

aspirant

简单之美，难得简单，享受简单的唯美。

博客园 首页 新随笔 联系 订阅 管理

Docker 入门 到部署Web 程序- (阿里面试常用的docker命令和优点)

最近阿里的面试官问我Docker是做什么用的，我记得之前360和美团，京东的都问过，但是一直没时间看，最近有时间了，系统的学习了一下Docker，在此做一下记录，方便各位看官学习交流

一、Docker概述：

Docker: 本意是码头工人，言外之意是集装箱；

Java号称“一次编译，到处运行”，因为java虚拟机解决平台的兼容性问题，所以有java虚拟机的地方就能跑java代码；

Docker是：“一次封装，到处运行”，因为docker决绝了应用环境的问题，安装了docker的平台就能跑“docker包”，这样就决绝了“开发环境能跑，一上线就崩”的尴尬。

一句话 应用打包工具

公告

昵称: aspirant
园龄: 8年3个月
粉丝: 119
关注: 20
[+加关注](#)

<	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	

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，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。

一个完整的Docker有以下几个部分组成：

- DockerClient客户端
- Docker Daemon守护进程
- Docker Image镜像
- DockerContainer容器 [2]

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；**
- 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。

二、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：

搜索

找找看

谷歌搜索

我的标签

C 替换字符 strtok(1)

随笔分类

Android(32)

Android Studio(16)

Android-Java --接收发送消息的系统

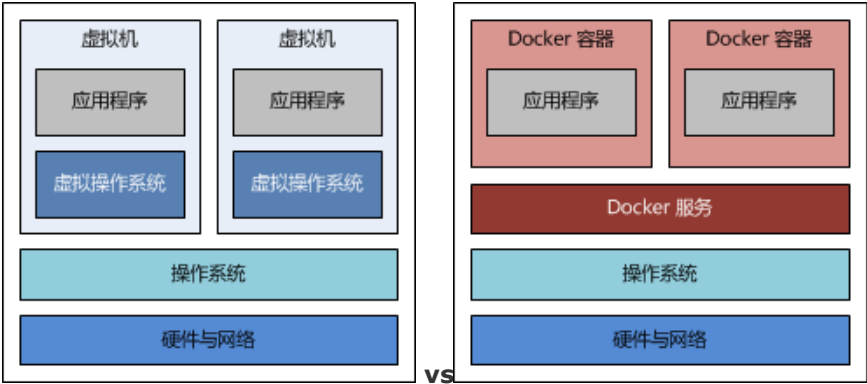
ANT MAKE 部署问题(1)

Architecture_架构知识(12)

Bat 脚本(1)

Bugzilla

C 设计模式(1)



就像一个集装箱一样，不管需要运送服装还是化工产品，还是汽车，只需要放在集装箱里面，不管到那个船上，也不管到那个铁路上，我直接机械化的搬运就行了。很简单

三、真正的部署一下Dock的开发环境

首先要安装VMware 虚拟机和 并下载Centos 操作系统，然后安装上 Linux 环境，

此处如果不懂的话，可直接百度，

安装完之后，我这边不能上网，后来做了一下配置：

参考：[VMware 虚拟机 不能上网 CentOS 6.5 Windows 7上面安装了VMware，然后安装了CentOS系统，安装完了无法上网；](#)

然后可以上网了，好了linux环境 准备好了，开始安装了

(1)直接使用yum 安装 docker:

首先肯定是要机器中安装了docker，如果没安装就是用yum install -y docker 命令安装一下

```
yum install -y docker
```

既然是部署web应用，那么当然少不了Tomcat了，所以我们应该先拉取Tomcat镜像。

命令如下,(开始的时候，我还跑到网上去下载tomcat，后来发现，根本不需要，直接用docker拉取就行了)

```
1 | docker pull tomcat
```

C 语言知识(13)
C++ 问题
C--基础知识(5)
Data Structure-数据结构算法(18)
DB_数据库(40)
DCloud 问题(1)
Deployment 部署知识(19)
DOS命令(2)
Dubbo_Spring Cloud等(13)
Eclipse(11)
ETL_数据挖掘(8)
Fitness_健身(2)
Hibernate(1)
HTTP-TCP-Socket IO NIO(21)

