How to install and run a simple Asp.Net 5 Application in a Docker Container

Preface

This is my first time to write this technical document in English, maybe it has some grammar or explanation errors, pointed out and tell me, thank you from the bottom of my heart.

Let's proceed with our topic. As we know, .Net technology comes from Microsoft, in early period, this technology only can be used in Windows, but as the Open Source respective is becoming more and more popular, Microsoft changed attitude towards it. With the help of Mono, .Net Application can be run not only on Windows but also on Linux and Mac OS.

Docker is a small and convenience cloud compute framework feverish to developers. It puts forward a concept of container and provides a simple mechanism to run and manage each container. For instance, if we want to run a hello-word application in a Docker container, we just type ***docker run hello-word*** in terminal, the ***hello-word*** is an image will be run (or pull & run) in a container. Microsoft has seen the prospect. They built an image called ***microsoft/aspnet*** in Docker Hub which will be helpful to deploy our .Net application on Docker.

Some methods reference to

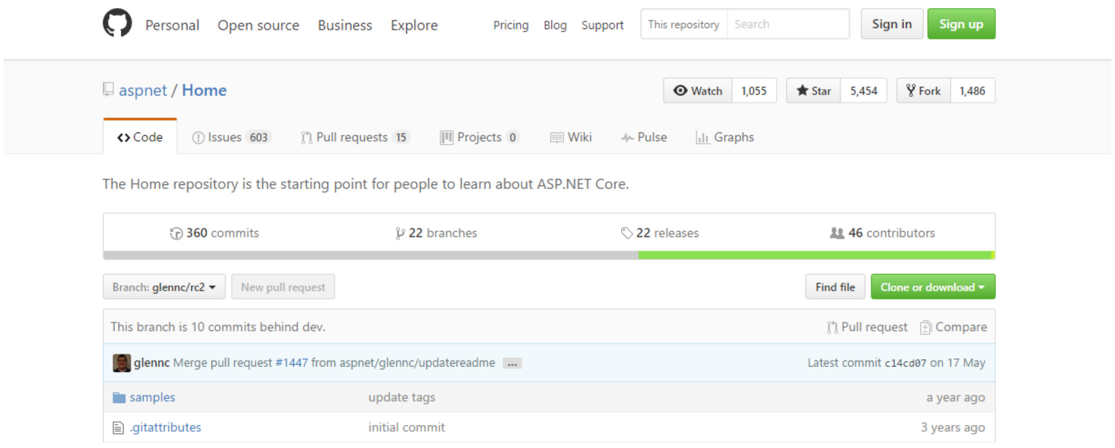
http://www.open-open.com/lib/view/open1422492505689.html

Martin Huang

Preparation

Download aspnet Resource

Type ***https://github.com/aspnet/Home/tree/glennc/rc2*** into your browser address bar, press Enter



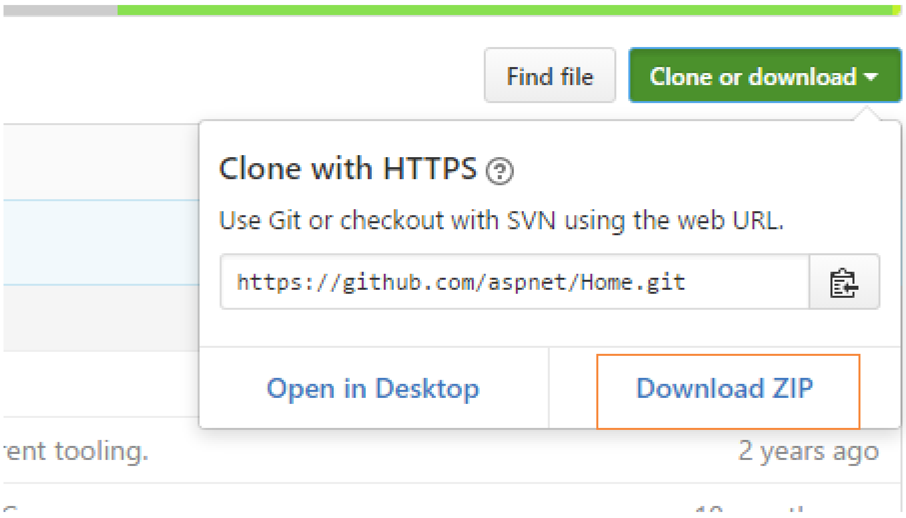
The Home repository is the starting point for people to learn about ASP.NET Core.

