ASP.NET Core Docker部署

2016-07-12 savorboard [dotNET跨平台](javascript:void(0);)

**前言**

在前面文章中，介绍了 ASP.NET Core在 macOS，Linux 上基于Nginx和Jexus的发布和部署，本篇文章主要是如何在Docker容器中运行ASP.NET Core应用程序。

[Asp.Net Core 发布和部署（ MacOS + Linux + Nginx ）](http://mp.weixin.qq.com/s?__biz=MzAwNTMxMzg1MA==&mid=2654067730&idx=2&sn=2cda7a7f0d82d7f2bfebb63ddc09737b&scene=21#wechat_redirect)

[Asp.Net Core 发布和部署（Linux + Jexus ）](http://mp.weixin.qq.com/s?__biz=MzAwNTMxMzg1MA==&mid=2654067730&idx=1&sn=1ac40760cc0273b7b0df0a0dbc6b45bb&scene=21#wechat_redirect)

**目录**

* Docker 常用命令
* Docker 安装Asp.Net Core镜像
* 使用Dockerfile部署

**Docker 常用命令**

docker info  
检查Docker的安装是否正确，如果没有找到此命令，则表示Docker没有正确安装

docker pull busybox  
拉取一个预建的镜像

sample\_job=$(docker run -d busybox /bin/sh -c "while true; do echo Docker; sleep 1; done")  
以后台进程的方式运行hello docker  
sample\_job命令会隔一秒打印一次Docker，使用Docker logs可以查看输出。如果没有起名字，那这个job会被分配一个id，以后使用命令例如Docker logs查看日志会变得比较麻烦。

docker logs $sample\_job  
运行Docker logs命令来查看job的当前状态：

docker help  
所有Docker命令

docker stop $sample\_job  
停止名为sample\_job的容器

docker restart $sample\_job  
重新启动该容器

docker stop $sample\_job docker rm $sample\_job  
如果要完全移除容器，需要将该容器停止，然后才能移除

docker commit $sample\_job job1  
将容器的状态保存为镜像

docker images  
令查看所有镜像的列表

**下面分享一些我经常使用的一些命令：**

移除所有的容器和镜像，用一行命令大扫除：  
docker kill $(docker ps -q) ; docker rm $(docker ps -a -q) ; docker rmi $(docker images -q -a)

仅仅想删除所有的容器，你可以运行如下命令：  
docker kill $(docker ps -q) ; docker rm $(docker ps -a -q)

清除名称为none的镜像，一般都是下载一般失败的残留：

docker ps -a | grep "Exited" | awk '{print $1 }'|xargs docker stop

docker ps -a | grep "Exited" | awk '{print $1 }'|xargs docker rm

docker images|grep none|awk '{print $3 }'|xargs docker rmi

清除单个镜像：  
docker rmi -f <image id>

**Docker 安装Asp.Net Core镜像**

有关Docker的安装请参考官网：  
https://www.docker.com/products/docker

到这里，假设你的Docker已经安装完成，以下环境为 Liunx（Ubuntu 15.02）。

打开 https://hub.docker.com 右上角搜索dotnet，选择第一个搜索结果microsoft/dotnet，点进去如下图：

得到命令了吧，切换到命令行窗口运行以下命令来安装asp.net core 镜像。

sudo docker pull microsoft/dotnet

什么？ 没响应，速度慢？ 你不会买个高速VPN啊～  
什么？ 没钱？ 你不会找一个国内的Docker镜像站啊～ DaoCloud镜像啊，阿里云Docker镜像啊。。自己动手找啊。。

镜像下载完成之后，输入sudo docker iamges我们检查一下：

至此，准备工作已经全部完成了

**使用 Dockerfile 部署**

Dockerfile 是什么东西，Dockerfile有什么用，自行百度。

博主直接教你怎么写吧：

总不能一直用5000端口吧，这次我们使用60000端口

1、程序命令行切换到publish文件目录中。  
2、sudo touch Dockerfile 名称不要写错了，没有任何后缀名,来新建一个Dockerfile文件。  
3、sudo vim Dockerfile 使用Vim来编辑Dockerfile。如果Vim不会用就用gedit吧。  
4、输入以下内容：

Dockerfile

#基于 `microsoft/dotnet:1.0.0-core` 来构建我们的镜像FROM microsoft/dotnet:1.0.0-core#拷贝项目publish文件夹中的所有文件到 docker容器中的publish文件夹中  COPY . /publish#设置工作目录为 `/publish` 文件夹，即容器启动默认的文件夹WORKDIR /publish#设置Docker容器对外暴露60000端口EXPOSE 60000#使用`dotnet HelloWebApp.dll`来运行应用程序CMD ["dotnet", "HelloWebApp.dll", "--server.urls", "http://\*:60000"]

使用:wq来保存并退出 (gedit无视掉 )。

构建镜像

运行 docker build -t hellowebapp:1.0 . 来构建一个镜像

运行构建的镜像

使用docker run --name hellowebapp -d -p 60000:60000 hellowebapp:1.0

打开浏览器测试一下：

**下一篇怎么在 Linux 为 dotnet 创建守护进程(Linux 部署必备知识)，已经写好了，明天发。**