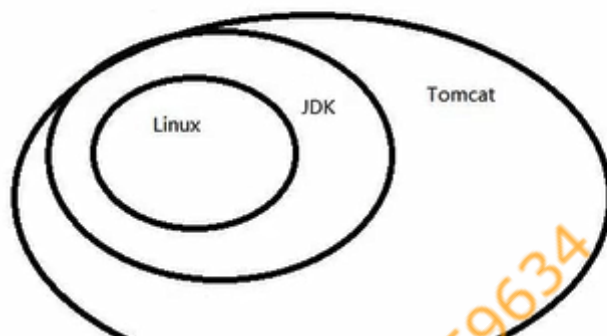
5.测试

```
# docker info
```


可以看到我们刚才配置的仓库信息

```
seccomp
  Profile: default
Kernel Version: 3.10.0-514.26.2.el7.x86_64
Operating System: CentOS Linux 7 (Core)
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 1.796GiB
Name: xk1181259634
ID: L7CY:VVF6:KQHY:ECJZ:F8GV:LTJP:AJO4:WB2H:FXT2:LFPB:BPZN:CQ53
Docker Root Dir: /var/lib/docker
Debug Mode: false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Registry Mirrors:
  https://p80ww07s.mirror.aliyuncs.com/
Live Restore enabled: false
[root@xk1181259634 ~]#
```

镜像的抽象层次结构



```
Debug Mode: false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Registry Mirrors:
  https://p80ww07s.mirror.aliyuncs.com/
Live Restore Enabled: false

[root@xk1181259634 ~]# docker pull tomcat:9.0.27-jdk8-adoptopenjdk-hotspot
9.0.27-jdk8-adoptopenjdk-hotspot: Pulling from library/tomcat
7ddb47eeb70: Downloading [=====] 17.65MB/26.69MB
c1bbdc448b72: Download complete
8c3b70e39044: Download complete
45d437916d57: Download complete
da4c04c54fa4: Waiting
2e619a1c4087: Waiting
9d52f3e37b59: Waiting
b31b11e9063: Waiting
379f59589cd7: Waiting
```

https://hub.docker.com/_/tomcat?tab=tags

afa3b125221e	linux/s390x	143.48 MB
jdk11-adoptopenjdk-hotspot Last updated 8 days ago by doijanky		
DIGEST	OS/ARCH	COMPRESSED SIZE
4944e621d94c	linux/amd64	237.58 MB
d98beca613f2	linux/arm/v7	219.41 MB
4b87f6fe2dd9	linux/ppc64le	226.89 MB
9.0.27-jdk8-adoptopenjdk-hotspot Last updated 8 days ago by doijanky		
DIGEST	OS/ARCH	COMPRESSED SIZE
459f037f438f	linux/amd64	149.79 MB
ea4816d8615	linux/ppc64le	151.29 MB
e6979ca0984c	linux/s390x	144.17 MB

Docker基本命令

名称	命令
启动docker	systemctl start docker
停止docker	systemctl stop docker
重启docker	systemctl restart docker
开机自启docker	systemctl enable docker
查看docker概要信息	docker info
查看docker版本信息	docker version
查看docker帮助文档	docker --help

Docker镜像

名称	命令
列出本机所有镜像	docker images
搜索镜像	docker search
下载镜像	docker pull
删除镜像	docker rmi

```
run      Run a command in a new container
save     Save one or more images to a tar archive (streamed to STDOUT by default)
search   Search the Docker Hub for images
start    Start one or more stopped containers
stats    Display a live stream of container(s) resource usage statistics
stop     Stop one or more running containers
tag      Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top      Display the running processes of a container
unpause  Unpause all processes within one or more containers
update   Update configuration of one or more containers
version  Show the Docker version information
wait     Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.
[root@xk1181259634 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
tomcat              7.0.96-jdk8-adoptopenjdk-openj9  2aebb055ce2c       11 days ago        345MB
tomcat              8.5.47-jdk8-adoptopenjdk-hotspot  50d274ef4fb3       11 days ago        324MB
tomcat              9.0.27-jdk8-adoptopenjdk-hotspot  f6152767e78c       11 days ago        325MB
mysql               5.7.28             cd3ed0dfff7e       4 weeks ago        437MB
hello-world         latest             fce289e99eb9       10 months ago      1.04kB
[root@xk1181259634 ~]#
```

名称	解释
REPOSITORY	镜像的仓库源
TAG	镜像的标签（版本） 同一个仓库有多个TAG的镜像，多个版本，我们用 REPOSITORY:TAG来定义不同的镜像
IMAGE ID	镜像ID， 镜像的唯一标识
CREATED	镜像创建时间
SIZE	镜像的大小

docker images的可选参数:

参数	解释
-a	显示所有镜像 (包括中间层)
-q	只显示镜像ID
-qa	可以组合
--digests	显示镜像的摘要信息
--no-trunc	显示完整的镜像信息

docker search和<https://hub.docker.com/>的搜索效果一样

可选参数:

参数	解释
--no-trunc	显示完整的镜像信息
--automated	只列出Docker Hub自动构建类型的镜像
-s	列出收藏数不小于指定值的镜像

docker pull 镜像名称:[TAG]

tips:不加TAG, 默认下载最新版本latest

1.删除单个镜像: docker rmi 镜像名称:[TAG]或者docker rmi 镜像ID (推荐)

如果不写TAG, 默认删除最新版本latest

有镜像创建的容器在运行的时候, 会删除失败并报错

```
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

[root@xk1181259634 ~]# docker rmi hello-world
Error response from daemon: conflict: unable to remove repository reference "hello-world" (must force) - container ea7ecc055ce6 is using its referenced
image fce289e955c6
[root@xk1181259634 ~]#
```

加上参数-f即可强制删除(加的-f参数相当于先把运行的容器停止再删除)

2.删除多个镜像: docker rmi 镜像名称1:[TAG] 镜像名称2:[TAG]

中间用空格隔开

3.删除所有镜像: docker rmi -f \$(docker images -qa)

Docker容器

名称	命令
创建并启动容器	<code>docker run [options] IMAGE [command] [ARG...]</code>
列出容器	<code>docker ps [options]</code>
退出容器	<code>exit</code> (停止退出)或 <code>Ctrl+P+Q</code> (不停止退出)
进入容器	<code>docker attach 容器ID或容器名</code>
启动容器	<code>docker start 容器ID或容器名</code>
重启容器	<code>docker restart 容器ID或容器名</code>
停止容器	<code>docker stop 容器ID或容器名</code>
删除容器	<code>docker rm -f 容器ID或容器名</code>

docker run参数详解：

参数	解释
<code>-d, (--detach=false)</code>	指定容器运行于前台还是后台，默认为false
<code>-i, (--interactive=false)</code>	打开STDIN，用于控制台交互
<code>-t, (--tty=false)</code>	分配tty设备，该可以支持终端登录，默认为false
<code>-p, (--publish)</code>	指定容器暴露的端口
<code>--name</code>	指定容器名字，后续可以通过名字进行容器管理，links特性需要使用名字
<code>-v, (--volume=[])</code>	给容器挂载存储卷，挂载到容器的某个目录

示例

- 运行一个在后台执行的容器，同时，还能用控制台管理：`docker run -i -t -d ubuntu:latest`
- 运行一个带命令在后台不断执行的容器，不直接展示容器内部信息：`docker run -d ubuntu:latest ping www.docker.com`
- 运行一个在后台不断执行的容器，同时带有命令，程序被终止后还能重启继续跑，还能用控制台管理，`docker run -d --restart=always ubuntu:latest ping www.docker.com`
- 为容器指定一个名字，`docker run -d --name=ubuntu_server ubuntu:latest`
- 容器暴露80端口，并指定宿主主机80端口与其通信(:之前是宿主主机端口，之后是容器需暴露的端口)，`docker run -d --name=ubuntu_server -p 80:80 ubuntu:latest`
- 指定容器内目录与宿主主机目录共享(:之前是宿主主机文件夹，之后是容器需共享的文件夹)，`docker run -d --name=ubuntu_server -v /etc/www:/var/www ubuntu:latest`

```
redis latest de25a81a5a0b 4 weeks ago 98.2MB
[root@xk1181259634 ~]# docker run -ti --name centos-xiaokang-01 5e35e350aded /bin/bash
[root@14aa4b1bf69c /]# ll
total 56
-rw-r--r-- 1 root root 12123 Oct 1 01:16 anaconda-post.log
lrwxrwxrwx 1 root root 7 Oct 1 01:15 bin -> usr/bin
drwxr-xr-x 5 root root 360 Nov 20 01:41 dev
drwxr-xr-x 1 root root 4096 Nov 20 01:41 etc
drwxr-xr-x 2 root root 4096 Apr 11 2018 home
lrwxrwxrwx 1 root root 7 Oct 1 01:15 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Oct 1 01:15 lib64 -> usr/lib64
drwxr-xr-x 2 root root 4096 Apr 11 2018 media
drwxr-xr-x 2 root root 4096 Apr 11 2018 mnt
drwxr-xr-x 2 root root 4096 Apr 11 2018 opt
dr-xr-xr-x 104 root root 0 Nov 20 01:41 proc
dr-xr-xr-x 2 root root 4096 Oct 1 01:16 root
drwxr-xr-x 11 root root 4096 Oct 1 01:16 run
lrwxrwxrwx 1 root root 8 Oct 1 01:15 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Apr 11 2018 srv
dr-xr-xr-x 13 root root 0 Apr 11 2018 sys
drwxrwxrwt 7 root root 4096 Oct 1 01:16 tmp
drwxr-xr-x 13 root root 4096 Oct 1 01:15 usr
```

docker ps参数详解：

参数	解释
-a	显示所有的容器，包括未运行的
-n	列出最近创建的几个容器
--no-trunc	显示容器完整信息
-q	只显示容器ID

示例

查看正在运行的容器：`docker ps`

查看所有容器：`docker ps -a`

显示最近创建的2个容器：`docker ps -n 2`

查看停止的容器：`docker ps -f status=exited`

进阶操作

启动普通容器：`docker run --name 别名 镜像ID`

启动交互式容器：`docker run -ti --name 别名 镜像ID`

守护式方式创建并启动容器：`docker run -di --name 别名 镜像ID`

名称	命令
宿主机和容器之间文件拷贝	docker copy
查看容器日志	docker logs 容器ID
查看容器进程	docker top 容器ID
进入容器执行命令	docker exec [options] 容器ID [command]

容器日志推荐大家直接去文件里面找，位置`/var/lib/docker/containers/`

```
drwxr-xr-x 13 root root 4096 Oct 1 01:15 usr
drwxr-xr-x 18 root root 4096 Oct 1 01:15 var
[root@xk1181259634 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
00736846c4d9       5e35e350aded       "/bin/bash"        5 minutes ago       Up 5 minutes                centos-Test
b8de8914c4a6       5e35e350aded       "/bin/bash"        7 minutes ago       Up 7 minutes                centos-TTY
[root@xk1181259634 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
00736846c4d9       5e35e350aded       "/bin/bash"        5 minutes ago       Up 5 minutes                centos-Test
b8de8914c4a6       5e35e350aded       "/bin/bash"        7 minutes ago       Up 7 minutes                centos-Test
1d6efd413da3       5e35e350aded       "/bin/bash"        10 minutes ago      Exited (137) 9 minutes ago   centos-AccountBack
aa77b1f6d8a7       5e35e350aded       "ping www.baidu.com" 26 minutes ago      Exited (0) 12 minutes ago    centos-Ping
[root@xk1181259634 ~]# docker logs 00736846c4d9
[root@xk1181259634 ~]# docker logs aa77b1f6d8a7
PING www.a.shifen.com (220.181.38.149) 56(84) bytes of data:
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=1 ttl=52 time=4.04 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=2 ttl=52 time=4.17 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=3 ttl=52 time=4.14 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=4 ttl=52 time=4.13 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=5 ttl=52 time=4.24 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=6 ttl=52 time=4.14 ms
64 bytes from 220.181.38.149 (220.181.38.149): icmp_seq=7 ttl=52 time=4.10 ms
```

