

Docker 调研报告

冯海雄

2015/05/20 09:05

内容列表

说明	3
docker的安装	3
centos	3
ubuntu	3
docker常用命令概览	4
docker镜像	4
Dockerfile	5
docker容器	5
启动一个容器	5
容器的网络设置	5
容器的重启策略	5
容器对硬件的控制	5
容器的日志	5
容器的链接	6
容器的数据管理	6
docker 仓库	6
公有仓库	6
私有仓库	6
docker 使用实例	6
docker mysql服务	6
docker 目前存在的问题	7

- [说明](#)
 - [docker的安装](#)
 - [centos](#)
 - [ubuntu](#)
 - [docker常用命令概览](#)
 - [docker镜像](#)
 - [Dockerfile](#)
 - [docker容器](#)
 - [启动一个容器](#)
 - [容器的网络设置](#)
 - [容器的重启策略](#)
 - [容器对硬件的控制](#)
 - [容器的日志](#)
 - [容器的链接](#)
 - [容器的数据管理](#)
 - [docker 仓库](#)
 - [公有仓库](#)
 - [私有仓库](#)
 - [docker 使用实例](#)
 - [docker mysql服务](#)
 - [docker 目前存在的问题](#)

说明

Docker 是一个方便开发人员和系统管理员开发、打包、运行程序的开放式平台，Docker是 PaaS 提供商 dotCloud 开源的一个基于 LXC 的高级容器引擎，2013年将源代码托管在 Github 上，使用go语言编写遵从Apache2.0协议开源，docker可以简化应用环境的配置和维护。

docker的安装

centos

- 目前只支持centos6.5及centos7
- 内核要求 2.6.32-431 以及这个版本之上的内核
- 只支持64位的系统，docker不能运行在32位的系统上

1. centos6.5上安装docker

1. 配置yum源
wget http://mirrors.yun-idc.com/epel/6/x86_64/epel-release-6-8.noarch.rpm
sudo rpm -ivh epel-release-6-8.noarch.rpm
2. 如果之前有安装docker则执行sudo yum -y remove docker
sudo yum -y install docker-io
3. 启动docker服务
sudo service docker start
4. 获取一个镜像
sudo docker pull centos
5. 查看下载好的centos镜像
sudo docker images centos
6. 启动一个shell容器
sudo docker run -i -t centos /bin/bash

ubuntu

docker支持的ubuntu系统有

- Ubuntu Trusty 14.04 (LTS)
 - Ubuntu Precise 12.04 (LTS)
 - Ubuntu Saucy 13.10
- 要求内核不低于**3.10**系统必须是**64**位系统

1. ubuntu 上装docker

1. 安装docker
wget -qO- <https://get.docker.com/> | sh
2. docker 更新
wget -N <https://get.docker.com/> | sh

3. 启动docker 服务
`sudo service docker start`
4. 获取一个镜像
`sudo docker pull centos`
5. 查看下载好的centos镜像
`sudo docker images centos`
6. 启动一个shell容器
`sudo docker run -i -t centos /bin/bash`

docker常用命令概览

命令	描述
build	通过一个dockerfile脚本构建镜像
commit	将一个容器提交成一个镜像
cp	将容器中的文件拷贝到宿主系统
exec	在正在运行的容器中执行一个命令
export	将指定名称的镜像导出到一个tar包中
images	列出本地所有可用的镜像
import	从一个tar包中导入镜像
kill	杀掉一个正在运行的容器
load	从tar包中导入镜像与import相同
login	登录一个仓库
logs	查看容器的运行日志
ps	列出正在运行的容器
pull	拉取一个镜像到本地
push	推送一个镜像到远程仓库
restart	重启一个正在运行中的容器
rm	删除一个容器
rmi	删除一个镜像
run	启动一个容器
save	保存一个镜像到tar包中
search	在 Docker Hub 中搜索镜像
start	启动一个容器
stop	停止一个容器
tag	给镜像打标签
top	查看容器内的进程

