

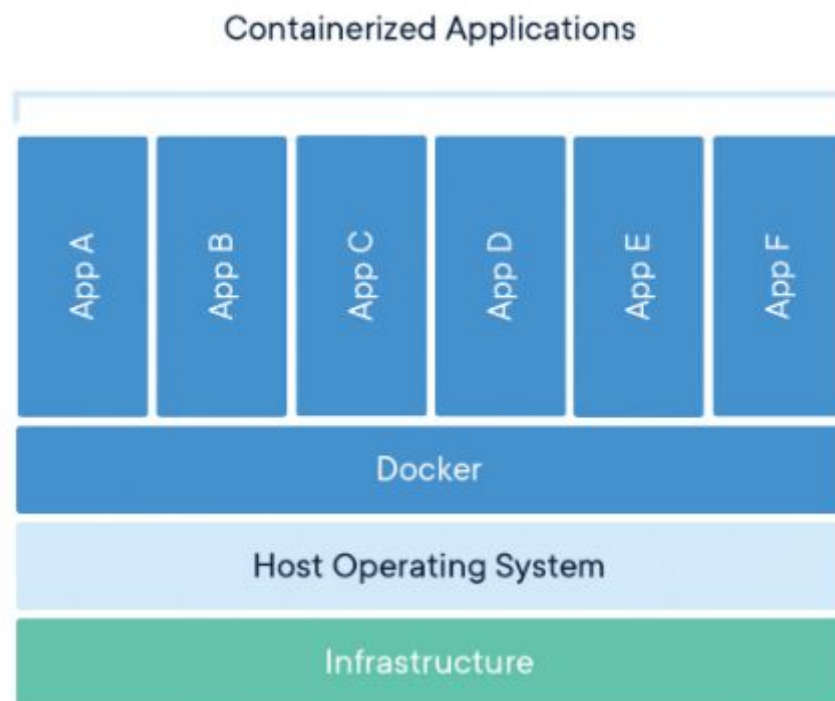
BAT面试14: 谈谈 docker 在深度学习任务中的应用

原创: zhupc

Docker是一种容器技术，它就像一个沙盒把应用程序隔离开来，不管有没有遇到过你至少听到某些应用程序不能兼容，最常见的就是升级某个系统，老版本跟新版本不能兼容，必须把老版本完全卸载掉。比如说oracle服务，如果把oracle安装到物理主机上,如果需要升级那将会比较痛苦。再比如说新手学习各种软件，apache、mysql、Python搞的电脑上乱七八糟的环境，想要重新安装都很痛苦。

有没有一种技术能够把这个环境隔离开？有那就是虚拟机，如果你为了一个小服务单独开一个虚拟机是不是太奢侈？这时候docker出现了，docker是一种容器区别于虚拟机，使用docker你可以**打包、发布、运行任何应用程序**，如果你经常碰到开发环境与生产环境不一致，或者经常需要配置复杂的环境，那你就可以考虑使用它，“它是一处打包到处运行”。再介绍一个使用环境，那就是我们熟悉的云，很常见我们经常去阿里云、腾讯云购买服务器，当我们使用的时候完全就是一个物理主机的状态，与其他服务器完全隔离，这些大厂不会为了一个用户单独开启一个虚拟机，因为太奢侈。而容器就不一样，用多少分配多少，如果虚拟机占用10G资源，而可能有9G是空余出来的，而容器如果占用的10G那么它就可能只空余出来1G或者更少。按照笔者的理解，容器仅仅隔离应用程序，而其他资源是容器间共享的，虚拟机是一种完全霸占的状态（个人理解）。

