CLOUD
DOCKER 2

Arranged By:

Rajendra Rakha Arya Prabaswara

(1941720080/21)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

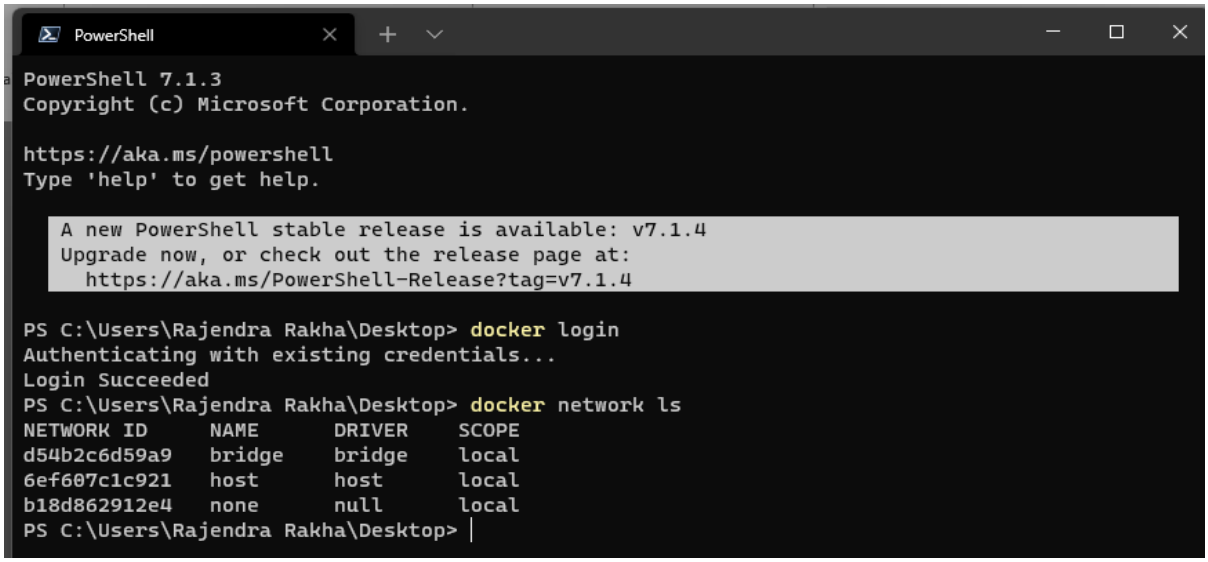
POLITEKNIK NEGERI MALANG

Practicum 1

(Docker Networks)

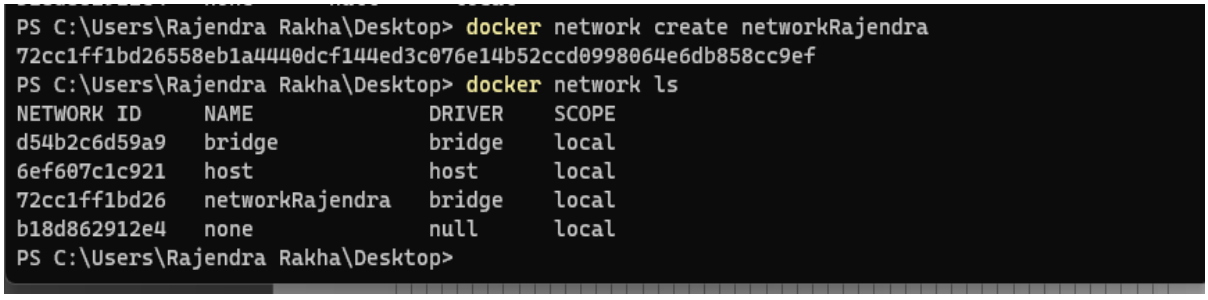
1. Create & Check Docker Networks , Type Syntax below :

