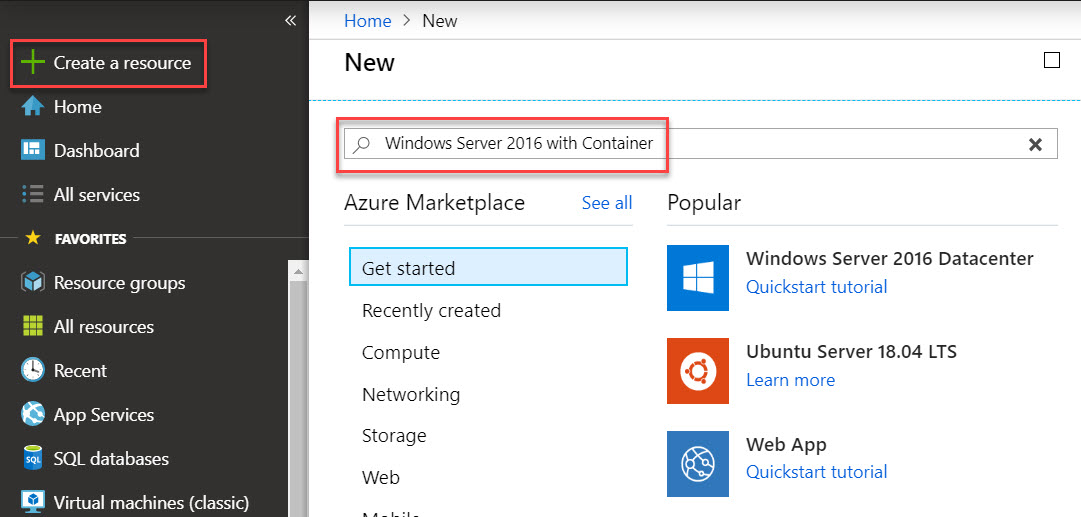
**Create & Deploy Docker Images**

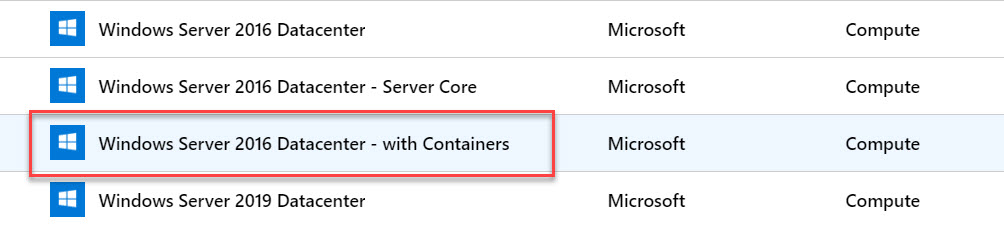
Step 1: Open Microsoft Azure Portal

[https://portal.azure.com](https://portal.azure.com/)

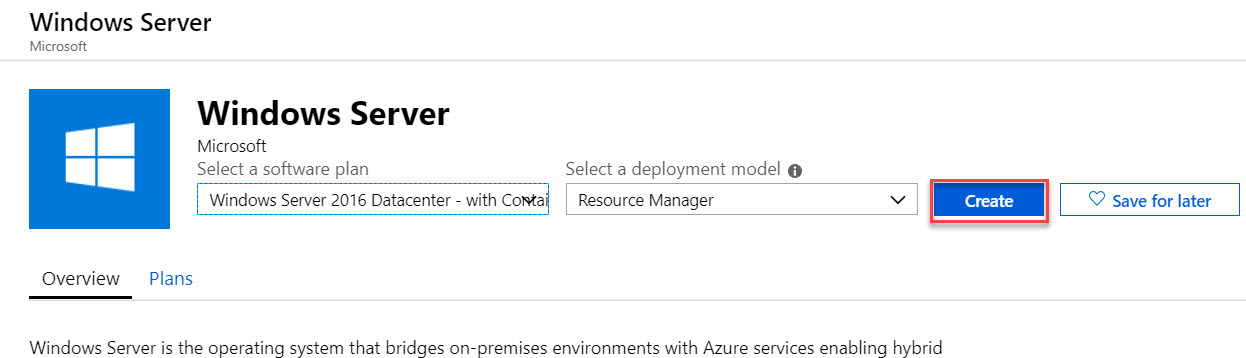
Step 2: Search with **Windows Server 2016 with Container**



Step 3: Select **Windows Server 2016 Datacenter with Containers**

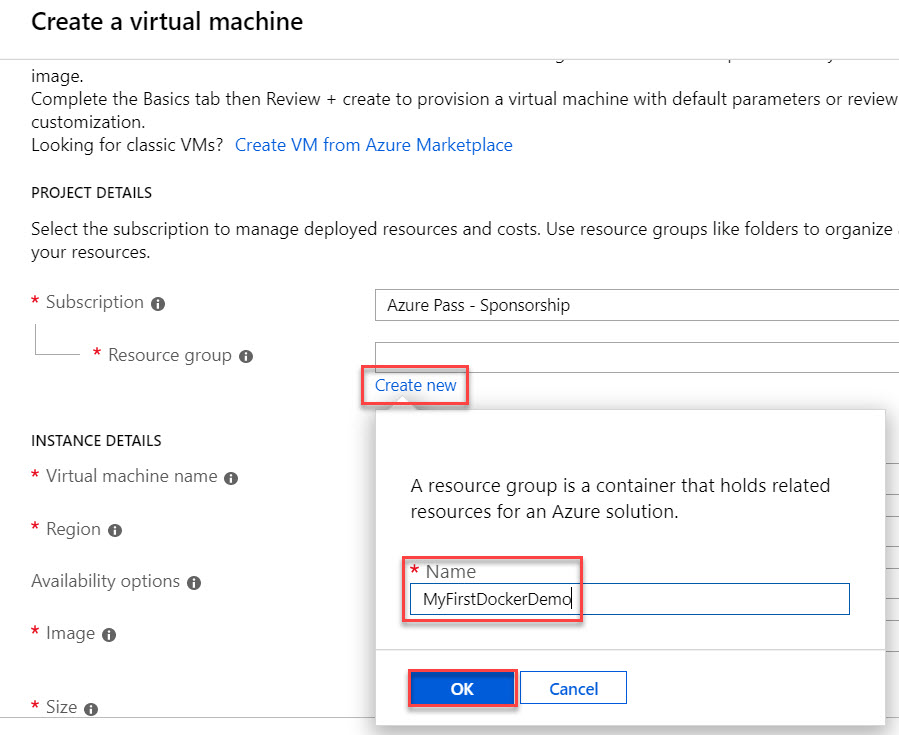


Step 4: Click on **Create** button.



Step 5: Create New Virtual Machine.

