

Leo_wlCnBlogs

导航

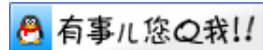
[博客园](#)[首页](#)[新随笔](#)[联系](#)[订阅 XML](#)[管理](#)

< 2018年9月 >						
日	一	二	三	四	五	六
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

公告

我的标签

2014 2013下 2013上 201
2下 LVS中文 开源 反向代理
CUDA



微信订阅号: HackerVirus

Docker搭建Java Web运行环境

阅读目录

- 一、这周末体验了一下挺火的Docker技术，简单记录自己的学习笔记。
- 二、> Docker是干什么的
- 三、> Docker和传统虚拟化技术的对比
- 四、
- 五、> 搭建Docker环境
 - 1.>>> 快速安装Docker
- 六、
- 七、> 创建第一个Docker镜像
- 八、>>> 下载镜像文件
- 九、>>> 启动容器并修改镜像
- 十、>>> 安装JDK和Tomcat等
- undefined、
- undefined、>>> 编写启动脚本
- undefined、
- undefined、>>> 构建镜像
- undefined、>>> 运行新创建的镜像

统计

随笔 - 16775

文章 - 28

评论 - 1650

引用 - 0

我的好友

[Artech](#)[asp.net](#)[Banq](#)[DNN](#)[Domain Driven Design](#)[hacker2012](#)[infoworld](#)[Jianqiang Bao](#)[jv9](#)[Manavi](#)[Martin Fowler](#)[Muhammad Mosa](#)[MVP](#)[programmer](#)[ScottGu博客\[英文\]](#)[ScottGu博客\[中文\]](#)[sql mag](#)[wcf](#)[阿捷](#)[冯大辉](#)[侯伯薇](#)[吉日嘎拉](#)[李天平](#)[李永京](#)



技术QQ群:114818988
 欢迎点击访问个人网站
<http://hackervirus.sxl.cn/>

107269

hit counter html code

Visitors



FLAG counter

Visitors



Pageviews: 5,254,680

FLAG coun

昵称: HackerVirus

园龄: 8年8个月

粉丝: 2922

关注: 247

+加关注

- undefined、
- undefined、>>>提交至docker仓库
- undefined、
- undefined、>使用体验
- undefined、添加Docker用户组，避免sudo输入
- undefined、Docker常用命令

阅读目录

- Docker搭建Java Web运行环境

回到目录

Docker搭建Java Web运行环境

返回顶部

这周末体验了一下挺火的Docker技术，简单记录自己的学习笔记。

返回顶部

>Docker是干什么的

Docker 是一个基于Linux容器(LXC-linux container)的高级容器引擎，基于go语言开发，源代码托管在 Github 上，遵从Apache2.0协议开源。Docker的目标是实现轻量级的操作系统虚拟化解决方案。

学习Docker首先要了解几个概念：

镜像—Docker的镜像和常见的系统ISO镜像类似，包含了应用程序的信息；

灵动生活

刘铁猛

秋色园

圣殿骑士

图灵书籍

微软论坛

伍迷

小洋 (燕洋天)

徐磊

徐明璐

张逸

赵劫 Jeffrey Zhao

周金根

周雪峰

周银辉

关注 | 顶部 | 评论

5

0

容器—容器相当于一个可以运行起来的虚拟机，应用程序运行在容器中，Docker运行在“Docker”上；

仓库—仓库是存放镜像的地方，有类似git的版本控制，同样分为公开仓库(Public)和私有仓库(Private)两种形式；

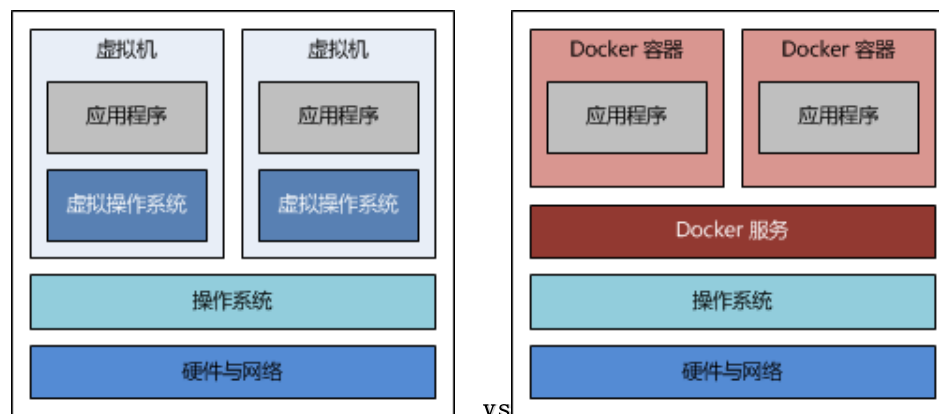
Docker支持大部分的Linux发行版，通过使用Docker容器，就可以在不同的操作系统，不同的机器上运行自己的应用，不用关心硬件、运行环境之类的配置，应用程序的迁移变得非常简单。