   ➔ **Docker login**
   ➔ **Docker network ls**



2. Create new Network

   ➔ **Docker network create networkRajendra**
   ➔ **Docker network ls**



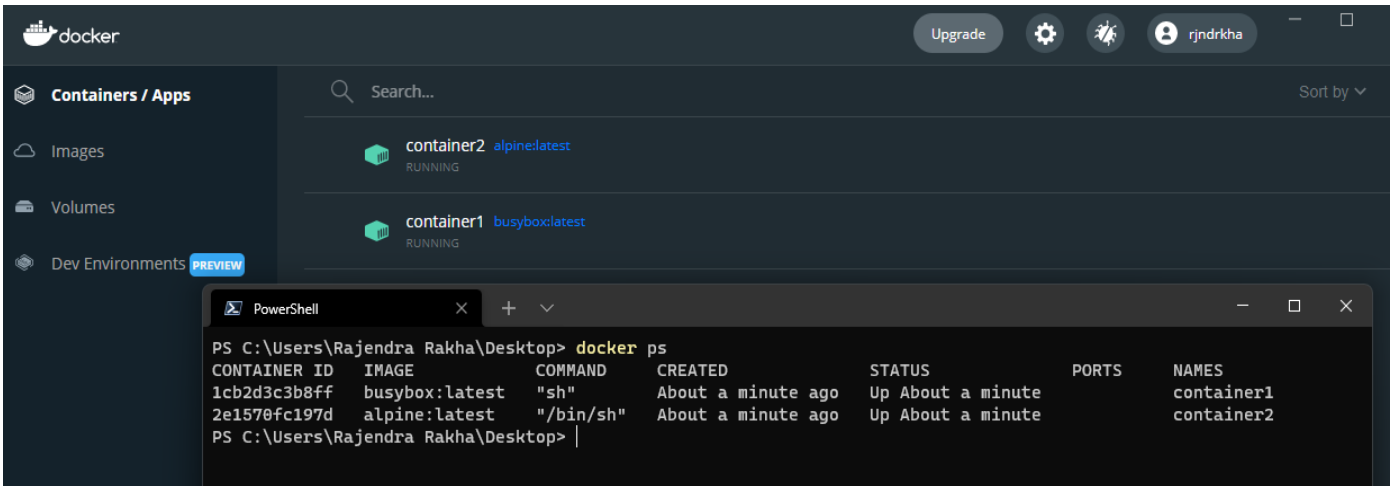
3. Check Detailed Network have been created

➔ **Docker network inspect networkRajendra**

```
PS C:\Users\Rajendra Rakha\Desktop> docker network inspect networkRajendra
[
    {
        "Name": "networkRajendra",
        "Id": "72cc1ff1bd26558eb1a4440dcf144ed3c076e14b52ccd0998064e6db858cc9ef",
        "Created": "2021-11-15T03:49:29.74231482",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
```

4. Create Container & Connect ContainerTo Network

➔ **docker run -it --name container1 --net=networkRajendra --rm busybox:latest /**
➔ **docker run -it --name container2 --net=networkRajendra --rm alpine:latest /**