This branch is 10 commits behind dev.

glennc Merge pull request #1447 from aspnet/glennc/updatereadme … Latest commit c14cd07 on 17 May

samples	update tags	a year ago
.gitattributes	initial commit	3 years ago

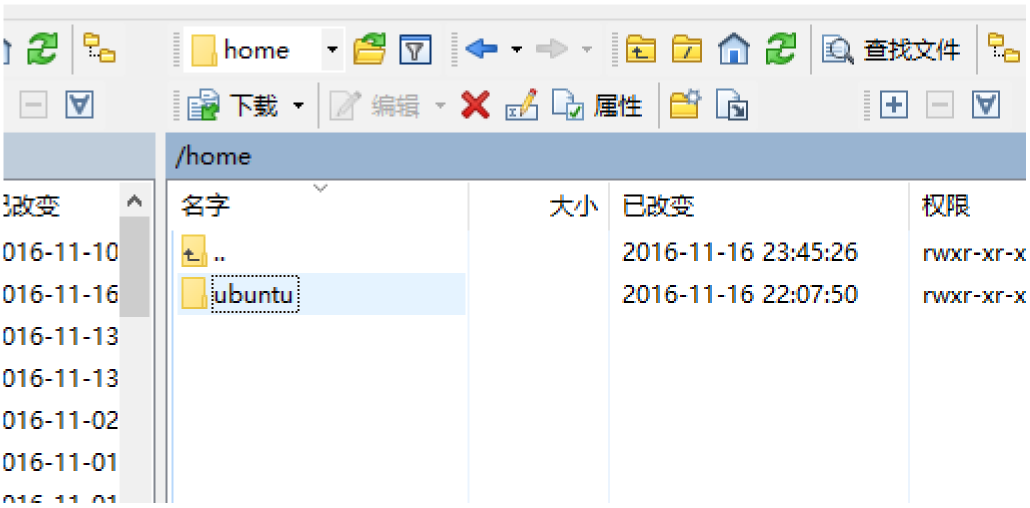
Then chose ***Clone or download*** and click ***Download ZIP***



Wait a moment until the download task finished.

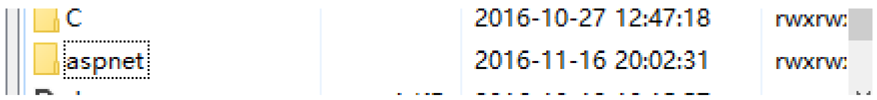
Upload the source into your Docker Server

Log in your Docker Server by means of the *WinSCP*



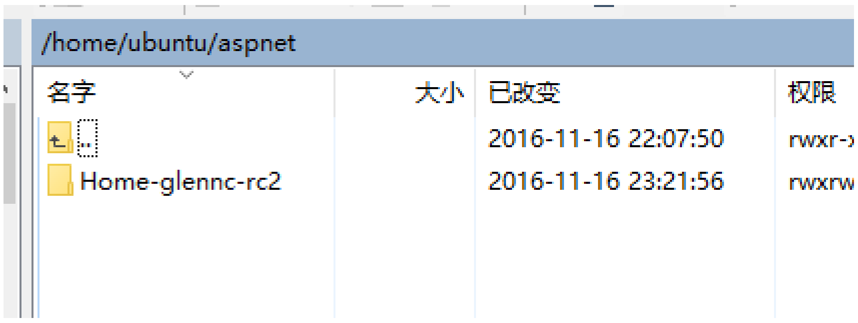
名字	大小	已改变	权限
..		2016-11-16 23:45:26	rwxr-xr-x
ubuntu		2016-11-16 22:07:50	rwxr-xr-x

Then, go into *ubuntu* directory, create a folder named *aspnet* at the same time.



