

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.


```
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
```

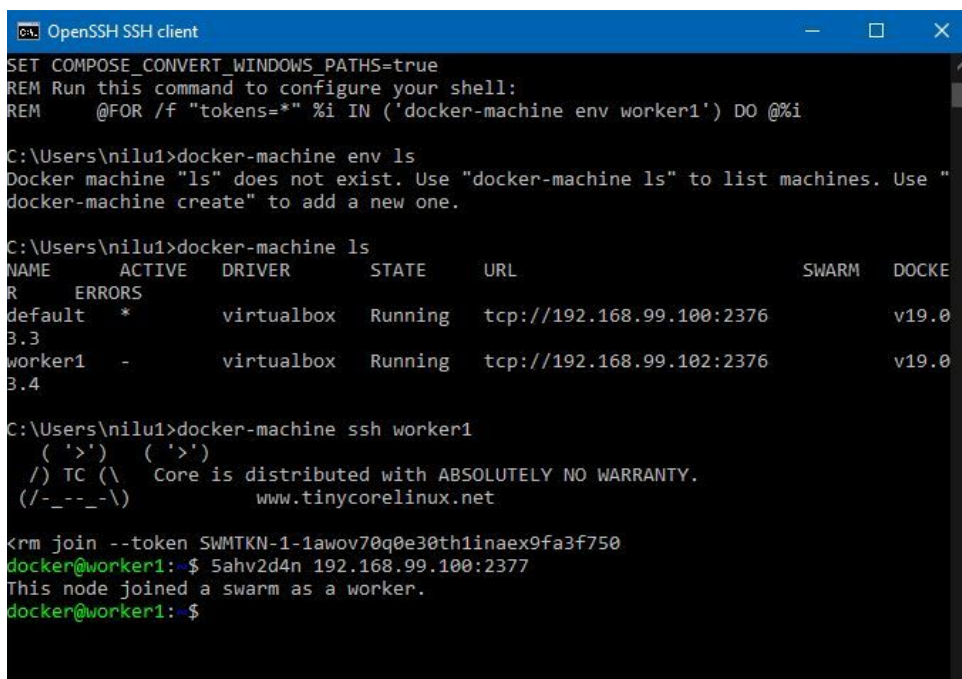
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
b5d93d869e0d	hello-world	"/hello"	2 minutes ago	Exited (0) 2 minutes ago	
ce7937b0a779	hello-world	"/hello"	4 minutes ago	Exited (0) 4 minutes ago	

3. Create an image and container (Web Application):

- 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
- Write a DockerFile
- 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 .
- Start a container based on your new image:
docker container run --publish 8000:8080 --detach --name bb
bulletinboard:1.0
- 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-8vxv8rssmk74
3ojnwacrr2e7c 192.168.99.100:2377**



```
OpenSSH SSH client
SET COMPOSE_CONVERT_WINDOWS_PATHS=true
REM Run this command to configure your shell:
REM @FOR /f "tokens=*" %i IN ('docker-machine env worker1') DO @%i

C:\Users\nilu1>docker-machine env ls
Docker machine "ls" does not exist. Use "docker-machine ls" to list machines. Use "
docker-machine create" to add a new one.

C:\Users\nilu1>docker-machine ls
NAME      ACTIVE   DRIVER        STATE     URL                  SWARM   DOCKE
R        ERRORS
default   *        virtualbox    Running   tcp://192.168.99.100:2376   v19.0
B.3
worker1   -        virtualbox    Running   tcp://192.168.99.102:2376   v19.0
B.4

C:\Users\nilu1>docker-machine ssh worker1
( ' > ' ) ( ' > ' )
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
(/-_-_-\)   www.tinycorelinux.net

krm join --token SWMTKN-1-1awov70q0e30th1inaex9fa3f750
docker@worker1:~$ 5ahv2d4n 192.168.99.100:2377
This node joined a swarm as a worker.
docker@worker1:~$
```

5. Manage application data using volumes and bind mounts:

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

```
OpenSSH SSH client
Monitoring Period: 5s
Max failure ratio: 0
Rollback order: stop-first
ContainerSpec:
  Image: alpine:latest@sha256:c19173c5ada610a5989151111163d28a67368362762534d8a8121ce95cf2bd5a
  Args: ping docker.com
  Init: false
Resources:
Endpoint Mode: vip

docker@default:~$ docker node ls
ID                                HOSTNAME    STATUS    AVAILABILITY
MANAGER STATUS    ENGINE VERSION
utknhzs56lyv17tqzei3tw4m3 *    default    Ready    Active
Leader
4i3k1akfttdg5imgx02wp448        worker1    Ready    Active
19.03.4

docker@default:~$ docker volume create my-vol
my-vol
docker@default:~$ docker volume ls
DRIVER    VOLUME NAME
local     my-vol
docker@default:~$ docker volume inspect my-vol
[
  {
    "CreatedAt": "2019-10-21T21:22:07Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/mnt/sda1/var/lib/docker/volumes/my-vol/_data",
    "Name": "my-vol",
    "Options": {},
    "Scope": "local"
  }
]
```

