# 概述

* Docker是一个构建在LXC之上的,基于进程容器(Processcontainer)的轻量级VM解决
* Docker是一个开源的引擎，可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署，包括VMs（虚拟机）、[bare metal](http://www.whatis.com.cn/word_5275.htm)、OpenStack 集群和其他的基础应用平台。
* Docker的初衷也就是将各种应用程序和他们所依赖的运行环境打包成标准的container/image,进而发布到不同的平台上运行。
* 在LXC的基础上, Docker额外提供的Feature包括：标准统一的打包部署运行方案， 历史版本控制， Image的重用，Image共享发布等等。

# 与系统级VM区别

* Docker container和普通的虚拟机Image相比, 最大的区别是它并不包含操作系统内核。

# 使用场景

* web应用的自动化打包和发布；
* 自动化测试和持续集成、发布；
* 在服务型环境中部署和调整数据库或其他的后台应用；
* 从头编译或者扩展现有的OpenShift或Cloud Foundry平台来搭建自己的PaaS环境。

# 安装过程

1. 使用yum源安装docker

|  |
| --- |
| yum install docker -y |

1. 在centos7安装后提示如下错误

|  |
| --- |
| Transaction check error:  file /usr/lib/systemd/system/blk-availability.service from install of device-mapper-7:1.02.107-5.el7.x86\_64 conflicts with file from package lvm2-7:2.02.105-14.el7.x86\_64  file /usr/sbin/blkdeactivate from install of device-mapper-7:1.02.107-5.el7.x86\_64 conflicts with file from package lvm2-7:2.02.105-14.el7.x86\_64  file /usr/share/man/man8/blkdeactivate.8.gz from install of device-mapper-7:1.02.107-5.el7.x86\_64 conflicts with file from package lvm2-7:2.02.105-14.el7.x86\_64 |

1. 说明包发生了冲突，则使用如下命令卸载；

|  |
| --- |
| rpm -qa | grep lvm2  yum -y remove lvm2-libs-2.02.105-14.el7.x86\_64 |

启动docker服务

|  |
| --- |
| systemctl start docker.service 启动服务  systemctl enable docker.service 开机启动 |

# Docker使用

## 搜索可用docker镜像

|  |
| --- |
| docker search tutorial |

## 创建容器

|  |
| --- |
| sudo docker run -i -t ubuntu /bin/bash |

上例中用于创建ubuntu镜像。

|  |
| --- |
| apt-get update && apt-get install vim |

安装vim工具。

## 退出docker镜像

|  |
| --- |
| Exit |

## 重新启动并进入镜像

|  |
| --- |
| sudo docker start docker-name  sudo docker attach docker-name |

## 创建守护进程

使容器在后台运行，不随着shell命令行的关闭而关闭。

|  |
| --- |
| sudo docker run --name docker\_dave -d ubuntu /bin/sh -c "while true;do echo hello world ;sleep 1;done" |

## 官网查找镜像

|  |
| --- |
| sudo docker search centos |

## 查看本地当前已创建的容器

|  |
| --- |
| Sudo docker ps –a |

## 进入运行中的容器

|  |
| --- |
| docker exec -i -t 5a9627882d0b /bin/bash |

## 查看容器的运行日志

|  |
| --- |
| 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容器内外互相拷贝数据

|  |
| --- |
|  |

# 镜像管理

## 构建镜像

### 创建docker hub账号

去<https://hub.docker.com/>网站按照提示完成注册

使用如下命令按照提示完成登录

|  |
| --- |
| Sudo docker login  Username ：renyi3033  Password：  Email: |

### Docker commit

|  |
| --- |
| Docker commit 323444 renyi3033/docker-demo |

### Docker dockerfile

|  |  |
| --- | --- |
| Mkdir static\_web  touch dockerfile   |  | | --- | | #version 1.0  FROM ubuntu:14.04  MAINTAINER James Turnbull "james@example.com"  RUN apt-get update  RUN apt-get install -y nginx  RUN echo 'Hi,I am in your contanier'\  >/usr/share/nginx/html/index.html  EXPOSE 80 |   Docker build –t=”renyi3033/static\_web” |