C	2016-10-27 12:47:18	rwxrw:
aspnet	2016-11-16 20:02:31	rwxrw:

Next, extract the file *Home-glennc-rc2.zip* we downloaded just now to the Desktop and drag the folder *Home-glennc-rc2* into *aspnet* which in our Docker Server



```
/home/ubuntu/aspnet
名字                 大小  已改变                 权限
..                         2016-11-16 22:07:50  rwxr-
Home-glennc-rc2            2016-11-16 23:21:56  rwxrw
```

Install

Pull the *microsoft/aspnet:1.0.0-rc1-update1-coreclr* image

Type the Docker Command below into your Putty terminal

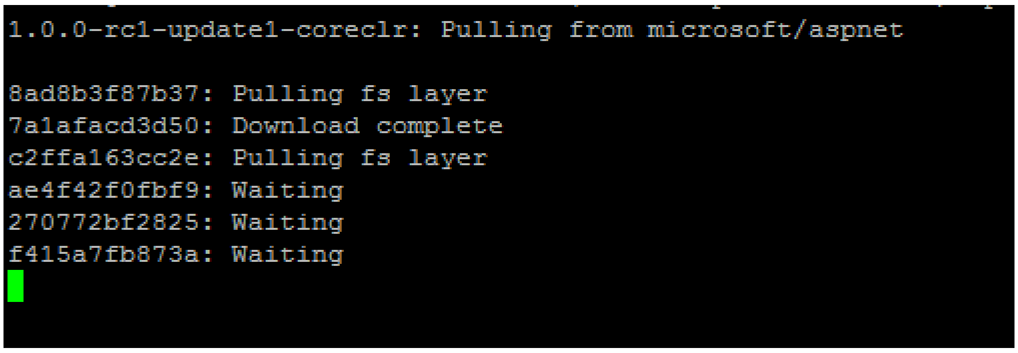
sudo docker pull microsoft/aspnet:1.0.0-rc1-update1-coreclr

Note:

The image will be pulled SLOWLY because of the Chinese Internet.

Be patient!

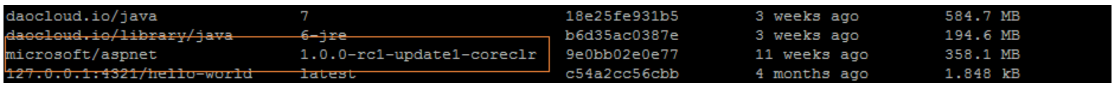
You may see the terminal like the picture below.



```
1.0.0-rc1-update1-coreclr: Pulling from microsoft/aspnet

8ad8b3f87b37: Pulling fs layer
7a1afacd3d50: Download complete
c2ffa163cc2e: Pulling fs layer
ae4f42f0fbf9: Waiting
270772bf2825: Waiting
f415a7fb873a: Waiting
```

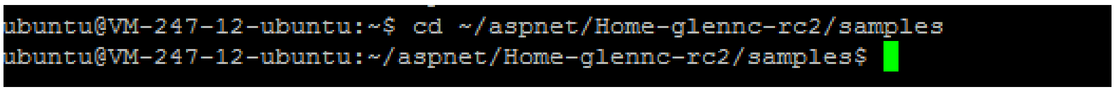
After pulling successfully, you can use the command ***docker images*** to see the ***microsoft/aspnet*** image.



```
daocloud.io/java                7                            18e25fe931b5   3 weeks ago    584.7 MB
daocloud.io/library/java        6-jre                        b6d35ac0387e   3 weeks ago    194.6 MB
microsoft/aspnet                1.0.0-rc1-update1-coreclr    9e0bb02e0e77   11 weeks ago   358.1 MB
127.0.0.1:4321/hello-world      latest                       c54a2cc56cbb   4 months ago   1.848 kB
```