[返回顶部](#)

> Docker和传统虚拟化技术的对比

相比传统虚拟机技术，Docker资源占用少，启动更快，很大的方便了项目的部署和运维。Docker是在操作系统层面上实现虚拟化，复用本地主机的操作系统，传统方式是在硬件的基础上，虚拟出多个操作系统，然后在系统上部署相关的应用。

下面的这张图片参考相关博文，很形象的说明了Docker和VM之类的传统虚拟化技术的区别：

[关注](#) | [顶部](#) | [评论](#)[返回顶部](#)

5

0

[返回顶部](#)

> 搭建Docker环境

Fork me on GitHub

我使用的是Ubuntu 14.04，在这上面安装Docker服务。

>>>快速安装Docker

14.04版本的Ubuntu仓库中已经支持Docker的安装，
可以使用快速安装的方式，

```
sudo apt-get update  
sudo apt-get install docker.io
```

启动服务和守护进程

```
service docker.io status  
service docker.io start
```

这种方式安装通常不是Docker的最新版本，
如果想要安装最新版本，可以去Docker官网下载安装。

[返回顶部](#)[关注](#) | [顶部](#) | [评论](#)[返回顶部](#)

>创建第一个Docker镜像

构建Docker镜像的一般过程是首先创建一个容器，
并在容器里修改镜像，配置相关环境等，最后将修改提交为一个新镜像。

[返回顶部](#)

>>>下载镜像文件

下载用于制作镜像的系统，

```
sudo docker pull index.alauda.cn/alauda/ubuntu
```

5

0

```
latest: Pulling from index.alauda.cn/alauda/ubuntu
511136ea3c5a: Pull complete
53f858aaaf03: Downloading 28.03 MB/67.49 MB
837339b91538: Download complete
615c102e2290: Download complete
53f858aaaf03: Downloading 30.19 MB/67.49 MB
53f858aaaf03: Downloading 31.81 MB/67.49 MB
a5ec8ea8f4d9: Downloading 33.42 MB/34.46 MB
a5ec8ea8f4d9: Downloading 33.77 MB/34.46 MB
d37760e835dc: Download complete
a5ec8ea8f4d9: Download complete
673f1fe20a3a: Download complete
77421a5e2b9c: Download complete
delf78ad6493: Download complete
ae983d5e88ce: Download complete
ae983d5e88ce: Pulling fs layer
```

这里我从灵雀云的镜像中心拉取。

或者也可以直接从Docker的镜像中心拉取，不过貌似非常慢：

```
sudo docker pull ubuntu
```

下载成功后，使用images命令查看本地的镜像列表：

```
docker images
```

```
bingyue@ubuntu:/$ sudo docker images
REPOSITORY              TAG                IMAGE ID           CREATED
VIRTUAL SIZE
index.alauda.cn/alauda/ubuntu latest            ae983d5e88ce       10 months ago
255.1 MB
```

这里需要注意，使用Docker的时候请添加sudo。

默认安装完 docker 后，每次执行 docker 都需要运行 sudo 命令，如果不跟 sudo，直接执行 docker 命令会报一些权限错误。

[返回顶部](#)

Fork me on GitHub

[关注](#) | [顶部](#) | [评论](#)

5

0

>>> 启动容器并修改镜像

镜像下载到本地以后，就可以使用Docker运行，通过下面的命令参数启动容器，

`docker run <相关参数> <镜像 ID> <初始命令>`

-i: 表示以“交互模式”运行容器

-t: 表示容器启动后会进入其命令行

-v: 表示需要将本地哪个目录挂载到容器中，

格式: `-v <宿主机目录>:<容器目录>`

我的相关程序都在当前机器的/data/software/目录下，并且想把它挂载到容器的相同目录下：

```
sudo docker run -i -t -v /data/software:/data/software/ ae983d5e88ce /bin/bash
```