文件名	大小	类型	修改时间	权限	用户/用户组
checkpoints		文件夹	2019/11/20 10:39	drwx-----	root/root
mounts		文件夹	2019/11/20 10:39	drwx-----	root/root
00736846c4d96d3a2a795bfc9f489f6fb547d	0	文本文件	2019/11/20 10:39	-rw-r-----	root/root
config.v2.json	2.6 KB	JSON 文件	2019/11/20 10:39	-rw-----	root/root
hostconfig.json	1.4 KB	JSON 文件	2019/11/20 10:39	-rw-r--r--	root/root
hostname	13 B	文件	2019/11/20 10:39	-rw-r--r--	root/root
hosts	174 B	文件	2019/11/20 10:39	-rw-r--r--	root/root
resolv.conf	149 B	CONF 文件	2019/11/20 10:39	-rw-r--r--	root/root
resolv.conf.hash	71 B	HASH 文件	2019/11/20 10:39	-rw-r--r--	root/root

docker inspect 容器ID

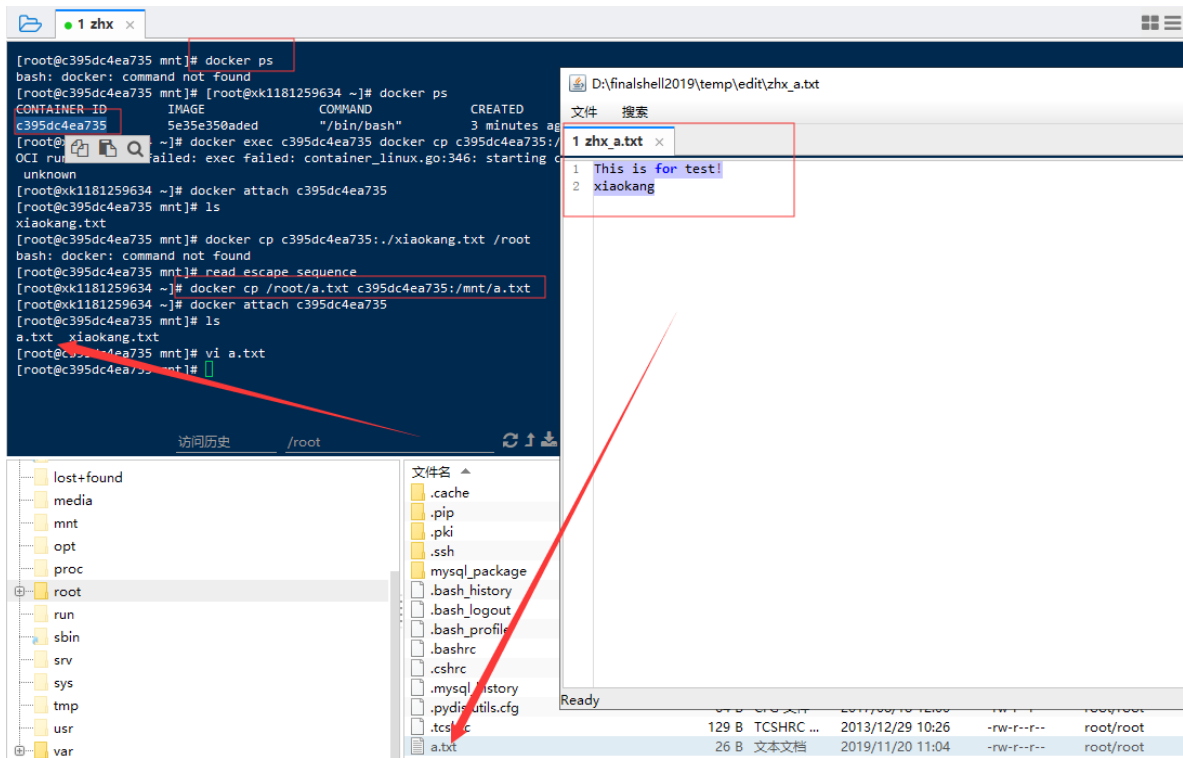
docker inspect 容器ID | grep IP //查看当前容器ip信息

```
[root@xk1181259634 ~]# docker inspect 00736846c4d9 | grep IP
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.18.0.3",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"IPAMConfig": null,
"IPAddress": "172.18.0.3",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
```

docker copy详解:

1.从宿主机拷贝到容器

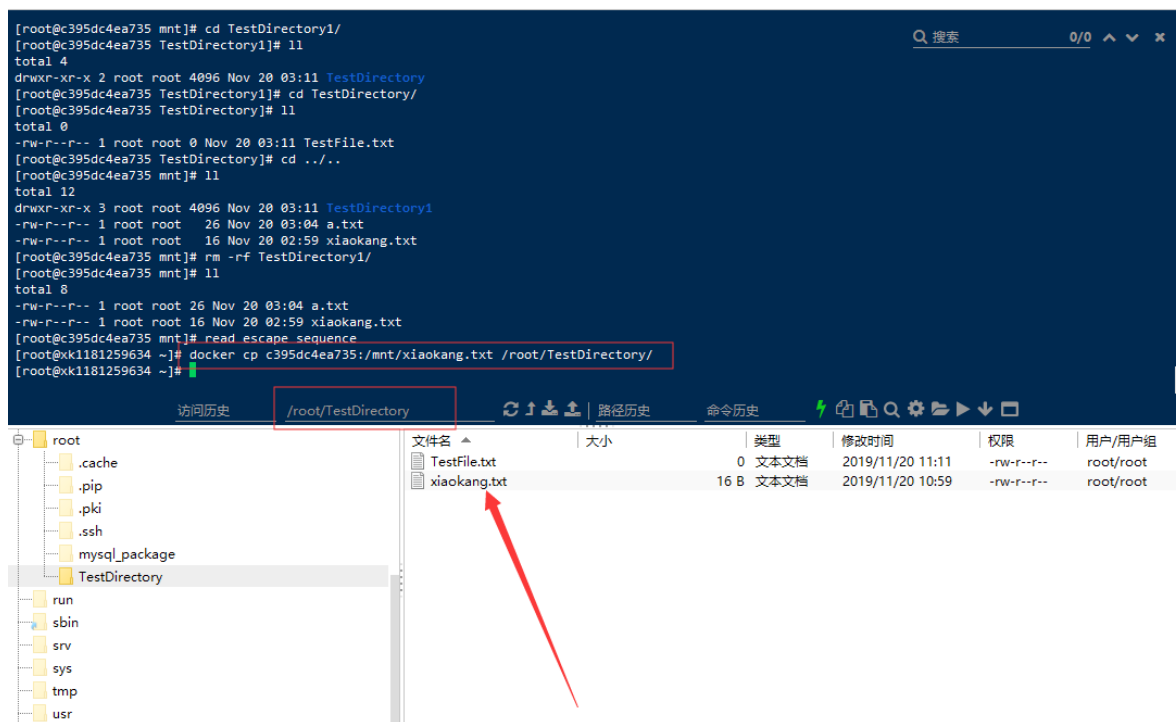
docker cp 宿主机目录 容器ID:容器目录



2. 从容器拷贝到宿主机（命令必须在宿主机内进行）

docker cp 容器ID:容器目录 宿主机目录

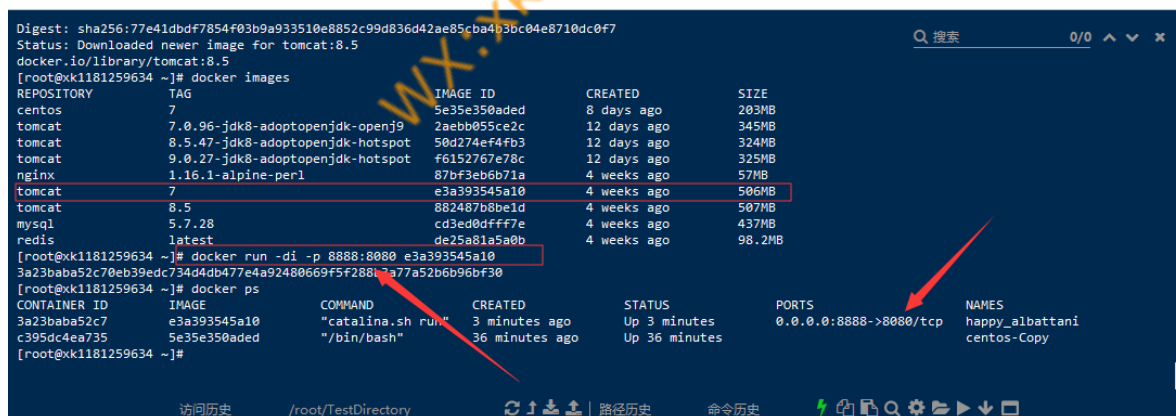




Docker容器与宿主机端口映射(需要镜像ID)

参数	命令
-p	随机端口映射，容器内部端口随机映射到主机的端口
-P	指定端口映射，格式为宿主机端口：容器端口

docker run -di --name 别名 宿主机端口号:容器端口号 镜像ID



提交运行时的Docker容器成为镜像（命令在宿主机内进行），命令 `docker commit [options]` 运行时的容器ID 新镜像名称

docker commit参数详解：

参数	解释
-a	作者
-m	备注信息


```
[root@xk1181259634 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
23824c2ee669       882487b8be1d      "catalina.sh run"   5 minutes ago       Up 5 minutes       0.0.0.0:32768->8080/tcp    tomcat8.5
3a23baba52c7       e3a393545a10      "catalina.sh run"   14 minutes ago      Up 14 minutes       0.0.0.0:8888->8080/tcp    happy_albattani
c395dc4ea735       5e35e350aded      "/bin/bash"         47 minutes ago      Up 47 minutes
[root@xk1181259634 ~]# docker exec -ti 23824c2ee669 /bin/bash
root@23824c2ee669:/usr/local/tomcat# cd webapps/
root@23824c2ee669:/usr/local/tomcat/webapps# cd ROOT/
root@23824c2ee669:/usr/local/tomcat/webapps/ROOT# ls
RELEASE-NOTES.txt  asf-logo-wide.svg  bg-middle.png  index.jsp  tomcat.css  tomcat.png
WEB-INF            bg-button.png      bg-nav.png     favicon.ico  tomcat-power.gif  tomcat.gif  tomcat.svg
root@23824c2ee669:/usr/local/tomcat/webapps/ROOT# rm -rf index.jsp
root@23824c2ee669:/usr/local/tomcat/webapps/ROOT#
```

想要commit, 必须要运行时的容器

这里我删掉了index.jsp

```
</div>
<div class="col120">
  <div class="container">
    <h4>Apache Software Foundation</h4>
    <ul>
      <li><a href="https://tomcat.apache.org/whoweare.html">Who We Are</a></li>
      <li><a href="https://tomcat.apache.org/heritage.html">Heritage</a></li>
      <li><a href="https://www.apache.org">Apache Home</a></li>
      <li><a href="https://tomcat.apache.org/resources.html">Resources</a></li>
    </ul>
  </div>
</div>
<br class="separator" />
</div>
<p class="copyright">Copyright ©1999-2019 Apache Software Foundation. All Rights Reserved</p>
</div>
</body>
</html>
[root@xk1181259634 ~]# docker commit -a 'xiaokang.188@qq.com' -m 'tomcat8.5 without index.jsp' 23824c2ee669 xiaokang/tomcat8.5:1.0
sha256:08aa9903b3314558807168f784a9d0a8670de49bd05b0c1c2fb2a7a12f8f0de8
[root@xk1181259634 ~]#
```