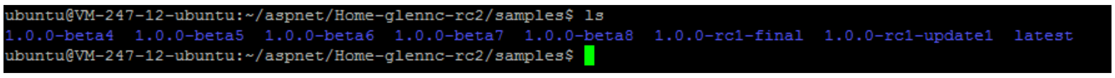
Write Dockerfile

Enter the directory ***~/aspnet/Home-glennc-rc2/samples***



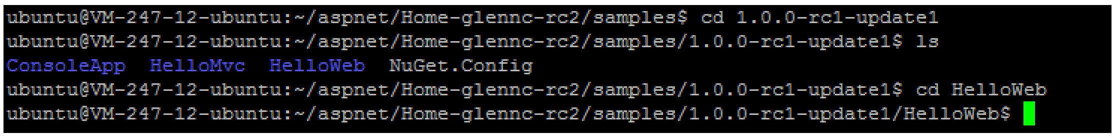
```
ubuntu@VM-247-12-ubuntu:~$ cd ~/aspnet/Home-glennc-rc2/samples
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples$
```

Explore the structure of the directory, use ***ls*** instruction



```
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples$ ls
1.0.0-beta4  1.0.0-beta5  1.0.0-beta6  1.0.0-beta7  1.0.0-beta8  1.0.0-rc1-final  1.0.0-rc1-update1  latest
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples$ 
```

We chose ***1.0.0-rc1-update1*** and go into the sub directory *HelloWeb*



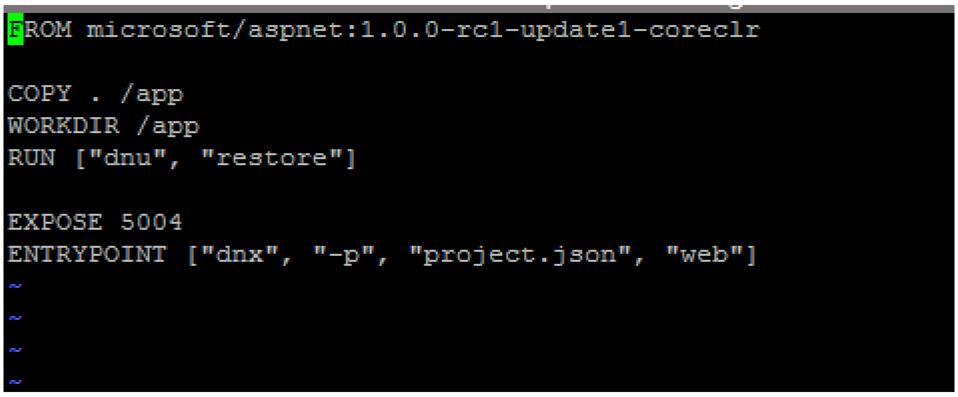
```
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples$ cd 1.0.0-rc1-update1
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples/1.0.0-rc1-update1$ ls
ConsoleApp  HelloMvc  HelloWeb  NuGet.Config
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples/1.0.0-rc1-update1$ cd HelloWeb
ubuntu@VM-247-12-ubuntu:~/aspnet/Home-glennc-rc2/samples/1.0.0-rc1-update1/HelloWeb$ 
```

Use the ***vi Dockerfile*** instruction to edit Dockerfile

Transfer the FIRST line INTO:

FROM microsoft/aspnet:1.0.0-rc1-update1-coreclr

The document finally may be like this:



```
FROM microsoft/aspnet:1.0.0-rc1-update1-coreclr

COPY . /app
WORKDIR /app
RUN ["dnu", "restore"]

EXPOSE 5004
ENTRYPOINT ["dnx", "-p", "project.json", "web"]
~
~
~
~
```