这个镜像有点大，所以可以事先拉取好，节省时间。

接下来我们就用Tomcat镜像来启动一个容器

```
1 | docker run -it --name webdemo -p 80:8080 tomcat /bin/bash
```

docker run -i -t <IMAGE_ID> /bin/bash: -i: 标准输入给容器 -t: 分配一个虚拟终端 /bin/bash: 执行bash脚本

这里我们启动一个交互式的容器，名字是webdemo，-p 80:8080表示将容器的8080端口映射到主机的80端口，这样我们只要访问主机的80端口就可以访问到容器的服务了。

```
[root@localhost ~]# docker run -it --name webdemo -p 80:8080 tomcat /bin/bash
root@a2f2091a661f:/usr/local/tomcat# ls
LICENSE NOTICE RELEASE-NOTES RUNNING.txt bin conf include lib logs native-jni-lib temp webapps work
root@a2f2091a661f:/usr/local/tomcat#
```

容器创建好了之后是会进入容器里面的，这时我们刚好可以看看内部的文件结构。那里有个webapps文件，我们只要把我们的web应用程序保存成war包的格式，然后再复制到这个文件中就可以了。**因为Tomcat会自动帮我们解压并且部署war包。**

如何将文件从主机复制到容器中？

因为我之前的终端在容器里面，这里我开启第二个终端进行操作。我的war包文件我放在/mnt/的目录下

```
root@localhost:/mnt
[root@localhost mnt]# cd /mnt/
[root@localhost mnt]# ls
LoginRegister webdemo.war
[root@localhost mnt]#
```

从主机复制到容器 `sudo docker cp host_path containerID:container_path`

从容器复制到主机 `sudo docker cp containerID:container_path host_path`

我们这里用的命令是：

Interview_面试(14)
IPV6
Java 基础知识(45)
Java-调度(3)
Java-缓存(4)
Java-加载类-内存分配回收-调优(14)
Java设计模式(22)
Linux C知识(16)
Linux挂载点(1)
Linux知识(18)
MAC Idea 环境搭建(7)
MAVEN(10)
MQ消息队列-MQ(7)
Multi_Thread多线程(25)

```
docker cp /mnt/webdemo.war
a2f2091a661fa51e02c0be54f252fc46fc604932526b17038ccc267affcef12c:/usr/local/tomcat/webapps
```