敲完命令后即可查看刚才提交的镜像了

```
<br class="separator" />
</div>
<p class="copyright">Copyright ©1999-2019 Apache Software Foundation. All Rights Reserved</p>
</div>
</body>
</html>
[root@xk1181259634 ~]# docker commit -a 'xiaokang.188@qq.com' -m 'tomcat8.5 without index.jsp' 23824c2ee669 xiaokang/tomcat8.5:1.0
sha256:08aa9903b3314558807168f784a9d0a8670de49bd05b0c1c2fb2a7a12f8f0de8
[root@xk1181259634 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
xiaokang/tomcat8.5  1.0                08aa9903b331      About a minute ago  507MB
centos               7                  e3a3e350aded       8 days ago         203MB
tomcat               7.0.96-jdk8-adop   2aebb055ce2c       12 days ago        345MB
tomcat               8.5.47-jdk8-adop   50d274ef4fb3       12 days ago        324MB
tomcat               9.0.27-jdk8-adop   f6152767e78c       12 days ago        325MB
nginx                1.16.1-alpine-perl 87bf3eb6b71a       4 weeks ago        57MB
tomcat               7                  e3a393545a10       4 weeks ago        506MB
tomcat               8.5                882487b8be1d       4 weeks ago        507MB
mysql                5.7.28             cd3ed0dfff7e       4 weeks ago        437MB
redis                latest             de25a81a5a0b       4 weeks ago        98.2MB
[root@xk1181259634 ~]#
```

推送镜像到Hub服务器

1. <https://hub.docker.com/>注册, 得到自己的账号和密码

2. 使用docker登录到Hub服务器

```
[root@xk1181259634 ~]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: xiaokangxxs
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
[root@xk1181259634 ~]#
```

3.docker push推送

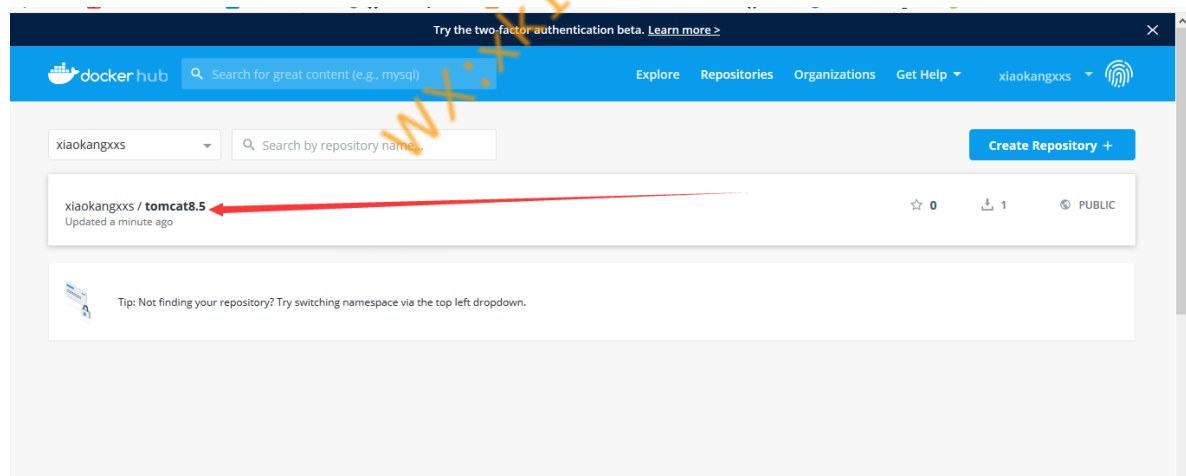
docker push 名称:TAG

这里小编踩了个坑，在push的时候镜像需要指定格式的名称

登录账号前缀 /具体镜像名，如xiaokangxxs/tomcat8.5

```
[root@xk1181259634 ~]# docker push xiaokangxxs/tomcat8.5:1.0
The push refers to repository [docker.io/xiaokangxxs/tomcat8.5]
b508840cf9c7: Pushing [=====] 106.5kB
65e5e74a1404: Pushing [=====] 2.048kB
38d8d468142f: Pushing [=====] 603.1kB/18.58MB
08579474bb30: Pushing [=====] 2.56kB
a8902d6047fe: Pushing [=====] 1.058MB/205MB
99557920a7c5: Waiting
7e3c900343d0: Waiting
b8f8aef56a8: Waiting
687890749166: Waiting
2f77733e9824: Waiting
97041f29baff: Waiting
```

```
[root@xk1181259634 ~]# docker push xiaokangxxs/tomcat8.5:1.0
The push refers to repository [docker.io/xiaokangxxs/tomcat8.5]
b508840cf9c7: Pushed
65e5e74a1404: Pushed
38d8d468142f: Pushed
08579474bb30: Pushed
a8902d6047fe: Pushed
99557920a7c5: Pushed
7e3c900343d0: Pushed
b8f8aef56a8: Pushed
687890749166: Pushed
2f77733e9824: Pushed
97041f29baff: Pushed
1.0: digest: sha256:f689158420252113038b533bc9485cd88ed0dc67f753f7a6ce927cabaffa1946 size: 2631
[root@xk1181259634 ~]#
```



推送镜像到阿里云服务器

1.进入：<https://cr.console.aliyun.com/>阿里云镜像控制台，需要注册，使用支付宝或淘宝登录，镜像控制台密码需要单独设置

2.进入控制台，先创建命名空间，再创建镜像

(1). 登录阿里云Docker Registry

```
$ sudo docker login --username=xiaokang.188@qq.com registry.cn-beijing.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。

```
97041f29baff: Pushed
1.0: digest: sha256:f68915842025113038b533bc9485cd88ed0dc67f7537a6ce927cabaffa1946 size: 2631
[root@xk1181259634 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
xiaokangxxs/tomcat  8.5                08aa9903b331       4 hours ago        507MB
centos               7                  5e35e350aded       8 days ago         203MB
tomcat               7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c       12 days ago        345MB
tomcat               8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3       12 days ago        324MB
tomcat               9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c       12 days ago        325MB
nginx                1.16.1-alpine-perl 87bf3eb6b71a       4 weeks ago        57MB
tomcat               7                  e3a393545a10       4 weeks ago        506MB
tomcat               8.5                882487b8be1d       4 weeks ago        507MB
mysql                5.7.28             cd3ed0dfff7e       4 weeks ago        437MB
redis                latest             de25a81a5a0b       4 weeks ago        98.2MB
[root@xk1181259634 ~]# docker login --username=xiaokang.188@qq.com registry.cn-beijing.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
[root@xk1181259634 ~]#
```

(2). 将镜像推送到Registry

```
$ docker tag [ImageId] registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:[镜像版本号]
$ docker push registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:[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。

```
Login Succeeded
[root@xk1181259634 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
xiaokangxxs/tomcat  8.5                08aa9903b331       4 hours ago        507MB
centos               7                  5e35e350aded       8 days ago         203MB
tomcat               7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c       12 days ago        345MB
tomcat               8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3       12 days ago        324MB
tomcat               9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c       12 days ago        325MB
nginx                1.16.1-alpine-perl 87bf3eb6b71a       4 weeks ago        57MB
tomcat               7                  e3a393545a10       4 weeks ago        506MB
tomcat               8.5                882487b8be1d       4 weeks ago        507MB
mysql                5.7.28             cd3ed0dfff7e       4 weeks ago        437MB
redis                latest             de25a81a5a0b       4 weeks ago        98.2MB
[root@xk1181259634 ~]# docker tag 08aa9903b331 registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
[root@xk1181259634 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
xiaokangxxs/tomcat  8.5                08aa9903b331       4 hours ago        507MB
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat  8.5                08aa9903b331       4 hours ago        507MB
centos               7                  5e35e350aded       8 days ago         203MB
tomcat               7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c       12 days ago        345MB
tomcat               8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3       12 days ago        324MB
[root@xk1181259634 ~]# docker push registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
The push refers to repository [registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat]
b508840cf9c7: Pushed
65e5e74a1404: Pushed
38d8d468142f: Pushing [=====>]          ] 9.167MB/18.58MB
08579474bb30: Pushed
a8902d6047fe: Pushing [====>]             ] 15.77MB/205MB
99557920a7c5: Pushed
7e3c900343d0: Pushed
b0f8aef56a8: Pushing [====>]             ] 13.1MB/141.8MB
607890749166: Pushed
2f77733e9824: Pushing [=====>]          ] 8.289MB/23.25MB
97041f29baff: Pushing [=====>]          ] 7.583MB/100.7MB
```

```
tomcat      8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3 12 days ago 324MB
tomcat      9.0.27-jdk8-adoptopenjdk-hotspot    f6152767e78c 12 days ago 325MB
nginx       1.16.1-alpine-perl                     87bf3eb6b71a 4 weeks ago 57MB
tomcat      7                                         e3a393545a10 4 weeks ago 506MB
tomcat      8.5                                     882487b8be1d 4 weeks ago 507MB
mysql       5.7.28                                 cd3ed0dfff7e 4 weeks ago 437MB
redis       latest                                 de25a81a5a0b 4 weeks ago 98.2MB

[root@xk1181259634 ~]# docker push registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
The push refers to repository [registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat]
b508840cf9c7: Pushed
65e5e74a1404: Pushed
38d8d468142f: Pushed
08579474bb30: Pushed
a8902d6047fe: Pushed
99557920a7c5: Pushed
7e3c900343d0: Pushed
b8f8aef56a8: Pushed
687890749166: Pushed
2f77733e9824: Pushed
97041f29baff: Pushed
8.5: digest: sha256:f689158420252113038b533bc9485cd88ed0dc67f753f7a6ce927cabaffa1946 size: 2631
[root@xk1181259634 ~]#
```

tomcat
华北2（北京） | 私有 | 本地仓库 | 正常

部署应用

镜像版本

刷新

版本	镜像ID	状态	Digest	镜像大小	最后更新时间	操作
8.5	08aa9903b331...	正常	f689158420252113038b533bc9485cd88ed0dc67f753f7a6ce927cabaffa1946	220.197 MB	2019-11-20 16:08:42	安全扫描 层信息 同步 删除

(3). 从Registry中拉取镜像

```
$ docker pull registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:[镜像版本号]
```

先删除刚才上传的本地镜像

```
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat 8.5 08aa9903b331 4 hours ago 507MB
centos 7 5e35e350aded 8 days ago 203MB
tomcat 7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c 12 days ago 345MB
tomcat 8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3 12 days ago 324MB
tomcat 9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c 12 days ago 325MB
nginx 1.16.1-alpine-perl 87bf3eb6b71a 4 weeks ago 57MB
tomcat 7 e3a393545a10 4 weeks ago 506MB
tomcat 8.5 882487b8be1d 4 weeks ago 507MB
mysql 5.7.28 cd3ed0dfff7e 4 weeks ago 437MB
redis latest de25a81a5a0b 4 weeks ago 98.2MB

[root@xk1181259634 ~]# docker rmi -f registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat
Error: No such image: registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat
[root@xk1181259634 ~]# docker rmi -f registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
Untagged: registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
Untagged: registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat@sha256:f689158420252113038b533bc9485cd88ed0dc67f753f7a6ce927cabaffa1946
[root@xk1181259634 ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
xiaokangxxs/tomcat:8.5 1.0 08aa9903b331 4 hours ago 507MB
centos 7 5e35e350aded 8 days ago 203MB
tomcat 7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c 12 days ago 345MB
tomcat 8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3 12 days ago 324MB
tomcat 9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c 12 days ago 325MB
```