Save & exit

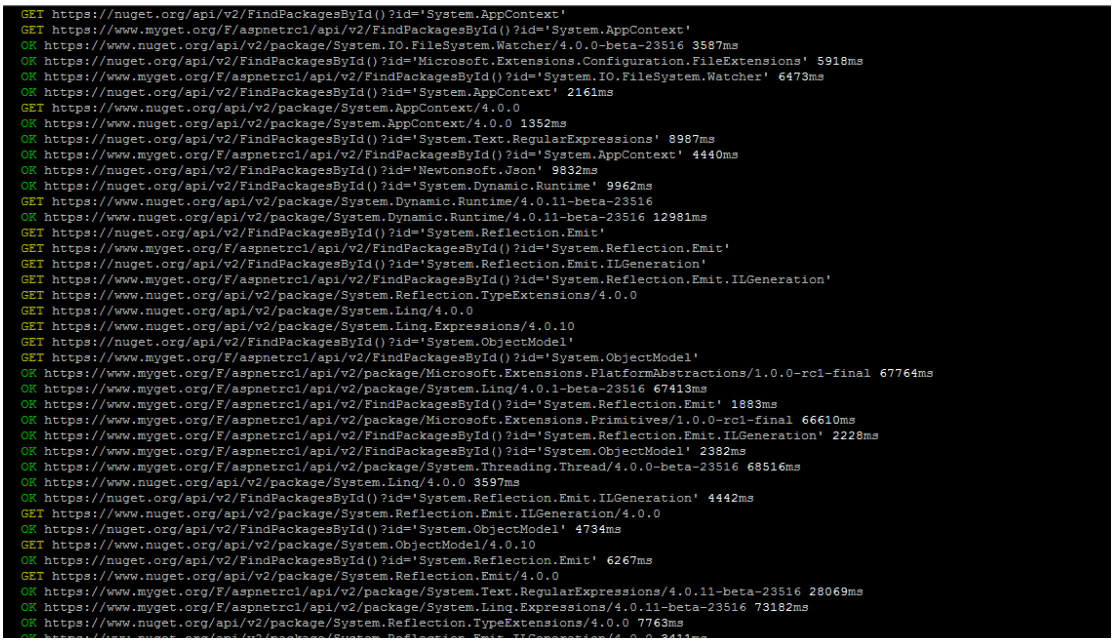
Build the hello-asp image

Use the command ***docker build -t hello-asp .*** to build

Note:

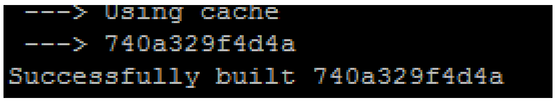
Be patient!

The process look like the picture below



```
  GET https://nuget.org/api/v2/FindPackagesById()?id='System.AppContext'
  GET https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.AppContext'
  OK https://www.nuget.org/api/v2/package/System.IO.FileSystem.Watcher/4.0.0-beta-23516 3587ms
  OK https://nuget.org/api/v2/FindPackagesById()?id='Microsoft.Extensions.Configuration.FileExtensions' 5918ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.IO.FileSystem.Watcher' 6473ms
  OK https://nuget.org/api/v2/FindPackagesById()?id='System.AppContext' 2161ms
  GET https://www.nuget.org/api/v2/package/System.AppContext/4.0.0
  OK https://www.nuget.org/api/v2/package/System.AppContext/4.0.0 1352ms
  OK https://nuget.org/api/v2/FindPackagesById()?id='System.Text.RegularExpressions' 8987ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.AppContext' 4440ms
  OK https://nuget.org/api/v2/FindPackagesById()?id='Newtonsoft.Json' 9832ms
  OK https://nuget.org/api/v2/FindPackagesById()?id='System.Dynamic.Runtime' 9962ms
  GET https://www.nuget.org/api/v2/package/System.Dynamic.Runtime/4.0.11-beta-23516
  OK https://www.nuget.org/api/v2/package/System.Dynamic.Runtime/4.0.11-beta-23516 12981ms
  GET https://nuget.org/api/v2/FindPackagesById()?id='System.Reflection.Emit'
  GET https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.Reflection.Emit'
  GET https://nuget.org/api/v2/FindPackagesById()?id='System.Reflection.Emit.ILGeneration'
  GET https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.Reflection.Emit.ILGeneration'
  GET https://www.nuget.org/api/v2/package/System.Reflection.TypeExtensions/4.0.0
  GET https://www.nuget.org/api/v2/package/System.Linq/4.0.0
  GET https://www.nuget.org/api/v2/package/System.Linq.Expressions/4.0.10
  GET https://nuget.org/api/v2/FindPackagesById()?id='System.ObjectModel'
  GET https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.ObjectModel'
  OK https://www.myget.org/F/aspnetrc1/api/v2/package/Microsoft.Extensions.PlatformAbstractions/1.0.0-rc1-final 67764ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/package/System.Linq/4.0.1-beta-23516 67413ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.Reflection.Emit' 1883ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/package/Microsoft.Extensions.Primitives/1.0.0-rc1-final 66610ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.Reflection.Emit.ILGeneration' 2228ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/FindPackagesById()?id='System.ObjectModel' 2382ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/package/System.Threading.Thread/4.0.0-beta-23516 68516ms
  OK https://www.nuget.org/api/v2/package/System.Linq/4.0.0 3597ms
  OK https://nuget.org/api/v2/FindPackagesById()?id='System.Reflection.Emit.ILGeneration' 4442ms
  GET https://www.nuget.org/api/v2/package/System.Reflection.Emit.ILGeneration/4.0.0
  OK https://nuget.org/api/v2/FindPackagesById()?id='System.ObjectModel' 4734ms
  GET https://www.nuget.org/api/v2/package/System.ObjectModel/4.0.10
  OK https://nuget.org/api/v2/FindPackagesById()?id='System.Reflection.Emit' 6267ms
  GET https://www.nuget.org/api/v2/package/System.Reflection.Emit/4.0.0
  OK https://www.myget.org/F/aspnetrc1/api/v2/package/System.Text.RegularExpressions/4.0.11-beta-23516 28069ms
  OK https://www.myget.org/F/aspnetrc1/api/v2/package/System.Linq.Expressions/4.0.11-beta-23516 73182ms
  OK https://www.nuget.org/api/v2/package/System.Reflection.TypeExtensions/4.0.0 7763ms
  OK https://www.nuget.org/api/v2/package/System.Reflection.Emit.ILGeneration/4.0.0 3411ms
```