### 查看镜像构建过程

|  |
| --- |
| docker history aa924a669ab5 |

### 从新镜像启动容器

|  |
| --- |
| docker run -d -p 45678:80 --name static\_web\_demo renyi3033/static\_web nginx -g "daemon off;"  通过-p命令，将容器的80端口映射到主机的45678端口上 |

# 容器操作

## Mysql镜像

### 查看可使用镜像

|  |
| --- |
| docker pull mysql:5.7 |

### 检查镜像是否下载

|  |
| --- |
| docker images |

### 将宿主机的目录挂载到容器中

|  |
| --- |
| docker run -it -v /opt/Downloads:/opt/Downloads centos /bin/bash  docker run –i -t -v /opt/Downloads:/opt/Downloads centos /bin/bash |

### 启动Mysql的docker容器

|  |
| --- |
| docker run --name mysql-db -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=mytest -d mysql:5.7 |

注意，这里的容器名字叫：mysql-db，mysql的root用户密码是：mytest，映射宿主机子的端口3306到容器的端口3306，仓库名mysql和标签(tag）唯一确定了要指定的镜像，其实如果这里只有一个mysql也有必要要tag，执行该命令返回的结果是：

92b1912564f3195856599cf88b781ffec7b195e7613971b7a0cec407d76ccc31

### 启动程序并挂载本地目录

|  |
| --- |
| docker run --name tomcat -d --link mysql:mysql --link activemq:mq --link redis:redis --link zookeeper:zookeeper --privileged -v /root/tomcat/webapps/:/usr/local/tomcat/webapps/ -v /root/tomcat/logs/:/usr/local/tomcat/logs/ -v /root/tomcat/work/:/usr/local/tomcat/work/ -v /root/tomcat/temp/:/usr/local/temp/ -p 8080:8080 tomcat:7.0.70-jre7 |

|  |
| --- |
| docker run -d --name mysql --privileged -v /root/mysql/data:/var/lib/mysql -v /root/mysql/my.cnf:/etc/mysql/my.cnf -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=11111111 mysql:5.6.32 |

## 安装Apache

|  |
| --- |
| docker run -d --name apache --link tomcat:casweb --privileged -v /root/apache/conf/httpd.conf:/usr/local/apache2/conf/httpd.conf -p 80:80 -p 8086:8086 -p 8087:8080 docker.io/httpd |

## 安装docker ui

### Pull docker ui

|  |
| --- |
| docker pull uifd/ui-for-docker  docker images  docker run -it -d --name docker-web -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock docker.io/uifd/ui-for-docker |

# 其它

## 关闭Selinux

|  |
| --- |
| cd /etc/Selinux  将SELINUX=enforcing改为SELINUX=disabled  设置后需要重启才能生效 |

## 对正在运行的容器打快照

|  |
| --- |
| docker commit -p 0648627a7ec5 docker-webui-back |

## 压缩快照

|  |
| --- |
| docker save -o ~/mysql-back.tar mysql-back |

## 加载镜像

|  |
| --- |
| docker load -i ~/container-backup.tar |

## 运行镜像

|  |
| --- |
| docker run --name tomcat8081 -d --link mysql5:mysql5 --link activemq:mq --link redis:redis --link zookeeper:zookeeper --privileged -v /root/tomcat8081/webapps/:/usr/local/tomcat/webapps/ -p 8081:8080 -p 8283:8283 tomcat |

