Labs

<https://labs.play-with-docker.com/>

**Chapter 1: Installation and configuration of docker on Ubuntu vm**

#apt update or #apt-get update

#apt install docker.io

#docker info

# docker run hello-world

**Chapter 2: Making hands dirty with docker commands**

# sudo usermod -aG docker

# sudo systemctl restart docker

Login docker

$ docker help

$ docker version

$ docker info

$ docker run hello-world

$ docker ps

$ docker ps -a

$ docker images

$ docker run -d hello-world

$ docker logs 6e // 6e -> ist two letters of docker id from ps -a command

$ docker pull ubuntu

$ docker pull busybox

$ docker images

$ docker run -it ubuntu echo “hello world”

$ docker run -it busybox echo “hello world”

$ docker ps -a

$ docker run -d ubuntu echo “hello world”

$ docker run -d busybox echo “hello world”

$ docker logs a5

$ docker logs bc

$ docker ps

$ docker ps -a

**Chapter 3: interactive and detached mode**

$ docker search busybox

$ docker run -t -i busybox

Cntr pq

$ docker run -it busybox ls -l

$ docker run -it --entrypoint /bin/sh busybox

$ docker run -t -d busybox

$ docker ps

$ docker ps -a

**Chapter 4: Exec command on a running container**

$ docker exec 04c ls -l

$ docker exec 852 vmstat 2 4

$ docker exec e50 vmstat 2 4

$ docker exec e50 apt update

$ docker exec e50 apt -y upgrade

$ docker exec e50 ifconfig

$ docker exec e50 apt install -y net-tools

$ docker exec e50 ifconfig

$ ping 172.17.0.5

**Chapter 5: Entering a running container and exit without stopping a container**

**Part 1**

$ docker attach e50

apt update

ifconfig

ping 172.17.0.5

cntr pq

**Part 2**

$ docker stop e50

$ docker start e50

$ docker attach e50

cntr pq

**Part 3**

$docker ps

$docker stop e50

$docker ps -a |grep e50

$docker start e50

$docker ps |grep e50

$docker attach e50

#ls -l

#ifconfig

cntr pq

$docker exec e50 ls -l

$docker exec e50 ifconfig

**Chapter 6: Docker human friendly names**

#docker run -d busybox sleep 30000

#docker run -d busybox sleep 30000

#docker run -d busybox sleep 30000

#docker run -d busybox sleep 30000

#docker ps

# docker ps|awk '{print $1,$2,$11}'

**Chapter 7: PID Name Space**

$docker run -d --name server1 busybox sh -c “nc -l -p 0.0.0.0:7070”

$docker run -d --name server2 busybox sh -c “nc -l -p 0.0.0.0:8080”

docker exec server1 ps

docker exec server2 ps

docker rm server1

docker rm server2

$docker run -d --name server1 busybox nc -l -p 0.0.0.0:7070

$docker run -d --name server2 busybox nc -l -p 0.0.0.0:8080

docker exec server1 ps -ef

docker exec server2 ps -ef

**Chapter 8: More ways to get Container ID**

#docker create busybox

#CID=$(docker create busybox)

#CID=`docker create busybox`

#echo $CID

#docker create --cidfile /tmp/mycidfile busybox

#cat /tmp/mycidfile

**Chapter 9: Remove containers**

#docker ps

#docker rm 06 // through warning or use -f

#docker stop 06

#docker rm 06

#docker ps -a

**Chapter 10: Reuse the same name after deleting a container**

#docker run -it –name dispdate busybox date // this container will display date

#docker run -it –name dispdate busybox date // throw warning name already in use

#docker rm dispdate

#docker run -it --rm –name dispdate busybox date

#docker run -it --rm –name dispdate busybox date

#docker run -it --rm –name dispdate busybox date

#docker run -it --rm –name dispdate busybox date

#docker run -it --rm –name dispdate busybox date

**Chapter 11: Docker process life cycle examples**

docker create --name contaner-name image-name

docker run -it -d --name contaner-name image-name shell

docker pause <container-id/name>

docker unpause <container-id/name>

docker start <container-id/name>

docker stop <container-id/name>

docker stop $(docker ps -a -q) //stop all the containers

docker restart <container-id/name>

docker kill <container-id/name>

docker rm <container-id/name>

docker rm $(docker ps -q -f status=exited)

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

docker create --name mybusy busybox

docker run -it -d --name mybusy1 busybox sh

docker run -t -d --name mybusy2 busybox sh

docker ps

docker ps -a

docker pause mybusy1

docker unpause mybusy1

docker stop mybusy1

docker start mybusy1

docker stop $(docker ps -a -q) //stop all the containers

docker start $(docker ps -a -q) //stop all the containers

docker restart mybusy1

docker kill mybusy1

docker start mybusy1

docker stop mybusy1;docker rm mybusy1

docker rm mybusy2 -f

docker rm $(docker ps -q -f status=exited)

docker rm $(docker ps -q ) -f