Click on Create New Resource Group: **MyFirstDockerDemo**

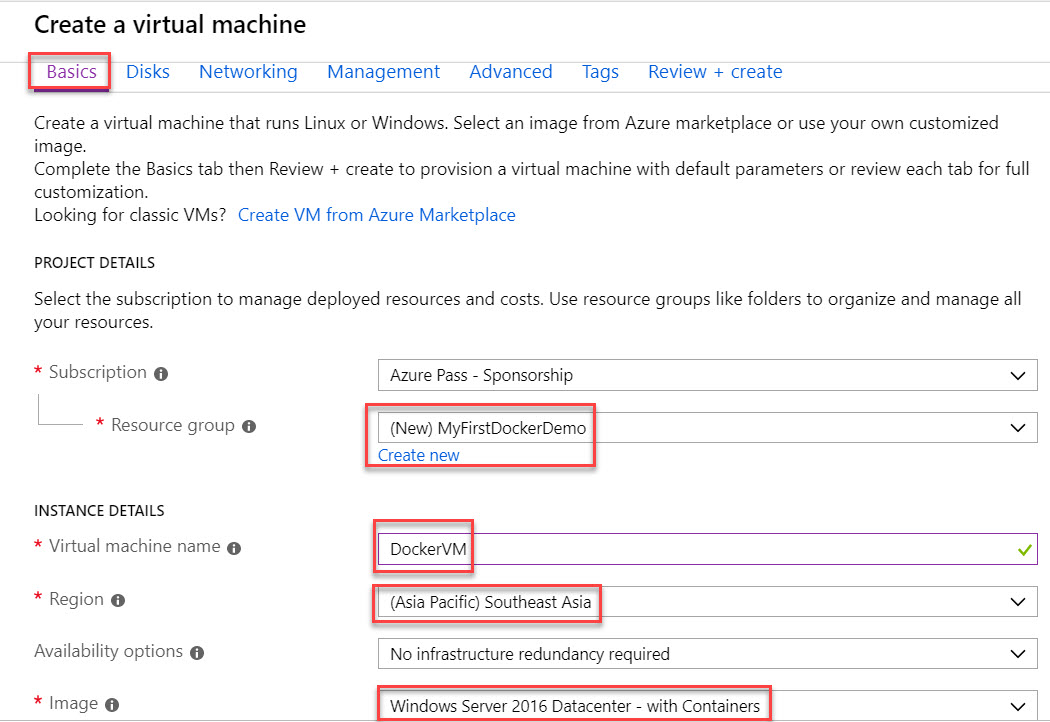


Step 6: Provide Instance Details

Virtual Machine Name: **DockerVM**

Region: Choose any nearest region

Image: **Windows Server 2016 Datacenter – with Containers**



Step 7:

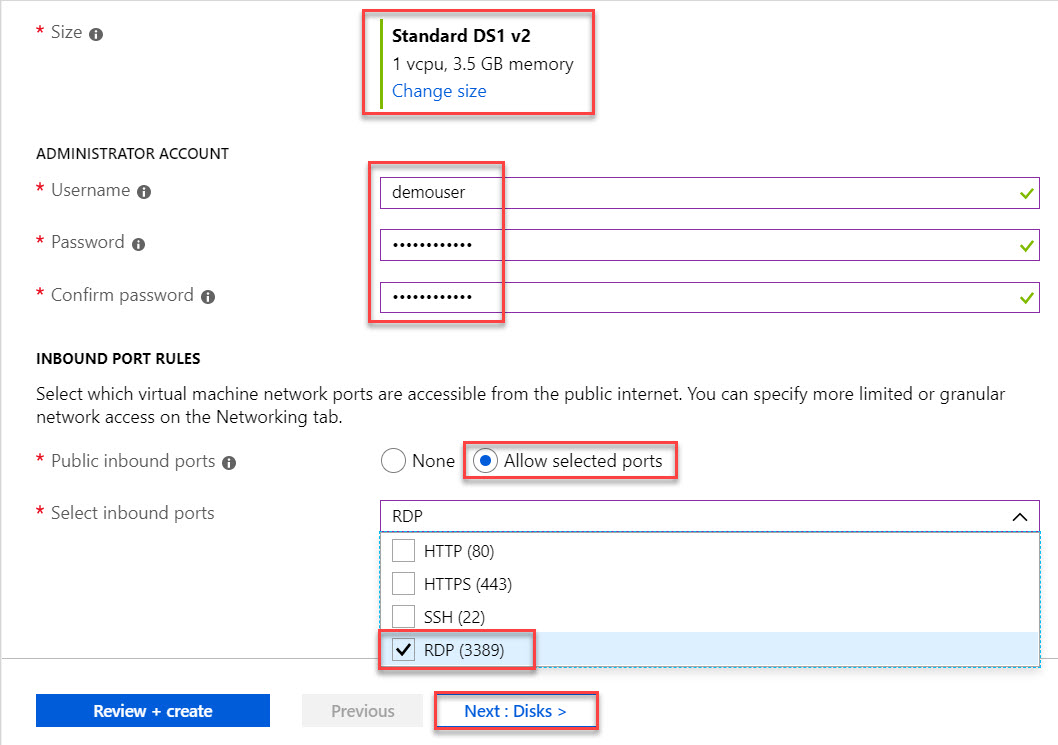
Size: Select any size Ex. Standard DS1 v2

Username: **demouser**

Password: **Demo@pass123**

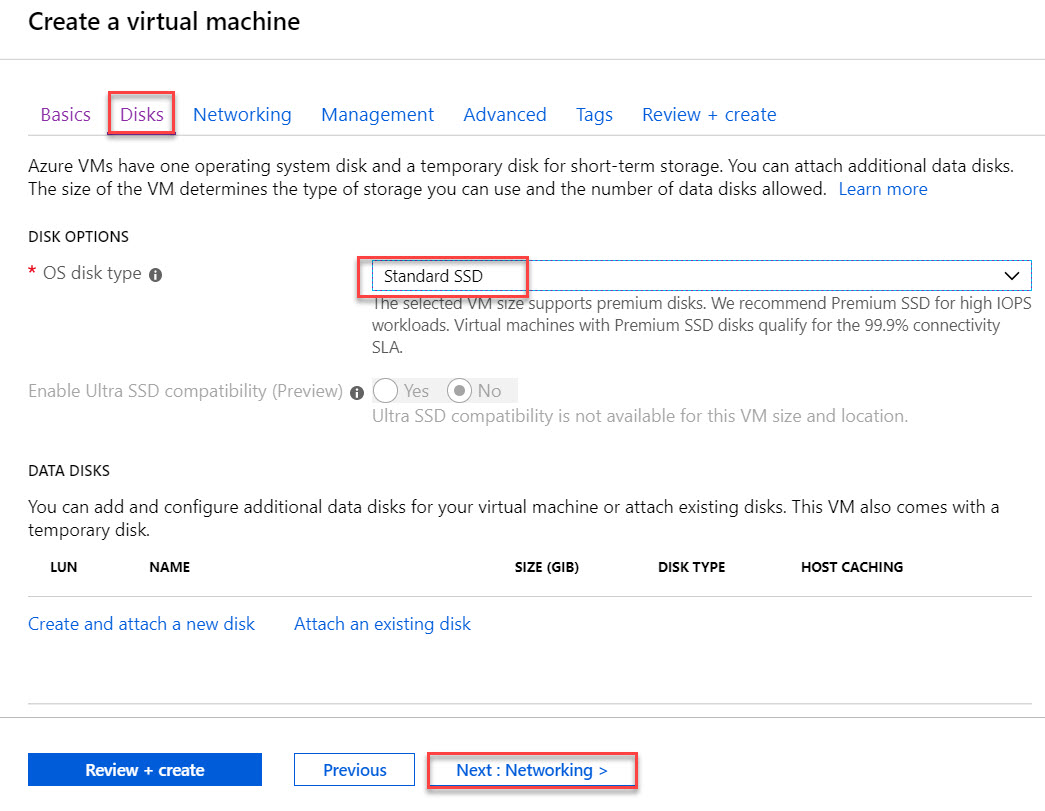
Public inbound ports: Allow Select Ports Ex. **RDP (3389)**