其中那个长的字符串是容器的id，自己去查一下。后面的路径是容器的内部路径，实在不懂可以复制过去，这里要注意：冒号后面没有空格，我之前有空格，一直复制不过去。

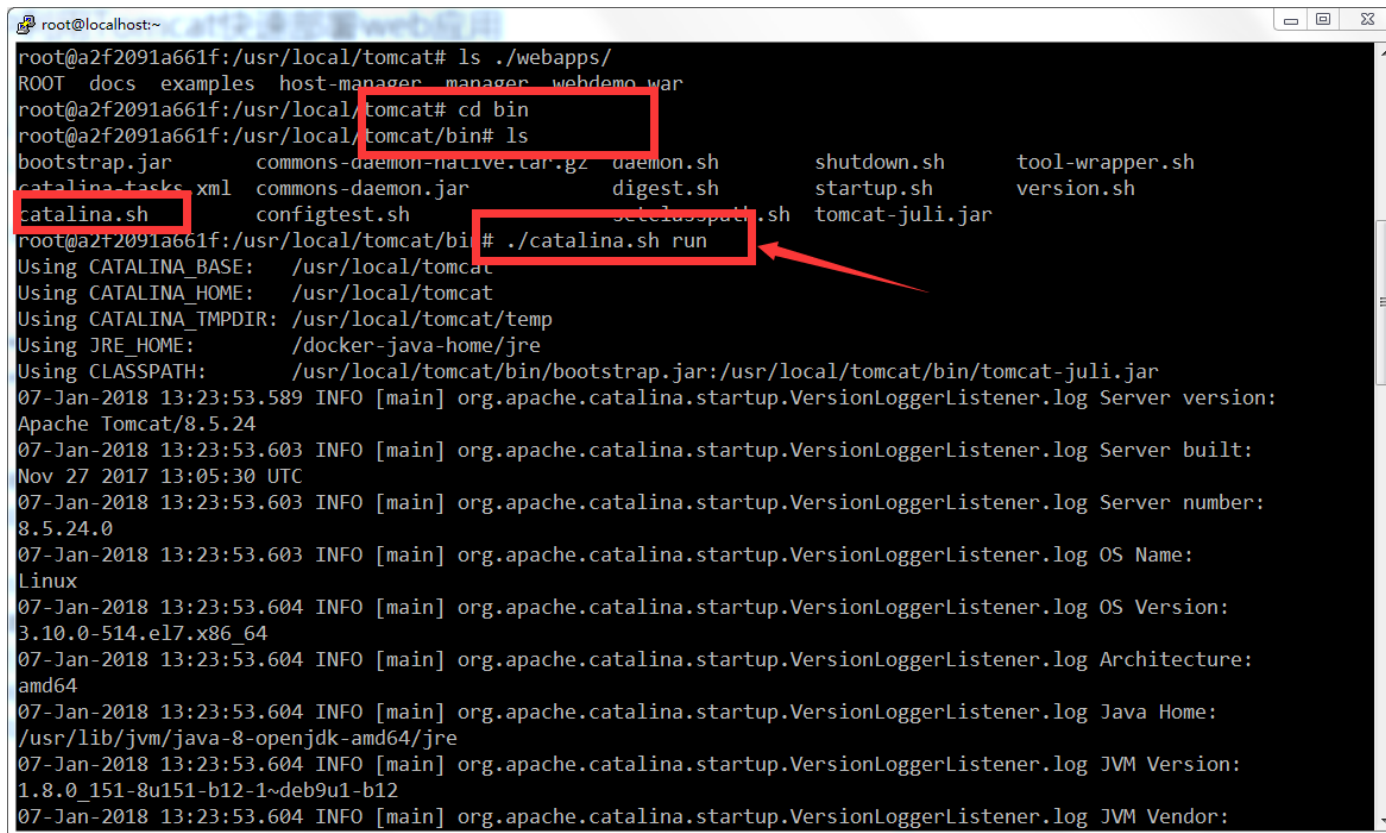
接下来就是启动Tomcat了。

war包已经导入容器中了，这个时候我们可以去第一个终端查看一下

```
[root@localhost ~]# docker run -it --name webdemo -p 80:8080 tomcat /bin/bash
root@a2f2091a661f:/usr/local/tomcat# ls
LICENSE NOTICE RELEASE-NOTES RUNNING.txt bin conf include lib logs native-jni-lib temp webapps work
root@a2f2091a661f:/usr/local/tomcat# ls ./webapps/
ROOT docs examples host-manager manager webdemo.war
root@a2f2091a661f:/usr/local/tomcat#
```

可以看到war包已经导入了。但是这个时候Tomcat服务其实是没有启动的，下面我们就启动Tomcat服务，让Tomcat帮我们完成war包的解压和部署。

[Political_时事政治\(4\)](#)[Python\(1\)](#)[Search-搜索引擎\(1\)](#)[Servlet\(2\)](#)[SPRING\(21\)](#)[Thrift\(1\)](#)[博客杂谈\(4\)](#)[个人创业项目\(4\)](#)[语言编码\(3\)](#)[随笔档案](#)[2018年9月 \(2\)](#)[2018年8月 \(12\)](#)[2018年7月 \(6\)](#)