“镜像 ID”，也可以使用“仓库名:标签名”，例如：`index.alauda.cn/alauda/ubuntu :latest`。

上面的命令，可以使用指定的镜像运行一个shell，如果想退出该终端，可以使用`exit`命令，或者依次按下`CTRL -p+CTRL -q`，即可切换到宿主机。不过这种方式，容器依然在后天运行。

[关注](#) | [顶部](#) | [评论](#)

启动终端后，进入/data/software/目录，可以发现当前机器目录下的文件已经同步过来：

```
root@39b2cf60a4c1:/# cd /data/software/
root@39b2cf60a4c1:/data/software# ls
apache-activemq-5.12.0-bin.tar.gz      openssl-0.9.8zg.tar.gz
apache-maven-3.3.3-bin.tar.gz          pcre-8.38.tar.gz
apache-tomcat-7.0.54.zip                redis-3.0.3.tar.gz
jdk-7u25-linux-x64.tar.gz              standard-1.1.2.jar
mongodb-linux-i686-latest.tgz          wls1033_dev.zip
mysql-server_5.6.27-1ubuntu15.04_amd64.deb-bundle.tar zlib-1.2.8.tar.gz
nginx-1.8.0.tar.gz                     zookeeper-3.4.6
```

[返回顶部](#)

>>> 安装JDK和Tomcat等

5

0

安装相关的JDK等程序，这里全部安装到/data/目录：

```
tar -zxvf jdk-7u25-linux-x64.tar.gz -C /data/  
mv jdk1.7.0_25 jdk  
  
unzip apache-tomcat-7.0.54.zip -d /data/  
mv apache-tomcat-7.0.54 tomcat
```

配置环境变量

```
vi /etc/profile
```

添加下面的配置：

```
#set java environment  
export JAVA_HOME=/data/jdk  
export JRE_HOME=${JAVA_HOME}/jre  
export CLASSPATH=.:JAVA_HOME/lib:${JRE_HOME}/lib  
export PATH=JAVA_HOME/bin:PATH  
  
export CATALINA_HOME=/data/tomcat  
export CATALINA_BASE=/data/tomcat
```

保存并退出，设置立即生效：

```
source /etc/profile
```

[返回顶部](#)

[返回顶部](#)

»»» 编写启动脚本

启动tomcat时必须通过TOMCATHOME/bin/catalina.sh实现，不能使用TOMCAT_HOME/bin/startup.sh启动，否则脚本执行后容器会马上退出。

```
vi /data/start.sh
```

添加以下内容：

5

0

```
#!/bin/bash
# Export environment variable
source /etc/profile
# Start tomcat
bash /data/tomcat/bin/catalina.sh run
```

添加可执行权限:

```
chmod u+x /data/start.sh
```

返回顶部

返回顶部

>>>构建镜像

使用Docker构建镜像的两种方法:

使用docker commit 命令, 更直观一些;

使用docker build命令和Dockerfile文件, 可以模板化镜像构建过程;

这里使用docker commit的方式创建镜像。

查看容器列表:

```
sudo docker ps -a
```

	1	2	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
			39b2cf60a4c1	ae983d5e88ce:latest	"/bin/bash"	5 hours ago	Exited (0)	9 seconds ago	dreamy_euclid

提交一个新的镜像:

```
sudo docker commit 39b2cf60a4c1 bingyue/docdemo
```

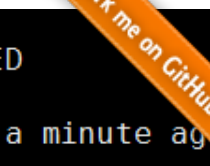
如果有Docker账号, 可以将镜像推送到Docker Hub或资金的私有Registry中。

现在查看本地的docker镜像,

5 0


```
sudo docker images
```

```
bingyue@ubuntu:/$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
bingyue/docdemo      latest             bfc7ed316d42       About a minute ago
528.2 MB
index.alauda.cn/alauda/ubuntu latest             ae983d5e88ce       10 months ago
255.1 MB
```



可以看到本地仓库已经有刚刚创建的docker镜像。

```
1 REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE
2 bingyue/docdemo latest bfc7ed316d42 About a minute ago 528.2 MB
3 index.alauda.cn/alauda/ubuntu latest ae983d5e88ce 10 months ago 255.1 MB
```

docker inspect可以查看新创建的镜像的详细信息：

```
sudo docker inspect bingyue/docdemo
```

```
bingyue@ubuntu:/$ sudo docker inspect bingyue/docdemo
[{"Architecture": "amd64",
  "Author": "",
  "Comment": "",
  "Config": {
    "AttachStderr": false,
    "AttachStdin": false,
    "AttachStdout": false,
    "Cmd": [
      "/bin/bash"
    ]
  }
}]
```