Click on **Next: Disks >**



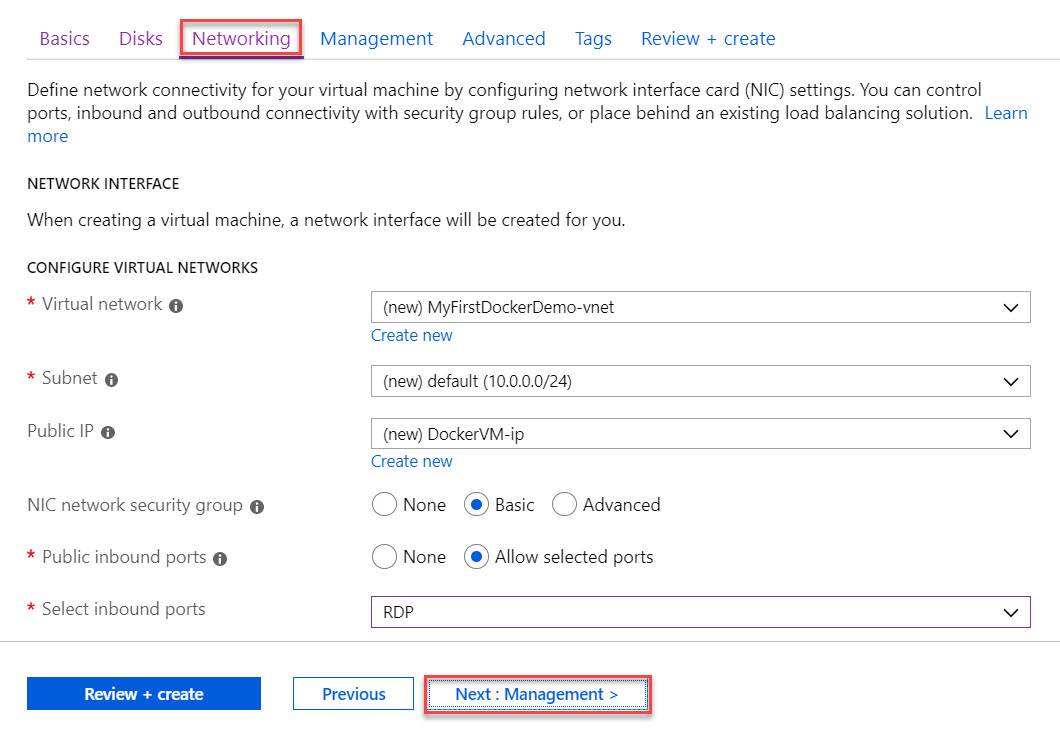
Step 8: Select Disk Type: **Standard SSD**