A terminal window showing the steps to start Tomcat. The user navigates to the bin directory and runs catalina.sh. The output shows various system properties and version information. A red box highlights the 'catalina.sh' command, and a red arrow points to the 'run' argument.

```
root@a2f2091a661f:/usr/local/tomcat# ls ./webapps/
ROOT docs examples host-manager manager webdemo war
root@a2f2091a661f:/usr/local/tomcat# cd bin
root@a2f2091a661f:/usr/local/tomcat/bin# ls
bootstrap.jar      commons-daemon-native.tar.gz  daemon.sh      shutdown.sh      tool-wrapper.sh
catalina_tasks.xml commons-daemon.jar             digest.sh       startup.sh        version.sh
catalina.sh        configtest.sh                  setclasspath.sh tomcat-juli.jar
root@a2f2091a661f:/usr/local/tomcat/bin# ./catalina.sh run
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /docker-java-home/jre
Using CLASSPATH:        /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
07-Jan-2018 13:23:53.589 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version:
Apache Tomcat/8.5.24
07-Jan-2018 13:23:53.603 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built:
Nov 27 2017 13:05:30 UTC
07-Jan-2018 13:23:53.603 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server number:
8.5.24.0
07-Jan-2018 13:23:53.603 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name:
Linux
07-Jan-2018 13:23:53.604 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version:
3.10.0-514.el7.x86_64
07-Jan-2018 13:23:53.604 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture:
amd64
07-Jan-2018 13:23:53.604 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home:
/usr/lib/jvm/java-8-openjdk-amd64/jre
07-Jan-2018 13:23:53.604 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version:
1.8.0_151-b12-1~deb9u1-b12
07-Jan-2018 13:23:53.604 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor:
```

这里我们进入bin目录把目录下的catalina.sh文件run起来，这样Tomcat就跑起来了，而且Tomcat跑起来是在前端运行的，这也是我为什么打开第二个终端的原因。

最后我们可以在浏览器中查看效果：

2018年6月 (18)

2018年5月 (44)

2018年4月 (43)

2018年3月 (40)

2018年2月 (1)

2018年1月 (6)

2017年12月 (4)

2017年11月 (7)

2017年10月 (8)

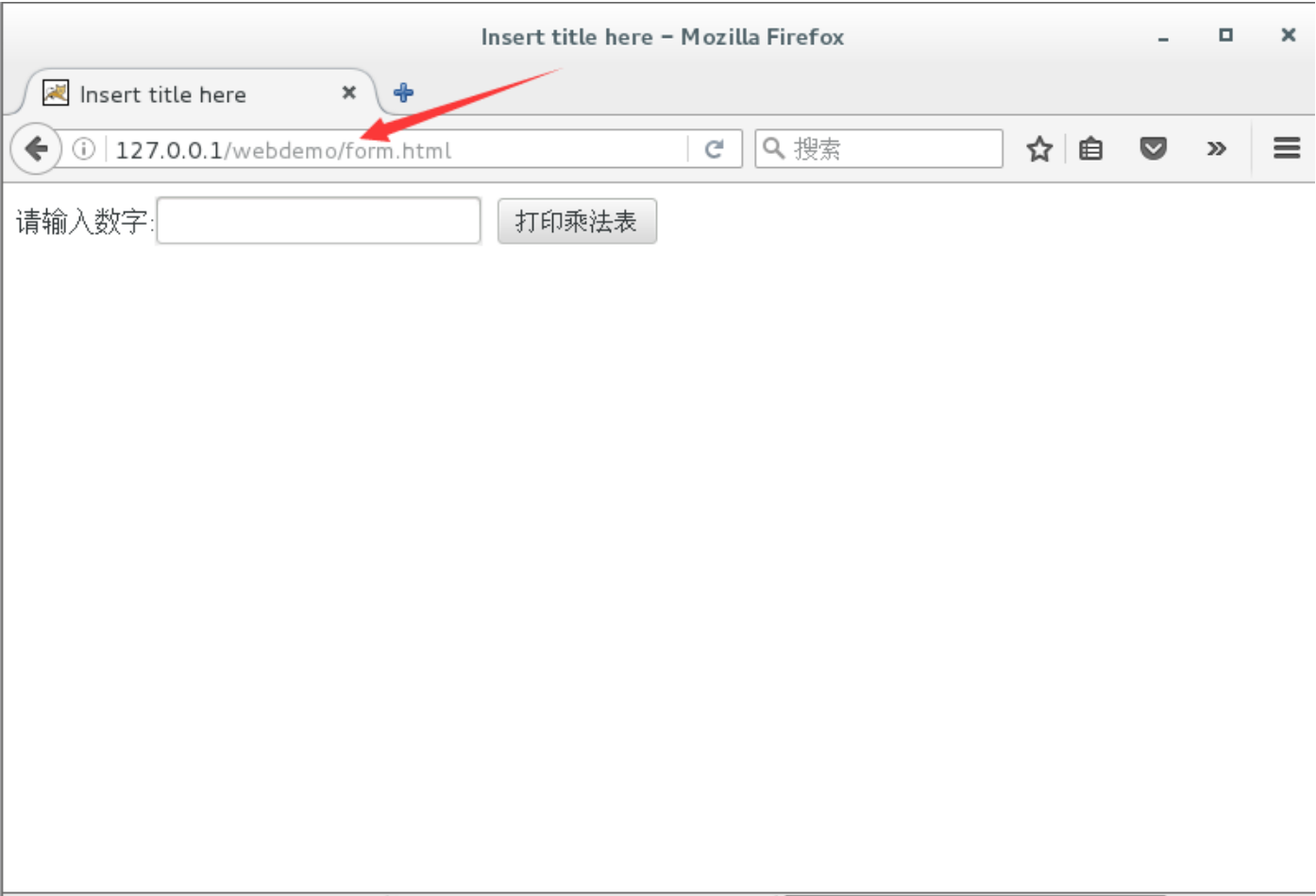
2017年9月 (7)

