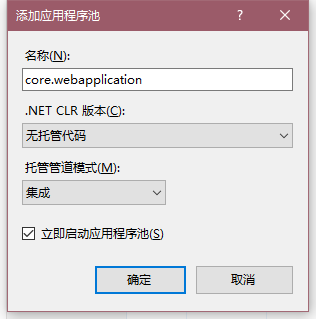
## 部署Asp.net Core Web程序到IIS

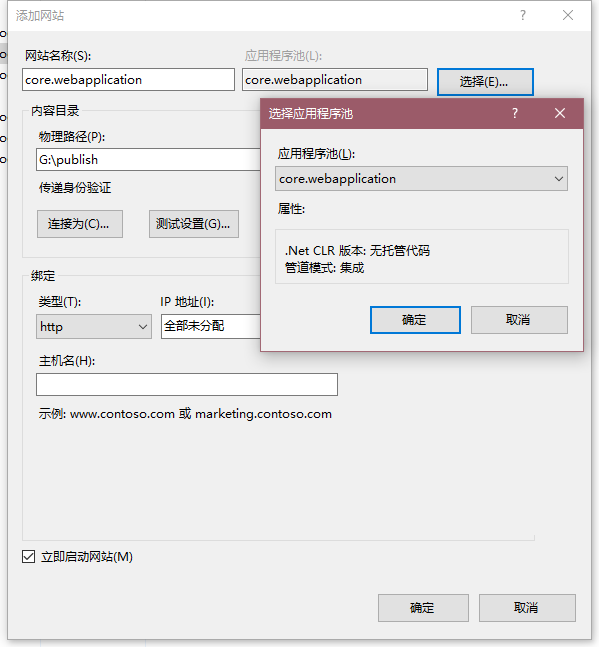
1:创建一个最简单的Asp.Net Core Web项目->右键->发布到一个文件夹中