Click on **Next: Networking >**



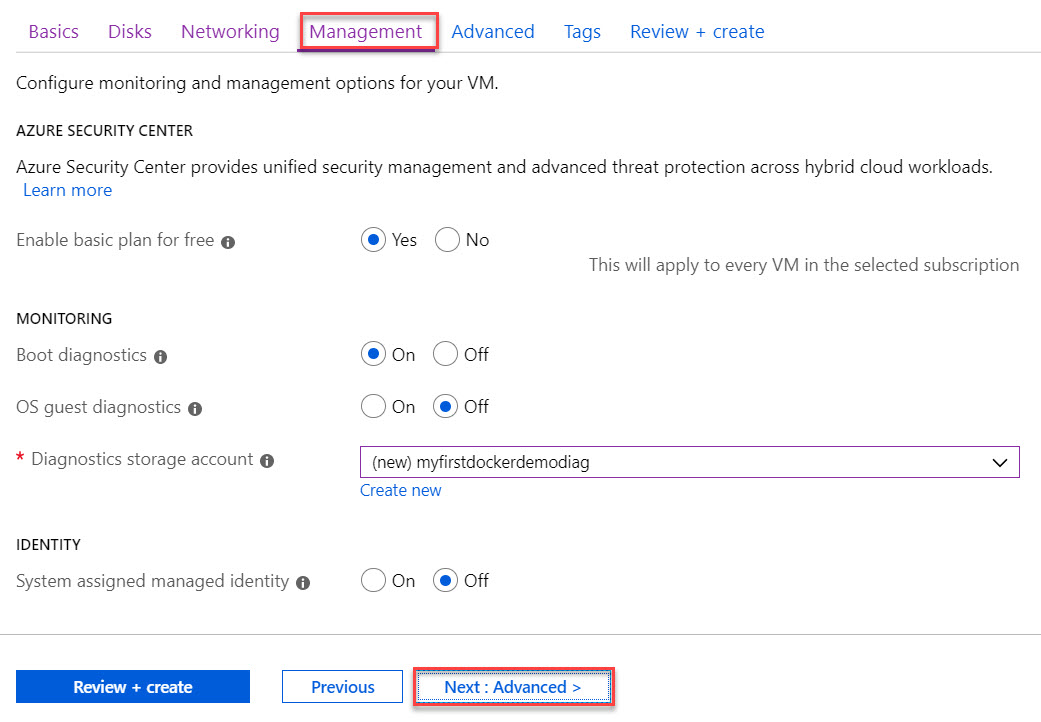
Step 9: Networking option

Go with Default option and click on **Next: Management >**

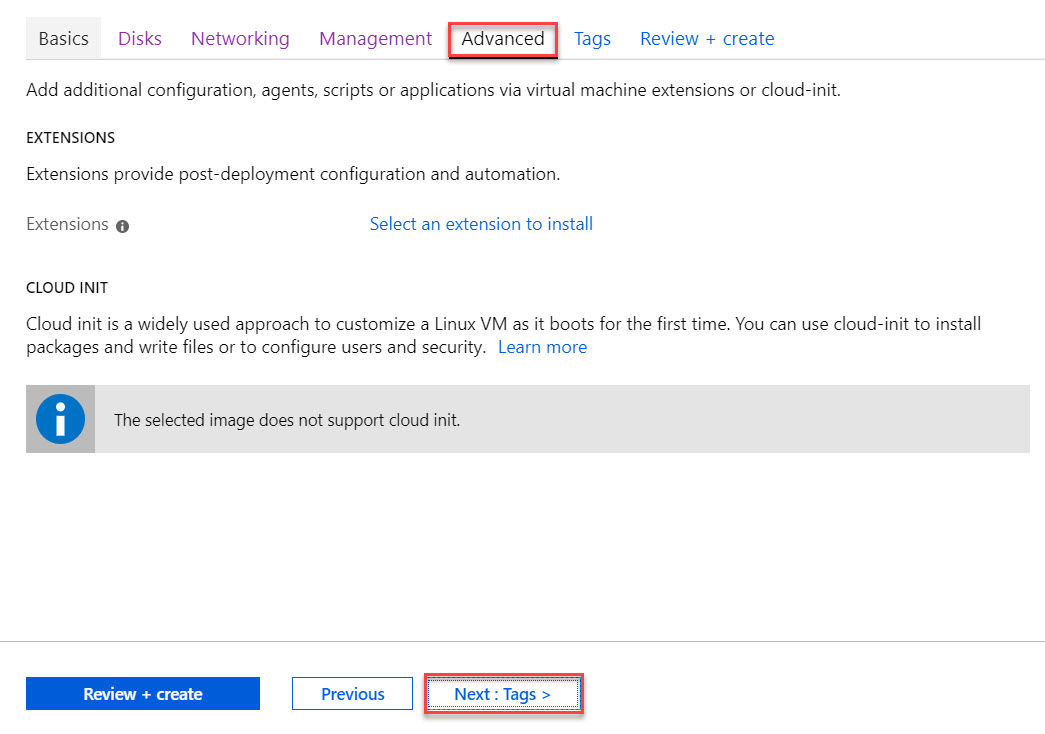


Step 10: Management option

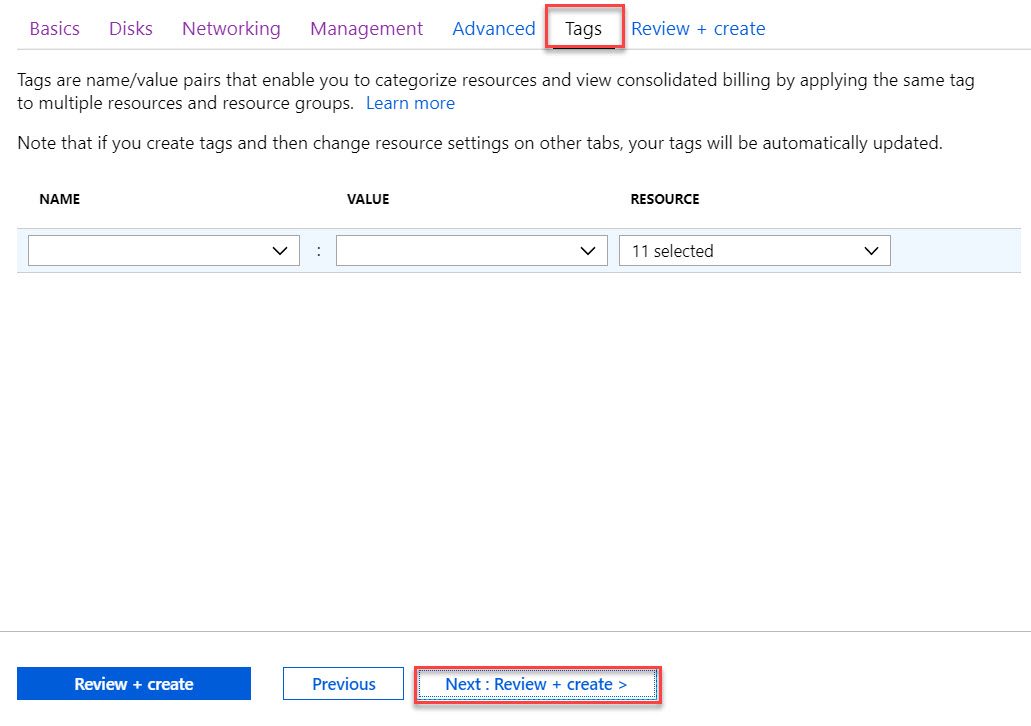
Go with default and click on **Next: Advanced >**



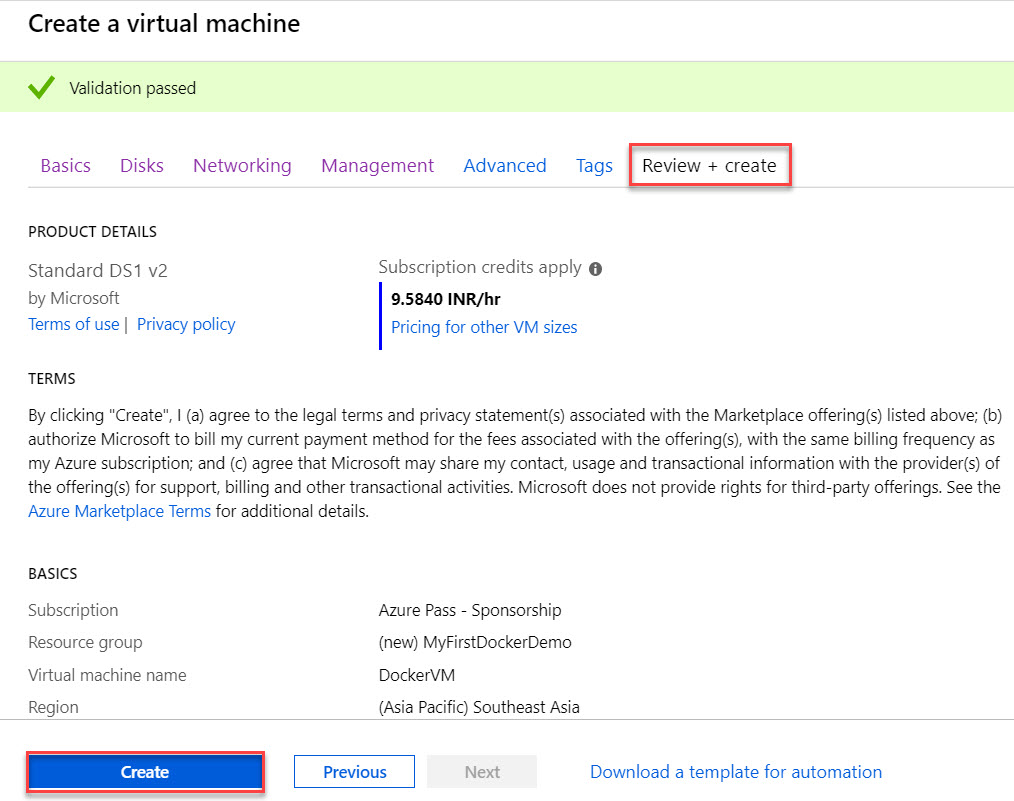
Step 11: Click on **Next: Tags >**



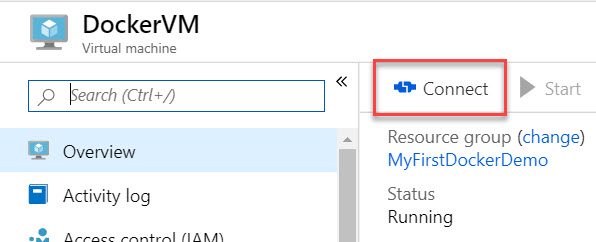
Step 12: Click on Next: **Review + Create >**



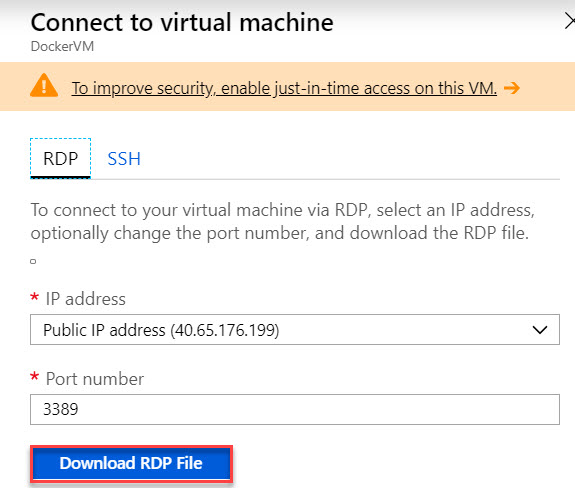
Step 13: Click on **Create** button.



Step 14: Select **Virtual Machine** and click on **Connect** option.



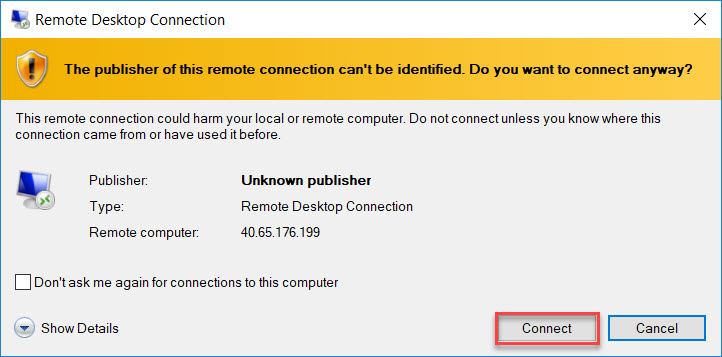
Step 15: Click on **Download RDP File**



It will download RDP file and double click on that.