下图是介绍docker运行机制，可以看出应用程序在相互独立的同时，分享物理机的资源



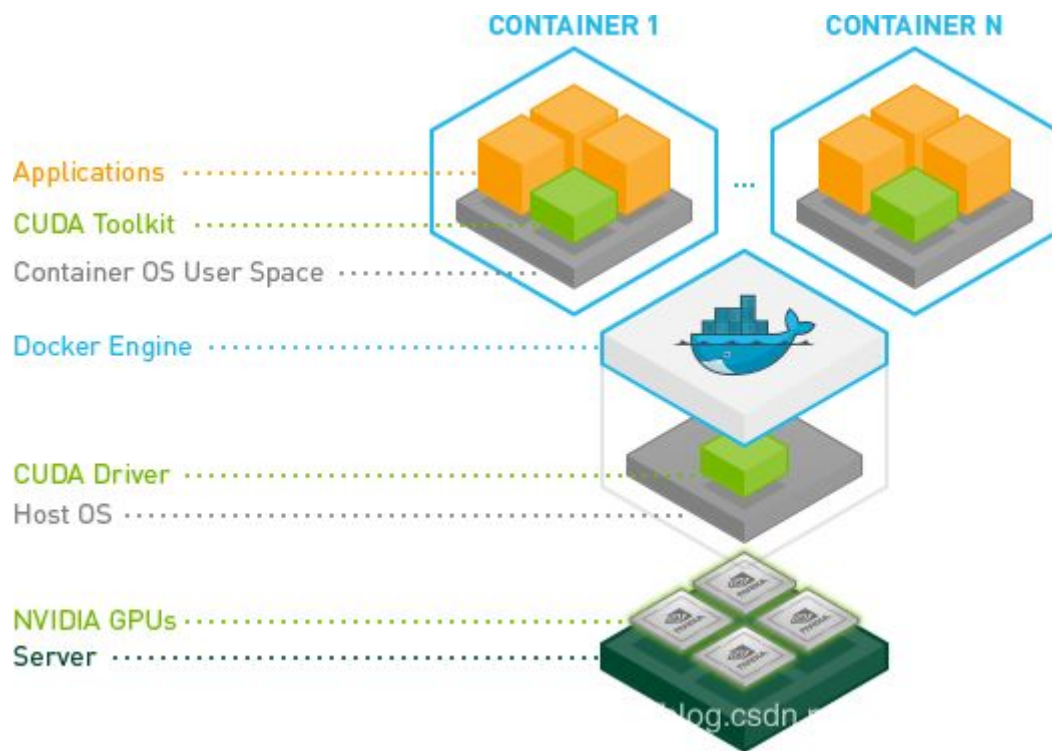
Docker最先是一种开源产品，现在有了社区版和企业版docker-ce与docker-ee。社区版基本已经满足我们。

笔者强烈推荐docker一个原因，那就是可以共享环境，只要有一个人把复杂的环境配置并打包成镜像，类似git一样，commit一个版本，那么其他人就可以直接使用这个环境，而不需要做任何配置，减少人力物力不说，还霸道的使得所有人的生产环境一致。

正是这个原因，我想到了入门机器学习时候，需要配置caffe/tensorflow，并且cuda版本号要对应上，cuda与cudnn版本也要对应上。当时就设想能不能使用docker来解决这个问题，遗憾的是不行，docker无法使用GPU。

但是办法总比困难多，docker又发布出一个插件nvidia-docker,nvidia-docker是基于docker安装的，可以使得深度学习任务也能隔离开，这就意味着，我可以在一台物理主机上跑各种框架的深度学习任务，环境随机配，你开心就好。唯一的限制就是物理机上的cuda版本号要与docker容器中的cuda版本号一致，我觉得这个要求不过分。

