Docker containerization

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

$ sudo apt install docker.io

“proxy2024.scansafe.net:8080” <- Add this in Docker desktop proxy. Or

“[http://docker.for.win.localhost:3128](http://docker.for.win.localhost:3128/).”

## **Configure Docker to start on boot**

Most current Linux distributions (RHEL, CentOS, Fedora, Ubuntu 16.04 and higher) use [systemd](https://docs.docker.com/install/linux/linux-postinstall/#systemd) to manage which services start when the system boots. Ubuntu 14.10 and below use [upstart](https://docs.docker.com/install/linux/linux-postinstall/#upstart).

### systemd

$ sudo systemctl enable docker

To disable this behavior, use disable instead.

$ sudo systemctl disable docker

If you need to add an HTTP Proxy, set a different directory or partition for the Docker runtime files, or make other customizations, see [customize your systemd Docker daemon options](https://docs.docker.com/engine/admin/systemd/).

### upstart

Docker is automatically configured to start on boot using upstart. To disable this behavior, use the following command:

$ echo manual | sudo tee /etc/init/docker.override

### chkconfig

$ sudo chkconfig docker on

Notes from Docker tutorials Plural sight,

1. Docker image will have one to many relationships with containers.
2. Image provide,
   1. File System
   2. MetaData
   3. A Command
3. Ways of crating docker images,
   1. An image is build using a sequence of instructions contained in a special file called a Docker file.
   2. Changes to a containers file system during execution, are committed to a brand new image.
4. Choosing the right base image is important, Is the image fit for purpose?, can you trust the content on the base image?
5. Container is invoked using “docker container run” command.
6. Typical Docker File contains
   1. Configuration object
   2. Ports exposed
   3. Root Filesystem object
   4. Ordered list of references of individual layers of file system content.
   5. Each reference is a 256-bit hash of the contents of the layer.
7. Image Size matters
8. An docker image build is invoked with docker image build command, the build context is sent as an archive from client to demon.
9. sudo apt install unzip to install unzip, unzip filename.zip
10. How to reset password in ubunt linux, glcoud machine

vfa\_gianglt@instance-1:~$ sudo -i

root@instance-1:~# sudo bash

root@instance-1:~# sudo passwd root

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

root@instance-1:~# cd /root

root@instance-1:~#

1. Restart nginx after default settings,

sudo fuser -k 80/tcp,sudo fuser -k 443/tcp, Then execute sudo service nginx restart

//Command to build docker file and –t stands for tagging with a name

docker build –t app .

//to run container , -it stands for interactive terminal

Docker run –it 4da53(image id)

docker run --name aspnetcore\_sample --rm -it -p 8000:80 mcr.microsoft.com/dotnet/core/samples:aspnetapp

# **ASP.NET Core deployment using Docker, Nginx and Ubuntu Server**

<https://www.youtube.com/watch?v=WUzEJuBYtY0>

<https://docs.microsoft.com/en-us/nuget/reference/nuget-config-file#config-section>

<https://codefresh.io/docker-tutorial/build-docker-image-dockerfiles/>

Command to build project with proxy settings,

“ docker build --build-arg http\_proxy=http://proxy2024.scansafe.net:8080 --build-arg https\_proxy=http://proxy2024.scansafe.net:8080 -t aspnetapp .”

//To stop docker running

Docker stop ‘containerid (first 3- letter)’

//Check containers with images

docker ps –as

//Remove Container arms

Docker rm “container id”

//Remove dockre images

Docker rmi “image id

devenv.exe.config

<system.net>

<defaultProxy enabled="true" useDefaultCredentials="true">

<proxy usesystemdefault="true" bypassonlocal="true" />

</defaultProxy>

<settings>

<ipv6 enabled="true"/>

</settings>

</system.net>

NugetConfig (%appdata%\Nuget)

<config>

<add key="http\_proxy" value="http://proxy2024.scansafe.net" />

<add key="https\_proxy" value="https://proxy2024.scansafe.net" />

</config>

Docker running on Linux container on windows machine

<https://github.com/dotnet/dotnet-docker/tree/master/samples/aspnetapp>

## Build and run the sample with Docker

You can build and run the sample in Docker using the following commands. The instructions assume that you are in the root of the repository.

cd samples

cd aspnetapp

docker build --pull -t aspnetapp .

docker run --name aspnetcore\_sample --rm -it -p 8000:80 aspnetapp

## Try a pre-built ASP.NET Core Docker Image

You can quickly run a container with a pre-built [sample ASP.NET Core Docker image](https://hub.docker.com/_/microsoft-dotnet-core-samples/), based on this [sample](https://github.com/dotnet/dotnet-docker/blob/master/samples/aspnetapp/Dockerfile).

Type the following command to run a sample with [Docker](https://store.docker.com/editions/community/docker-ce-desktop-windows):

docker run --name aspnetcore\_sample --rm -it -p 8000:80 mcr.microsoft.com/dotnet/core/samples:aspnetapp

After the application starts, navigate to http://localhost:8000 in your web browser. In some scenarios, and with earlier versions of Windows, you need to access the container via IP address. See the following section for instructions on how to do that.

## View ASP.NET Core apps via IP address

After the ASP.NET Core application starts, navigate to the container IP in your web browser with the the following instructions:

Note: These instructions rely on using the --name aspnetcore\_sample argument with docker run. The --name argument makes it possible to access the container by name. If you used a different name, then use it instead in the following steps.

1. Open up a command prompt.
2. Run docker exec aspnetcore\_sample ipconfig.
3. Copy the container IP address and paste into your browser (for example, 172.29.245.43).

See the following example of how to get the IP address of a running Windows container.

C:\git\dotnet-docker\samples\aspnetapp>docker exec aspnetcore\_sample ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : contoso.com

Link-local IPv6 Address . . . . . : fe80::1967:6598:124:cfa3%4

IPv4 Address. . . . . . . . . . . : 172.29.245.43

Subnet Mask . . . . . . . . . . . : 255.255.240.0

Default Gateway . . . . . . . . . : 172.29.240.1

Note: [docker exec](https://docs.docker.com/engine/reference/commandline/exec/) supports identifying containers with name or hash. The container name is used in the preceding instructions. docker exec runs a new command (as opposed to the [entrypoint](https://docs.docker.com/engine/reference/builder/#entrypoint)) in a running container.

docker inspect can also be used for this same purpose, as demonstrated in the following example.

C:\git\dotnet-docker\samples\aspnetapp>docker inspect -f "{{ .NetworkSettings.Networks.nat.IPAddress }}" aspnetcore\_sample

172.25.157.148

Best Practices to write a docker file

<https://docs.docker.com/develop/develop-images/dockerfile_best-practices/>