Documentation of Installing Docker on Ubuntu

1. sudo apt update -y

*To update the existing packages

2. sudo apt install apt-transport-https ca-certificates curl software-propertiescommon -y

* Allows our operating system to access the Docker repositories over HTTPS and download the dependencies.

apt-transport-https

*Allows the package manager to transfer files and data over https.

ca-certificates :

*Allows the system to check security certificates.

curl :

*A tool for transferring data.

software-properties-common : Adds scripts for managing software

3. curl -fsSL | sudo apt-key add

* Let's add GPG keys to ensure the software we're installing is authentic

4. sudo add-apt-repository "deb [arch=amd64] bionic stable"

*To Install docker Repository.

* $(lsb_release –cs) scans and returns the codename of our ubuntu installation – In our case, its bionic and the stable is the type of docker release

5. sudo apt update

*Update the repository currently added

6. sudo apt install docker-ce -y

*Installing latest docker community edition

7. sudo systemctl status docker

* To check the status of the docker

8. sudo docker pull nginx

* Get the Nginx Docker Image

It downloads all the necessary components for the container. Docker will cache these, so when we run the container, we don't need to download the container image(s) each time.

9. sudo docker run --name docker-nginx -p 80:80 nginx

run : The command to create a new container

The --name flag : Specify the name of the container

-p : Specifies the port we are exposing in the format of -p. In this case we are mapping Port 80 in the container to Port 80 on the server

Nginx : Name of the image on dockerhub

1. sudo docker container run -d -p 3306:3306 --name dp -e MYSQL_ROOT_PASSWORD=yes mysql

• docker run will automatically run a created docker container

• --name dp : the flag --name will instruct Docker to create a container named dp

• -d : this optional flag stands for the detouch mode. When included, the MySQL database will run in the background as a docker demon.

• -p 3306:3306 - this port number will map the MySQL server to its default port which is 3306 (Port for Outside World: Actual Port of service in container)

• -e MYSQL_ROOT_PASSWORD = yes : the flag e stands for environment variables. In this case, we need a root password to access the MySQL server. We will assign the root password as an environment variable.

• Mysql - specifies the image that we want to include in this container. This image must be downloaded/pulled and made available in a local docker.

2. sudo docker container logs dp

Docker logs: command retrieves logs present at the time of execution.

3. sudo docker container logs dp > db_log

Copy log files to db_log

4. sudo docker container run -d -p 8080:80 --name apache httpd

-d : this optional flag stands for the detouch mode. When included, the apache will run in the background as a docker demon.

-p 8080:80 - this port number will map the apache server to its default port which is 8080 (Port for Outside World: Actual Port of service in container)

5. sudo docker container run -d -p 80:80 --name web nginx

Miscellaneous Commands

1. sudo docker container ls

*List the running containers

2. sudo docker container stop web apache dp

*To stop the running container

3. sudo docker container ls -a

* List all the containers

4. sudo docker container rm web apache dp

* To remove the container

5. sudo docker images

*Manage Image

6. sudo docker image ls

*List the image

7. sudo docker container top

* To display running processes of the container

8. sudo docker container inspect

*Display detailed information on one or more containers

9. sudo docker container stats

* Display a live stream of container(s) resource usage statistics