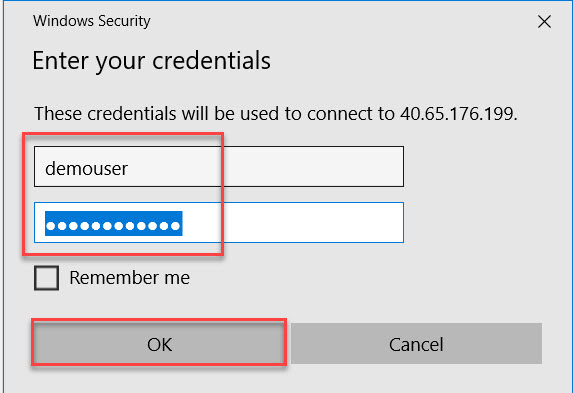
Step 16: Click on **Connect** button.



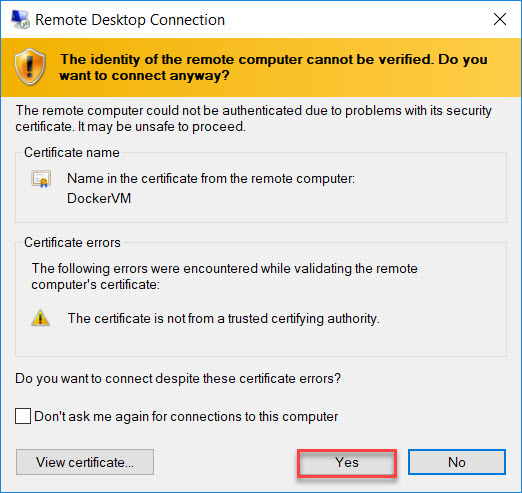
Step 17: Enter Virtual Machine Credentials:

Username: **demouser**

Password: **Demo@pass123**

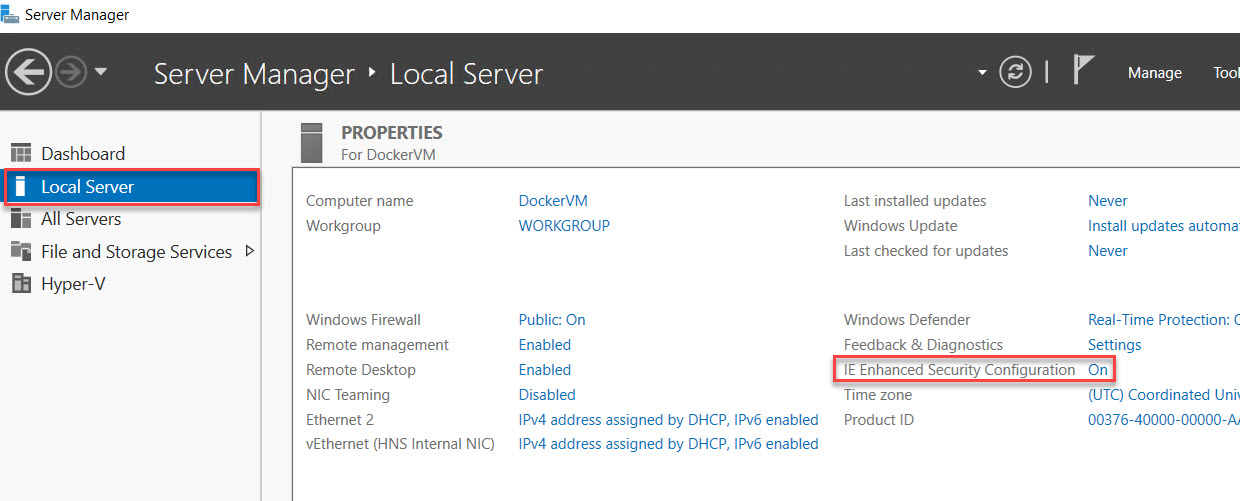


Step 18: Click on **Yes** button.

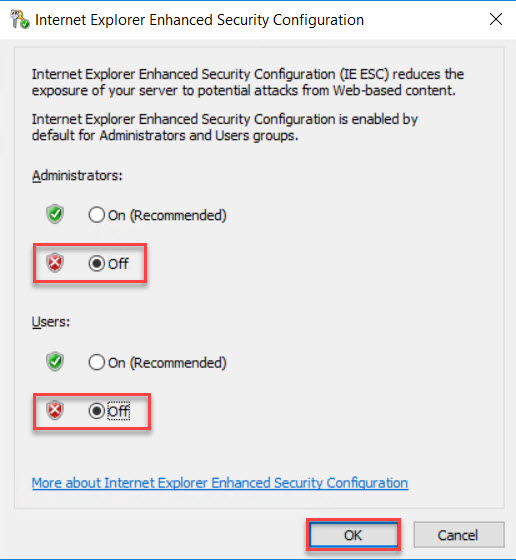


Step 19: Windows Server 2016 Datacenter will start.

Within couple of second Server Manager will open and Turn off IE Enhanced Security Configuration.



**Administrators: Off & Users: Off.**

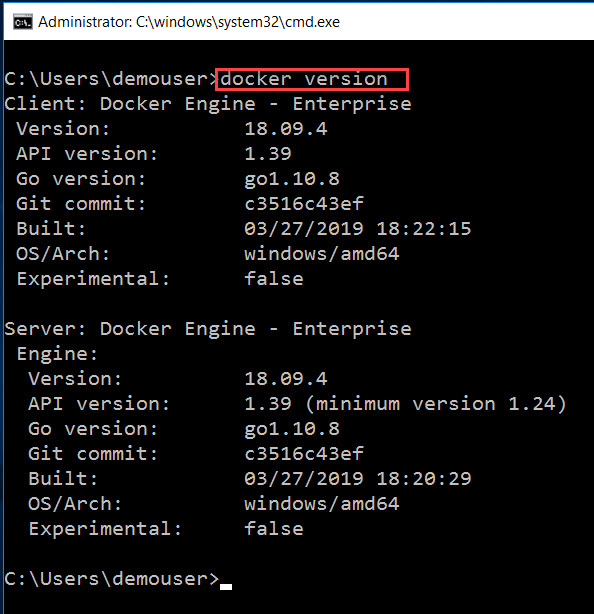


Step 20: Start Command Prompt and type below commands:

docker version

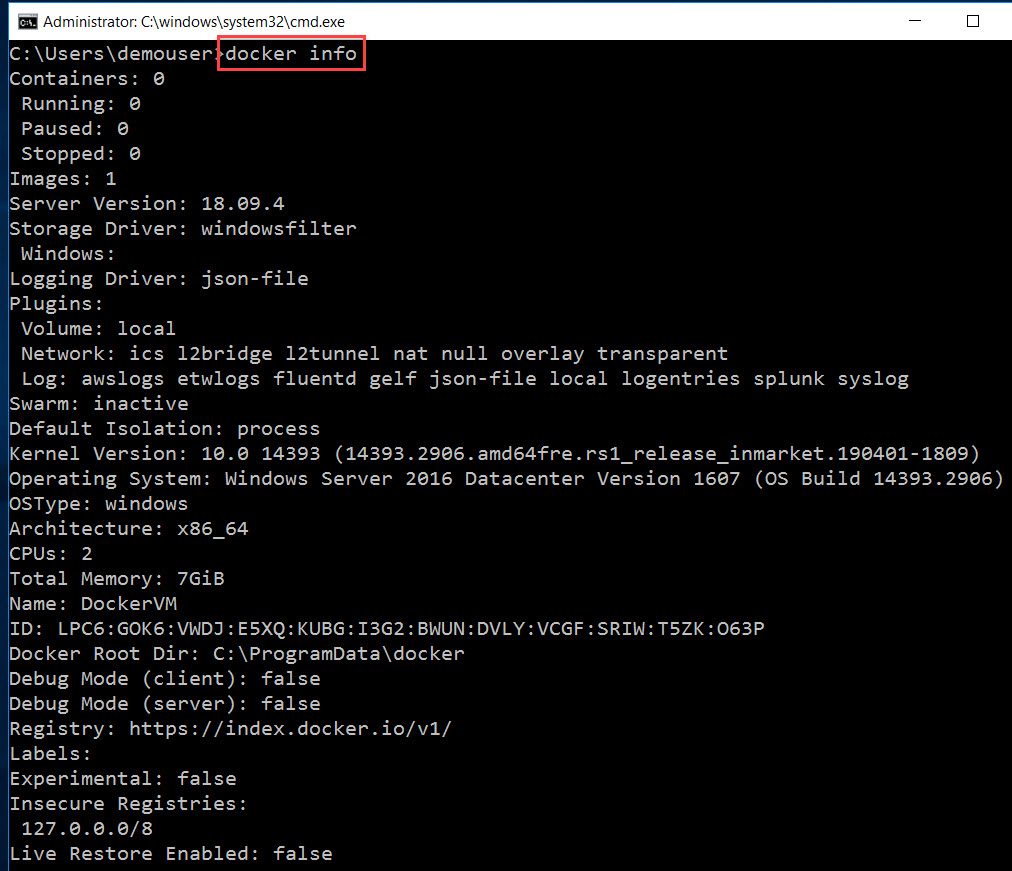
It will show Client and Server versions.

Here Client and Server installed on same machine. There any two edition available: Community and Enterprise. Community edition is free of cost.



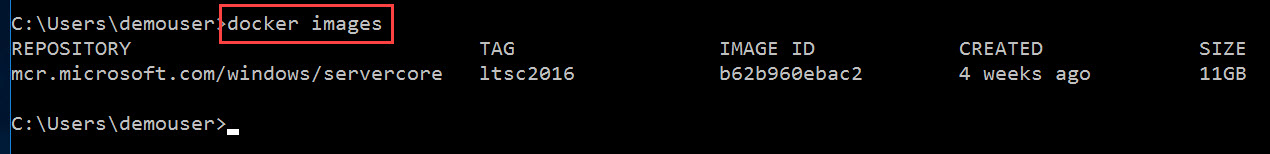
Step 21: To get system information run below command

docker info



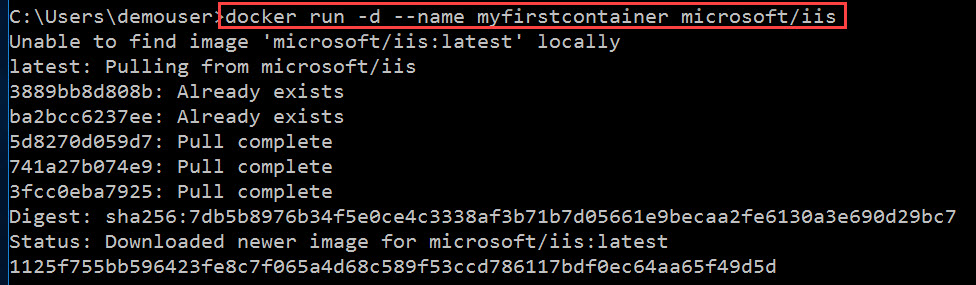
Step 22: Check available images on machine

Docker images



Step 23: Run Container

docker run -d --name myfirstcontainer microsoft/iis

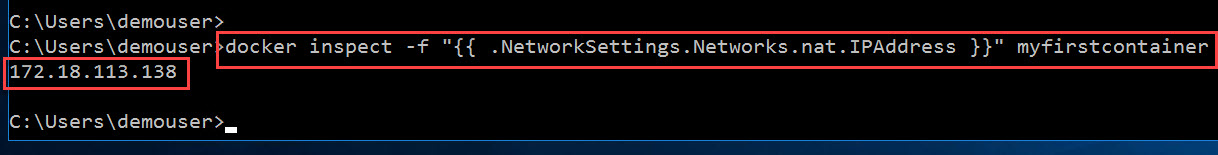


The output from Docker can be divided into several sections:

1. In the first line, Docker is telling you that it couldn't find the image you specified, **microsoft/iis**, on the Docker server. The *latest* portion after the colon (:) is a tag. The tag identifies which version of the image to use. By default, it looks for the *latest* version.
2. In the next line, it notifies you that it automatically pulled the image. You could manually perform that task using the command docker pull microsoft/iis. The *microsoft/iis* is the repository it's pulling from inside the Docker Hub registry. *microsoft*is the account name for Microsoft on Docker Hub. In general, images will come from repositories identified using the pattern *account/repository*. The account name for official Docker images is *library*.
3. The next six lines show the status of layers inside an image. The first two layers were already present and the next four had to be *pulled*/downloaded.
4. The next two lines confirm the image has been downloaded.
5. The last line is the ID of the container that started running. Usually you won't see the long form of the ID. You will usually only see the first 12 characters.

Step 24: To check IP Address of Container

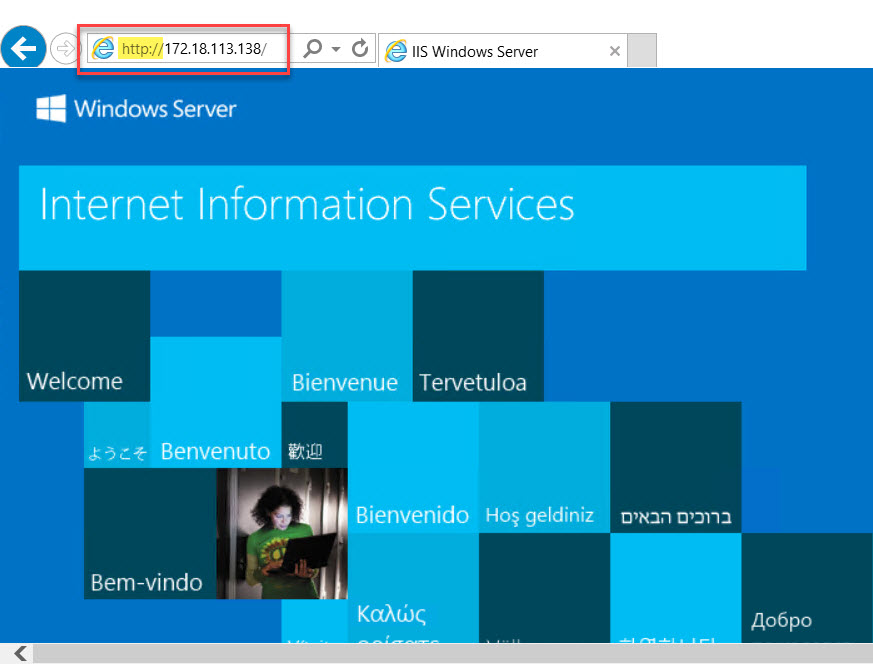
docker inspect -f "{{ .NetworkSettings.Networks.nat.IPAddress }}" myfirstcontainer



Step 25: Open Internet Explorer.