2017年8月 (2)

2017年7月 (28)

2017年6月 (25)

2017年5月 (17)



我照着网上的应用去做，上面的都没问题，成功了，不过后来把原来创建的容器ssh关闭了，然后又打开，如何进入已经打开的容器呢，两种方法

a:使用docker attach命令

```
docker attach db3 或者 docker attach d48b21a7e439
```

db3是后台容器的NAMES,d48b21a7e439是容器的进程ID CONTAINER ID
然后就进去了这个容器的ssh界面。

2017年4月 (11)
2017年3月 (3)
2017年2月 (2)
2017年1月 (2)
2016年12月 (1)
2016年10月 (1)
2016年9月 (2)
2016年8月 (5)
2016年7月 (11)
2016年6月 (2)
2016年5月 (1)
2016年4月 (2)
2016年3月 (2)
2016年2月 (1)

但是它有一个缺点，只要这个连接终止，或者使用了exit命令，容器就会退出后台运行

b:使用docker exec命令

这个命令使用exit命令后，不会退出后台，一般使用这个命令，使用方法如下

```
docker exec -it db3 /bin/sh 或者 docker exec -it d48b21a7e439 /bin/sh
```

db3是后台容器的NAMES,d48b21a7e439是容器的进程ID CONTAINER ID

/bin/sh 是固定写法
它也能进入这个容器

鉴于阿里的面试官问这个问题 Docker常用的命令，有必要对Docker常用的命令做一次梳理：

(1) 安装 docker:

```
yum install -y docker
```

(2) 启动 docker服务：

```
service docker start //启动docker service
```

如果想要关闭docker服务：

```
service docker stop
```

(3) 获取镜像：

```
sudo docker pull NAME[:TAG]  
sudo docker pull centos:latest
```

比如获取tomcat 镜像：

2016年1月 (2)
2015年12月 (2)
2015年9月 (5)
2015年3月 (2)
2014年12月 (1)
2014年11月 (7)
2014年10月 (1)
2014年9月 (1)
2014年7月 (6)
2014年6月 (3)
2014年5月 (8)
2014年4月 (5)
2014年3月 (12)
2014年2月 (25)


```
docker pull tomcat //命令格式: docker pull NAME
```

(4) 罗列所有镜像:

```
docker images
```

(5)、删除镜像, 从本地删除一个已经下载的镜像

```
sudo docker rmi IMAGE [IMAGE...]  
sudo docker rmi centos:latest
```

(5)创建docker容器 并进入 webdemo1:

```
docker run -it --name webdemo1 -p 81:8080 tomcat /bin/bash
```

docker run -i -t <IMAGE_ID> /bin/bash: -i: 标准输入给容器 -t: 分配一个虚拟终端 /bin/bash: 执行bash脚本