再从阿里云私人仓库中拉下来

```
tomcat 8.5 882487b8be1d 4 weeks ago 507MB
mysql 5.7.28 cd3ed0dfff7e 4 weeks ago 437MB
redis latest de25a81a5a0b 4 weeks ago 98.2MB

[root@xk1181259634 ~]# docker pull registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
8.5: Pulling from xiaokangxxs/tomcat
Digest: sha256:f689158420252113038b533bc9485cd88ed0dc67f753f7a6ce927cabaffa1946
Status: Downloaded newer image for registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat:8.5
[root@xk1181259634 ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
xiaokangxxs/tomcat:8.5 1.0 08aa9903b331 4 hours ago 507MB
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat 8.5 08aa9903b331 4 hours ago 507MB
centos 7 5e35e350aded 8 days ago 203MB
tomcat 7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c 12 days ago 345MB
tomcat 8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3 12 days ago 324MB
tomcat 9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c 12 days ago 325MB
nginx 1.16.1-alpine-perl 87bf3eb6b71a 4 weeks ago 57MB
tomcat 7 e3a393545a10 4 weeks ago 506MB
tomcat 8.5 882487b8be1d 4 weeks ago 507MB
mysql 5.7.28 cd3ed0dfff7e 4 weeks ago 437MB
redis latest de25a81a5a0b 4 weeks ago 98.2MB
[root@xk1181259634 ~]#
```

Docker容器目录挂载

简介

我们可以在创建容器的时候，将主机的目录与容器内的目录进行映射，这样我们就可以实现宿主机和容器目录的**双向数据自动同步**。

作用

前面我们可以通过cp命令来实现宿主机和容器之间的数据传递，这种方式比较麻烦

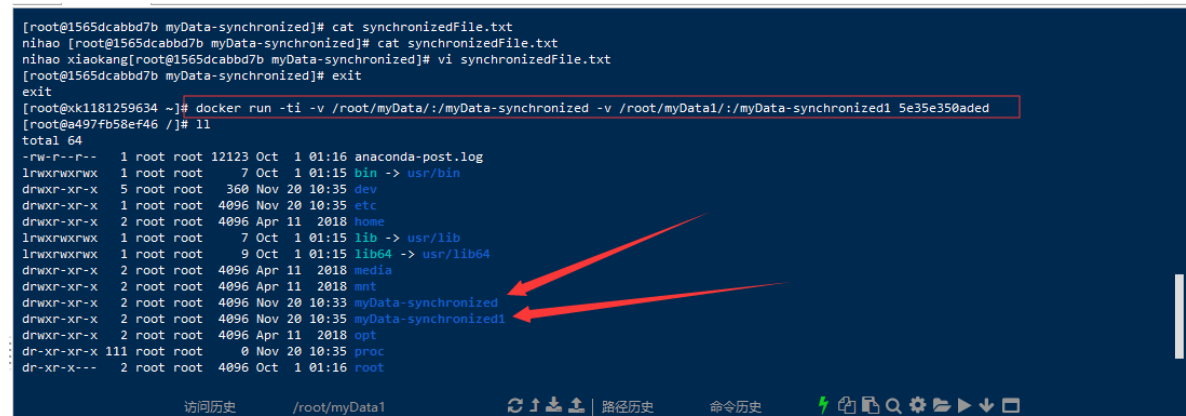
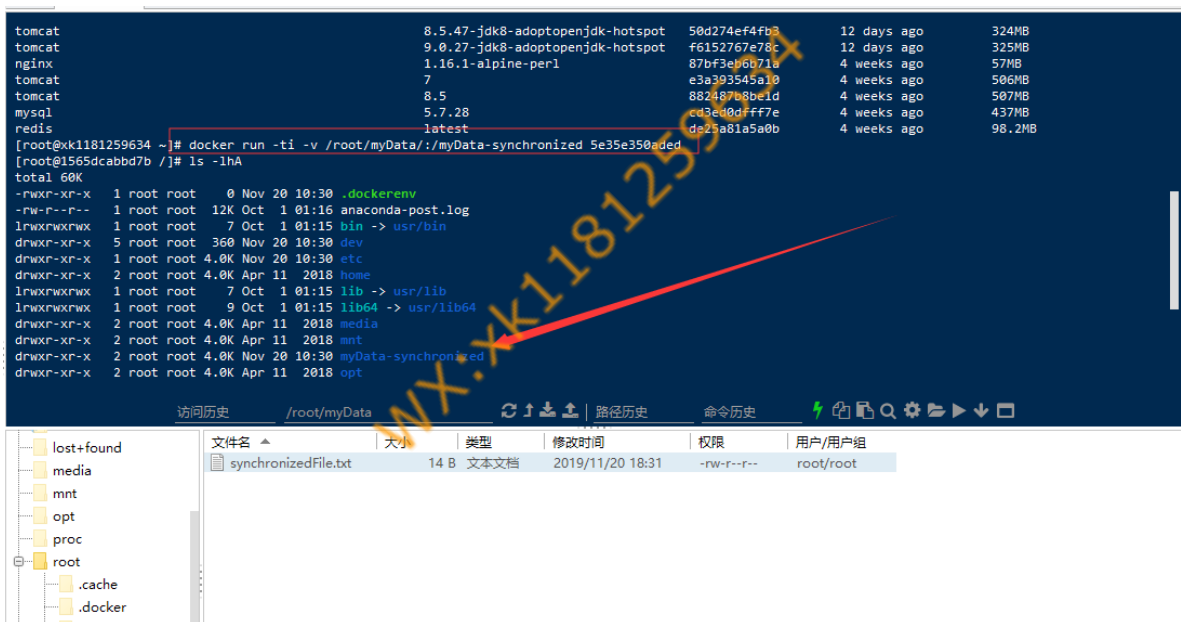
我们可以通过容器目录挂载，能够轻松实现代码上传、配置修改、日志同步等需求。

实现

语法: `docker run -ti -v /宿主机目录:/容器目录 镜像ID`

多目录挂载: `docker run -ti -v /宿主机目录:/容器目录 -v /宿主机目录1:/容器目录1 镜像ID`

如果同步的是多级目录，可能会出现权限不足的提示，这是因为CentOS7中的安全模块selinux把权限禁掉了，我们需要添加--privileged=true来解决挂载的目录没有权限的问题。



挂载目录只读 (ro-readonly)

`docker run -ti -v /宿主机目录:/容器目录:ro 镜像ID`


```
[root@xk1181259634 bin]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
xiaokangxxs/tomcat8.5      1.0                08aa9903b331       7 hours ago        507MB
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat      8.5                08aa9903b331       7 hours ago        507MB
centos                 7                  5e35e350aded       8 days ago         203MB
tomcat                  7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c       12 days ago        345MB
tomcat                  8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3       12 days ago        324MB
tomcat                  9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c       12 days ago        325MB
nginx                   1.16.1-alpine-perl 87bf3eb6b71a       4 weeks ago        57MB
tomcat                  7                  e3a393545a10       4 weeks ago        506MB
tomcat                  8.5                882487b8be1d       4 weeks ago        507MB
mysql                   5.7.28             cd3ed0dfff7e       4 weeks ago        437MB
redis                   latest             de25a81a5a0b       4 weeks ago        98.2MB

[root@xk1181259634 bin]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[root@xk1181259634 bin]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[root@xk1181259634 bin]# docker run -d -p 8080:8080 f6152767e78c
d4e31c8da7b4df2fdee173f7cb1eb810c2e13bec20ab3eb5882ae8071d8f677e
[root@xk1181259634 bin]# cd /home
[root@xk1181259634 home]#
```

2.在宿主机的/home目录下新建tomcat目录，复制容器里面conf、webapps到宿主机
进入容器并执行命令

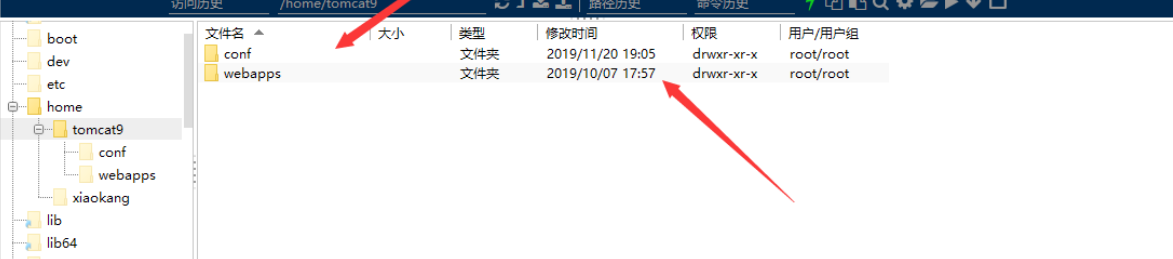
```
# docker exec -ti d4e31c8da7b4 /bin/bash
```

```
[root@xk1181259634 ~]# docker exec -ti d4e31c8da7b4 /bin/bash
root@d4e31c8da7b4:/usr/local/tomcat# ll
total 160
drwxr-xr-x 1 root root 4096 Nov  8 02:53 ./
drwxr-xr-x 1 root root 4096 Nov  8 02:52 ../
drwxr-xr-x 2 root root 4096 Nov  8 02:53 bin/
-rw-r--r-- 1 root root 18982 Oct  7 09:59 BUILDING.txt
drwxr-xr-x 1 root root 4096 Nov 20 11:05 conf/
-rw-r--r-- 1 root root 5408 Oct  7 09:59 CONTRIBUTING.md
drwxr-xr-x 2 root root 4096 Nov  8 02:53 include/
drwxr-xr-x 2 root root 4096 Nov  8 02:52 lib/
-rw-r--r-- 1 root root 57092 Oct  7 09:59 LICENSE
drwxrwxrwx 1 root root 4096 Nov 20 11:05 logs/
drwxr-xr-x 3 root root 4096 Nov  8 02:53 native-jni-lib/
-rw-r--r-- 1 root root 2333 Oct  7 09:59 NOTICE
-rw-r--r-- 1 root root 3255 Oct  7 09:59 README.md
-rw-r--r-- 1 root root 6849 Oct  7 09:59 RELEASE-NOTES
-rw-r--r-- 1 root root 16262 Oct  7 09:59 RUNNING.txt
drwxrwxrwx 2 root root 4096 Nov  8 02:52 temp/
drwxr-xr-x 7 root root 4096 Oct  7 09:57 webapps/
drwxrwxrwx 1 root root 4096 Nov 20 11:05 work/
root@d4e31c8da7b4:/usr/local/tomcat#
```

```
# docker cp d4e31c8da7b4:/usr/local/tomcat/conf /home/tomcat9/
```

```
# docker cp d4e31c8da7b4:/usr/local/tomcat/webapps /home/tomcat9/
```

```
[root@xk1181259634 home]# docker cp d4e31c8da7b4:/usr/local/tomcat/conf /home/tomcat9/
[root@xk1181259634 home]# docker cp d4e31c8da7b4:/usr/local/tomcat/webapps /home/tomcat9/
[root@xk1181259634 home]#
```



3.把容器里面的tomcat里的webapps、logs、conf挂载到宿主机tomcat目录下，方便代码上传、同步持久化日志以及方便配置tomcat


```
# docker run -d --name tomcat9 -p 8080:8080 -v
/home/tomcat9/conf:/usr/local/tomcat/conf:ro -v
/home/tomcat9/webapps:/usr/local/tomcat/webapps:ro -v
/home/tomcat9/logs:/usr/local/tomcat/logs:ro f6152767e78c
```

```
[root@xk1181259634 home]# docker run -d --name tomcat9 -p 8080:8080 -v /home/tomcat9/conf:/usr/local/tomcat/conf:ro -v /home/tomcat9/webapps:/usr/local/tomcat/webapps:ro -v /home/tomcat9/logs:/usr/local/tomcat/logs:ro f6152767e78c
[root@xk1181259634 home]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
1f43bc6a11a7       f6152767e78c      "catalina.sh run"   44 seconds ago     Created            0.0.0.0:8080->8080/tcp   tomcat9
d4e31c8da7b4       f6152767e78c      "catalina.sh run"   14 minutes ago     Up 14 minutes      0.0.0.0:8080->8080/tcp   awesome_aryabhata
[root@xk1181259634 home]# docker rm -f d4e31c8da7b4
d4e31c8da7b4
[root@xk1181259634 home]# docker run -d --name tomcat9 -p 8080:8080 -v /home/tomcat9/conf:/usr/local/tomcat/conf:ro -v /home/tomcat9/webapps:/usr/local/tomcat/webapps:ro -v /home/tomcat9/logs:/usr/local/tomcat/logs:ro f6152767e78c
docker: Error response from daemon: Conflict. The container name "/tomcat9" is already in use by container "1f43bc6a11a77bf6b2c163c1f999dca3f1604ab3dcb9ef67b0c9d4ae42c52d8b". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@xk1181259634 home]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
1f43bc6a11a7       f6152767e78c      "catalina.sh run"   About a minute ago   Created            0.0.0.0:8080->8080/tcp   tomcat9
1f43bc6a11a7
[root@xk1181259634 home]# docker rm -f 1f43bc6a11a7
1f43bc6a11a7
[root@xk1181259634 home]# docker run -d --name tomcat9 -p 8080:8080 -v /home/tomcat9/conf:/usr/local/tomcat/conf:ro -v /home/tomcat9/webapps:/usr/local/tomcat/webapps:ro -v /home/tomcat9/logs:/usr/local/tomcat/logs:ro f6152767e78c
34b97835dad18de843a81653961a3c6960891d85e7160015e23554bf72dda00e
[root@xk1181259634 home]#
```

