Docker学习

笔记本:	冲突的修改		
创建时间:	2019/12/10 17:21	**更新时间:**	2019/12/19 22:06
作者:	duanxx9156@163.com		

Docker基础教程

一、Docker基础认识

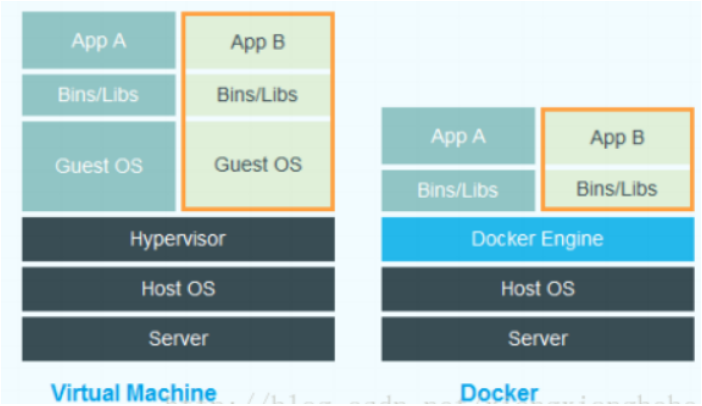
1. 什么是虚拟化

将计算机的各种实体资源，如服务器，网络，内存及存储等经过抽象，转换后呈现出来，新虚拟的部分不受现有资源的架设方式，地域或物理组态所限制，一 般所指的虚拟化资源包括计算机能力以及存储资源。

2. 什么是Docker

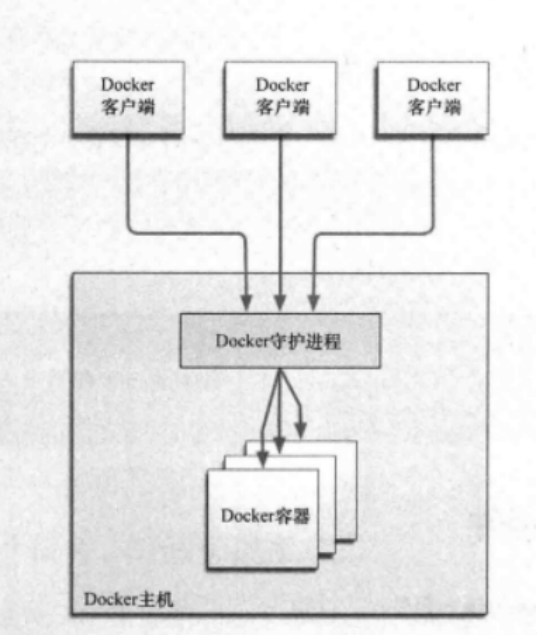
Docker是一个开源项目，目标是实现轻量级的操作系统虚拟化解决方案，其基础是Linux容器等技术。