docker镜像

docker镜像基于一个只读的文件系统，docker上的一切操作都基于镜像，

1. 获得镜像有以下几种方法
 1. 用docker pull 从docker Hub获取
 2. 基于当前系统构建基础镜像
 3. 使用dockerfile构建镜像
`build -t mysql ./`
其中-t指定构建的镜像名称，./表示从当前目录构建当前目录中必须有dockerfile
 4. 使用commit将容器固化成镜像
1. 镜像的管理
 1. docker images 列出本地所有镜像
 2. docker rmi 删除一个指定的镜像
 3. docker push 推送镜像到仓库
 4. 给镜像添加标签管理镜像
如 `docker tag hello-world:latest admin/test:devel` 将hello-world:latest打上 admin/test:devel标签

Dockerfile

FROM 指定父级镜像
MAINTAINER 指定维护者的信息
RUN 执行要执行的命令
CMD 执行容器启动时的命令
LABEL 指定标签
EXPOSE 指定镜像对外开放的端口
ENV 指定环境变量
ADD 添加本地文件到镜像中去 如果本定文件为URL可以先下载再拷贝，如果本定文件为压缩包文件，可以执行解压操作
COPY 拷贝本地文件到镜像中去
ENTRYPOINT 镜像入口程序设定
VOLUME 添加数据存储的卷
USER 致死那个运行的用户
WORKDIR 切换工作目录类似cd
ONBUILD 表示下一个继承镜像构建的时候执行

docker容器

容器是一个动态的运行进程,容器依赖于指定的镜像,容器可以看作一个轻量级动态镜像

docker ps - Lists containers.
docker logs - Shows us the standard output of a container.
docker stop - Stops running containers.

运行ubuntu镜像下的bash终端
sudo docker run -i -t ubuntu /bin/bash
docker的版本信息
sudo docker version

启动一个容器

docker run 可以启动一个容器例如
docker run -it - -rm ubuntu /bin/bash
表示启动一个ubuntu的终端交互容器，其中-i指定容器为可交互的 -t指定为一个终端容器 - - rm表示容器退出后就删除容器

容器的网络设置

host模式，使用- -net=host指定。
container模式，使用- -net=container:name
none模式，使用- -net=none指定。
bridge模式，使用- -net=bridge指定，默认设置。
容器中的 /etc/hosts可以通过- -add-hosts来添加内容
容器可以通过-dns来指定dns服务器

容器的重启策略

- -restart 可以指定容器的重启策略
no 表示不重启
onfailure 表示容器如果以失败状态结束则重启容器
always 表示只要退出就重启容器
- -rm 选项表示运行完毕就删除容器
- -rm 和- -restart参数不能同时设置

容器对硬件的控制

-m="": Memory limit (format: <number><optional unit>, where unit = b, k, m or g)
-memory-swap="": Total memory limit (memory + swap, format: <number><optional unit>, where unit = b, k, m or g)
-c, - -cpu-shares=0 CPU shares (relative weight)

容器的日志

- -log-driver= none禁用日志，容器运行过程中不会有日志输出
- -log-driver= json-file 容器日志以json格式的形式输出，默认为json格式
- -log-driver= syslog 容器日志直接以系统日志的形式记录，这种情况下docker logs 指令不可用。

容器的链接

- link 可以将运行了不同服务的容器关联在一起容器可以使用被关联过来的容器的服务实例如下:
启动一个运行了mysql的容器
`sudo docker start sqlubuntu`
新建一个关联了mysql容器的终端容器
`sudo docker run -it -rm -link sqlubuntu:db base:6.6 /bin/bash`
通过新建的终端访问容器中的数据库

容器的数据管理

在容器中管理数据的 2 个主要方式:

- Data volumes
- Data volume containers.