- 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

```

OpenSSH SSH client
node        6.11.5      852391892b9f    23 months ago    662MB
sequence/static-site    latest      f589ccde7957    3 years ago      191MB
$ -d --name devtest --mount source=my-vol,target=/app
Unable to find image 'nginx:latest' locally
^[[Adocker: Error response from daemon: pull access denied for nginx, repository does not exist or may require 'docker login': denied: requested access to the resource
is denied.
See 'docker run --help'.
$ kvttest --mount source=my-vol,target=/app nginx:latest^C
$ kvttest --mount source=my-vol,target=/app nginx:latest
cdf0867c6350e945515203d8529e3656ceffeb4ae15a63220238a6260e269d0b
docker@default: $ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS              NAMES
cdf0867c6350       nginx:latest        "nginx -g 'daemon of..." 4 seconds ago      Up 2 seconds        80/tcp             devtest
2c8e640cc19a       alpine:latest       "ping docker.com"         25 minutes ago     Up 25 minutes        80/tcp             helloworld.1.ysk2rfe6z205dr7319qqif6u6
fb22d77efaba       bulletinboard:1.0   "npm start"               34 minutes ago     Up 34 minutes        0.0.0.0:8000->8080/tcp    bb
docker@default: $ docker stop cdf0867c6350
cdf0867c6350
docker@default: $ docker rm dectest
Error: No such container: dectest
docker@default: $ docker rm devtest
devtest
docker@default: $ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
bulletinboard       1.0                1368d46db968       36 minutes ago    681MB
alpine               <none>             965ea09ff2eb       4 hours ago       5.55MB
nginx               latest            5a9061639d0a       4 days ago        126MB
hello-world         latest            fce289e99eb9       9 months ago      1.84kB
node                6.11.5            852391892b9f       23 months ago     662MB
sequence/static-site    latest      f589ccde7957       3 years ago       191MB
docker@default: $ docker run -d --name devtest --mount source=my-vol,target=/app nginx:latest
277d1d2388176f69b71aef597462330f925def5195aa7be1cc0e173efe84987
docker@default: $ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS              NAMES
277d1d238817       nginx:latest        "nginx -g 'daemon of..." 2 seconds ago      Up 1 second         80/tcp             devtest
2c8e640cc19a       alpine:latest       "ping docker.com"         26 minutes ago     Up 26 minutes        80/tcp             helloworld.1.ysk2rfe6z205dr7319qqif6u6
fb22d77efaba       bulletinboard:1.0   "npm start"               35 minutes ago     Up 35 minutes        0.0.0.0:8000->8080/tcp    bb
docker@default: $

```

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

```

OpenSSH SSH client
"GraphDriver": {
  "Data": {
    "LowerDir": "/mnt/sda1/var/lib/docker/overlay2/11efe40dfc1b28a8ee39b25822cae7e6562815ef8b0302d7694deaf8a1fc859-init/diff:/mnt/sda1/var/lib/docker/overl
ay2/8a34b1bc791e4dc8b4b29f312d201d5a6c1251bf0b9fd25dda704a3f39fd05a/diff:/mnt/sda1/var/lib/docker/overlay2/3d3ed12d8048486bfb4bead3b59d1cb1b2785545abef51f7b7cb3eeef730
0886/diff:/mnt/sda1/var/lib/docker/overlay2/4818079e3e632d7601b8019b7935dac3f6fe747a079a8aef0d96a412e0e68cb6/diff",
    "MergedDir": "/mnt/sda1/var/lib/docker/overlay2/11efe40dfc1b28a8ee39b25822cae7e6562815ef8b0302d7694deaf8a1fc859/merged",
    "UpperDir": "/mnt/sda1/var/lib/docker/overlay2/11efe40dfc1b28a8ee39b25822cae7e6562815ef8b0302d7694deaf8a1fc859/diff",
    "WorkDir": "/mnt/sda1/var/lib/docker/overlay2/11efe40dfc1b28a8ee39b25822cae7e6562815ef8b0302d7694deaf8a1fc859/work"
  },
  "Name": "overlay2"
},
"Mounts": [
  {
    "Type": "volume",
    "Name": "my-vol",
    "Source": "/mnt/sda1/var/lib/docker/volumes/my-vol/_data",
    "Destination": "/app",
    "Driver": "local",
    "Mode": "z",
    "RW": true,
    "Propagation": ""
  }
],
"Config": {
  "Hostname": "277d1d238817",
  "Domainname": "",
  "User": "",
  "AttachStdin": false,
  "AttachStdout": false,
  "AttachStderr": false,
  "ExposedPorts": {
    "80/tcp": {}
  },
  "Tty": false,
  "OpenStdin": false,
  "StdinOnce": false,
  "Env": [
    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
    "NGINX_VERSION=1.17.4",
    "NJS_VERSION=0.3.5",
    "PKG_RELEASE=1~buster"
  ],
  "Cmd": [
    "nginx",