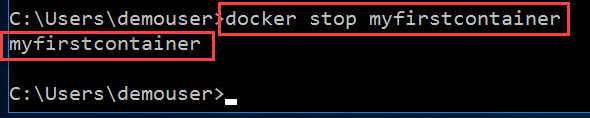
Type http://<IPADDRESS>

Ex. http://1XX.XX.XXX.XXX



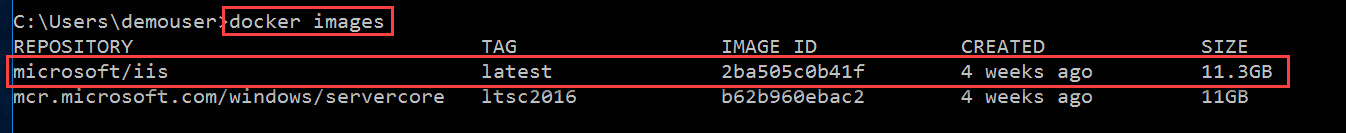
Step 26: Now stop this container

docker stop myfirstcontainer



Step 27: Check all the images

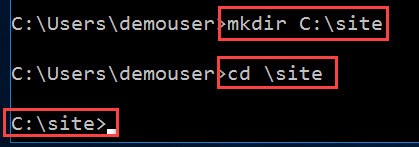
docker images



Step 28: Create one folder

mkdir c:\site

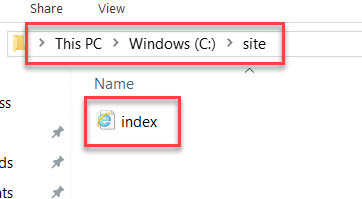
cd \site



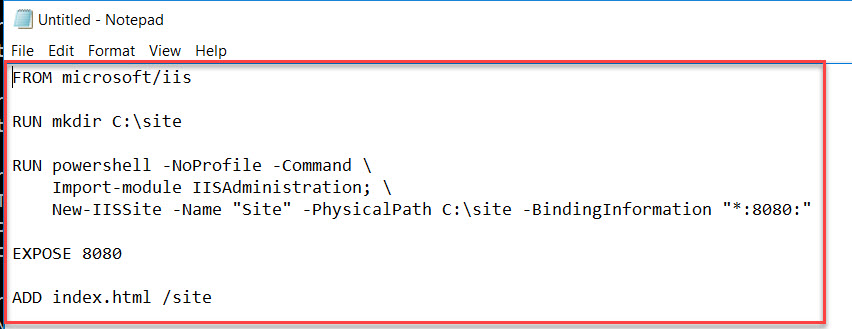
Step 29: Type below command to create web page

echo ^<!doctype html^>^<body^>^<h3^>My first web page on Docker^</h3^>^</body^>^</html^> > index.html

Step 30: navigate to C Drive and open site folder, one index.html page available.



Step 31: Now open notepad and add below code to docker file



FROM microsoft/iis

RUN mkdir C:\site

RUN powershell -NoProfile -Command \

Import-module IISAdministration; \

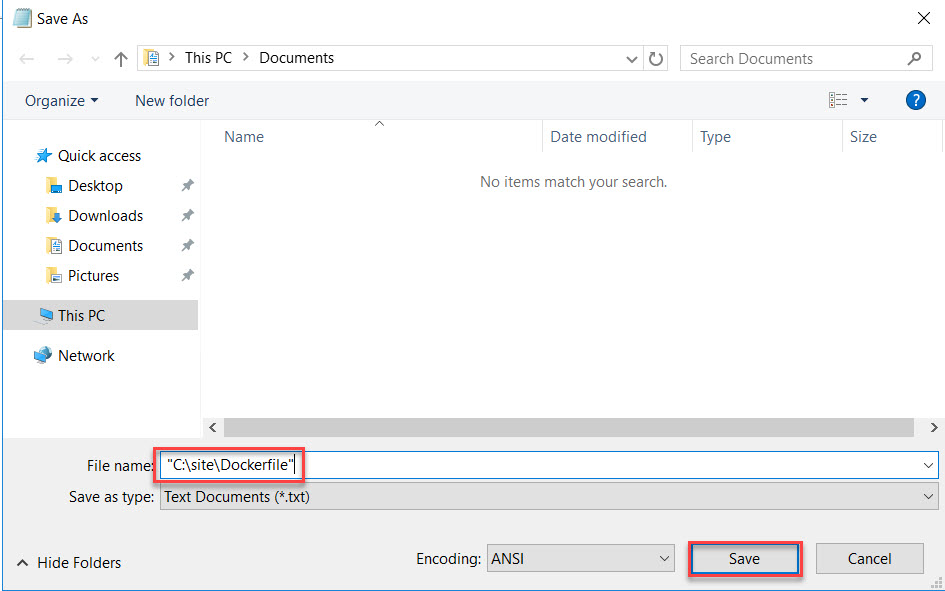
New-IISSite -Name "Site" -PhysicalPath C:\site -BindingInformation "\*:8080:"

EXPOSE 8080

ADD index.html /site

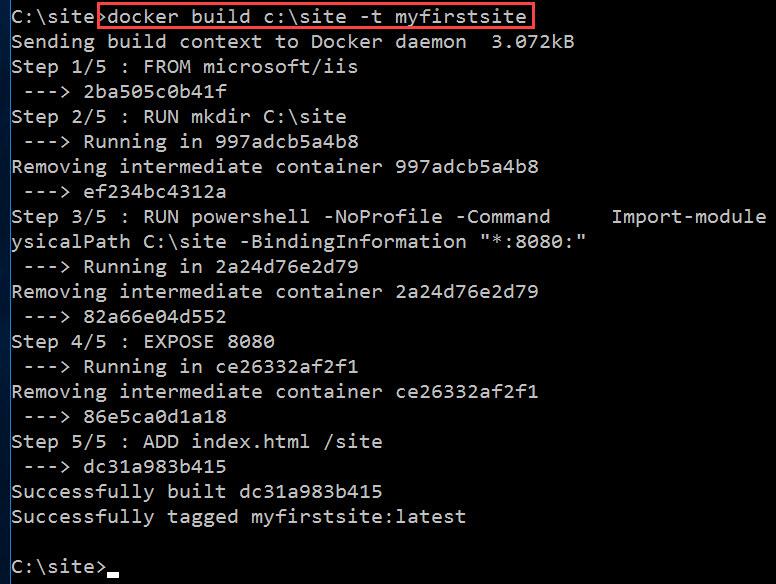
Step 32: Save the file and type this file name **"C:\site\Dockerfile"**

Note: include the quotation marks



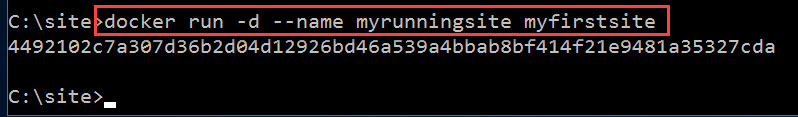
Step 33: Time to build an image to create container

docker build c:\site -t myfirstsite



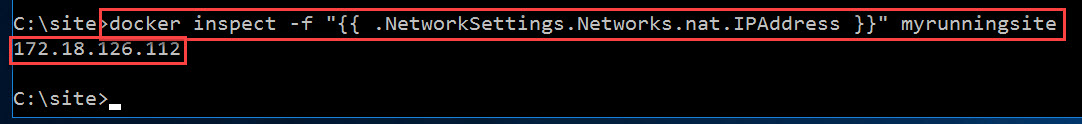
Step 34: Now run container using an image

docker run -d --name myrunningsite myfirstsite



Step 35: Get IP Address assigned to the container

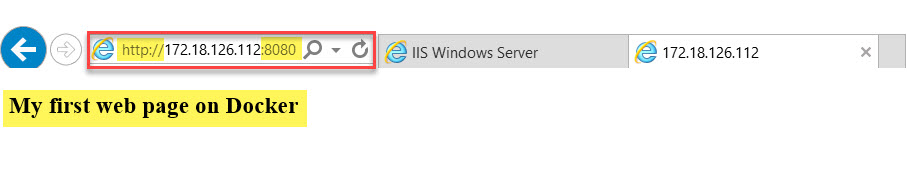
docker inspect -f "{{ .NetworkSettings.Networks.nat.IPAddress }}" myrunningsite



Step 36: Navigate to Internet Explorer and type.