注意: name前面是两个-, 端口前面有-p, 后面有个tomcat;

这里我们启动一个交互式的容器, 名字是webdemo1, -p 81:8080表示将容器的8080端口映射到主机的81端口, 这样我们只要访问主机的81端口就可以访问到容器的服务了。如果还想创建其他的容器, 那么名称 要改, 而且不能再挂载到主机的81上了, 可以82上挂载;

(6)启动一个停止的容器:

```
docker start webdemo
```

(7) 罗列所有的docker容器: 包含了启动的Docker和没有启动的容器Docker

```
docker ps -a;
```

(8) 启动容器:

```
docker start webdemo
```

可能端口会被占用, 我们可以先把运行的docker实例停掉, 然后再删掉,

```
docker kill webdemo  
  
docker rm webdemo
```

2014年1月 (6)

2013年12月 (7)

2013年10月 (13)

2010年6月 (1)

相册

IOS(5)

美女(4)

最新评论

1. Re:分布式与集群的区别是什么?

顶楼主个, 写的通俗易懂

--最咸的咸鱼

2. Re:Docker 入门 到部署Web 程序-
(阿里面试常用的docker命令和优点)

写的不错, 无废话

(9) 进入其中一个容器：使用docker exec命令

这个命令使用exit命令后，不会退出后台，一般使用这个命令，使用方法如下

```
docker exec -it webdemo /bin/bash
```

(10)拷贝文件

从主机复制到容器 `sudo docker cp host_path containerID:container_path`

从容器复制到主机 `sudo docker cp containerID:container_path host_path`

请注意，以上这两个命令都是在主机中执行的，不能再容器中执行

```
docker cp /root/software/docker.war webdemo:/
```

```
docker cp demo:/abc.txt /root/software/
```

(10) 在容器中启动tomcat:

```
cd /usr/local/tomcat/bin
```

```
./catalina.sh run
```

(11)记住一个服务器可以有多个容器，但是一个服务器只需要安装一个tomcat,然后每个容器可以挂载到不同的tomcat的端口上面，就是这关系。前台访问：

```
http://192.168.163.128:80/docker //这是第一个容器webdemo
```

```
http://192.168.163.128:81/docker; //这是第二个容器webdemo1
```

Docker的优点：

Docker五大优势：持续集成、版本控制、可移植性、隔离性和安全性

【编者的话】 随着Docker技术的不断成熟，越来越多的企业开始考虑使用Docker。Docker有很多的优势，本文主要讲述了Docker的五个最重要优势，即持续集成、版本控制、可移植性、隔离性和安全性。

--loongchao

3. Re:分布式与集群的区别是什么？

楼主，网上辣鸡文章太多了，是个码农都出来发文章！赶紧删了吧！

--我比光速更快

4. Re:Java探针-Java Agent技术-阿里面试题

学习.

--NewSea

5. Re:Java 类加载机制(阿里面试题)-何时初始化类

请教一个问题可以吗？String a = new ("abc")；可能会产生两个对象吗？一个可能是在编译期生成？另一个是运行期间生成？那么编译期间将“abc”放进类的常量池 只是utf-8 符号引用.....

--天才小熊猫666

阅读排行榜

1. 分布式与集群的区别是什么？ (53430)

对于Docker，应该不需要进行详细的介绍了。它是最火热的开源项目之一，通过在容器中增加一个抽象层（a layer of abstraction），就可以将应用程序部署到容器中。在看似稳定而成熟的场景下，使用Docker的好处越来越多。在这篇文章中，我不谈论Docker是什么或者Docker是怎么工作的，取而代之，我会提出使用这个不断成长的平台的五大好处。

持续部署与测试

Docker在开发与运维的世界中具有极大的吸引力，因为它能保持跨环境的一致性。在开发与发布的生命周期中，不同的环境具有细微的不同，这些差异可能是由于不同安装包的版本和依赖关系引起的。然而，Docker可以通过确保从开发到产品发布整个过程环境的一致性来解决这个问题*Docker容器通过相关配置，保持容器内部所有的配置和依赖关系始终不变。最终，你可以在开发到产品发布的整个过程中使用相同的容器来确保没有任何差异或者人工干预。