下图是nvidia-docker的运行机制，我们可以看到他们直接的层次结构。应用程序共享GPU,且被隔离开。



安装docker这里不再累赘，官网安装地址：

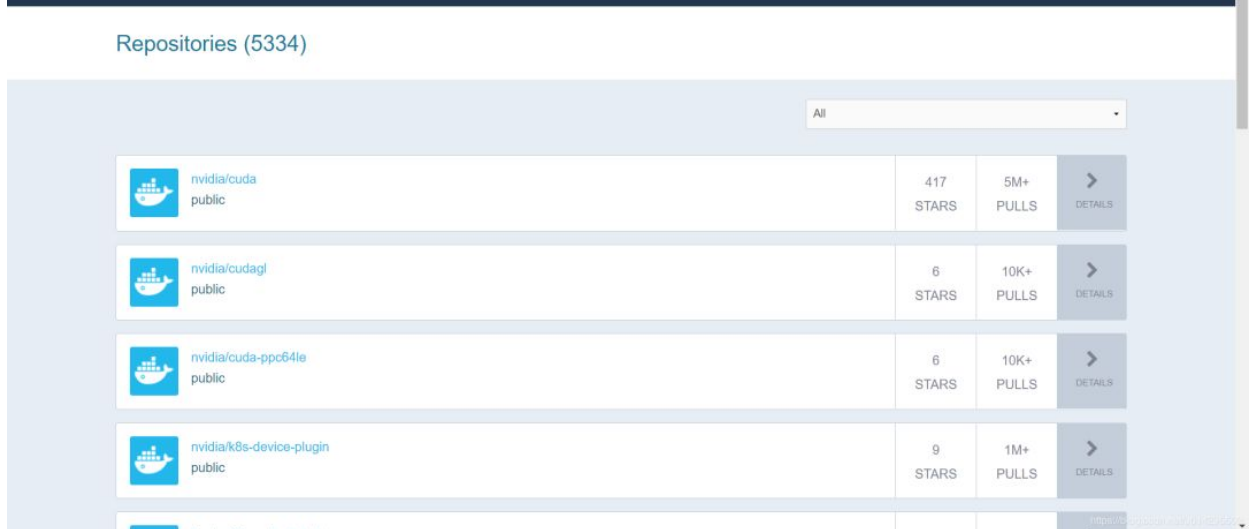
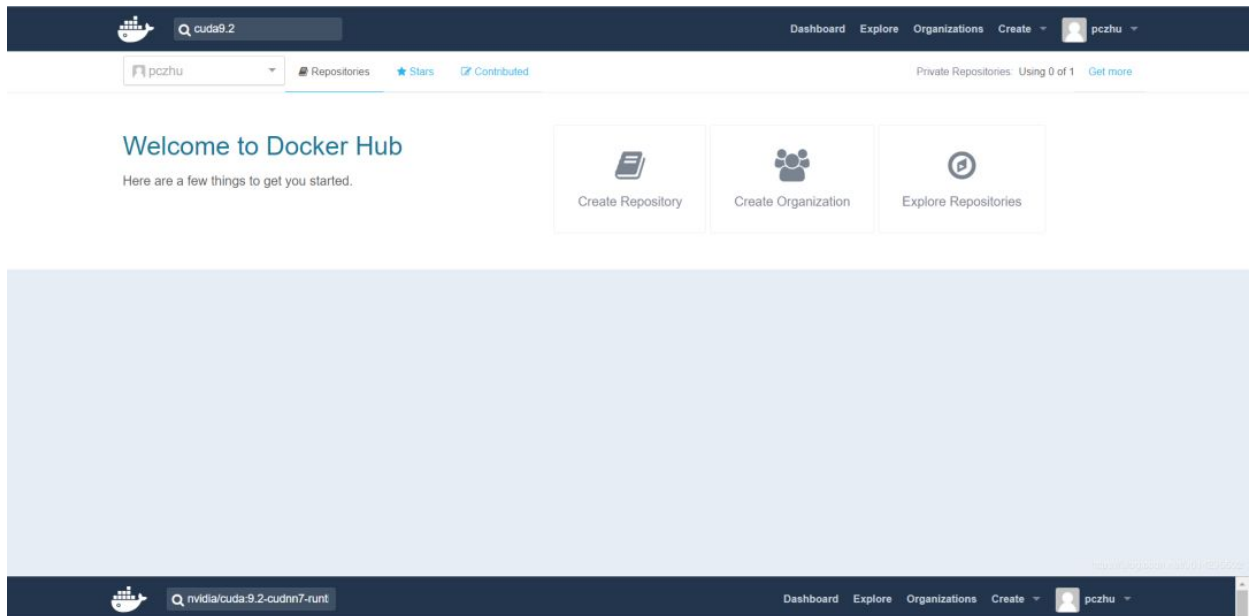
<https://docs.docker.com/install/linux/docker-ce/ubuntu/>。