4.配置tomcat中server.xml以及同步上传war包

server.xml中添加映射

```
<Context path="/" docBase="/usr/local/tomcat/webapps/FaceDemo" debug="0"
reloadable="true" />
```

```
136
137
138 <!-- Use the LockOutRealm to prevent attempts to guess user passwords
139 via a brute-force attack -->
140 <Realm className="org.apache.catalina.realm.LockOutRealm">
141 <!-- This Realm uses the UserDatabase configured in the global JNDI
142 resources under the key "UserDatabase". Any edits
143 that are performed against this UserDatabase are immediately
144 available for use by the Realm. -->
145 <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
146 resource="UserDatabase"/>
147 </Realm>
148 <Host name="localhost" appBase="webapps"
149 unpackWARs="true" autoDeploy="true">
150 <Context path="/" docBase="/usr/local/tomcat/webapps/FaceDemo" debug="0" reloadable="true" />
151 <!-- SingleSignOn valve, share authentication between web applications
152 Documentation at: /docs/config/valve.html -->
153 <!--
154 <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
155 -->
156
157 <!-- Access Log processes all example.
158 Documentation at: /docs/config/valve.html
159 Note: The pattern used is equivalent to using pattern="common" -->
160 <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
161 prefix="localhost_access_log" suffix=".txt"
162 pattern="%h %l %u %t &quot;%r&quot; %s %b" />
163
164 </Host>
165 </Engine>
166 </Service>
```

Text found; occurrences marked: 18



修改完server.xml后重启容器才能生效

```
# docker restart 34b97835dad1
```

```
d4e31c8da7b4 f6152767e78c "catalina.sh run" 14 minutes ago Up 14 minutes 0.0.0.0:8080->8080/tcp awesome_aryabhata
[root@kk1181259634 home]# docker rm -f d4e31c8da7b4
d4e31c8da7b4
[root@kk1181259634 home]# docker run -d --name tomcat9 -p 8080:8080 -v /home/tomcat9/conf:/usr/local/tomcat/conf:/ro -v /home/tomcat9/webapps:/usr/local/tomcat/webapps:/ro -v /home/tomcat9/logs:/usr/local/tomcat/logs:/ro f6152767e78c
docker: Error response from daemon: Conflict. The container name "/tomcat9" is already in use by container "1f43bc6a11a77b6b2c163c1f999dca3f1604ab3dcb0ef67b0c9d4ae42c52d8b". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@kk1181259634 home]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
1f43bc6a11a7        f6152767e78c       "catalina.sh run"   About a minute ago   Created            0.0.0.0:8080->8080/tcp   tomcat9
[root@kk1181259634 home]# docker rm -f 1f43bc6a11a7
1f43bc6a11a7
[root@kk1181259634 home]# docker run -d --name tomcat9 -p 8080:8080 -v /home/tomcat9/conf:/usr/local/tomcat/conf:/ro -v /home/tomcat9/webapps:/usr/local/tomcat/webapps:/ro -v /home/tomcat9/logs:/usr/local/tomcat/logs:/ro f6152767e78c
34b97835dad18de843a81653961a3c6960891d85e7160015e23554bf72dda00e
[root@kk1181259634 home]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
34b97835dad1        f6152767e78c       "catalina.sh run"   3 minutes ago       Up 3 minutes        0.0.0.0:8080->8080/tcp   tomcat9
[root@kk1181259634 home]# docker restart 34b97835dad1
34b97835dad1
[root@kk1181259634 home]#
```

安装MySQL5.7和配置

1.运行容器

先以守护进程方式启动并运行一个容器,这里mysql的默认密码是123456

```
# docker run -d --name mysql5.7-config -p 80:3306 -e MYSQL_ROOT_PASSWORD=123456 cd3ed0dfff7e
```

```
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
Active: inactive (dead) since Wed 2019-11-20 18:57:21 CST; 1h 24min ago
Docs: man:httpd(8)
man:apachectl(8)
Process: 19529 ExecStop=/bin/kill -W INCH $(MAINPID) (code=exited, status=0/SUCCESS)
Process: 8484 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
Main PID: 8484 (code=exited, status=0/SUCCESS)
Status: "Total requests: 33; Current requests/sec: 0; Current traffic: 0 B/sec"

Nov 20 15:21:28 xk1181259634 systemd[1]: Starting The Apache HTTP Server...
Nov 20 15:21:28 xk1181259634 httpd[8484]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.1... message
Nov 20 15:21:28 xk1181259634 systemd[1]: Started The Apache HTTP Server.
Nov 20 18:57:20 xk1181259634 systemd[1]: Stopping The Apache HTTP Server...
Nov 20 18:57:21 xk1181259634 systemd[1]: Stopped The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@kk1181259634 ~]# docker run -d --name mysql5.7-config -p 80:3306 -e MYSQL_ROOT_PASSWORD=123456 cd3ed0dfff7e
43924a07f9d8119db10f501fc95839aa003557a957efa5864a8c875ae1c7fae1
[root@kk1181259634 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
43924a07f9d8        cd3ed0dfff7e       "docker-entrypoint.s..."  14 seconds ago     Up 13 seconds        33060/tcp, 0.0.0.0:80->3306/tcp   mysql5.7-config
[root@kk1181259634 ~]#
```

2.在宿主机的/home目录下新建mysql5.7目录,复制容器里面conf.d(配置)、mysql(数据)到宿主机
进入容器并执行命令

```
# docker exec -ti 43924a07f9d8 /bin/bash
```

```
Process: 8484 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
Main PID: 8484 (code=exited, status=0/SUCCESS)
Status: "Total requests: 33; Current requests/sec: 0; Current traffic: 0 B/sec"

Nov 20 15:21:28 xk1181259634 systemd[1]: Starting The Apache HTTP Server...
Nov 20 15:21:28 xk1181259634 httpd[8484]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.1... message
Nov 20 15:21:28 xk1181259634 systemd[1]: Started The Apache HTTP Server.
Nov 20 18:57:20 xk1181259634 systemd[1]: Stopping The Apache HTTP Server...
Nov 20 18:57:21 xk1181259634 systemd[1]: Stopped The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@xk1181259634 ~]# docker run -d --name mysql5.7-config -p 80:3306 -e MYSQL_ROOT_PASSWORD=123456 cd3ed0dfff7e
43924a07f9d8119db10f501fc95839aa0b3557a957efa5804a8c875a61cfae1
[root@xk1181259634 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
43924a07f9d8        cd3ed0dfff7e       "docker-entrypoint.s..."  14 seconds ago      Up 13 seconds      33060/tcp, 0.0.0.0:80->3306/tcp   mysql5.7-con
fig
[root@xk1181259634 ~]# docker exec -ti 43924a07f9d8 /bin/bash
root@43924a07f9d8:/# ls
bin boot dev docker-entrypoint-initdb.d entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@43924a07f9d8:/# ll
bash: ll: command not found
root@43924a07f9d8:/#
```

```
# docker cp 43924a07f9d8:/etc/mysql/conf.d /home/mysql5.7/
# docker cp 43924a07f9d8:/var/log /home/mysql5.7/
# docker cp 43924a07f9d8:/var/lib/mysql /home/mysql5.7/
```

```
连接成功
Last login: Wed Nov 20 19:05:47 2019 from 223.71.70.8

Welcome to Alibaba Cloud Elastic Compute Service !

[root@xk1181259634 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
43924a07f9d8        cd3ed0dfff7e       "docker-entrypoint.s..."  18 minutes ago      Up 18 minutes      33060/tcp, 0.0.0.0:80->3306/tcp   mysql5.7-con
fig
[root@xk1181259634 ~]# docker cp 43924a07f9d8:/etc/mysql/conf.d /home/mysql5.7/
[root@xk1181259634 ~]# docker cp 43924a07f9d8:/var/log /home/mysql5.7/
[root@xk1181259634 ~]# docker cp 43924a07f9d8:/var/lib/mysql /home/mysql5.7/
[root@xk1181259634 ~]#
```

3.运行一个带挂载的MySQL5.7容器

```
# docker run -d -p 3306:3306 --name mysql5.7 -v
/home/mysql5.7/conf.d:/etc/mysql/conf.d:ro -v /home/mysql5.7/log:/var/log:ro
-v /home/mysql5.7/mysql:/var/lib/mysql:ro -e MYSQL_ROOT_PASSWORD=123456
cd3ed0dfff7e
```

注意连接数据库的ip需要填写的是mysql容器的ip, 使用 `docker inspect 镜像ID | grep IP`

备份镜像

```
# docker save -o 备份镜像名称 源镜像名称:tag版本
```

默认备份到当前路径

```
# docker save -o tomcat9.0.27.tar xiaokangxxs/tomcat9:1.0
```

CONTAINER ID

IMAGE

COMMAND

CREATED

STATUS

PORTS

NAMES

34b97835dad1

f6152767e78c

"catalina.sh run"

13 hours ago

Up 9 minutes

0.0.0.0:8080->8080/tcp

tomcat9

[root@xk1181259634 ~]# docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
xiaokangxxs/tomcat9	1.0	a1adffb24a42	13 seconds ago	325MB
xiaokangxxs/tomcat8.5	1.0	08aa9903b331	20 hours ago	507MB
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat8.5	8.5	08aa9903b331	20 hours ago	507MB
centos	7	5e35e350aded	9 days ago	203MB
tomcat	7.0.96-jdk8-adoptopenjdk-openj9	2aebb055ce2c	12 days ago	345MB
tomcat	8.5.47-jdk8-adoptopenjdk-hotspot	50d274ef4fb3	12 days ago	324MB
tomcat	9.0.27-jdk8-adoptopenjdk-hotspot	f6152767e78c	12 days ago	325MB
nginx	1.16.1-alpine-perl	87bf3eb6b71a	4 weeks ago	57MB
tomcat	7	e3a393545a10	4 weeks ago	506MB
tomcat	8.5	882487b8be1d	4 weeks ago	507MB
mysql	5.7.28	cd3ed0dfff7e	4 weeks ago	437MB
redis	latest	de25a81a5a0b	4 weeks ago	98.2MB

[root@xk1181259634 ~]# docker save -o tomcat9.0.27.tar xiaokangxxs/tomcat9:1.0

unknown shorthand flag: 'o' in -o

See 'docker save --help'.

[root@xk1181259634 ~]# docker save -o tomcat9.0.27.tar xiaokangxxs/tomcat9:1.0

[root@xk1181259634 ~]# ^C

[root@xk1181259634 ~]#

访问历史

/root

文件名称

大小

类型

修改时间

权限

用户/用户组

lost+found

media

mnt

opt

proc

root

run

sbin

srv

sys

tmp

usr

var

myData

myData1

mysql_package

TestDirectory

.bash_history

.bash_logout

.bash_profile

.bashrc

.cshrc

.mysql_history

.pydistutils.cfg

.tcshrc

a.txt

tomcat9.0.27.tar

14.9 KB

18 B

176 B

176 B

100 B

156 B

64 B

129 B

26 B

314.1 MB

BASH_HIS...

