Docker commands

Install Docker CE on UBUNTU

Install the latest version of Docker CE and ContainerD
Command: Sudo apt-get install docker-ce docker-ce-cli containerd.io

Second Method

Command: curl -fsL get.docker.com -o get-docker.sh
Command: sh get-docker.sh
Use docker as a non root user
Command: sudo usermod -aG docker username

To check version

Command: sudo docker--version
Command: docker version
Verified cli can talk to engine

To check Info
Command: docker info
Most config values of engine

Installing Docker Machine

https://docs.docker.com/machine/install-machine/

```
base=https://github.com/docker/machine/releases/download/v0.16.0 &&
```

```
curl -L $base/docker-machine-$(uname -s)-$(uname -m) >/tmp/docker-machine &&
sudo mv /tmp/docker-machine /usr/local/bin/docker-machine &&
chmod +x /usr/local/bin/docker-machine
```

For checking Docker machine version
Command: docker-machine version

Installing Docker Compose

https://docs.docker.com/compose/install

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose &&
sudo chmod +x /usr/local/bin/docker-compose
```

Docker Command line Structure

Old (still work): docker <command> (Options)
new : docker <command> <sub-command> (Options)

Image:

Here we use alpine image

Pulling Image
Command: docker image pull alpine:latest

Listing Image
Command: docker image ls

Removing Image
You can not delete image until the last container using it has been stopped and destroyed.
Command: docker image rm alpine:latest

Container

Command: docker container run --publish 8080:80 --name webhost -d nginx:1.11 nginx
--publish -p
-d –detach
--env -e

Check container list
Command: docker container ls
Run container again
Then press ctrl +pq

Check All container list
Command: docker container ls -a

Check container list again with ps command
Command: docker container ps

To stop a container
Command: docker container stop name

To restart stopped container
Command: docker container start name

To remove container permanently
Command: docker container rm name

To remove running container
Command: docker container rm -f name

To remove or stop all images, container volume
docker stop $(docker ps -aq)
docker rm -f $(docker ps -aq)
docker container prune   # Remove all stopped containers
docker volume prune   # Remove all unused volumes
docker image prune   # Remove unused images
docker system prune   # All of the above, in this order: containers, volumes, images

To check all running Process
Command: ps aux | grep
Grep – is use for search

Process List in One Container
Command: docker container top

Details of One Container config
Command: docker container inspect
it shows meta data about the containe r(startup,config,volumes,networking)

Performance stats for all containers
Command: docker container stats

Getting A Shell Inside Container

To run a container from an image in an interactive mode
Command: Docker run -it Image [command]
-**i** interactive keep session open to receive terminal input
-t pseudo-tty simulates a real terminal, like what SSH does
sh, bash shell if run with it it will give you a terminal inside the running container

To go back to stop container shell
Command: docker container start -ai Image

To go back to running container shell with different process
Command: docker container exec -it name sh

Docker Networks: CLI Management

To check port of container
Command: docker container port Image
-**p** (--publishing) publishing ports is always in Host: CONTAINER format

To check IP Address of Container
Command: Docker container inspect --format '{{ .NetworkSettings.IPAddress }} ' Image
--format A common option for formatting the output of commands using
"GO templates"
By Default Container Ip Address is different from host IpAddress

Show Networks
Command: docker network ls
--network Bridge: DEFAULT Docker virtual network, which is NAT'ed behind the Host IP.
--network Host: it gains performance by skipping virtual networks but sacrifices security of container model
--network non: removes eth0 and only local host interface in container

Inspect A network
Command: docker network inspect networkName

Create a network
Command: docker network create –driver

Attach a network to a container
Command: docker network connect [networkName] [container]
Dynamically creates a NIC in a container on an existing virtual network

Detach a network from container
Command: docker network disconnect [networkName] [container]
Example:
Command: docker container run -d --name new_nginx --network my_app_net nginx

DNS and How Containers Find Each Other
docker container ls
docker network inspect TAB COMPLETION
docker container run -d --name my_nginx --network my_app_net nginx
docker container inspect TAB COMPLETION
docker container exec -it my_nginx ping new_nginx
docker container exec -it new_nginx ping my_nginx
docker network ls
docker container create --help

DNS Round Robin Testing

Remove container on close or exit
docker container run --rm -it centos:7 bash

docker network create dude
docker container run -d --net dude --net-alias search elasticsearch:2
docker container ls
docker container run --rm --net dude alpine nslookup search
docker container run --rm --net dude centos curl -s search:9200
docker container ls
docker container rm -f TAB COMPLETION

Docker Images

Images and Their Layers: Discover the Image Cache

Command: docker **history** nginx:latest
Command: docker image **inspect** nginx

Image Tagging and Pushing to Docker Hub
Command: docker image tag –help
docker tag assign one or more tags to an image

Official Repos: they live at the “root namespace” of the registry, so they don’t need account name in front of repo name
docker pull mysql/sysql-server
docker pull nginx:mainline

Re tag existing docker images
Command: docker image tag Source_Image[:Tag] Target_Image[:Tag]

Docker image push
Uploads changed layers to a image registry(default is Hub)
Command: docker image push

Docker Login in TERMINAL(DOCKER HUB)
Command: docker login

You can see your docker login credential here
Command: Cat .docker/config.json
Command: docker logout
Command: dockerFile

==========

Docker Storage Driver

Default docker driver for Centos 8+ and ubuntu 18.6 lts is **File-Overlay 2 (File Based Storage)**
DeviceMapper for Centos 7 and Earlier you can check Storage Driver using **Docker info**

You can Change Default Storage driver as Follows:
Go to File
sudo vi /usr/lib/systemd/system/docker.service
Add storage driver in following Line
ExecStart = /usr/bin/dockerd --storage-driver devicemapper
sudo systemctl daemon-reload
sudo systemctl restart Daemon
Recommended Way to change StorageDriver
sudo vi /etc/docker/daemon.json