2:配置IIS

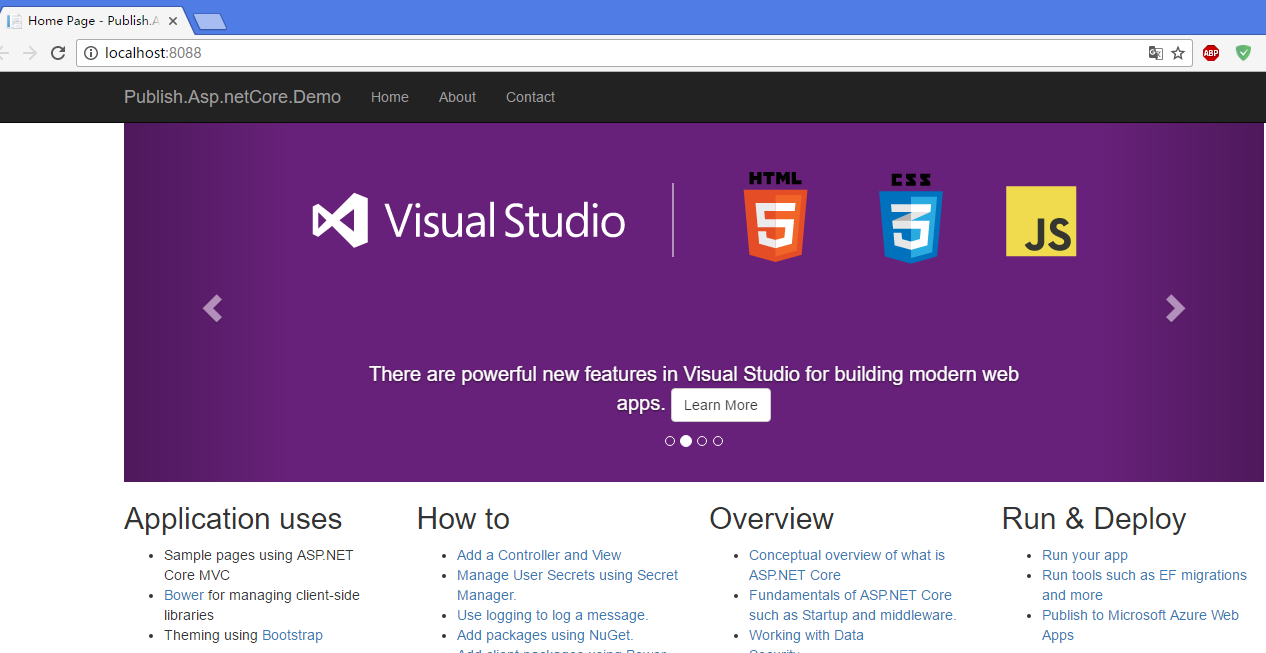
2.1 新建一个无托管的应该程序池



2:新建一个站点，应该程序池选择刚才新建的那个无托管代码的core.webapplication

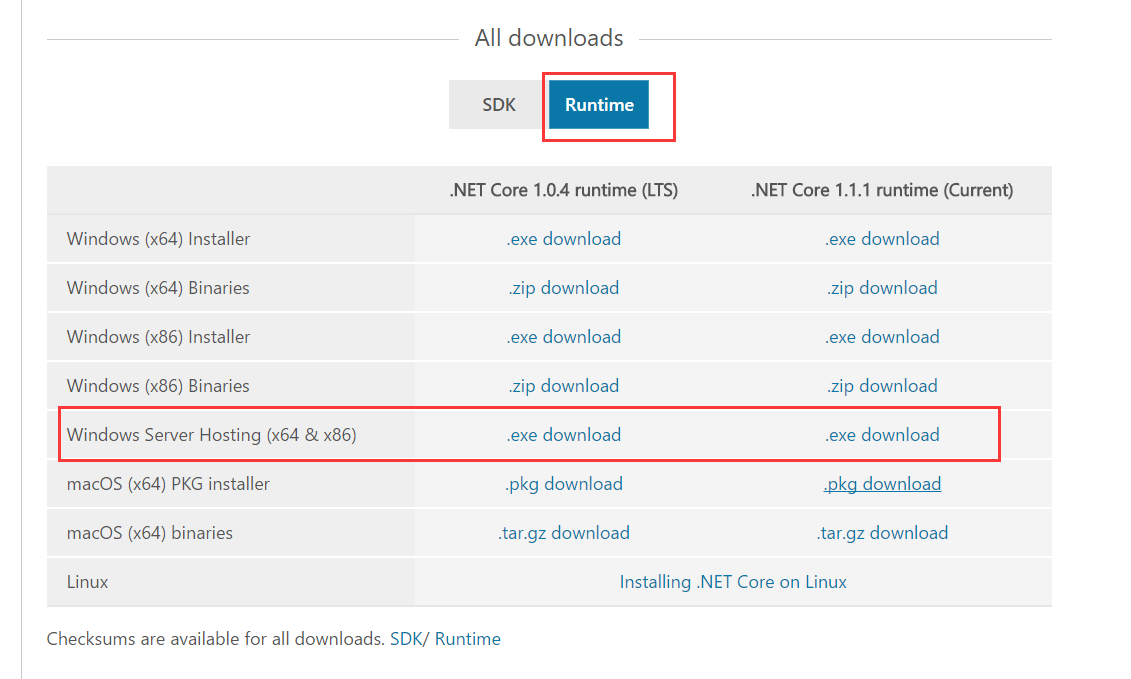


配置好文件路径和端口就Ok了。

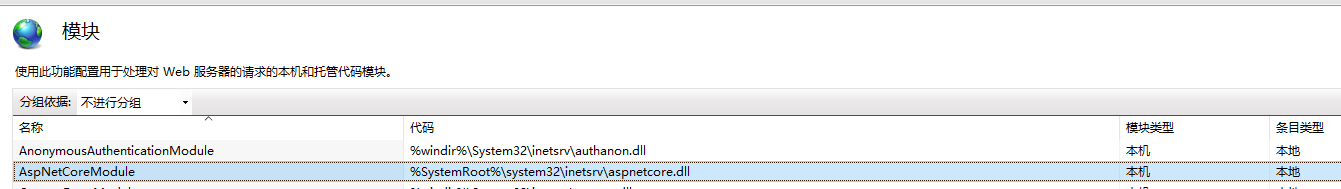


除了安装SDK之后，如果要在IIS中部署.netcore程序，还需要安装WindowsServerHosting

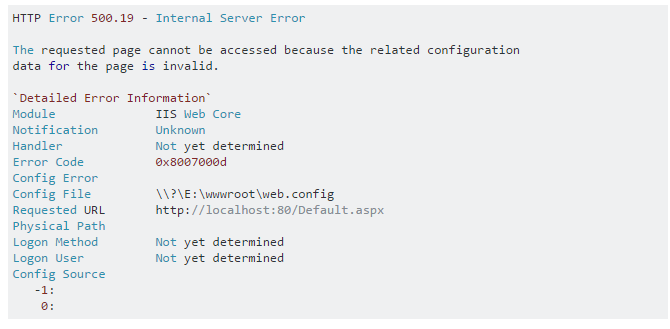
<https://www.microsoft.com/net/download/core#/runtime>