```


6. Scale your app as a swarm service:

- 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"
```

- Deploy your application to Swarm:
`docker stack deploy -c bb-stack.yaml demo`
- List your service:
`docker service ls`
- Scale up service

```
oxvsd8euighh  demo_bb-app  replicated  1/1  bulletinboard:1.0  *:8000->8080/tcp
p8ohqzqwc2z9  helloworld  replicated  1/1  alpine:latest

docker@default:~$ docker service scale oxvsd8euighh=5
oxvsd8euighh scaled to 5
overall progress: 5 out of 5 tasks
5: running
1/5: running
3/5: running
5/5: running
verify
Service converged
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
b557239e3602  bulletinboard:1.0  "npm start"             15 seconds ago  Up 10 seconds  demo_bb-app.5.1wq10qe5jjguj4cbztrbazfy
a8da254d579c  bulletinboard:1.0  "npm start"             15 seconds ago  Up 10 seconds  demo_bb-app.3.u0ny6zpgpkobjblh470b07yc
739b7342608f  bulletinboard:1.0  "npm start"             15 seconds ago  Up 10 seconds  demo_bb-app.2.jl34cfzgulgmftjkrghr6h2
7170a974632b  bulletinboard:1.0  "npm start"             23 seconds ago  Up 22 seconds  demo_bb-app.4.xke5zowmz3ip5wtrfufaijg
762616eaa37b  bulletinboard:1.0  "npm start"             7 minutes ago   Up 7 minutes   demo_bb-app.1.x1il4ji421zy97d8ls3vfadi
1a62311b3726  nginx:latest      "nginx -g 'daemon of..." 42 minutes ago  Up 42 minutes  80/tcp         devtest1
277d1d238817  nginx:latest      "nginx -g 'daemon of..." 44 minutes ago  Up 44 minutes  80/tcp         devtest
2c8e640cc19a  alpine:latest     "ping docker.com"          About an hour ago  Up About an hour  helloworld.1.ysk2rfe6z205dr7319qq1f6u6
fb22d77efaba  bulletinboard:1.0  "npm start"             About an hour ago  Up About an hour  0.0.0.0:8000->8080/tcp  bb

docker@default:~$ docker service ps oxvsd8euighh
ID            NAME          IMAGE      NODE     DESIRED STATE  CURRENT STATE        CURRENT STATE        ERROR
PORTS
x1il4ji421zy  demo_bb-app.1  bulletinboard:1.0  default  Running         Running 7 minutes ago
1yjj9kp5uzia  \ demo_bb-app.1  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
ve4n4lf6vv5c  \ demo_bb-app.1  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
z0e1z2prpqpy  \ demo_bb-app.1  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
ws4gvctcm6vok  \ demo_bb-app.1  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
jl34cfzgulgm  demo_bb-app.2  bulletinboard:1.0  default  Running         Running 48 seconds ago
vni3kno1tte4  \ demo_bb-app.2  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
p1zq49xbs382  \ demo_bb-app.2  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
u0ny6zpgpkobj  demo_bb-app.3  bulletinboard:1.0  default  Running         Running 48 seconds ago
u2g3072c51s4  \ demo_bb-app.3  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
mghj36pdjkoz  \ demo_bb-app.3  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
xke5zowmz3ip  demo_bb-app.4  bulletinboard:1.0  default  Running         Running about a minute ago
1wq10qe5jjgu  demo_bb-app.5  bulletinboard:1.0  default  Running         Running 48 seconds ago
fj9xy89pgnk9  \ demo_bb-app.5  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."
2q7g16k1al66  \ demo_bb-app.5  bulletinboard:1.0  worker1  Shutdown       Rejected less than a second ago  "No such image: bulletinboard:..."

docker@default:~$
```

