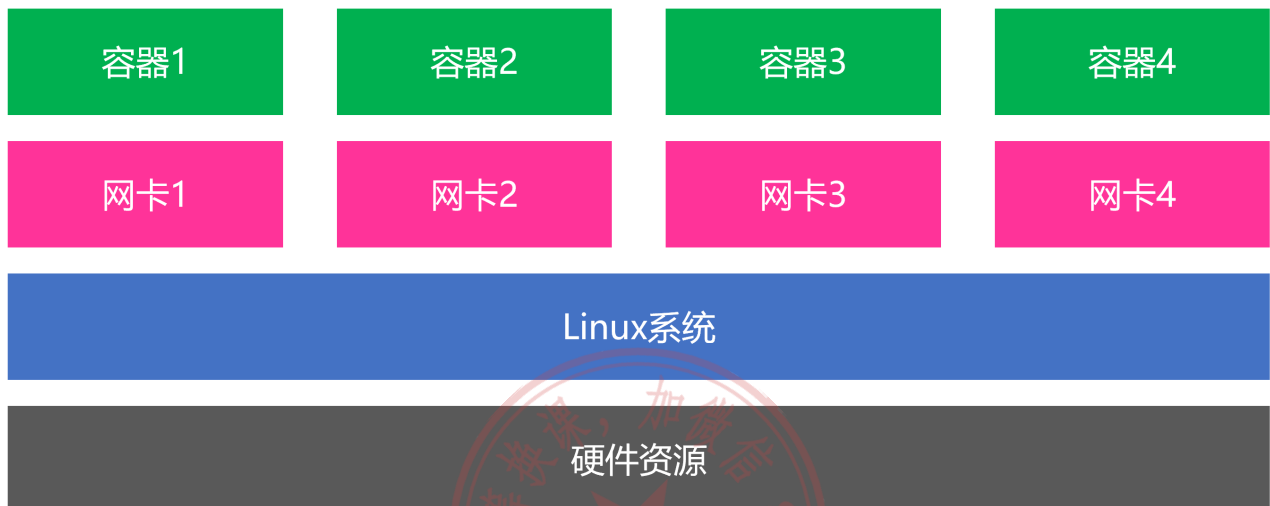


## 4-15 在Docker中安装人脸识别镜像

140.143.132.225:8000/project-1/doc-63

### 一、Docker是什么？

**Docker**是轻量级的虚拟机产品，我们在使用**Docker**的时候，它并不会为每个虚拟化实例（容器）创建完整的虚拟硬件环境，而是为每个实例虚拟化少量的硬件环境（例如网卡）。这些虚拟化实例直接共享使用Linux系统的CPU、内存和硬盘资源。因此说，采用**Docker**的Linux主机，可以轻松创建几十个容器。



**VMware**这种重量级的虚拟机，无法做到这一点。**VMware**虚拟机创建每个实例，都要虚拟化完整的硬件环境，所以在Linux或者Windows系统上面根本无法创建多个这样的虚拟机实例。

### 二、Docker只能使用Linux吗？

所有的**Docker**容器都是建立在Linux系统之上的，所以建议大家在Linux环境中安装**Docker**程序。如果你本地不是Linux系统，可以先通过虚拟机安装Linux，然后在Linux上面安装**Docker**程序。

虽然**Docker**也提供了Windows版本和MacOS版本，但是安装程序都包含了简化版的Linux内核，这个简化版的Linux功能有限，所以无法发挥出**Docker**所有的功能，所以建议大家在Linux系统上面安装**Docker**，才能发挥出**Docker**全部的功能。

### 三、为什么要使用Docker？

使用**Docker**主要有两点原因：

首先，**Docker**容器之间是相互隔离的，我们可以在容器中部署程序的同时，不用担心跟其他容器产生冲突。例如某公司几年前开发的Java程序用的是JDK1.6，但是现在要部署的Java项目用JDK1.8开发的，现在我们无法在服务器上面同时安装JDK1.6和JDK1.8环境。如果我们用上了**Docker**程序，先创建出两个容器，然后分别安装JDK1.6和JDK1.8，接下来分别部署新旧程序。这就能做到在一个服务器上面可以部署若干程序的时候，不用担心发生矛盾冲突。其实你在阿里云和腾讯上面购买的云主机实例，就是这种虚拟化的容器。

其次，**Docker**能为我们提供快速化部署。比如说有人在网上提供了**PHP**镜像，里面包括了**PHP**环境、**Apache**服务器和**MySQL**数据库，我们从网上下载镜像，然后创建容器，就直接可以部署**PHP**项目了，非常的方便。再比如，你自己也可以创建镜像，把要部署的程序和运行环境打成**Docker**镜像，分发给其他人，他们创建容器就可以直接运行程序了。

总结成一句话，**Docker**有毒，用上就会让人上瘾。

## 四、镜像和容器

**Docker**为了最大化的共享资源，减少资源浪费，创造出了镜像技术。镜像是虚拟实例之间相同的部分，差异化的部分叫做容器。比如说一个**JDK**镜像，里面安装了**JDK1.8**，然后我为这个镜像创建出3个容器，这三个容器共享使用**JDK**镜像中的**Java**环境，并且我可以在每个容器中部署不同的**Java**程序。如果**Docker**没有镜像技术，我们只能分别在三个容器中安装**JDK**环境，这就造成了不必要的重复浪费。

需要注意的是，容器只能读取镜像中的数据，不能修改镜像的内容，也就是说，镜像对于容器是只读的。



## 五、关闭SELINUX

**SELINUX**是**CentOS**自带的安全服务，因为晦涩难用，并且跟很多程序冲突，所以强烈建议大家关闭这个服务。

找到 `/etc/sysconfig/selinux` 文件，把其中的 `SELINUX` 设置为 `disabled`，保存文件之后重启**CentOS**系统。

## 六、安装Docker程序

执行下面的指令，稍等片刻，**Docker**程序就安装好了

```
1. yum install docker -y
```

管理**Docker**程序的命令也非常简单，如下：

1. `service docker start`
2. `service docker stop`
3. `service docker restart`

## 七、导入人脸识别镜像

---

1. 把 `face.tar.gz` 文件上传到CentOS系统

2. 把镜像导入Docker环境

1. #导入镜像文件
2. `docker load < face.tar.gz`
3. #查看安装的镜像
4. `docker images`
5. #删除镜像
6. `docker rmi face`