3. 容器和虚拟机的区别



4. Docker组件

（1）Docker客户端与服务器

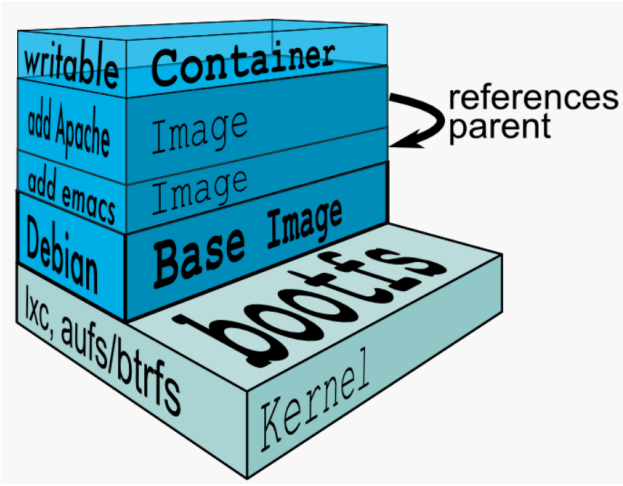


（2）Docker镜像

镜像是一个只读的容器模板，含有启动docker容器所需的文件系统结构及内容。

docker的镜像机制是有层次感的，一个镜像可以放到另一个镜像的顶部。位于下端的为父镜像，以此类推；最底部的镜像可称为基础镜像。

镜像采用分层构建，每个镜像由一系列的镜像层组成, 当需要修改容器内的某个文件时，只对处于最上方的读写层进行变动，不覆盖下面已有文件系统的内容。 当提交这个修改过的容器文件系统为一个新的镜像时，保存的内容仅为最上层读写文件系统中被更新过的文件。



（3）仓库

Docker用Registry来保存用户构建的镜像。Register分为公共和私有两种。Docker公司运营的公共Registry叫做Docker Hub。用户可以在Docker Hub注 册账户，分享并保持自己的镜像。

用户也可以在Docker Hub上保存自己的私有镜像。

（4） 容器

Docker 可以帮用户构建和部署容器，用户只需把自己的应用程序或者服务打包放进容器即可。容器是基于镜像启动起来的，容器中可以运行一个或

者多个进程。我们可以认为，镜像是Docker生命周期中的构建或者打包阶段，而容器则是启动或执行阶段。

二、Docker 安装

安装步骤如下：

（1）从https://download.docker.com/linux/ubuntu/dists/，选择合适的Ubuntu版本，索引至对应路径，选择amd64、armhf、arm64、ppc64el或者s390x，下载想要的Docker-CE version对应的.deb文件。

此处下载了docker-ce_17.03.3_ce-0_ubuntu-xenial_amd64.deb

（2）在linux放置.deb文件的目录下执行：

```
$ sudo dpkg -i docker-ce_17.03.3_ce-0_ubuntu-xenial_amd64.deb
```

（3）测试能否运行docker测试镜像：

```
$ sudo docker run hello-world
```

查看docker版本：

```
$ docker --version
```

（4）安装完成后，使用镜像加速器地址修改配置文件，来加快下载速度。

```
$ sudo mkdir -p /etc/docker
$ sudo tee /etc/docker/daemon.json <<-'EOF'

{
  "registry-mirrors": ["你的镜像加速器地址"]
}
EOF

$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
```

三、安装Nvidia-Docker

Nvidia-Docker为Docker的插件，当镜像支持CUDA，CUDNN时，镜像或容器可使用GPU加速。
安装步骤如下：
（1）卸载 nvidia-docker 1.0 及其他GPU容器：

```
$ sudo docker volume ls -q -f driver=nvidia-docker | xargs -r -I{} -n1 docker ps -q -a -f volume={} | xargs -r docker rm -f
$ sudo apt-get purge -y nvidia-docker
```

（2）添加package repositories

```
$ sudo curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey
| sudo apt-key add -
$ sudo distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
$ curl -s -L https://nvidia.github.io/nvidia-
docker/$distribution/nvidia-docker.list | sudo tee
/etc/apt/sources.list.d/nvidia-docker.list
$ sudo apt-get update
```

（3）安装 nvidia-docker2

```
$ sudo apt-get install -y nvidia-docker2
$ sudo pkill -SIGHUP dockerd
```

四、Docker常用命令

1. **Docker启动与停止**

```
$ sudo systemctl start docker

$ sudo systemctl stop docker

$ sudo systemctl restart docker
# 开机自启动
$ sudo systemctl enable docker
# 帮助信息
$ sudo docker --help
```

2. **查看Docker状态**

```
$ sudo systemctl status docker
```

3. **镜像命令**

```
# 查看本地镜像
$ sudo docker images
# 搜索远程仓库镜像
$ sudo docker search name
# 拉取镜像
$ docker pull name
```

```
# 删除镜像
$ sudo docker rmi name
# 删除所有镜像
$ sudo docker rmi 'docker images -q'
```

