# Docker入门

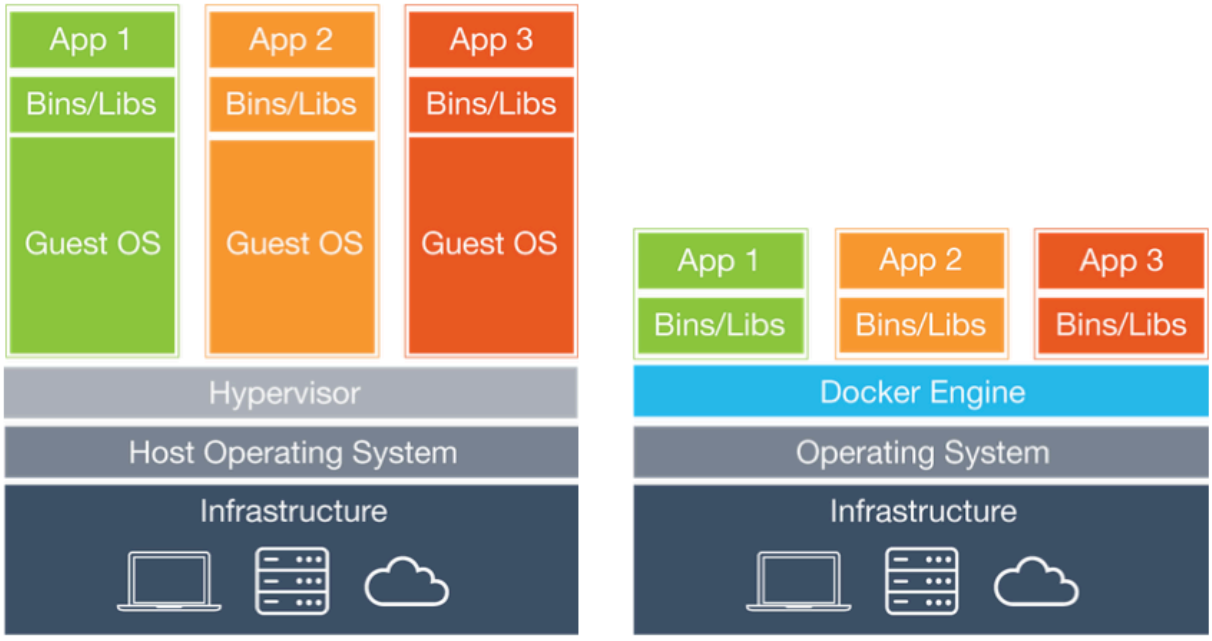
## Docker简介

软件开发中最为麻烦的事情可能就是配置环境了。由于用户使用的操作系统具有多样性，即便使用跨平台的开发语言（如Java和Python）都不能保证代码能够在各种平台下都可以正常的运转，而且可能在不同的环境下我们的软件需要依赖的其他软件包也是不一样的。

那么问题来了，我们再安装软件的时候可不可以把软件运行的环境一并安装课？也就是说在安装软件的时候，我们是不是可以把原始环境一模一样地复制过来呢？

虚拟机（virtual machine）就是带环境安装的一种解决方案，它可以在一种操作系统里面运行另一种操作系统，比如在Windows系统里面运行Linux系统，在macOS上运行Windows，而应用程序对此毫无感知。使用过虚拟机的人都知道，虚拟机用起来跟真实系统一模一样，而对于虚拟机的宿主系统来说，虚拟机就是一个普通文件，不需要了就删掉，对宿主系统或者其他的程序并没有影响。但是虚拟机通常会占用较多的系统资源，启动和关闭也非常的缓慢，总之用户体验没有想象中的那么好。

Docker属于对Linux容器技术的一种封装，它提供了简单易用的容器使用接口，是目前最流行的 Linux 容器解决方案。Docker将应用程序与该程序的依赖打包在一个文件里面，运行这个文件，就会生成一个虚拟容器。程序在这个虚拟容器里运行，就好像在真实的物理机上运行一样。有了Docker就再也不用担心环境问题了。




目前，Docker主要用于几下几个方面：

1. 提供一次性的环境。

2. 提供弹性的云服务（利用Docker很容易实现扩容和收缩）。
3. 实践微服务架构（隔离真实环境在容器中运行多个服务）。

## CentOS下的安装和使用

下面的讲解以CentOS为例，使用Ubuntu、macOS或Windows的用户可以通过点击链接了解这些平台下如何安装和使用Docker。

1. 在CentOS下使用yum安装Docker并启动。

```
yum -y install docker-io

systemctl start docker
```

2. 检视Docker的信息和版本。

```
docker version
docker info
```

3. 运行Hello-World项目来测试Docker。第一次运行时由于本地没有hello-world的镜像因此需要联网进行下载。

```
docker run hello-world
```

也可以先用下面的命令下载镜像，然后再来运行。

```
docker pull <name>
```

4. 运行镜像文件。

```
docker run <image-id>
docker run -p <port1>:<port2> <name>
```

6. 查看镜像文件。

```
docker image ls
docker images
```

7. 删除镜像文件。

```
docker rmi <name>
```

8. 查看正在运行容器。

```
docker ps
```

9. 停止运行的容器。

```
docker stop <container-id>
docker stop <name>
```

对于那些不会自动终止的容器，就可以用下面的方式来停止。

```
docker container kill <container-id>
```

## Docker实战

Docker的使用肯定不止上面这点东西，但是有了这些知识之后，我们已经可以开始感受Docker的强大之处。下面我们就基于Docker来搭建HTTP服务器（Nginx）环境。

```
 docker container run -d -p 80:80 --rm --name mynginx nginx
```

> *说明：上面的参数-d表示容器在后台运行；-p是用来映射容器的端口到宿主机的端口；--rm表示容器停止后自动删除容器，例如通过docker container stop mynginx以后，容器就没有了；--name是自定义容器的名字。*

如果需要将自己的页面部署到Nginx上，可以使用容器的拷贝命令将当前文件夹下所有的文件和文件夹拷贝到容器的指定目录中。当然也可以从容器中拷贝文件到我们指定的路径下。

```
docker container cp ./* mynginx:/usr/local/nginx/html
```

如果不愿意拷贝文件也可以将文件夹映射到Nginx保存页面文件的目录。

```
docker container run -d -p 80:80 --rm --name mynginx --volume
"$PWD/html":/usr/share/nginx/html nginx
```