使用Docker，你还可以确保开发者不需要配置完全相同的产品环境，他们可以在他们自己的系统上通过VirtualBox建立虚拟机来运行Docker容器。Docker的魅力在于它同样可以让你在亚马逊EC2实例上运行相同的容器。如果你需要在一个产品发布周期中完成一次升级，你可以很容易地将需要变更的东西放到Docker容器中，测试它们，并且使你已经存在的容器执行相同的变更。这种灵活性就是使用Docker的一个主要好处。和标准部署与集成过程一样，Docker可以让你构建、测试和发布镜像，这个镜像可以跨多个服务器进行部署。哪怕安装一个新的安全补丁，整个过程也是一样的。你可以安装补丁，然后测试它，并且将这个补丁发布到产品中。

多云平台

Docker最大的好处之一就是可移植性。在过去的几年里，所有主流的云计算提供商，包括亚马逊AWS和谷歌的GCP，都将Docker融入到他们的平台并增加了各自的支持。Docker容器能运行在亚马逊的EC2实例、谷歌的GCP实例、Rackspace服务器或者VirtualBox这些提供主机操作系统的平台上。举例来说，如果运行在亚马逊EC2实例上的Docker容器能够很容易地移植到其他几个平台上，比如说VirtualBox，并且达到类似的一致性和功能性，那这将允许你从基础设施层中抽象出来。除了AWS和GCP，Docker在其他不同的IaaS提供商也运行的非常好，例如微软的Azure、OpenStack和可以被具有不同配置的管理者所使用的Chef、Puppet、Ansible等。

环境标准化和版本控制

通过上面的讨论，Docker容器可以在不同的开发与产品发布生命周期中确保一致性，进而标准化你的环境。除此之外，Docker容器还可以像git仓库一样，可以让你提交变更到Docker镜像中并通过不同的版本来管理它们。设想如果你因为完成了一个组件的升级而导致你整个环境都损坏了，Docker可以让你轻松地回滚到这个镜像的前一个版本。这整个过程可以在几分钟内完成，如果和虚拟机的备份或者镜像创建流程对比，那Docker算相当快的，它可以让你快速地进行复制和实现冗余。此外，启动Docker就和运行一个进程一样快。

隔离性

Docker可以确保你的应用程序与资源是分隔开的。几个月前，Gartner发表了一篇报告，这份报告说明了运行Docker 容器进行资源隔离的效果和虚拟机（VM）管理程序一样的好，但是管理与控制方面还需要进行完善。

2. No space left on device 解决Linux系统磁盘空间满的办法(48899)
3. Java-线程池专题 (美团面试题)(20960)
4. Id returned 1 exit status"的解决办法(17741)
5. JNDI 和JDBC的区别(16342)

- 评论排行榜
1. 分布式与集群的区别是什么? (12)
2. 择一城终老，遇一人白首(4)
3. MySql数据库连接池专题(3)
4. Web网站的几个并发量级(3)
5. Java探针-Java Agent技术-阿里面试题(3)

- 推荐排行榜

我们考虑这样一个场景，你在你的虚拟机中运行了很多应用程序，这些应用程序包括团队协作软件（例如Confluence）、问题追踪软件（例如JIRA）、集中身份管理系统（例如Crowd）等等。由于这些软件运行在不同的端口上，所以你必须使用Apache或者Nginx来做反向代理。到目前为止，一切都很正常，但是随着你的环境向前推进，你需要在你现有的环境中配置一个内容管理系统（例如Alfresco）。这时候有个问题发生了，这个软件需要一个不同版本的Apache Tomcat，为了满足这个需求，你只能将你现有的软件迁移到另一个版本的Tomcat上，或者找到适合你现有Tomcat的内容管理系统（Alfresco）版本。

对于上述场景，使用Docker就不用做这些事情了。Docker能够确保每个容器都拥有自己的资源，并且和其他容器是隔离的。你可以用不同的容器来运行使用不同堆栈的应用程序。除此之外，如果你想在服务器上直接删除一些应用程序是比较困难的，因为这样可能引发依赖关系冲突。而Docker可以帮你确保应用程序被完全清除，因为不同的应用程序运行在不同的容器上，如果你不在需要一款应用程序，那你可以简单地通过删除容器来删除这个应用程序，并且在你的宿主机操作系统上不会留下任何的临时文件或者配置文件。