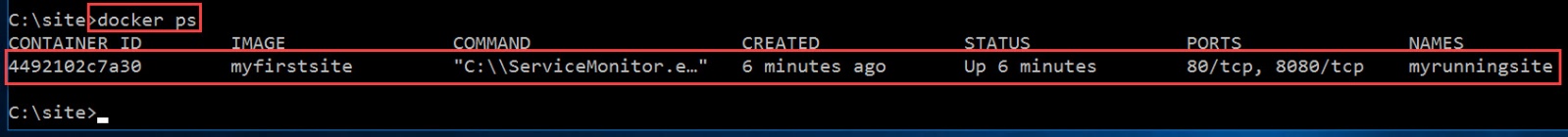
http://<IP-ADDRESS>:8080

Ex. http://1XX.XX.XX.XXX:8080



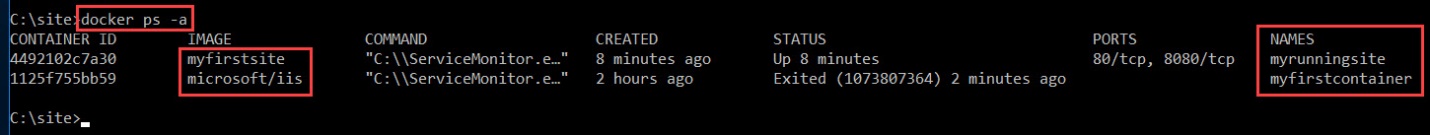
Step 37: To list running container

docker ps



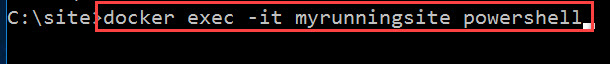
Step 38: To list running and stopped containers

docker ps -a

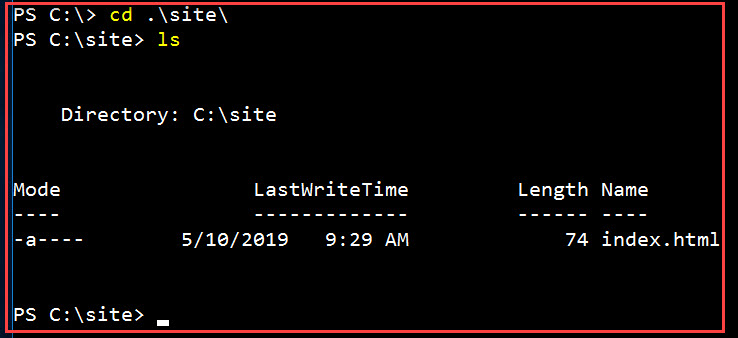


Step 39: Run other commands in a running container. Here PowerShell prompt will start

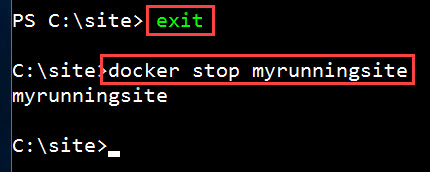
docker exec -it myrunningsite powershell



Type cd .\site\

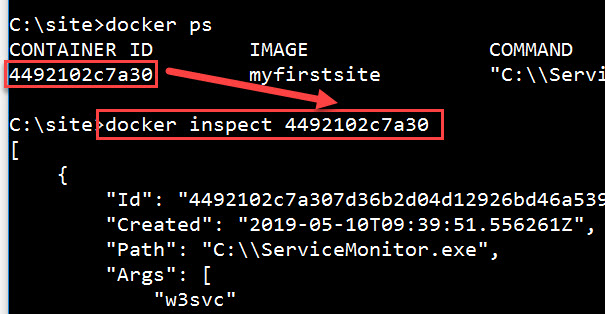


Step 40: type **Exit**



Step 41: also can check information from container id

docker inspect <container-id>

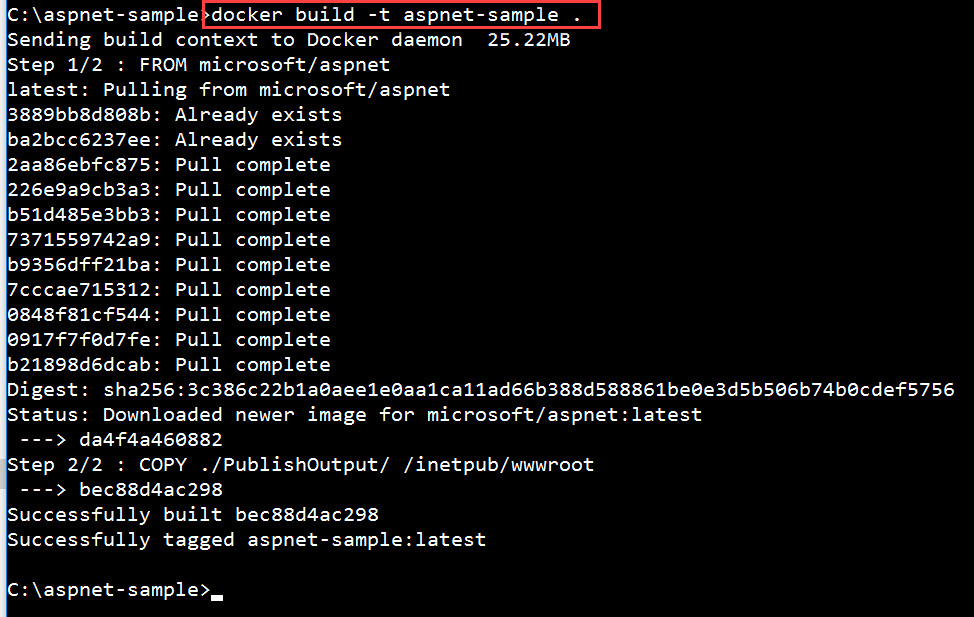


Step 42: Navigate to another folder

cd C:\aspnet-sample

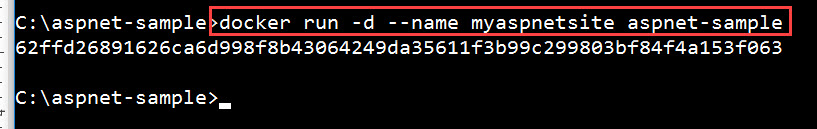
Build image

docker build -t aspnet-sample .



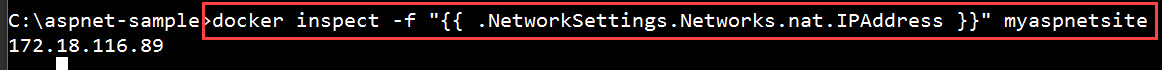
Step 43: run container

docker run -d –name myaspnetsite aspnet-sample



Step 44: Get IP Address

docker inspect -f "{{ .NetworkSettings.Networks.nat.IPAddress }}" myaspnetsite



Step 45: Open Browser and check the output

