Topics in Computing Lab Assignment 7 : Docker

Team R : Vandita Goyal (2016ucp1004)
Nidheesh Panchal (2016ucp1008)
G. Jahnvi (2016ucp1332)

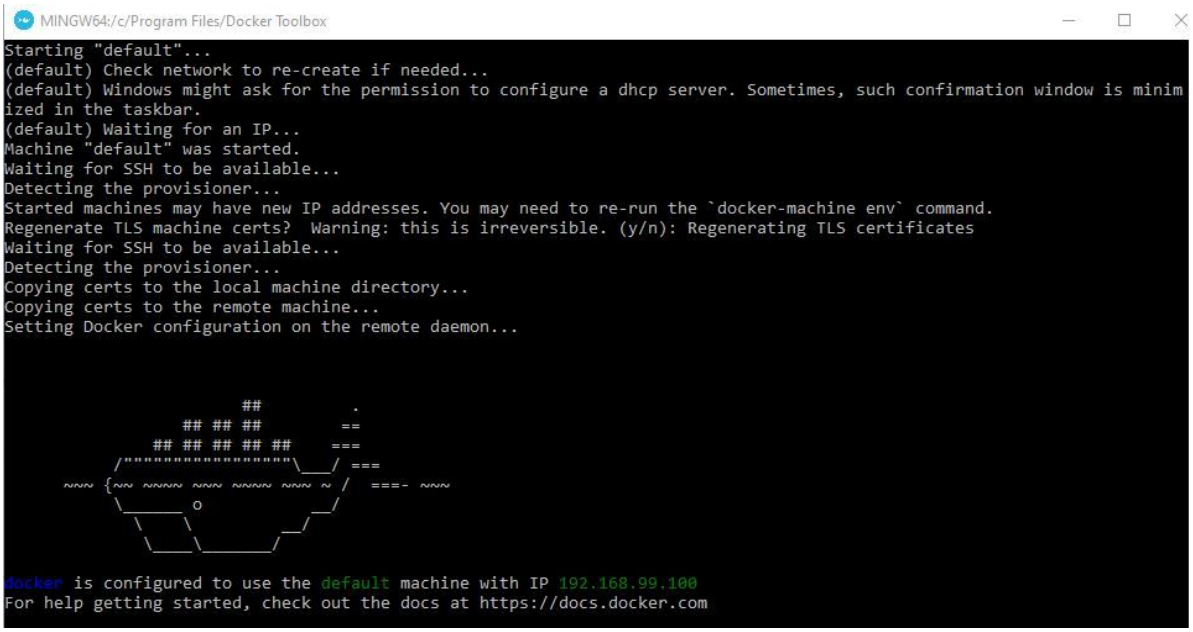
Objective:

1. Building a Docker container on host
2.Creating a Docker Swarm
3.Make use of Docker Swarm to join multiple Docker hosts to the cluster
4.Manage application data using volumes and bind mounts
5.Scale your app with kubernetes
6.Scale your app as a swarm service

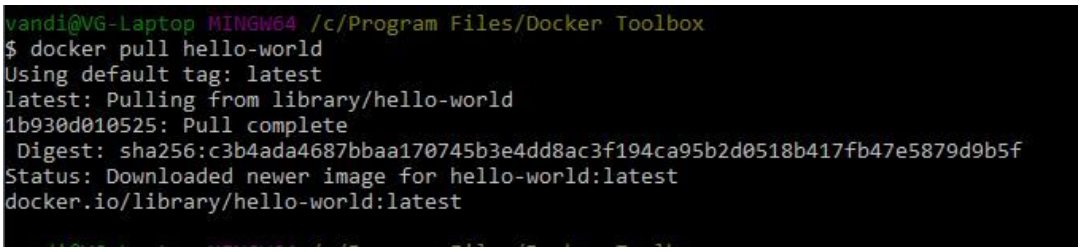
Docker
Docker swarm

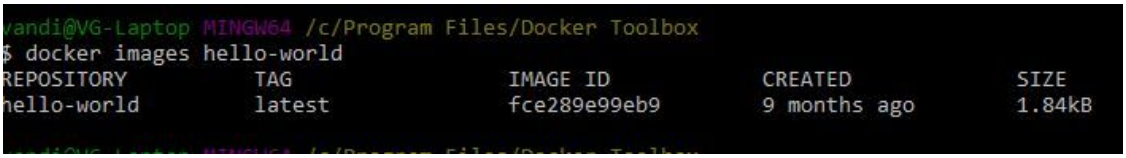
Implementation:

1. Installation of Docker Toolbox:
    a. Download the latest version of Docker Toolbox by going to Toolbox Releases and download the latest .exe file.
    b. Install Docker Toolbox by double-clicking the installer.
    c. Press Next to accept all the defaults and then Install.