If you see *Successfully built XXX* at the end of the process, congratulations!



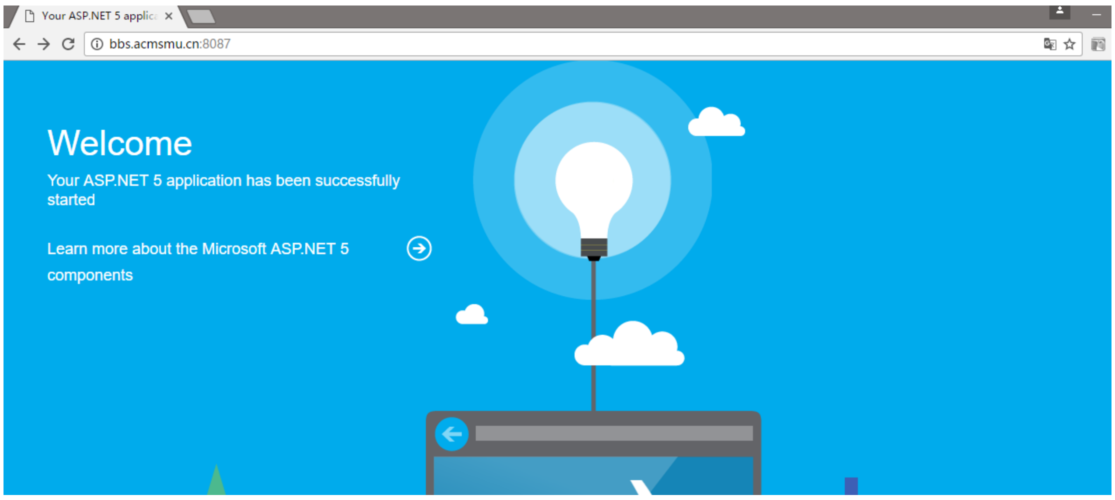
```
 ---> Using cache
 ---> 740a329f4d4a
Successfully built 740a329f4d4a
```

Enjoy

Now you can type the command below in Putty and expose the PORT **8087**(if not be allocated) in *console.qcloud.com* Security Groups

sudo docker run -t -d -p 8087:5004 hello-asp

Verify through your browser by type *ip:8087*


Your ASP.NET 5 applic
bbs.acmsmu.cn:8087
Welcome
Your ASP.NET 5 application has been successfully started
Learn more about the Microsoft ASP.NET 5 components