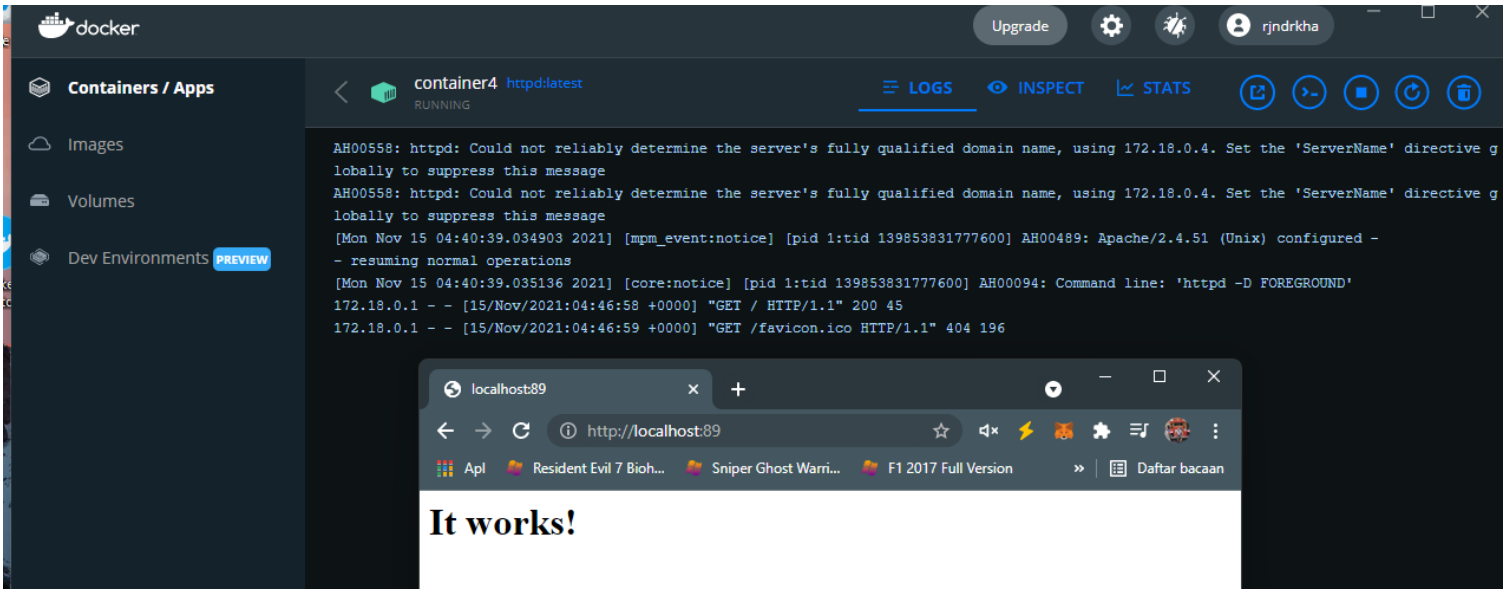


5. Run inspect to find out the detailed information of each container

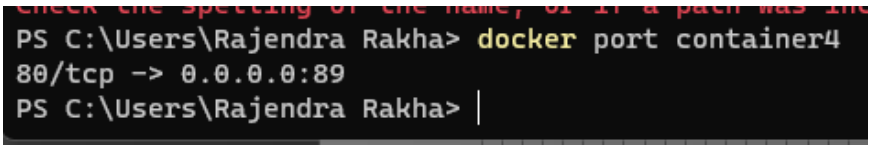
➔ **docker inspect container1 | findstr IPAddress**
➔ **docker inspect container2 | findstr IPAddress**

```
PS C:\Users\Rajendra Rakha\Desktop> docker inspect container1 | findstr IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.3",
                    "IPAddress": "172.17.0.3",
PS C:\Users\Rajendra Rakha\Desktop> docker inspect container2 | findstr IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.2",
                    "IPAddress": "172.17.0.2",
```

6. Connect & Run Docker Telnet



7. Check Container4 Port



Practicum 2

(Docker Volume)

1. Create folder namely docker-shared
2. Open in Terminal , Then Run Syntax below :

   ➔ **docker run -v "C:\Users\Rajendra Rakha\Downloads\docker-shared:\volume" -it busybox**
   ➔ **ls**
   ➔ **cd volume/**
   ➔ **touch log.txt**

```
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker run -v C:\Users\Rajendra Rakha\Downloads\docker-shared:\volum
e -it busybox
docker: invalid reference format: repository name must be lowercase.
See 'docker run --help'.
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker run -v "C:\Users\Rajendra Rakha\Downloads\docker-shared:\volu
me" -it busybox
/ # ls
\volume  bin      dev      etc      home     proc     root     sys      tmp      usr      var
/ # cd volume/
sh: cd: can't cd to volume/: No such file or directory
/ # cd volume\
> ls
sh: cd: can't cd to volumels: No such file or directory
/ # touch log.txt
/ #
```

3. Checking the volume that already exists.

```
/ # touch log.txt
/ # exit
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker volume ls
DRIVER    VOLUME NAME
local     87d2de2dffd6b702ca74f4f248c9f38c4a664af7bbacd39b0f754e6aae9b81c3
PS C:\Users\Rajendra Rakha\Downloads\docker-shared>
```

4. Create New Volume

   ➔ **docker volume create volumeRajendra**
   ➔ **docker volume ls**

```
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker volume create volumeRajendra
volumeRajendra
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker volume ls
DRIVER    VOLUME NAME
local     87d2de2dffd6b702ca74f4f248c9f38c4a664af7bbacd39b0f754e6aae9b81c3
local     volumeRajendra
PS C:\Users\Rajendra Rakha\Downloads\docker-shared>
```

5. Run the container by including the volume

➔ **docker run -it --name container3 -v volumeRajendra:\volume alpine:latest**
➔ **touch log.txt**

```
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker volume ls
DRIVER    VOLUME NAME
local     87d2de2dffd6b702ca74f4f248c9f38c4a664af7bbacd39b0f754e6aae9b81c3
local     volumeRajendra
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker run -it --name container3 -v volumeRajendra:\volume alpine:la
test
/ # touch log.txt
/ #
```

Create Another Container

➔ **docker run -it --name container4 -v volumeRajendra:\volume alpine:latest**

```
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker run -it --name container4 -v volumeRajendra:\volume alpine:la
test
/ # ls volume\
> log.txt
```

6. Delete Volume

```
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker volume rm volumeRajendra
volumeRajendra
PS C:\Users\Rajendra Rakha\Downloads\docker-shared> docker volume ls
DRIVER    VOLUME NAME
local     87d2de2dffd6b702ca74f4f248c9f38c4a664af7bbacd39b0f754e6aae9b81c3
PS C:\Users\Rajendra Rakha\Downloads\docker-shared>
```

