

## docker 笔记

### 虚拟化技术 VM

### 容器虚拟化 Docker

#### 安装

```
yum -y install docker
```

#### 启动

```
systemctl start docker
```

```
systemctl status docker
```

```
systemctl enable docker
```

#### docker运行状态

```
[root@docker ~]# docker ps
```

#### 关闭docker

```
docker stop CONTAINER ID
```

#### 网卡连接方式

```
[root@docker ~]# ifconfig docker0 |grep inet  
    inet 192.168.0.1 netmask 255.255.240.0 broadcast 0.0.0.0
```

#### 镜像管理:

##### 搜索镜像

```
[root@docker ~]# docker search 'mysql'
```

##### 下载镜像

```
[root@docker ~]# docker pull docker.io/mysql
```

##### 是否下载成功

```
[root@docker ~]# docker images |grep mysql  
mysql latest  
b7dc06006192 3 weeks ago
```

<https://www.certdepot.net/rhel7-get-started-docker/>

[https://access.redhat.com/documentation/en/red-hat-enterprise-linux-atomic-host/7/single/getting-started-with-containers/#get\\_started\\_with\\_docker\\_formatted\\_container\\_images](https://access.redhat.com/documentation/en/red-hat-enterprise-linux-atomic-host/7/single/getting-started-with-containers/#get_started_with_docker_formatted_container_images)

红帽官方镜像

```
[root@docker ~]# docker pull registry.access.redhat.com/rhel7  
centos
```

```
[root@docker ~]# docker pull centos
```

进入容器

```
[root@docker ~]# docker run -i -t registry.access.redhat.com/  
rhel7 /bin/bash
```

查看配置的详细信息

```
[root@docker ~]# docker inspect fd55f0c85d15
```

通过红帽的基础镜像 制作lamp架构

```
[root@c0207afe056b ~]# vi /etc/yum.repos.d/rhel6.repo
```

```
[root@c0207afe056b ~]# yum -y install httpd php php-mysql mysql-  
server mysql
```

制作之后 提交镜像

```
[root@docker ~]# docker commit --author=kanghuikai --  
message='contain lamp' c0207afe056b uplooking:vv01
```

实例1: 让lamp的服务开机自启动

```
vim /usr/bin/lamp
```

```
#!/bin/bash
```

```
service httpd start &> /dev/null
```

```
service mysql start &> /dev/null
```

```
tail -f /etc/passwd
```

```
chmod +x /usr/bin/lamp
```

—保存镜像，提交保存 `docker commit`

执行脚本

```
[root@docker ~]# docker run -d 04d8c471ed1f /usr/bin/lamp
```

可以查看 lamp是否运行

```
[root@docker ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
c559736ecde3	04d8c471ed1f	"/usr/bin/lamp"	3
minutes ago	Up 3 minutes		
kickass_bhabha			

查看ip地址

```
[root@docker ~]# docker inspect c559736ecde3
```

访问lamp页面

`elinks http://192.168.0.x`

实例2: 设置ssh管理服务

让sshd服务在前端运行

```
vim /usr/bin/lamp
```

```
#!/bin/bash
```

```
service httpd start &> /dev/null
```

```
service mysql start &> /dev/null
```

```
/usr/sbin/sshd -D
```

提交docker commit 然后执行脚本 `docker run -d xxx /usr/sbin/lamp`

实例3 安装页面数据《上传一套discuz》

以及mysql的数据

数据不能存放在docker之中，需要存放在宿主机的磁盘中

```
[root@docker ~]# mkdir /webroot
```

```
[root@docker ~]# mkdir /data
```

上传discuz页面

```
[root@docker ~]# docker run -d -v /webroot:/var/www/html/ -v /data:/var/lib/mysql/ -p 80:80 7c43f6c6b4d3 /usr/bin/lamp
```

```
[root@docker ~]# docker run -d -v /webroot:/var/www/html/ -v /data:/var/lib/mysql/ -p 81:80 7c43f6c6b4d3 /usr/bin/lamp
```

```
[root@docker ~]# docker run -d -v /webroot:/var/www/html/ -v /data:/var/lib/mysql/ -p 82:80 7c43f6c6b4d3 /usr/bin/lamp
```

```
[root@docker ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
084b910b33af	7c43f6c6b4d3	"/usr/bin/lamp"	About a minute ago
Up	About a minute	0.0.0.0:82->80/tcp	
naughty_hypatia			
b847e248705d	7c43f6c6b4d3	"/usr/bin/lamp"	About a minute ago
Up	About a minute	0.0.0.0:81->80/tcp	
backstabbing_curran			
91a0eec6852a	7c43f6c6b4d3	"/usr/bin/lamp"	18 minutes ago
Up	18 minutes	0.0.0.0:80->80/tcp	
furious_poincare			

将镜像导入到本地

```
[root@docker ~]# docker save --output=/root/rhel6.8.tar.gz 7c43f6c6b4d3
```

从镜像导入到其它主机

```
[root@a02 ~]# docker load -i /root/rhel6.8.tar.gz
```

定义一个基础镜像:

```
[root@docker ~]# mkdir /root/rhel7.2
[root@docker ~]# yum -y install vim-enhanced openssh-clients openssh-server net-tools --installroot=/root/rhel7.2/
```

```
[root@docker ~]# cd /root/rhel7.2/  
[root@docker rhel7.2]# tar czf /root/rhel7.2.tar.gz ./
```

导入容器镜像

```
[root@docker ~]# cat rhel7.2.tar.gz | docker import - test/  
rhel7.2:v1.0
```

此外，也可以通过指定 URL 或者某个目录来导入，例如

```
# docker import http://example.com/exampleimage.tgz  
example/imagerepo
```

注：用户既可以使用 `docker load` 来导入镜像存储文件到本地镜像库，也可以使用 `docker import` 来导入一个容器快照到本地镜像库。这两者的区别在于容器快照文件将丢弃所有的历史记录和元数据信息（即仅保存容器当时的快照状态），而镜像存储文件将保存完整记录，体积也要大。此外，从容器快照文件导入时可以重新指定标签等元数据信息。

## Docker 清理命令集锦

杀死所有正在运行的容器

代码如下：

```
docker kill $(docker ps -a -q)
```

删除所有已经停止的容器

代码如下：

```
docker rm $(docker ps -a -q)
```

删除所有未打 `dangling` 标签的镜像

代码如下：

```
docker rmi $(docker images -q -f dangling=true)
```

删除所有镜像

代码如下:

```
docker rmi $(docker images -q)
```

为这些命令创建别名

代码如下:

```
# ~/.bash_aliases
```

```
# 杀死所有正在运行的容器.
```

```
alias dockerkill='docker kill $(docker ps -a -q)'
```

```
# 删除所有已经停止的容器.
```

```
alias dockercleanc='docker rm $(docker ps -a -q)'
```

```
# 删除所有未打标签的镜像.
```

```
alias dockercleani='docker rmi $(docker images -q -f dangling=true)'
```

```
# 删除所有已经停止的容器和未打标签的镜像.
```

```
alias dockerclean='dockercleanc || true && dockercleani'
```