安装成功之后，在IIS的模块中能够看到AspnetcoreModule



如果不安装这个WindowsServerHosting，在IIS中部署是它会提示以下错误



安装WindowsServerHosting后即可解决

## 部署Asp.net Core Web程序到Ubuntu 16.04

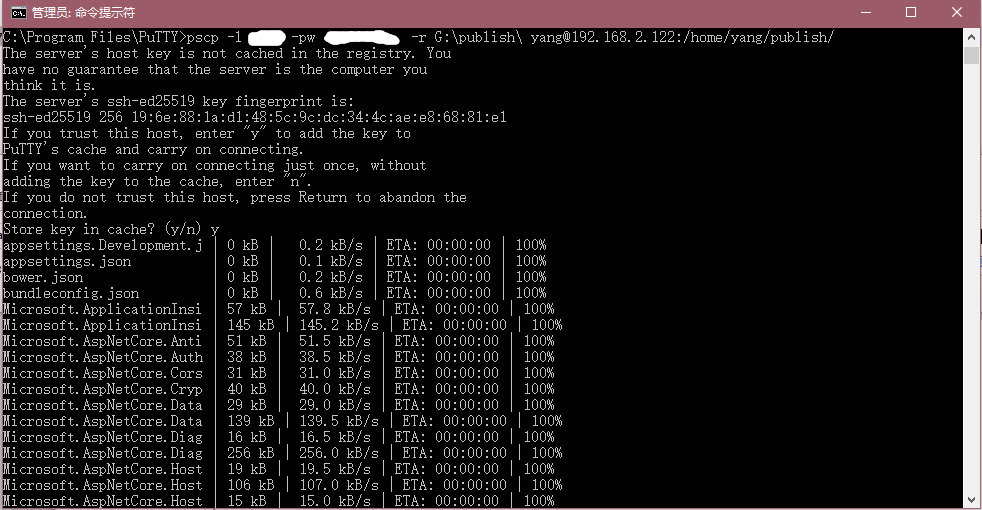
1：程序仍然使用上一步中的发布出来的文件

2：将windows中的publish文件夹拷贝到ubuntu上面

2.1 安装putty,cmd中切换到安装目录

运行命令

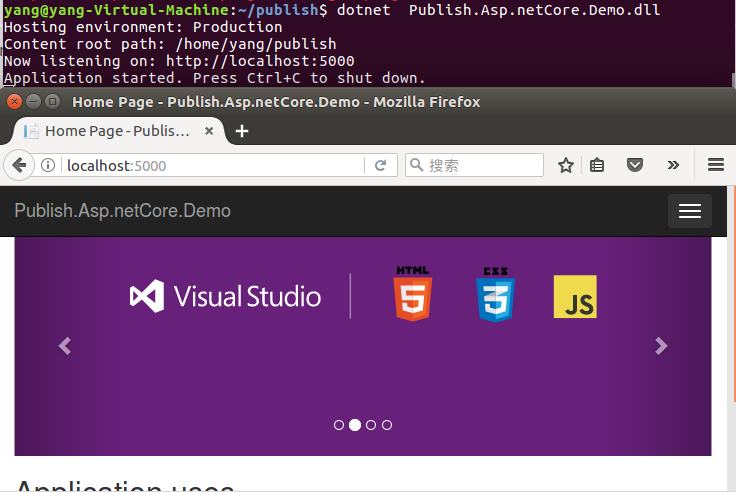
pscp -l yang -pw cms2016... -r G:\publish\ [yang@192.168.2.122:/home/yang/publish/](mailto:yang@192.168.2.122:/home/yang/publish/)



3：在ubuntu 16.04 上面安装.net core sdk

具体方法，参考官方文档<https://www.microsoft.com/net/download/linux>

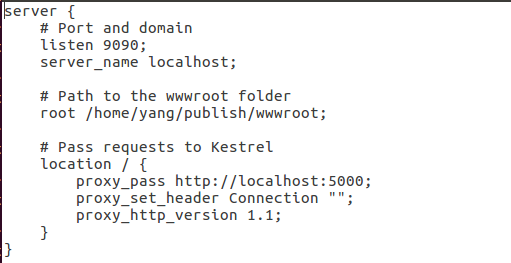
4：切换到home/publish 运行命令dotnet Publish.Asp.netCore.Demo.dll 就可以将web程序启动起来了<http://localhost:5000>