4. **容器命令**

```
# 查看正在运行的容器
$ sudo docker ps
# 查看所有容器
$ sudo docker ps -a
# 查看最后一次运行的容器
$ sudo ps -l
# 查看停止的容器
$ sudo docker ps -f status=exited
# 创建容器命令
  # 交互式方式创建容器
  $ sudo docker run -it --name=名称 镜像名称：标签 /bin/bash
   -i 表示运行容器
   -t 表示容器启动后进入命令行
   -v 目录映射，宿主机与与容器同步
   -d 守护式后台运行容器，不会创建后直接进入容
  # 守护式创建容器（略）
# 启动，停止容器
$ sudo docker start
$ sudo docker stop
# 文件拷贝
$ sudo docker cp 文件名称 容器名称：容器目录
# 目录挂载
#创建容器，添加-v参数，后面为  宿主机目录：容器目录，例：
$ docker run -di -v /usr/local/myhtml:/usr/local/myhtml --
name=mycon centos:7
# 创建GPU加速容器
$ sudo docker run -d -it $DEVICES--runtime=nvidia --
privileged=true --name=dxx dxx_image:latest /bin/bash
# 进入容器
$ sudo docker exec -it 名称 /bin/bash
```

四、迁移与备份

1. **将容器保存为镜像**

```
$ sudo docker commit 容器名称 镜像名称
```

2. **镜像备份，即保存为tar文件**

```
$ sudo docker save -o 文件名称 镜像名称
```

3. **将文件恢复为镜像**

```
$ sudo docker load -i 文件
```

五、项目交付完整流程：

方式1：手动安装镜像

```
########################软件安装#############################
# 1. 安装Docker
# 2. 安装Nvidia-Docker
########################镜容构建#############################
# 1.拉取远程对应镜像：cuda10.1,cudnn7.0, ubuntu16.04
$ sudo docker pull nvidia/cuda:10.1-cudnn7-devel-ubuntu16.04
# 2.运行镜像，生成启动GPU容器
$ sudo docker run -d -it $DEVICES--runtime=nvidia --
privileged=true \
--name=dxx dxx_image:latest /bin/bash
# 3. 创建容器工作路径，并导入或指定挂载的文件夹，文件
# 将requirements.txt以及代码接口模型文件夹拷入容器路径
$ sudo docker cp 文件名称 容器名称：容器目录
# 4. 进入容器
$  sudo docker exec -it 名称 /bin/bash
# 5. 更新apt
$ apt-get update
# 6. 安装python
$ apt-get install python3
# 7. 安装pip
$ apt-get install python3-pip
# 8. 更新pip
$ python3 -m pip install --upgrade pip
# 9. pip安装requirements.txt中的包
$ pip install -r requirements.txt
# 10.若报“x86 64-linux-gnu-gcc”错误,则按缺失文件安装
# 11. 保存镜像，导出文件
$ sudo docker commit 容器名称 镜像名称
$ sudo docker save -o 文件名称 镜像名称
$ sudo docker load -i 文件
```

```
###################交付测试#############################
# 1.将公司给的excel数据放入容器路径下
$ sudo docker cp 文件名称 容器名称：容器目录
# 2.创建容器
$ sudo docker run -d -it $DEVICES--runtime=nvidia --
privileged=true \
--name=dxx dxx_image:latest /bin/bash
# 3.进入容器
$ sudo docker exec -it 名称 /bin/bash
# 4.运行脚本DelDataGenerate.py，生成txt测试数据
$ python3 DelDataGenerate.py
# 5.运行测试脚本，PredictRun.py，得到结果
$python3 PredictRun.py --test_type(small_all,
small_besides,big,open)
--pretrained_model='./pretrained_models/small_sample/all/
                                        mtdnn/Bert_base.bin'
# 6.最后输出准确率与正确个数，最后将结果txt拷出
$ sudo docker cp 容器名称：容器目录 文件名称
```