2. Create a container (Hello World):
    a. Search on Docker Hub for an image
    b. Pull the said image





    c. Run the image (creates a container)

```
vandi@VG-Laptop MINGW64 /c/Program Files/Docker Toolbox
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

```
vandi@VG-Laptop MINGW64 /c/Program Files/Docker Toolbox
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                     PORTS
        NAMES
b5d93d869e0d        hello-world         "/hello"            2 minutes ago       Exited (0) 2 minutes ago
        furious_leakey
ce7937b0a779        hello-world         "/hello"            4 minutes ago       Exited (0) 4 minutes ago
        prickly_austin
```

3. Create an image and container (Web Application):
    a. Clone an example project from Github
    git clone -b v1 https://github.com/docker-training/node-bulletin-board
    cd node-bulletin-board/bulletin-board-app

    b. Write a DockerFile

    c. Make sure you're in the directory node-bulletin-board/bulletin-board-app in a terminal or powershell, and build your bulletin board image:
    docker image build -t bulletinboard:1.0 .

    d. Start a container based on your new image:
    docker container run --publish 8000:8080 --detach --name bb bulletinboard:1.0

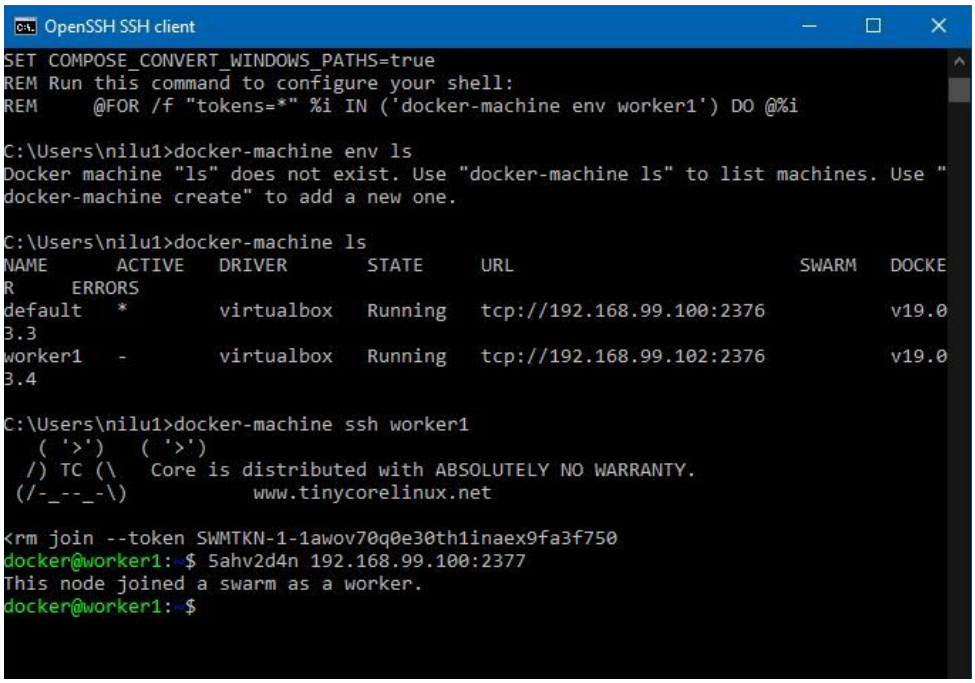
    e. Visit your application in a browser at localhost:8000.

4. Create a Docker Swarm and add workers to it:

a. Open a terminal and ssh into the machine where you want to run your manager node
docker-machine ssh manager1

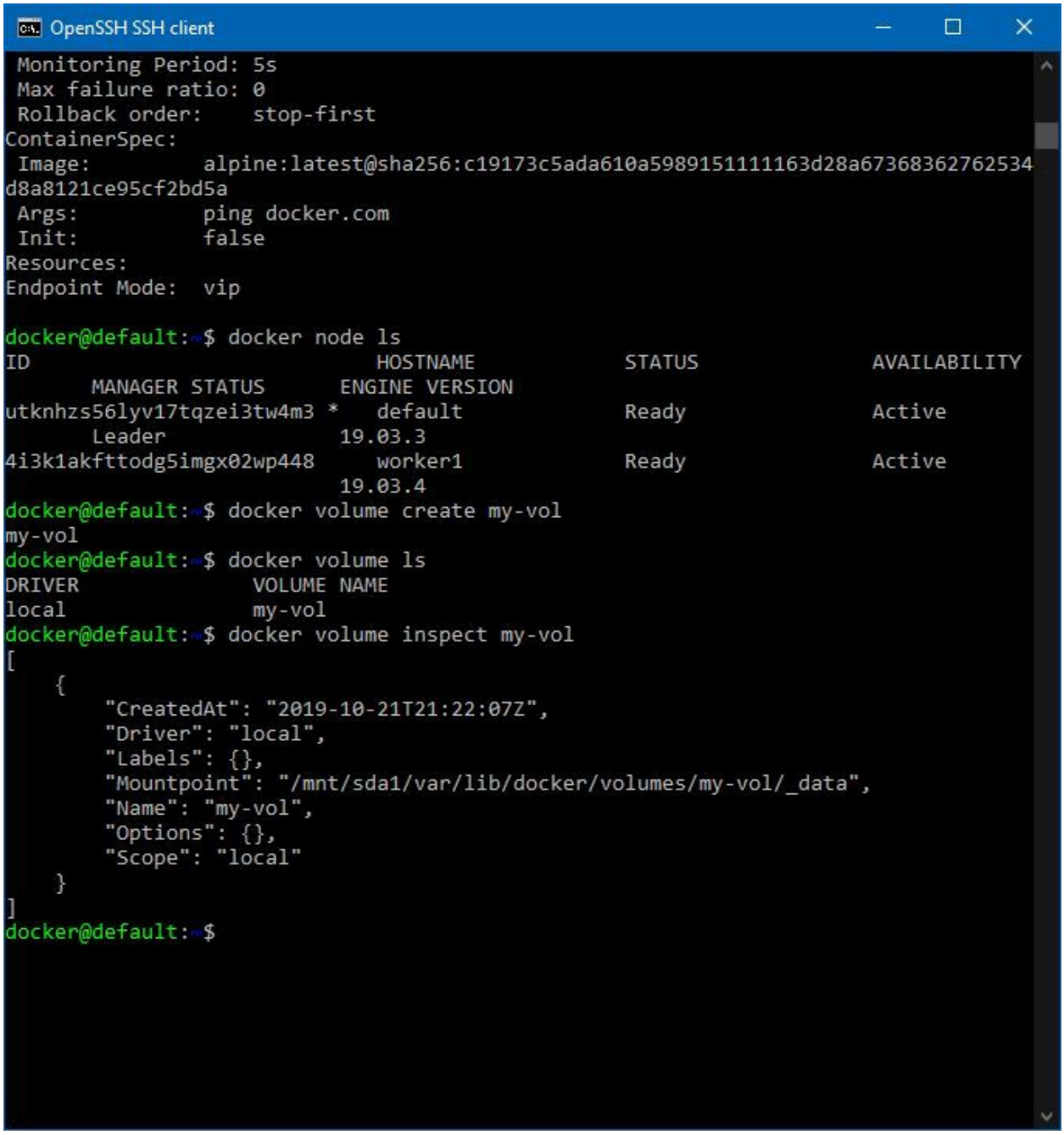
b. Run the following command to create a new swarm:
docker swarm init --advertise-addr <MANAGER-IP>

c. Run the command that comes as output of the above command in the worker Docker machine :
docker swarm join --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c 192.168.99.100:2377



5. Manage application data using volumes and bind mounts:

a. Create a volume:
docker volume create my-vol