5:安装nginx

sudo apt-get install nginx

5.1 配置nginx



#重启服务

sudo service nginx restart

*浏览*

<http://localhost:9090>

*注意事项：1:端口一定不能被其他程序占用，否则将无法启动nginx*

*2:通过<http://localhost:9090> 访问系统时,仍然需要*dotnet Publish.Asp.netCore.Demo.dll

否则访问<http://localhost:9090> 时会报错502 bad gateway

6:安装supervisor

sudo apt-get install supervisor

配置supervisor

安装好 supervisor 以后，下面就来配置。

定位到 /etc/supervisor/conf.d/ 文件夹，添加一个PublishDemo.conf 文件。

sudo gedit PublishDemo.conf

配置如下

[复制代码](javascript:void(0);)

[program:PublishDemo]

command=/usr/bin/dotnet /home/yang/publish/ Publish.Asp.netCore.Demo.dll

directory=/home/yang/publish

autostart=true

autorestart=true

stderr\_logfile=/var/log/ PublishDemo.err.log

stdout\_logfile=/var/log/ PublishDemo.out.log

environment=ASPNETCORE\_\_ENVIRONMENT=Production

user=root

stopsignal=INT

[复制代码](javascript:void(0);)

启动supervisor

sudo service supervisor restart

输入<http://localhost:9090> 就可以访问系统了

这样我们已经停止掉了dotnet Publish.Asp.netCore.Demo.dll,它仍然能够访问，主要就是supervisor起的作用。

Ubuntu重启之后，它仍然能够正常访问。

参考文档:

IIS中需要配置相关的Module,比如：AspNetCoreModule。<http://www.cnblogs.com/wangjieguang/p/core-iis.html>

<http://www.cnblogs.com/Dream618/p/5618787.html>

http://www.cnblogs.com/linezero/p/aspnetcoreubuntu.html

## 部署Asp.net Core Web程序到Docker

### Ubuntu 16.04上安装docker

查看ubuntu16.04 是32位还是64位，只有64位的才可以安装docker

getconf LONG\_BIT

1:安装docker

sudo apt-get install docker.io

安装完成后可以使用命令查看docker版本 docker –version

2:启动docker服务

Systemctl start docker.service

3:配置开机启动

Systemctl enable docker.service

4:查看docker是否运行成功,我们可以使用命令

docker images 看docker里面有那些镜像，结果报错

error:

Cannot connect to the Docker daemon. Is the docker daemon running on this host?

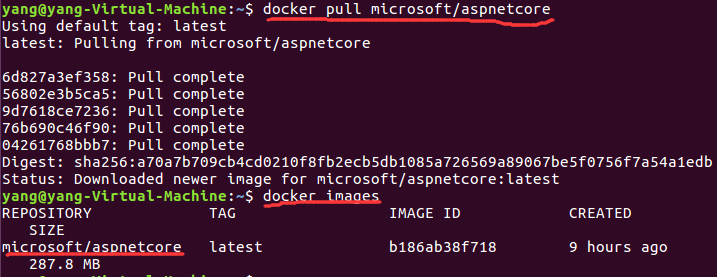
运行命令 sudo gpasswd –a yourUserName docker,再重启ubuntu就可以了



5.拉取.netcore镜像,

docker pull Microsoft/aspnetcore

拉取之后使用docker images查看.net core镜像是否拉取成功



6.新建一个文件夹dockerCoreApp,将上次publish文件夹的内容全部拷贝到此文件夹中

.这个文件夹和publish在同一个目录 /home/yang/下面

7.新建Dockfile

进入到dockerCoreApp目录,sudo gedit Dockerfile

FROM docker.io/microsoft/aspnetcore

#这句可以不要，因为我们已经手动将pushlish中的的文件拷到dockerCoreApp目录下了

#COPY ./publish ./dockerCoreApp

WORKDIR ./dockerCoreApp

#docker容器暴露出来的端口

EXPOSE 8080

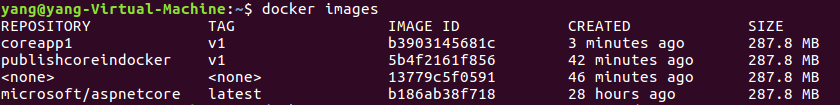
CMD ["dotnet", "Publish.Asp.netCore.Demo.dll"]

8.打包镜像

docker build –t coreApp1:v1 .

查看镜像是否打包成功

docker images



docker rmi publishcoreindocker(或者publishcoreindocker:v1) 可以删除不需要的镜像

9.启动容器

#将ubuntu的8088商品和docker的8088映射起来

docker run -it –d –p 8088:8088 coreapp1:v1

以下命令也可以启动窗口

docker run –name coreapp1 –p 8088:8088 –d coreapp1:v1

通过<http://localhost:8088>就可以访问系统了

我们还可以通过命令

docker ps 查看正在运行的镜像

.