Dockerizing In Ubuntu 18.04

Docker Installation

Docker Official Documentation

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

Quick Installation steps

```
sudo apt-get remove docker \
docker-engine docker.io containerd runc

sudo apt-get update

sudo apt-get install curl gnupg-agent \
apt-transport-https ca-certificates \
software-properties-common

sudo apt-key fingerprint 0EBFCD88

sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable"

sudo apt-get update

sudo apt-get install \
docker-ce docker-ce-cli containerd.io
```

These steps will install stable release. See doc for more info.

Test with Hellow-world

```
sudo docker run hello-world
```

Output

```
Hello from Docker!
This message shows that your
installation appears to be working
correctly.
```

Dockerfile

Getting Started

- Contains sequential sets of instructions.
- Each instruction creates a layers.
- Used to create/build Docker Images

Official Instruction Guide

https://docs.docker.com/engine/reference/builder/

Dockerizing Snakegame using Dockerfile

Make sure you are on the root/parent path of your project. Create your own Dockerfile as shown below

```
- Snakegame
        |__ index.html
        |__ default.conf
        |__ Dockerfile
```

Dockerfile

```
# Use nginx base image alpine version

FROM nginx:alpine

# Copying files in required path

COPY default.conf /etc/nginx/conf.d/

COPY index.html /usr/share/nginx/html/

# Command that keeps the container live

CMD ["nginx", "-g", "daemon off;"]
```

Docker Image

Build Docker Image

```
sudo docker build -t myimage .
```

Output

It creates layers with unique ID. Multiple Images can be created using same Dockerfile with same layers.

```
Sending build context to Docker daemon
10.75kB
Step 1/4 : FROM nginx:alpine
 ---> dd025cdfe837
Step 2/4 : COPY default.conf
/etc/nginx/conf.d/
 ---> Using cache
 ---> 03e0ef498d52
Step 3/4 : COPY index.html
/usr/share/nginx/html/
 ---> Using cache
 ---> cba61c47dd23
Step 4/4 : CMD ["nginx", "-g", "daemon
off;"]
 ---> Using cache
 ---> 4e6dc325dddc
Successfully built 4e6dc325dddc
Successfully tagged myimage:latest
```

Check the created image using command

```
sudo docker image ls
```

Outuput

```
REPOSITORY  TAG     IMAGE ID
myimage     latest  4e6dc325dddc

CREATED              SIZE
9 seconds ago        16.1MB
```

Docker Container

Run Container

```
sudo docker run -itd --name cont myimage
```

Check Container Status

```
sudo docker container ls
```

Outuput

```
CONTAINER ID  IMAGE  COMMAND
226909f6c962  myimage "nginx -g 'daemon
of…"
CREATED      STATUS         PORTS   NAMES
1 hour ago   Up 3 seconds   80/tcp  cont
```

Get inside the running Container using sh

```
sudo docker exec -it cont sh
```

Outuput

```
/ # ls
bin    dev    etc    home   lib    media
mnt    opt    proc   root   run    sbin
srv    sys    tmp    usr    var
```

Stop and Remove the Container and remove Image

```
sudo docker container stop cont
sudo docker container rm cont
Sudo docker image rm myimage
```

Container ID and Container Name are unique, refers a container.

For More Container Commands

https://docs.docker.com/engine/reference/commandline/container/

Facilities : Docker-compose, Swarm mode and Orchestraction.