BASH_LO...

BASH_PR...

BASHRC ...

CSHRC 文件

MYSQL_HI...

CFG 文件

TCSHRC ...

文本文件

WinRAR ...

2019/11/20 18:33

2019/11/20 18:35

2019/11/13 07:58

2019/11/20 11:16

2019/11/20 21:27

2013/12/29 10:26

2013/12/29 10:26

2013/12/29 10:26

2013/12/29 10:26

2019/11/13 13:29

2017/08/18 12:00

2013/12/29 10:26

2019/11/20 11:04

2019/11/21 08:02

drwxr-xr-x

drwxr-xr-x

drwxr-xr-x

drwxr-xr-x

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

恢复镜像

docker load -i 镜像文件

默认从当前路径找要恢复的镜像文件

docker save -i tomcat9.0.27.tar

Untagged: xiaokangxxs/tomcat9:1.0

Deleted: sha256:a1adffb24a420962728b7498f452e2aecdd9c69e814a76abfc55f4d60e14cb38

Deleted: sha256:86f9501c0c7547d06d3fdb5a9f10d39c7b814cdd09ca2446d5257b6efb74656

[root@xk1181259634 ~]# docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat8.5	8.5	08aa9903b331	20 hours ago	507MB
xiaokangxxs/tomcat8.5	1.0	08aa9903b331	20 hours ago	507MB
centos	7	5e35e350aded	9 days ago	203MB
tomcat	7.0.96-jdk8-adoptopenjdk-openj9	2aebb055ce2c	12 days ago	345MB
tomcat	8.5.47-jdk8-adoptopenjdk-hotspot	50d274ef4fb3	12 days ago	324MB
tomcat	9.0.27-jdk8-adoptopenjdk-hotspot	f6152767e78c	12 days ago	325MB
nginx	1.16.1-alpine-perl	87bf3eb6b71a	4 weeks ago	57MB
tomcat	7	e3a393545a10	4 weeks ago	506MB
tomcat	8.5	882487b8be1d	4 weeks ago	507MB
mysql	5.7.28	cd3ed0dfff7e	4 weeks ago	437MB
redis	latest	de25a81a5a0b	4 weeks ago	98.2MB

[root@xk1181259634 ~]# docker load -i tomcat9.0.27.tar

b0b228f76fff: Loading layer [=====] 139.8kB/139.8kB

Loaded image: xiaokangxxs/tomcat9:1.0

[root@xk1181259634 ~]# docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
xiaokangxxs/tomcat9	1.0	a1adffb24a42	9 minutes ago	325MB

访问历史

/root

文件名称

大小

类型

修改时间

权限

用户/用户组

lost+found

media

mnt

opt

proc

root

run

sbin

srv

sys

tmp

usr

var

myData

myData1

mysql_package

TestDirectory

.bash_history

.bash_logout

.bash_profile

.bashrc

.cshrc

.mysql_history

.pydistutils.cfg

.tcshrc

a.txt

tomcat9.0.27.tar

14.9 KB

18 B

176 B

176 B

100 B

156 B

64 B

129 B

26 B

314.1 MB

BASH_HIS...

BASH_LO...

BASH_PR...

BASHRC ...

CSHRC 文件

MYSQL_HI...

CFG 文件

TCSHRC ...

文本文件

WinRAR ...

2019/11/20 18:33

2019/11/20 18:35

2019/11/13 07:58

2019/11/20 11:16

2019/11/20 21:27

2013/12/29 10:26

2013/12/29 10:26

2013/12/29 10:26

2013/12/29 10:26

2019/11/13 13:29

2017/08/18 12:00

2013/12/29 10:26

2019/11/20 11:04

2019/11/21 08:02

drwxr-xr-x

drwxr-xr-x

drwxr-xr-x

drwxr-xr-x

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

-rw-r--r--

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

root/root

DockerFile简介

DockerFile是由一系列命令和参数构成的脚本，这些命令应用于操作系统（centos或ubuntu）基础镜像并最终创建的一个新镜像。

前面所讲的手工方式，修改配置文件、添加、删除文件目录的方式，来构建一种新镜像，这种手工方式麻烦还比较容易出错，而且不能复用

这里我们使用DockerFile，用脚本方式来构建自动化，可复用的，高效率的创建镜像的方式，是企业级开发的首选方式。

在软件开发生命周期中，采用DockerFile来构建镜像：

- 1.对于开发人员可以为开发团队提供一个完全一致的开发环境
- 2.对于测试人员可以直接拿开发时所构建的镜像或者通过DockerFile文件构建一个新的镜像开始工作
- 3.对于运维人员，在部署时，可以实现应用的无缝衔接

DockerFile常用指令

命令	解释
FROM image_name:tag	定义了用哪个基础镜像启动构建流程（相当于Java中的extends）
MAINTAINER user_info	声明镜像维护者信息
LABEL key value	镜像描述元信息（可写多条）
ENV key value	设置环境变量（可写多条）
RUN command	构建容器时需要运行的命令（可写多条）
WORKDIR path_dir	设置终端默认登录进来的工作目录
EXPOSE port	当前容器对外暴露的端口号
ADD source_dir/file dest_dir/file	将宿主机的文件复制到容器内，如果是一个压缩文件，将会在复制后自动解压
COPY source_dir/file dest_dir/file	和ADD相似，但是如果有压缩文件是不能解压
VOLUME	创建一个可以从本地主机或其它容器挂载的挂载点，一般用来存放数据库和需要保存的数据等
CMD	指定容器启动时要运行的命令，假如有多个CMD，最后一个生效
ENTRYPOINT	指定容器启动时要运行的命令
ONBUILD	当构建一个被继承的DockerFile时运行的命令，父镜像在被子镜像继承后父镜像的onbuild被出发，可以把ONBUILD理解为一个触发器

DockerFile构建自定义centos（相当于helloworld级别）

1.编写DockerFile

```
FROM centos
MAINTAINER xiaokang<xiaokang.188@qq.com>

LABEL name="xiaokangxxs CentOS Image" \
      build-date="2019年11月21日 08:57:33"

ENV CENTOS_HOME /root/
WORKDIR $CENTOS_HOME

RUN yum -y install net-tools
RUN yum -y install vim

EXPOSE 80
CMD ["/bin/bash"]
```

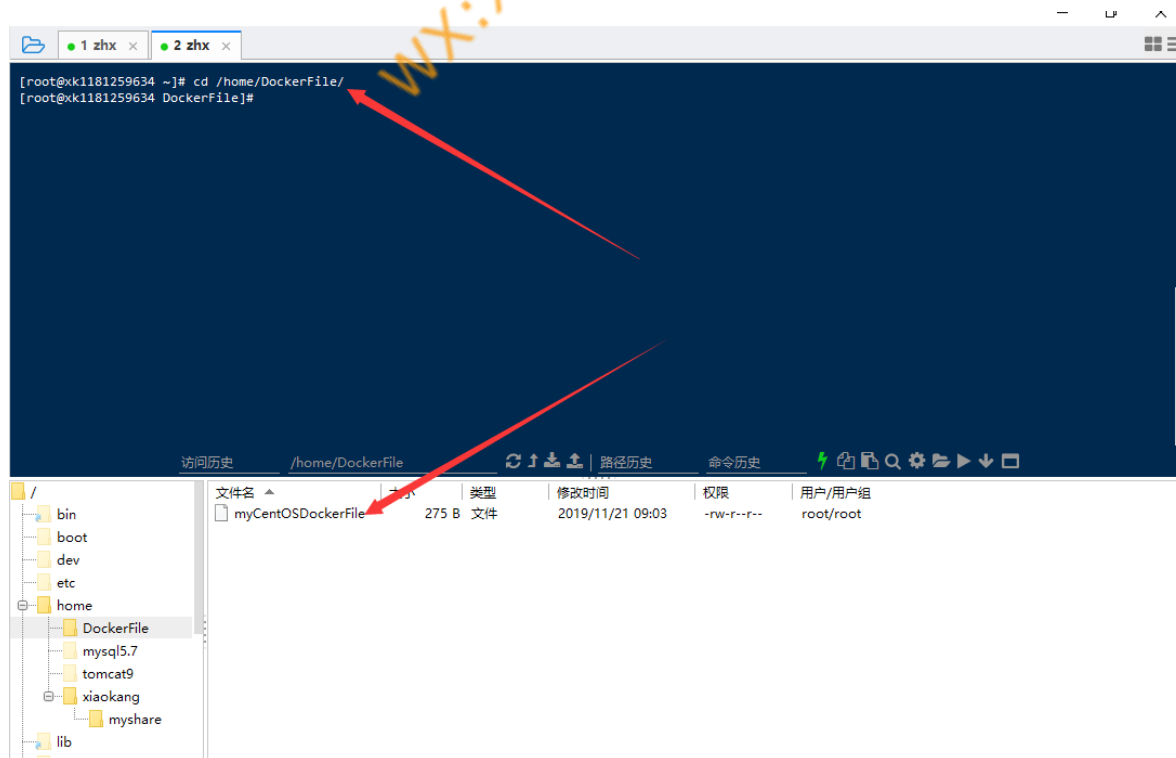
2.构建

docker build -f DockerFile文件名 -t 镜像名称:tag版本

文件默认从当前目录读取进行构建

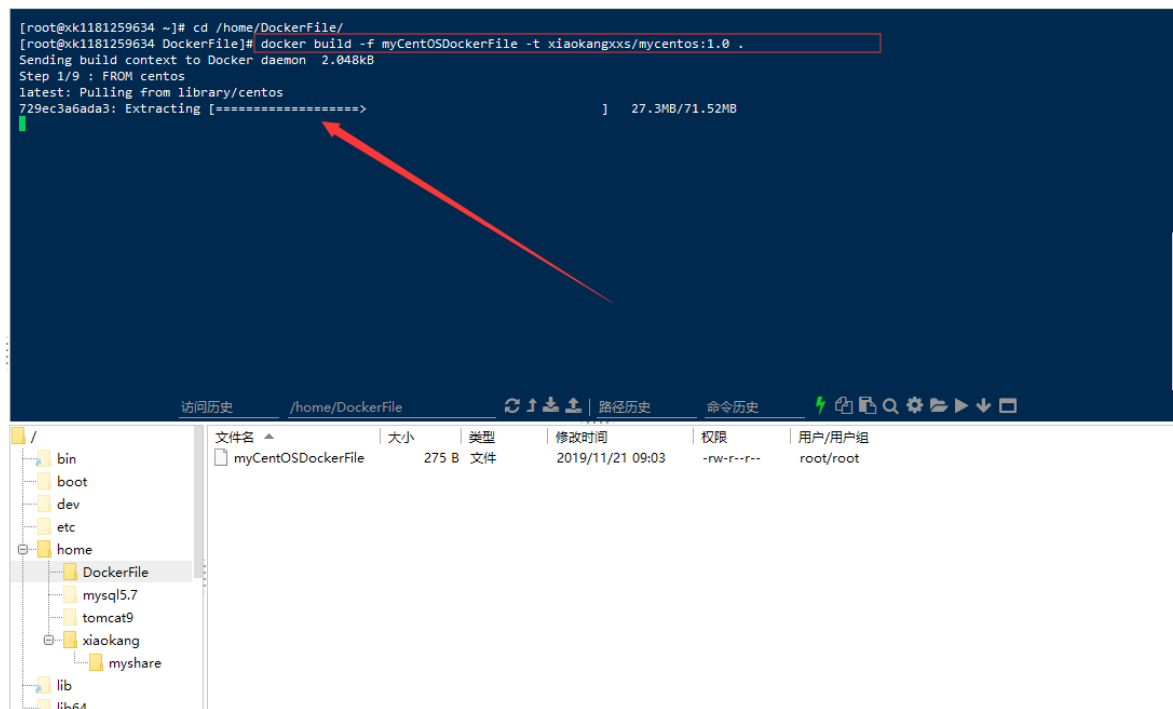
注意最后一定要有一个英文句号

```
# docker build -f 指定DockerFile文件名 -t 镜像名称:tag版本 .
```



```
# docker build -f myCentOSDockerFile -t xiaokangxxs/mycentos:1.0 .
```

```
[root@xk1181259634 ~]# cd /home/DockerFile/
[root@xk1181259634 DockerFile]# docker build -f myCentOSDockerFile -t xiaokangxxs/mycentos:1.0 .
Sending build context to Docker daemon 2.048kB
Step 1/9 : FROM centos
latest: Pulling from library/centos
729ec3a6ada3: Extracting [=====] 27.3MB/71.52MB
```