[返回顶部](#)

[>>>运行新创建的镜像](#)

```
docker run -d -p 18080:8080 --name docdemo bingyue/docdemo /data/start.sh
```

5


0

-p: 表示宿主机与容器的端口映射, 此时将容器内部的 8080 端口映射为宿主机的 18080 端口, 这样就向外界暴露了 18080 端口, 可通过 Docker 网桥来访问容器内部的 8080 端口了。

查看后台是否启动成功:


```
docker ps
```

测试访问:

 192.168.106.129:8080

Home Documentation Configuration Examples Wiki Mailing Lists

Apache Tomcat/7.0.54

 The Apache Software Foundation
http://www.apache.org

If you're seeing this, you've successfully installed Tomcat. Congratulations!

返回顶部

返回顶部

>>>提交至docker仓库

如果有Docker仓库的账户, 可以将本地创建的镜像提交至仓库。

返回顶部

返回顶部

>使用体验

Fork me on GitHub

5

0

到这一步，差不多完成了Docker的初体验，Docker应用还是比较简单的，真正复杂的应该是背后的虚拟化技术。

一步一步部署下来，的确Docker相比传统的虚拟机技术要简单了很多，有机会继续深入学习。

[返回顶部](#)

添加Docker用户组，避免sudo输入

默认安装完 docker 后，每次执行 docker 都需要运行 sudo 命令，影响效率。如果不跟 sudo，直接执行 docker images 命令会有如下问题：

```
Get http://var/run/docker.sock/v1.18/images/json: dial unix /var/run/docker.sock:
permission denied. Are you trying to connect to a TLS-enabled daemon without TLS?
```

把当前用户执行权限添加到相应的docker用户组里面就可以解决这个问题。

添加一个新的docker用户组

```
sudo groupadd docker
```

添加当前用户到docker用户组里

```
sudo gpasswd -a bingyue docker
```

重启Docker后台监护进程

```
sudo service docker restart
```

重启之后，尝试一下，是否生效

```
docker version
```

#若还未生效，则系统重启，则生效

```
sudo reboot
```

[返回顶部](#)

Docker常用命令

下载一个ubuntu镜像

```
sudo docker pull ubuntu
```

使用ubuntu运行一个交互性的shell

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

#docker ps命令

```
sudo docker ps #列出当前所有正在运行的container
```

5

0

```
sudo docker ps -l #列出最近一次启动的, 且正在运行的container
sudo docker ps -a #列出所有的container
#port命令
docker run -p 80:8080 <image> <cmd> #映射容器的8080端口到宿主机的80端口
#删除容器命令
sudo docker rm `sudo docker ps -a -q`#删除所有容器
sudo docker rm $CONTAINER_ID#删除容器id为CONTAINER_ID的容器
#其他命令快速参考:
sudo docker images #查看本地镜像
sudo docker attach $CONTAINER_ID #启动一个已存在的docker实例
sudo docker stop $CONTAINER_ID #停止docker实例
sudo docker logs $CONTAINER_ID #查看docker实例运行日志, 确保正常运行
sudo docker inspect $CONTAINER_ID #查看container的实例属性, 比如ip等等
```

分类: [15]Dev Tools

好文要顶

关注我

收藏该文



HackerVirus

关注 - 247

粉丝 - 2922

+加关注

« 上一篇: 创建型设计模式

» 下一篇: LVS + keepalived

posted on 2015-12-10 11:35 HackerVirus 阅读(23464) 评论(2) 编辑 收藏

评论

#1楼

青羽 Posted @ 2016-06-28 13:52

```
#set java environment
export JAVA_HOME=/data/jdk
```

5

0

```
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$PATH
```

支持(0) 反对(0)

#2楼

BookShu Posted @ 2017-04-28 09:56

不过，一般都不这样创建镜像，用Dockerfile更直观，还能复用。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【免费】要想入门学习Linux系统技术，你应该先选择一本适合自己的书籍

【前端】SpreadJS表格控件，可嵌入应用开发的在线Excel

【直播】如何快速接入微信支付功能

Powered by:

博客园

Copyright © HackerVirus

Fork me on GitHub

5

0