## 进入镜像内部

|  |
| --- |
| docker run -i -t mysql /bin/bash |

## 从容器内拷贝东西至虚拟机

|  |
| --- |
| docker cp 2f64d40fa697:/etc/mysql /usr/local |

## 安装MySQL并映射本地目录

|  |
| --- |
| docker run -d --name mysql --privileged -v /root/mysql/my.cnf:/etc/mysql/my.cnf -v /root/mysql/data:/var/lib/mysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=11111111 mysql-b |

## 安装zookeeper

|  |
| --- |
| docker run --name zookeeper -d -p 2181:2181 -p 2888:2888 -p 3888:3888 jplock/zookeeper:3.4.8 |

## 安装elasticsearch

|  |
| --- |
| docker run -d --name esearch \  --privileged \  -v "/root/esdata":/usr/share/elasticsearch/data \  -p 9200:9200 \  -p 9300:9300 \  elasticsearch:2.3.5 |

## 查询MySQL是否区分大小写

|  |
| --- |
| show Variables like '%table\_names' |

## 政务云监管平台docker部署

|  |
| --- |
| docker run -d --name mysql --privileged -v /root/mysql/data:/var/lib/mysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=11111111 mysql:5.6.32  docker run --name redis --privileged -v /root/redis:/data -d -p 6379:6379 redis:3.0.7  docker run --name='activemq' -d \  -e 'ACTIVEMQ\_NAME=amqp-srv1' \  -e 'ACTIVEMQ\_REMOVE\_DEFAULT\_ACCOUNT=true' \  -e 'ACTIVEMQ\_ADMIN\_LOGIN=admin' -e 'ACTIVEMQ\_ADMIN\_PASSWORD=admin' \  -e 'ACTIVEMQ\_WRITE\_LOGIN=producer\_login' -e 'ACTIVEMQ\_WRITE\_PASSWORD=producer\_password' \  -e 'ACTIVEMQ\_READ\_LOGIN=consumer\_login' -e 'ACTIVEMQ\_READ\_PASSWORD=consumer\_password' \  -e 'ACTIVEMQ\_JMX\_LOGIN=jmx\_login' -e 'ACTIVEMQ\_JMX\_PASSWORD=jmx\_password' \  -e 'ACTIVEMQ\_STATIC\_TOPICS=topic1;topic2;topic3' \  -e 'ACTIVEMQ\_STATIC\_QUEUES=queue1;queue2;queue3' \  -e 'ACTIVEMQ\_MIN\_MEMORY=1024' -e 'ACTIVEMQ\_MAX\_MEMORY=4096' \  -e 'ACTIVEMQ\_ENABLED\_SCHEDULER=true' \  --privileged \  -v /root/activemq:/data/activemq \  -v /root/log/activemq:/var/log/activemq \  -p 8161:8161 \  -p 61616:61616 \  -p 61613:61613 \  webcenter/activemq:5.13.2  docker run --name zookeeper -d -p 2181:2181 -p 2888:2888 -p 3888:3888 jplock/zookeeper:3.4.8  docker run -d --name esearch \  --privileged \  -v "/root/esdata":/usr/share/elasticsearch/data \  -p 9200:9200 \  -p 9300:9300 \  elasticsearch:2.3.5  docker run --name tomcat -d --link mysql:mysql --link activemq:mq --link redis:redis --link zookeeper:zookeeper --privileged -v /root/tomcat/webapps/:/usr/local/tomcat/webapps/ -p 8080:8080 tomcat:7.0.70-jre7  docker run --name tomcat81 -d --link mysql:mysql --link activemq:mq --link redis:redis --link zookeeper:zookeeper --privileged -v /root/tomcat8081/webapps/:/usr/local/tomcat/webapps/ -p 8081:8080 -p 8283:8283 tomcat:7.0.70-jre8  docker run -d --name nginx80 --restart=always --privileged -v /etc/localtime:/etc/localtime -v /root/nginx/log80/:/var/log/nginx/ -v /root/nginx/conf:/etc/nginx/ -v /root/nginx/html:/usr/share/nginx/html -p 80:80 nginx |