访问历史 /home/DockerFile

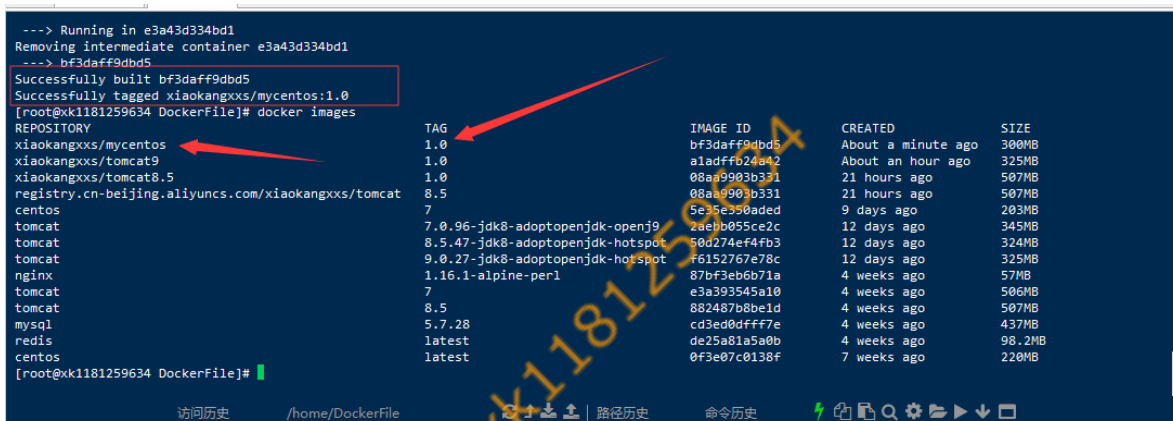
文件名	大小	类型	修改时间	权限	用户/用户组
myCentOSDockerFile	275 B	文件	2019/11/21 09:03	-rw-r--r--	root/root

bin boot dev etc home DockerFile mysql5.7 tomcat9 xiaokang myshare lib lib64

```
---> Running in e3a43d334bd1
Removing intermediate container e3a43d334bd1
---> bf3daff9dbd5
Successfully built bf3daff9dbd5
Successfully tagged xiaokangxxs/mycentos:1.0
[root@xk1181259634 DockerFile]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
xiaokangxxs/mycentos	1.0	bf3daff9dbd5	About a minute ago	300MB
xiaokangxxs/tomcat9	1.0	aladffb24a42	About an hour ago	325MB
xiaokangxxs/tomcat8.5	1.0	08aa9903b331	21 hours ago	507MB
registry.cn-beijing.aliyuncs.com/xiaokangxxs/tomcat	8.5	08aa9903b331	21 hours ago	507MB
centos	7	5e35e30aded	9 days ago	203MB
tomcat	7.0.96-jdk8-adopopenjdk-openj9	2aeb055ce2c	12 days ago	345MB
tomcat	8.5.47-jdk8-adopopenjdk-hotspot	50d274ef4fb3	12 days ago	324MB
tomcat	9.0.27-jdk8-adopopenjdk-hotspot	f6152767e78c	12 days ago	325MB
nginx	1.16.1-alpine-perl	87bf3eb6b71a	4 weeks ago	57MB
tomcat	7	e3a393545a10	4 weeks ago	506MB
tomcat	8.5	882487b8be1d	4 weeks ago	507MB
mysql	5.7.28	cd3ed0dfff7e	4 weeks ago	437MB
redis	latest	de25a81a5a0b	4 weeks ago	98.2MB
centos	latest	0f3e07c0138f	7 weeks ago	220MB

```
[root@xk1181259634 DockerFile]#
```



访问历史 /home/DockerFile

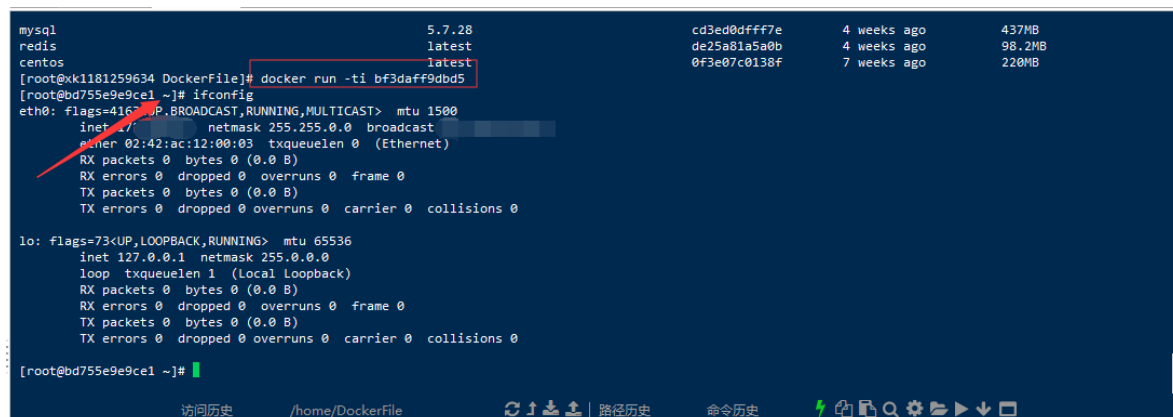
3.运行

```
# docker run -ti bf3daff9dbd5
```

```
mysql 5.7.28 cd3ed0dfff7e 4 weeks ago 437MB
redis latest de25a81a5a0b 4 weeks ago 98.2MB
centos latest 0f3e07c0138f 7 weeks ago 220MB
[root@xk1181259634 DockerFile]# docker run -ti bf3daff9dbd5
[root@bd755e9e9ce1 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.2 netmask 255.255.0.0 broadcast
    ether 02:42:ac:12:00:03 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@bd755e9e9ce1 ~]#
```



访问历史 /home/DockerFile

4.查看镜像历史

```
# docker history bf3daff9dbd5
```

[root@xk1181259634 DockerFile]# docker history bf3daf9dbd5					
IMAGE	CREATED	CREATED BY		SIZE	COMMENT
bf3daf9dbd5	7 minutes ago	/bin/sh -c #(nop) CMD ["/bin/bash"]		0B	
3e68afd6f08f	7 minutes ago	/bin/sh -c #(nop) EXPOSE 80		0B	
7b42f51a63ce	7 minutes ago	/bin/sh -c yum -y install vim		39.2MB	
0ef0425f39f7	7 minutes ago	/bin/sh -c yum -y install net-tools		41.2MB	
06da8a1f5f0f	7 minutes ago	/bin/sh -c #(nop) WORKDIR /root/		0B	
0efa77216fe2	7 minutes ago	/bin/sh -c #(nop) ENV CENTOS_HOME=/root/		0B	
f257b45a03ee	7 minutes ago	/bin/sh -c #(nop) LABEL name=xiaokangxxs Ce..		0B	
0501efb4cb43	7 minutes ago	/bin/sh -c #(nop) MAINTAINER xiaokang<xiaok..		0B	
0f3e07c0138f	7 weeks ago	/bin/sh -c #(nop) CMD ["/bin/bash"]		0B	
<missing>	7 weeks ago	/bin/sh -c #(nop) LABEL org.label-schema.sc..		0B	
<missing>	7 weeks ago	/bin/sh -c #(nop) ADD file:d6fdacc1972df524a..		220MB	
[root@xk1181259634 DockerFile]#					

DockerFile构建自定义tomcat

1.编写DockerFile

\$PATH是宿主机系统变量

```
FROM centos
MAINTAINER xiaokang<xiaokang.188@qq.com>

LABEL name="xiaokangxxs Tomcat8.5.9 Image" \
      build-date="2019年11月21日 08:57:33"

COPY copyright.txt /home/copyright.txt
ADD jdk-8u221-linux-x64.tar.gz /opt/
ADD apache-tomcat-8.5.9.tar.gz /opt/

ENV DEFAULT_HOME /opt/apache-tomcat-8.5.9/
WORKDIR $DEFAULT_HOME

ENV JAVA_HOME /opt/jdk1.8.0_221
ENV CLASSPATH $JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
ENV CATALINA_HOME /opt/apache-tomcat-8.5.9/
ENV CATALINA_BASE /opt/apache-tomcat-8.5.9/
ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/lib:$CATALINA_HOME/bin