Practicum 3

(Docker Compose)

1. Open VM then clone this synatax below to your VM

   ➔ **sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/dockercompose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose**

```
Last login: Mon Nov 15 04:27:20 2021 from 36.85.61.128
ubuntu@vm-ubuntu-rajendra:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/dockercompose-$(uname -s)-$(uname -m)" -
o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100     9  100     9    0     0     38      0 --:--:-- --:--:-- --:--:--    38
```

2. Change docker compose to executable using syntax below :

   ➔ **sudo chmod +x /usr/local/bin/docker-compose**

```
ubuntu@vm-ubuntu-rajendra:~$ sudo chmod +x /usr/local/bin/docker-compose
ubuntu@vm-ubuntu-rajendra:~$
```

3. Check Docker Compose Version & Clone aplikasi akademik From Github

   ➔ **git clone https://github.com/0d3ng/aplikasi-akademik.git**
   ➔ **ls**
   ➔ **cd aplikasi-akademik**
   ➔ **docker-compose --version**

```
ubuntu@vm-ubuntu-rajendra:~$ git clone https://github.com/0d3ng/aplikasi-akademik.git
Cloning into 'aplikasi-akademik'...
remote: Enumerating objects: 176, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 176 (delta 14), reused 41 (delta 7), pack-reused 122
Receiving objects: 100% (176/176), 67.55 KiB | 974.00 KiB/s, done.
Resolving deltas: 100% (40/40), done.
ubuntu@vm-ubuntu-rajendra:~$ ls
aplikasi-akademik  cloud-docker-java-sample
ubuntu@vm-ubuntu-rajendra:~$ cd aplikasi-akademik
```

```
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$
```

4. Build Image

➔ **Git checkout cloud-docker**
➔ **docker build -t 0d3ng/komputasi-awan-akademik .**

```
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$ git checkout cloud-docker
Branch 'cloud-docker' set up to track remote branch 'cloud-docker' from 'origin'.
Switched to a new branch 'cloud-docker'
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$ docker build -t 0d3ng/komputasi-awan-akad
emik .
Sending build context to Docker daemon  245.2kB
Step 1/12 : FROM maven:3.6.3-openjdk-8-slim AS build
3.6.3-openjdk-8-slim: Pulling from library/maven
75646c2fb410: Pull complete
875a154571f0: Pull complete
8d86e30204e0: Pull complete
6b9efcfa6e72: Pull complete
e5a0d12a178b: Pull complete
5933e326ee4e: Pull complete
1f98777813c0: Pull complete
7fdfc10cc758: Pull complete
Digest: sha256:195e9c227ad891282e80602cac2372a3085ecf4ceefbb395558ffe0f7bb0b9aa
Status: Downloaded newer image for maven:3.6.3-openjdk-8-slim
 ---> f3f54c8fc76f
Step 2/12 : RUN mkdir -p /workspace
 ---> Running in 3602629be882
```

5. Create Data Under /opt So data from MYSQL not lost when container removed & Run Docker Compose

➔ **sudo mkdir /opt/data/**
➔ **docker-compose up -d**

```
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$ sudo mkdir /opt/data/
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$ docker-compose up
Creating network "backend-network" with the default driver
Pulling mysql (mysql:latest)...
latest: Pulling from library/mysql
b380bbd43752: Downloading [================>                                  ]  8.956MB/27.14MB
f23cbf2ecc5d: Download complete
30cfc6c29c0a: Download complete
b38609286cbe: Waiting
8211d9e66cd6: Waiting
2313f9eeca4a: Waiting
7eb487d00da0: Waiting
4d7421c8152e: Waiting
77f3d8811a28: Waiting
cce755338cba: Waiting
69b753046b9f: Waiting
b2e64b0ab53c: Waiting
```

6. Check the results in the browser by accessing your VM

➔ **Oracle -> Compute -> IP PUBLIC (132.226.129.209)**

```
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$ docker-compose ps
    Name                  Command               State                          Ports
-------------------------------------------------------------------------------------------------------------
akademik-app   bash -c java -jar /opt/app.jar   Up      0.0.0.0:8090->8090/tcp,:::8090->8090/tcp
mysql-db       docker-entrypoint.sh mysqld      Up      0.0.0.0:3306->3306/tcp,:::3306->3306/tcp, 33060/tcp
ubuntu@vm-ubuntu-rajendra:~/aplikasi-akademik$
```