Docker Container Workshop

Install Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt install -y apt-transport-https
sudo apt-get update
apt-cache policy docker-ce
sudo apt-get install -y docker-ce
sudo systemctl status docker
```

Executing the Docker Command Without Sudo (Optional)

```
sudo usermod -aG docker ${USER}
sudo usermod -aG docker samsul
#logout and close terminal
```

Using the Docker Command

```
docker [option] [command] [arguments]
docker
docker [docker-subcommand] --help
docker info
```

Running a Docker Container

```
docker run ubuntu:14.04 echo "Hello World"
docker run ubuntu ps ax
docker run -d -it ubuntu bash
docker run -d -it ubuntu:14.04 bash
docker run -d -it debian bash
docker exec -it [Container ID] bash
docker inspect [Container ID] |grep IP

docker run -d -P nginx
docker run -d -p 8080:80 nginx
docker ps -a
#Buka browser masukan IP host:[port]
```

Images and Volumes

```
mkdir test
cd test

vim Dockerfile
------------------------------
FROM ubuntu:14.04
RUN apt update
RUN apt install -y curl
RUN apt install -y vim
------------------------------

docker build -t samsulmaarif/test-image:1.0 .
docker images
docker run -d -it samsulmaarif/test-image:1.0 bash
docker ps -a
docker exec -it [CONTAINER-ID] bash
which vim
which curl

#Tips
docker ps -a -f status=exited
docker rm $(docker ps -a -f status=exited -q)
docker ps -a -f "name=nostalgic_stallman"
```

Push Image to hub.docker.com

```
docker login
docker push samsulmaarif/test-image:1.0
docker pull [username]/[nama-image]:1.0
docker rmi [nama-image]
```

Volumes

```
mkdir myvolume
cd myvolume
echo "ini file dari volumes" > index.html
docker run -d -it --hostname nginx1 --name nginx1 -p 8080:80 -v
/home/samsul/myvolume:/usr/share/nginx/html nginx
```

Portainer

https://hub.docker.com/r/portainer/portainer/

```
docker run -d -p 9000:9000 --restart always -v
/var/run/docker.sock:/var/run/docker.sock -v /opt/portainer:/data
portainer/portainer
```

Docker compose

```
sudo apt install -y docker-compose apache2-utils curl

mkdir compose
cd compose
docker volume create db_data

vim docker-compose.yml
------------------------------
version: '2'
services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     ports:
       - "8000:80"
     restart: always
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
volumes:
    db_data:
------------------------------

docker-compose up -d
```

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

Remove All Container

```
docker rm -f $(docker ps -a -q)
```

Docker Swarm Initializing The Cluster Manager

```
ip addr
docker swarm init --advertise-addr [node_ip_address]
```

Adding Nodes to the Cluster

```
docker swarm join --token [your_swarm_token] [manager_node_ip_address]:2377
```

Managing The Cluster

```
docker node ls

docker node --help
docker node inspect self --pretty
docker node inspect [hostname-worker] --pretty
```

Promote manager

```
docker node promote [name-node]
docker node demote [name-node-leader]
```

Run the Visualizer

```
docker run -it -d -p 8080:8080 -v /var/run/docker.sock:/var/run/docker.sock
dockersamples/visualizer
# Buka browser [ip-manager]:8080
```

Running Services in the Docker Swarm

```
docker service create -p 8090:80 --name webserver nginx
docker service ls
docker service ps webserver
```

Scale out

```
docker service scale webserver=10
docker service ps webserver
```