RUN yum -y install net-tools
RUN yum -y install vim

EXPOSE 1124
CMD ["/opt/apache-tomcat-8.5.9/bin/catalina.sh","run"]
```

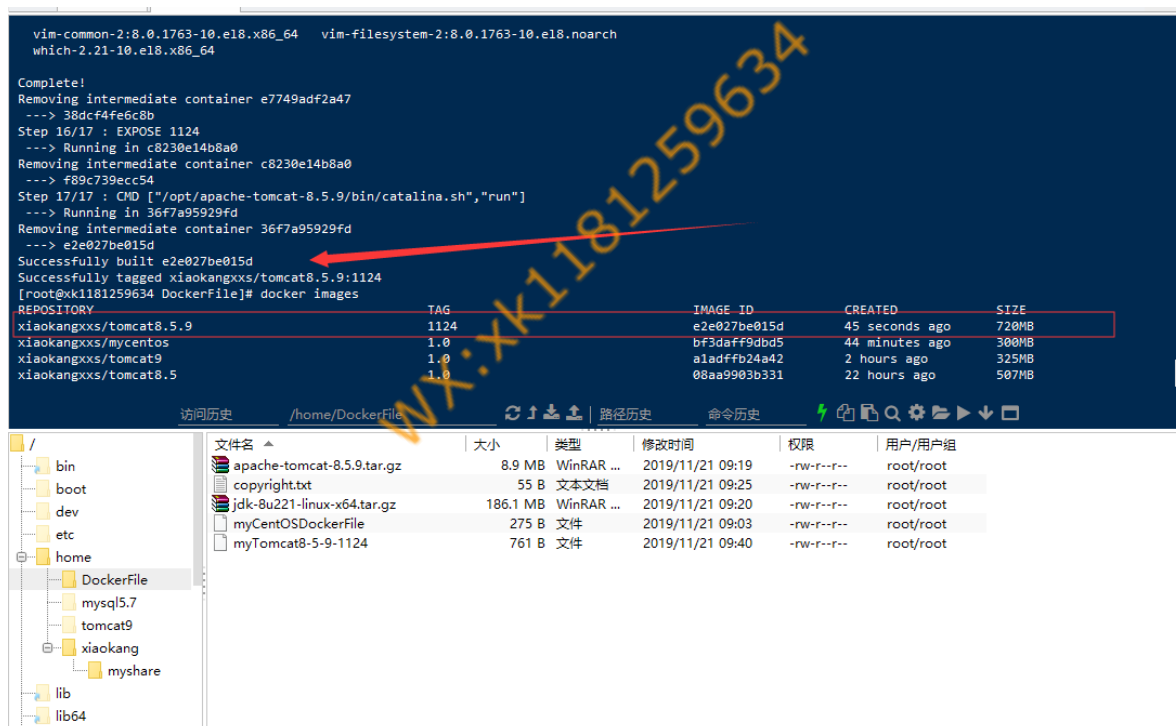
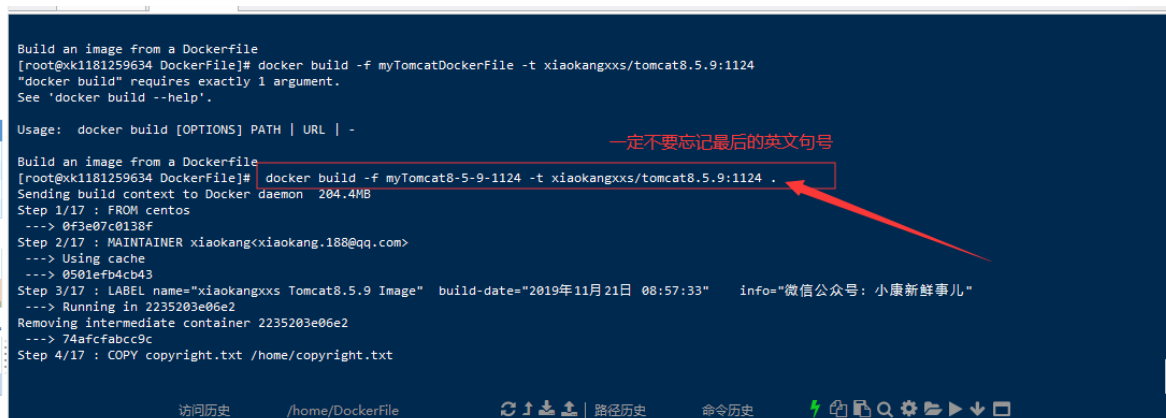
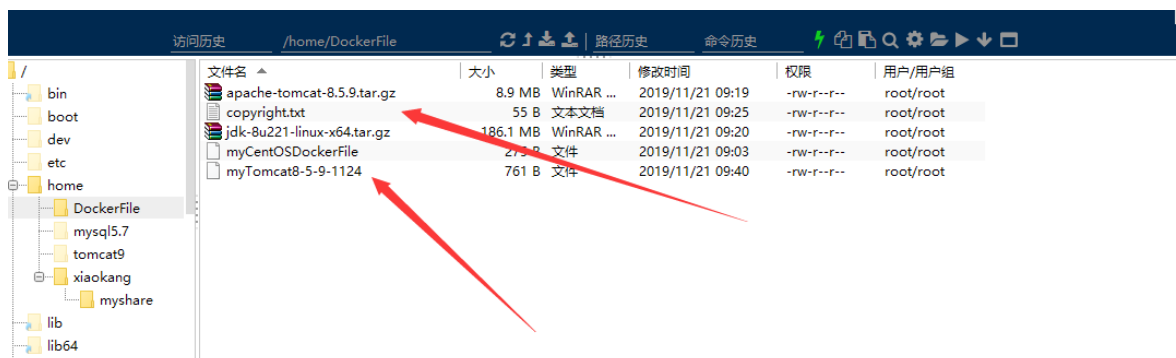
2.构建

docker build -f DockerFile文件名 -t 镜像名称:tag版本

文件默认从当前目录读取进行构建

注意最后一定要有一个英文句号，镜像名称必须小写

```
# docker build -f myTomcat8-5-9-1124 -t xiaokangxxs/tomcat8.5.9:1124 .
```

3.运行

```
# docker run -di -p 8080:1124 --name tomcat8.5.9-1124 e72295b32841
```



```
[root@xk1181259634 DockerFile]# docker run -di -p 8080:1124 --name tomcat8.5.9-1124 e72295b32841
eb0ad8648ed568cc81df6a05e4bece0919d58dc505e8ccbae04d515e4d9e1206
[root@xk1181259634 DockerFile]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
eb0ad8648ed5        e72295b32841       "/opt/apache-tomcat-..." 4 seconds ago       Up 3 seconds        0.0.0.0:8080->1124/tcp  tomcat8.5.9-1124

[root@xk1181259634 DockerFile]# docker exec -ti eb0ad8648ed5 /bin/bash
[root@eb0ad8648ed5 apache-tomcat-8.5.9]# ls
LICENSE NOTICE README CASE-NOTES RUNNING.txt bin conf lib logs temp webapps work
[root@eb0ad8648ed5 opt]# cd /opt/
[root@eb0ad8648ed5 opt]# ls
bash: ls: command not found
[root@eb0ad8648ed5 opt]# ls
apache-tomcat-8.5.9 jdk1.8.0_221
[root@eb0ad8648ed5 opt]# cd /home/
[root@eb0ad8648ed5 home]# cat copyright.txt
Powered by xiaokang!© 2018-2019 All rights reserved.
[root@eb0ad8648ed5 home]#
```

DockerFile通过VOLUME实现容器卷

VOLUME ["/home/v1","/home/v2"]

前面启动容器的时候可以通过 -v /宿主机目录:/容器目录:ro 来实现容器卷目录挂载

但由于定义DockerFile的时候，并不能保证在所有的宿主机上都有这样的特定目录，所以在DockerFile定义中，只能指定容器卷目录。

```
FROM centos
MAINTAINER xiaokang<xiaokang.188@qq.com>

LABEL name="xiaokangxxs volumeTest Image" \
      build-date="2019年11月21日 10:33:56"

VOLUME ["/home/conf", "/home/webapps"]

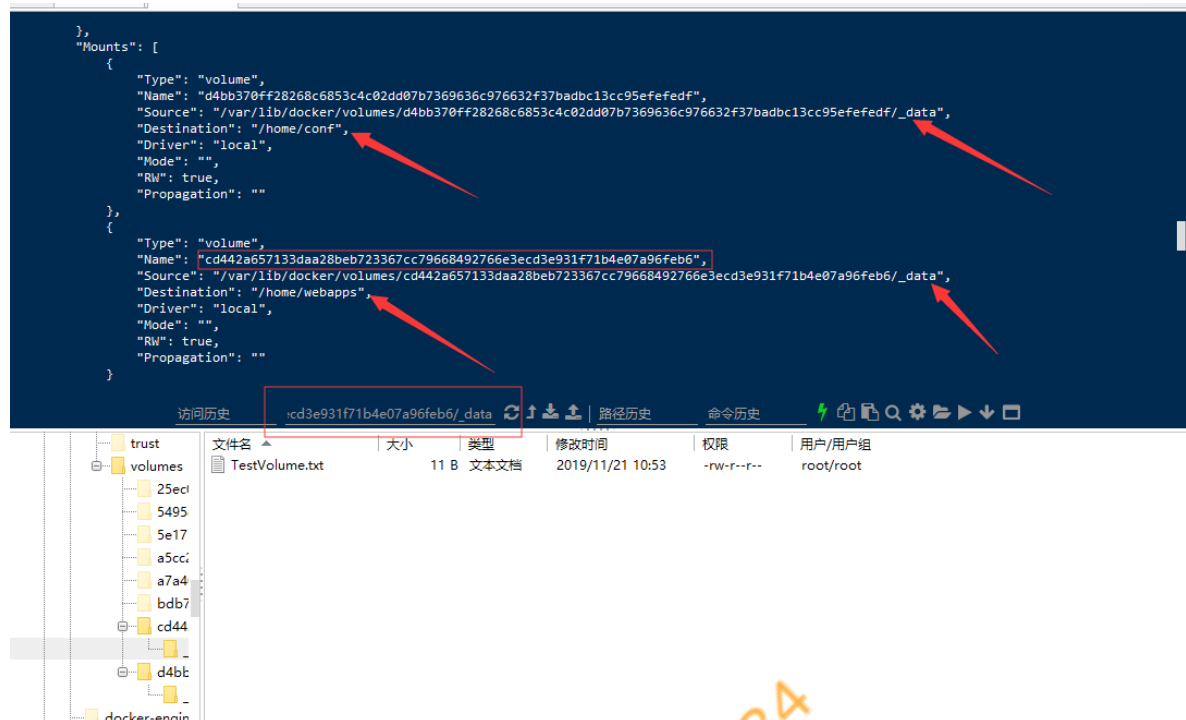
CMD ["/bin/bash"]
```

```
centos          7          5e35e350aded      9 days ago      203MB
tomcat          7.0.96-jdk8-adoptopenjdk-openj9 2aebb055ce2c      13 days ago      345MB
tomcat          8.5.47-jdk8-adoptopenjdk-hotspot 50d274ef4fb3      13 days ago      324MB
tomcat          9.0.27-jdk8-adoptopenjdk-hotspot f6152767e78c      13 days ago      325MB
nginx           1.16.1-alpine-perl 87bf3eb6b71a      4 weeks ago      57MB
tomcat          7          e3a393545a10      4 weeks ago      506MB
tomcat          8.5        882487b8be1d      4 weeks ago      507MB
mysql           5.7.28     cd3ed0dfff7e      4 weeks ago      437MB
redis           latest     de25a81a5a0b      4 weeks ago      98.2MB
centos          latest     0f3e07c0138f      7 weeks ago      220MB

[root@xk1181259634 DockerFile]# docker run -ti --name volumeTest d4da17dcb9af
[root@7daf09545b0f /]# cd /home/
[root@7daf09545b0f home]# ls
conf webapps
[root@7daf09545b0f home]# docker ps
bash: docker: command not found
[root@7daf09545b0f home]# cd conf/
[root@7daf09545b0f conf]# ls
[root@7daf09545b0f conf]# ls
[root@7daf09545b0f conf]# ls
[root@7daf09545b0f conf]# cd ..
[root@7daf09545b0f conf]# cd conf/
```

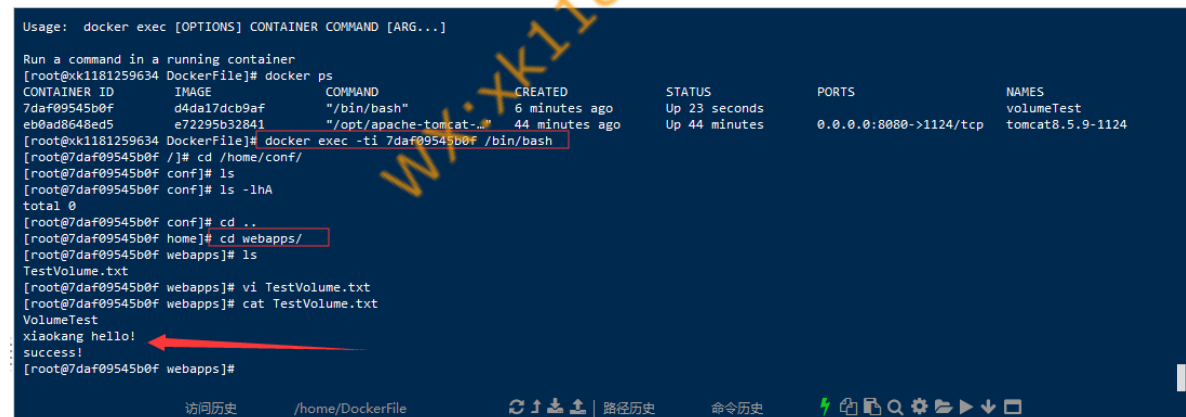
```
# docker inspect 7daf09545b0f
```

查看容器元信息，可以找到挂载的宿主机目录



进入到容器中,可以看到挂载的目录

```
# docker exec -ti 7daf09545b0f /bin/bash
```



CMD、ENTRYPOINT的区别和联系

CMD和ENTRYPOINT都是容器启动的时候执行命令；都支持exec和shell方式；一般用法是：单独一个CMD或者先ENTRYPOINT，后CMD结合使用。

假如有多个CMD，启动的时候带命令参数，会覆盖前面的CMD命令，最后一个命令生效，所以平时用CMD的时候，有一种情况就是单独一个CMD命令即可，启动命令也不带参数。