1、Data volumes 数据卷

数据卷是一个由 UFS 文件系统专门设计的特殊目录,它可以提供很多有用的特性:

- 数据卷可以在容器之间共享和重用
- 对数据卷的改变是立马生效
- 当你更新数据卷中的数据的时候,不会被包含到 image 中
- 卷会一直存在直到没有容器使用他们

1)添加一个数据卷

在用 `docker run` 命令的时候,使用 `-v` 标记来添加一个数据卷。在一次 `run` 中多次使用可以挂载多个数据卷,下面加载一个卷到 web 容器上。

```
$ sudo docker run -d -P -name web -v /webapp training/web
```

容器可以写入和删除挂上去的目录

docker 仓库

仓库用来存储和管理镜像的

公有仓库

DockerHub是目前最大的docker仓库,默认的情况下我们使用的仓库就是公有仓库

`docker pull ubuntu:latest` 从DockerHub获取镜像

`docker push base:6.6` 推送镜像至Dockerhub

私有仓库

为了提高网络速度方便使用docker可以通过docker提供的镜像搭建私有仓库私有仓库使用时必须指明仓库的名称如现在有一个名为yutu-v04.dsc:5000的仓库

则使用如下:

拉取本仓库的base镜像

```
docker pull yutu-v04.dsc:5000/base
```

推送本地镜像至私有仓库

```
docker tag hello-world yutu-v04.dsc:5000/hello-world
```

```
docker push yutu-v04.dsc:5000/hello-world
```

现在的**docker**使用的**api**版本为v1, 私有仓库目前权限控制还不完善不能控制用户为只读或者只写

docker 使用实例

docker mysql服务

1. dockerfile

```
FROM ubuntu
MAINTAINER fenghaixiong "fenghaixiong@nd.com.cn"
RUN apt-get update
RUN apt-get install -y openssh-server mysql-server supervisor
RUN sed -i 's/UsePAM yes/UsePAM no/g' /etc/ssh/sshd_config
RUN sed -i 's/TZ=Asia/Shanghai; export TZ /etc/profile
RUN sed -i 's/TZ=Asia/Shanghai; export TZ /etc/bash.bashrc
RUN useradd admin -m -s /bin/bash
RUN useradd guest -m -s /bin/bash
RUN echo "admin:admin" | chpasswd
RUN echo "guest:guest" | chpasswd
```

```
RUN echo "admin  ALL=(ALL)  ALL" >> /etc/sudoers
RUN sed -i '$a\TZ=Asia/Shanghai'; export TZ /home/admin/.profile
RUN mkdir /var/run/sshd
RUN sed -i '127\0\0/ s/bind/# bind/' /etc/mysql/my.cnf
RUN /etc/init.d/mysql start &&\
    mysql -e " CREATE USER 'admin'@'%' IDENTIFIED BY 'admin';"&&\
    mysql -e "grant all privileges on *.* to 'admin'@'%' identified by 'admin';"
RUN mkdir -p /var/log/supervisor
COPY supervisord.conf /etc/supervisord.conf
EXPOSE 22 3306
CMD ["/usr/bin/supervisord"]
supervisord.conf
[supervisord]
nodaemon=true
[program:sshd]
command=/usr/sbin/sshd -D
[program:mysql]
command=/usr/sbin/mysql
2. 构建镜像
    docker build -t mysql ./
3. 使用镜像
    docker run - -name=mysql_docker -d -p 5037:3306 -p 5036:22 mysql
```

docker 目前存在的问题

1. docker 在使用的过程中会导致系统莫名的卡死并重启
出现上述情况的具体描述：
 1. docker运行平台：
centos6.5
 2. 死机的表现：
电脑卡死，鼠标键盘无响应，持续20秒左右电脑自动重启
 3. 死机的触时间：
容器退出的时候。
 4. 死机产生的原因不明确，死机出现具有一定随机性。
2. docker 创建容器失败提示错误不明重启docker服务可恢复正常
3. docker 私有仓库缺少权限管理