nvidia-docker 安装官网地址：

<https://github.com/NVIDIA/nvidia-docker>。

现在我需要一个ubuntu16.04+cuda9.2+cudnn7的开发环境，我可以选择下载镜像再去配置各种驱动，安装cuda、安装cudnn。但是我要秀一下更“懒的方式”。

在docker hub里面查询cuda9.2,找到星星最多的一个镜像



里面很多镜像，我找到一个符合我自己要求的。



拉取镜像：`sudo docker pull nvidia/cuda:9.2-cudnn7-runtime-ubuntu16.04`

查看拉取的镜像：`sudo docker ps`

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e84f33f8bc9e	nvidia/cuda:9.2-cudnn7-runtime-ubuntu16.04	"bash"	44 minutes ago	Up 44 minutes		friendly_chatterj

基于这个镜像启动容器：

`sudo nvidia-docker run -ti nvidia/cuda:9.2-cudnn7-runtime-ubuntu16.04 bash`

```
ubuntu@bun:~/NVIDIA_CUDA-9.2_Samples/bin/x86_64/linux/release$ sudo nvidia-docker run -ti nvidia/cuda:9.2-cudnn7-runtime-ubuntu16.04 bash
[sudo] password for ubuntu:
root@8cdf8b079d13:/#
```

现在我已经进来了，完全就是一个干净的开发环境，等下，看看显卡驱动配置好了没。

```
[sudo] password for ubuntu:
root@8cdf8b079d13:/# nvidia-smi
Tue Nov 20 08:34:13 2018

+-----+
| NVIDIA-SMI 396.37                Driver Version: 396.37          |
+-----+-----+
| GPU Name Persistence-M| Bus-Id  Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|   0  GeForce GTX 108...  off          | 00000000:03:00:0 On          |             0%      N/A |
| 29%   22C   P8     9w / 250w|    195MiB / 11178MiB |             0%      Default |
+-----+-----+
|   1  GeForce GTX 108...  off          | 00000000:83:00:0 off         |             0%      N/A |
| 29%   24C   P8     7w / 250w|     2MiB / 11178MiB |             0%      Default |
+-----+-----+

+-----+
| Processes:                      GPU Memory |
|  GPU       PID    Type    Process name                     Usage    |
+-----+
root@8cdf8b079d13:/#
```

<https://blog.csdn.net/u014296502>

开心的不行，如果你对版本没有这个高的要求，那我们来github里面找找看看还有什么镜像，搜索 tensorflow.

The screenshot shows the GitHub repository for tensorflow/tensorflow. It highlights two Docker commands for running containers:

- Start CPU only container:** `$ docker run -it -p 8888:8888 tensorflow/tensorflow`
- Start GPU (CUDA) container:** `$ docker run --runtime=nvidia -it -p 8888:8888 tensorflow/tensorflow:latest-gpu`

Both commands are highlighted with red boxes. The page also shows the Docker Pull Command as `docker pull tensorflow/tensorflow` and the Owner as tensorflow.

GPU与CPU任你选，很开心有没有？好了，docker在深度学习中的应用就介绍到这，安装配置细节网络一大堆，笔者已经在使用docker解放劳动力的兴奋状态中无法自拔，顺便对windows用户用安慰一句，nvidia-docker不支持windows，但是好消息是，其他的都支持。

感谢zhupc的原创分享，总结的很不错，相信对大家大有裨益，查看更多BAT面试系列，请公众号界面点击：**资料->BAT面试**。



Python与机器学习算法频道，

230篇原创，8类技术文章，

更有精华资料，各种福利活动，问答社区，等你来。