Docker Installation on centos for community edition:

#########################

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

Install required packages for docker:

Step:1> sudo yum install -y device-mapper-persistent-data lvm2 yum-utils

Adding the Docker CE repo:

sudo yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

Installing the Docker CE packages and containerd.io:

sudo yum install -y docker-ce-18.09.5 docker-ce-cli-18.09.5 containerd.io

Start and enable the Docker service:

sudo systemctl start docker

sudo systemctl enable docker

TO VERIFY:

docker version

Add user to the docker group, giving the user permission to run docker commands:

sudo usermod -a -G docker username

After above step Log out and login in again.

docker version

Test the installation by running a simple container:

docker run hello-world

DOCKER COMMUNITY VERSION INSTALLATION ON UBUNTU

#############################################

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

Install required packages:

sudo apt-get update

sudo apt-get -y install \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common

Add the Docker GPG key and repo:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo apt-key fingerprint 0EBFCD88

sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

Installing Docker CE packages:

sudo apt-get update

sudo apt-get install -y docker-ce=5:18.09.5~3-0~ubuntu-bionic docker-ce-cli=5:18.09.5~3-0~ubuntu-bionic containerd.io

Give user permission to run docker commands:

sudo usermod -a -G docker user

Log out and log back in.

docker verssion

Test the installation by running a simple container:

docker run hello-world

WHAT IS STORAGE DRIVER:

ANS:- Storage driver provides a pluggable framework for managing the temporarily, internal storage of a container writable layar.

Docker supports a veriety of storage driver .The best storage driver to use depends on your environment and storage needs .

Overlay2: File based storage . Default for Ubuntu and CentOS 8+.

Devicemappaer: Block based storage . Default for Centos & and eairler .

We can check what storage driver we are using now by running

docker info

SELECTING A DOCKER STORAGE DRIVER

================================

This lesson was performed on a CentOS 7 server running Docker CE

Get the current storage driver:

docker info

Set the storage driver explicitly by providing a flag to the Docker daemon:

sudo vi /usr/lib/systemd/system/docker.service

Edit the ExecStart line, adding the --storage-driver devicemapper flag:

ExecStart=/usr/bin/dockerd --storage-driver devicemapper ...

After any edits to the unit file, reload Systemd and restart Docker:

sudo systemctl daemon-reload

sudo systemctl restart docker

We can also set the storage driver explicitly using the daemon configuration file. This is the method that Docker recommends. Note that we cannot do this and pass the --storage-driver flag to the daemon at the same time:

sudo vi /etc/docker/daemon.json

Set the storage driver in the daemon configuration file:

{

"storage-driver": "devicemapper"

}

Restart Docker after editing the file. It is also a good idea to make sure Docker is running properly after changing the configuration file:

sudo systemctl restart docker

sudo systemctl status docker

Run a simple container using the hello-world image:

docker run hello-world

<https://docs.docker.com/engine/reference/run/>

Run a container using a specific image tag:

docker run nginx:1.15.11

Run a container with a command and arguments:

docker run busybox echo hello world!

Run an Nginx container customized with a variety of flags:

docker run -d --name nginx --restart unless-stopped -p 8080:80 --memory 500M --memory-reservation 256M nginx

List any currently running containers:

docker ps

List all containers, both running and stopped:

docker ps -a

Stop the Nginx container:

docker container stop nginx

Start a stopped container:

docker container start nginx

Delete a container (but it must be stopped first):

docker container rm nginx

When using Docker to manage containers, it is important to be able to keep the Docker engine up-to-date. In this lesson, we will discuss the process of both downgrading and upgrading the Docker engine.

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

Downgrade to a previous version:

sudo systemctl stop docker

sudo apt-get remove -y docker-ce docker-ce-cli

sudo apt-get update

sudo apt-get install -y docker-ce=5:18.09.4~3-0~ubuntu-bionic docker-ce-cli=5:18.09.4~3-0~ubuntu-bionic

docker version

Upgrade to a new version:

sudo apt-get install -y docker-ce=5:18.09.5~3-0~ubuntu-bionic docker-ce-cli=5:18.09.5~3-0~ubuntu-bionic

docker version

#######################################

Storing and accessing container logs is an essential part of managing containers. Docker logging drivers allow us to choose our own logging implementation to fit our particular needs. In this lesson, we will discuss logging drivers. We will also see how to customize the system default logging driver configuration, as well as how to override the defaults for individual containers.

**Relevant Documentation**

* <https://docs.docker.com/config/containers/logging/configure/>

**Lesson Reference**

Check the current default logging driver:

docker info | grep Logging

Edit daemon.json to set a new default logging driver configuration:

sudo vi /etc/docker/daemon.json

Add the configuration to daemon.json:

{

"log-driver": "json-file",

"log-opts": {

"max-size": "15m"

}

}

Restart docker after editing daemon.json:

sudo systemctl restart docker

Run a docker container, overriding the system default logging driver settings:

docker run --log-driver json-file --log-opt max-size=50m nginx