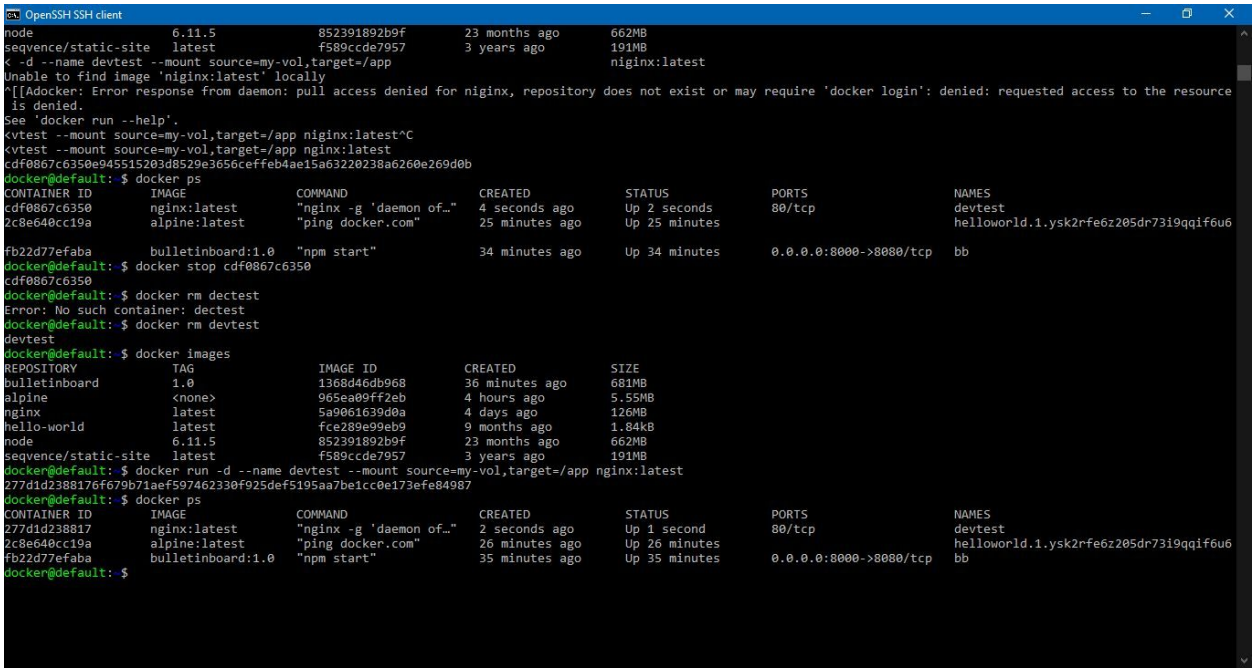


b. Run the following command to run the container with the volume
docker run -d --name devtest --mount source=myvol2,target=/app nginx:latest

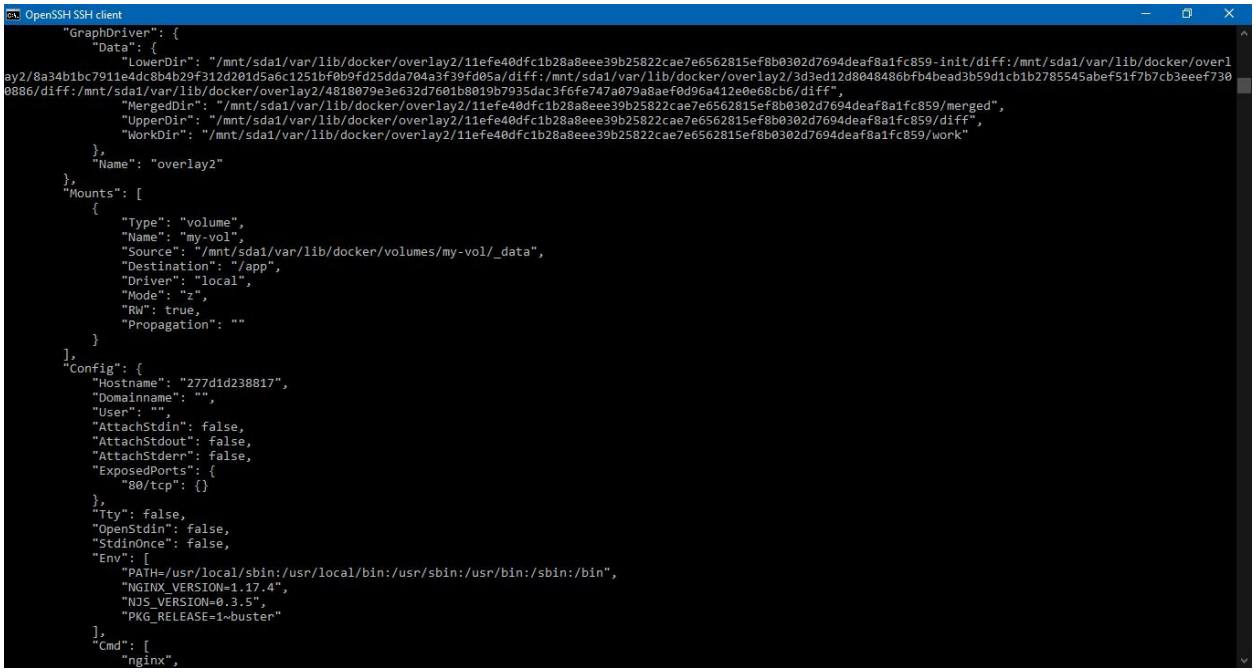
c. Use **docker inspect devtest** to verify that the volume was created and mounted correctly

d. To start a service with multiple containers of the same image with a shared volume:

docker service create -d --replicas=4 --name devtest-service --mount source=myvol2,target=/app nginx:latest



e. Use **docker service ps devtest-service** to verify that the service is running



6. Scale your app as a swarm service:

a. Write a simple stack file to run and manage our bulletin board. Place the following in a file called bb-stack.yaml

```
version: '3.7'

services:
  bb-app:
    image: bulletinboard:1.0
    ports:
      - "8000:8080"
```

b. Deploy your application to Swarm:
docker stack deploy -c bb-stack.yaml demo

c. List your service:
docker service ls

d. Scale up service