除了上述好处，Docker还能确保每个应用程序只使用分配给它的资源（包括CPU、内存和磁盘空间）。一个特殊的软件将不会使用你全部的可用资源，要不然这将导致性能降低，甚至让其他应用程序完全停止工作。

安全性

如上所述，Gartner也承认Docker正在快速地发展。从安全角度来看，Docker确保运行在容器中的应用程序和其他容器中的应用程序是完全分隔与隔离的，在通信流量和管理上赋予你完全的控制权。Docker容器不能窥视运行在其他容器中的进程。从体系结构角度来看，每个容器只使用着自己的资源（从进程到网络堆栈）。

作为紧固安全的一种手段，Docker将宿主机操作系统上的敏感挂载点（例如/proc和/sys）作为只读挂载点，并且使用一种写时复制系统来确保容器不能读取其他容器的数据。Docker也限制了宿主机操作系统上的一些系统调用，并且和SELinux与AppArmor一起运行的很好。此外，在Docker Hub上可以使用的Docker镜像都通过数字签名来确保其可靠性。由于Docker容器是隔离的，并且资源是受限制的，所以即使你其中一个应用程序被黑，也不会影响运行在其它Docker容器上的应用程序。

结语

将云计算一起考虑，上面提到的这些好处能够清楚地证明Docker是一个有效的开源平台。使用Docker的好处越来越多，今天我只想强调这前五大好处。如果你使用了Docker，欢迎分享你的使用案例或者任何你觉得使用Docker带来的好处。

- 参考：[Docker常用命令](#)
- 参考：[安装Docker和下载images镜像和常用Docker命令](#)
- 参考：[docker从容器里面拷文件到宿主机或从宿主机拷文件到docker容器里面](#)
- 参考：[如何通俗解释Docker是什么？](#)

1. 分布式与集群的区别是什么? (35)
2. IO复用,AIO,BIO,NIO,同步, 异步, 阻塞和非阻塞 区别(百度)(7)
3. 择一城终老, 遇一人白首(3)
4. No space left on device 解决Linux 系统磁盘空间满的办法(2)
5. Bat命令学习(2)

参考: [Docker搭建Java Web运行环境](#)

参考: [在Docker中如何利用Tomcat快速部署web应用?](#)

参考: [docker进入后台运行的容器](#)

参考: [Docker五大优势: 持续集成、版本控制、可移植性、隔离性和安全性](#)

分类: Deployment 部署知识

好文要顶

关注我

收藏该文



aspirant

关注 - 20

粉丝 - 119

+加关注

0

0

« 上一篇: Java探针-Java Agent技术-阿里面试题

» 下一篇: ETL技术(Extract-Transform-Load) 数据仓库技术-比如kettle

posted @ 2018-04-13 09:10 aspirant 阅读(1828) 评论(1) 编辑 收藏

评论列表

#1楼 2018-09-05 21:18 loongchao

写的不错, 无废话

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](#)。

**最新IT新闻:**

- 医生对四代Apple Watch赞赏有加 但担心用户可能反应过度
 - “iPhone X下架”登顶微博热搜 网友吐槽：史上寿命最短的手机
 - 刘强东人设崩塌：穷苦出身功成名就 夫妻恩爱感情好
 - 真双卡双待iPhone来了：只在中国卖不是没有原因的
 - 网秦林宇再度指责史文勇撒谎：拉拢收买少数工作人员
- » 更多新闻...



华为全联接大会 | 上海 | 2018.10.10-12

[大会门票+云服务器] 专属套餐0.35折起

**最新知识库文章:**

- 为什么说 Java 程序员必须掌握 Spring Boot ?
 - 在学习中，有一个比掌握知识更重要的能力
 - 如何招到一个靠谱的程序员
 - 一个故事看懂“区块链”
 - 被踢出去的用户
- » 更多知识库文章...