

## **Summary**

## Session No - 8

- Installing docker in a virtual machine
  - Step 1:- Installing yum-utils
    - Command:- yum install -y yum-utils

```
[root@localhost ~]# yum install -y yum-utils
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Repository 'dvdl' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
dvd1

2.7 MB/s | 2.7 kB 00:00
dvd2
2.7 MB/s | 2.8 kB 00:00
```

- Step 2:- Configuring Yum
  - Command :- yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/dockerce.repo

```
[root@localhost ~]# yum-config-manager --add-repo https://download.docker.com/linux/r
hel/docker-ce.repo
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to
register.

Repository 'dvd1' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
Adding repo from: https://download.docker.com/linux/rhel/docker-ce.repo
```

Step 3:- Checking the repository

```
[root@localhost ~]# yum repolist

Updating Subscription Management repositories.

Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Repository 'dvd1' is missing name in configuration, using id.

Repository 'dvd2' is missing name in configuration, using id.

Repository 'dvd2' is missing name in configuration, using id.

repo id

docker-ce-stable

docker-ce-stable

dvd1

dvd2

[root@localhost ~]#
```

- o Step 4:- Installing Docker
  - Command:- Yum install docker-ce

· Creating a network in the docker

```
[root@ip-172-31-3-96 ~] # docker network create lwnet cc2dd7ded56e9c578cedb37b1f53d7e26b82503fd8a112c2dff3db6df8961e46 [root@ip-172-31-3-96 ~] # docker network ls NETWORK ID NAME DRIVER SCOPE bb1ebc24578b bridge bridge local 7ad773827e76 host host local cc2dd7ded56e lwnet bridge local 141758ddf6bb none null local
```

If we don't give a network name docker gives the default range

```
"Name": "lwnet",

"Id": "cc2dd7ded56e9c578cedb37b1f53d7e26b82503fd8a112c2dff3db6df896le46",

"Created": "2022-10-20T15:42:13.920186377Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": {

"Subnet": "172.18.0.0/"6",

"Gateway": "172.18.0.1"

}
```

- Volume is a keyword in the docker-compose file for getting storage
- Networks is a keyword in the docker-compose file for creating a network
- Creating a network in the docker-compose file
  - Manifest /code file

```
[root@ip-172-31-3-96 mycode]# cat docker-compose.yml
version: "3.8"

networks:
   lwnet:
        driver: bridge
```

- Launching the Word press & MySQL application with the dockercompose file
  - Manifest/code file for MY-SQL database & word press application

```
version: "3.8"

services:
db:
    image: mysql:latest
    container_name: mydbl
    networks:
    - lwnet
    volumes:
    - /data:/var/lib/mysql
    environment:
    - MYSQL_ROOT_PASSWORD=redhat
    - MYSQL_DATABASE=blogdb
    - MYSQL_USER=vimal
    - MYSQL_PASSWORD=redhat
```

## **Docker Certified Associate**

```
wp:
   image: wordpress:latest
   container_name: mywpl
   networks:
   - lwnet
   ports:
   - 8080:80
```

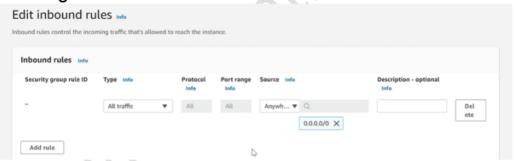
 Launching MY-SQL database & word press container from the docker-compose file



Checking the containers



o Allowing in firewall



Connecting database to word press application



 Docker-compose logs (service name) is used to see the logs of containers

Now with one click, we can stop the entire three-tier application

```
[root@ip-172-31-3-96 mycode]# docker-compose stop
[+] Running 0/0
" Container mywp1 Stopping
[+] Running 0/2s
" Container mywp1 Stopping
[+] Running 0/2s
" Container mywp1 Stopping
[+] Running 0/2s
" Container mywp1 Stopping
[+] Container mywp1 Stopping
0.4s
" Container mydb1 Stopping
```

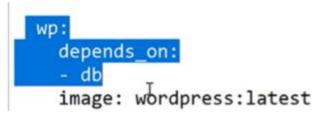
 Docker-compose is command will tell us which file we used to launch the container

```
CONFIG FILES
                      STATUS
NAME
                                            /mycode/docker-compose.yml
                      running(2)
[root@ip-172-31-3-96 mycode] # docker
                                         compose ps
                     COMMAND
                                                 SERVICE
                                                                        STATUS
                                                                                              3306/tcp, 33060/tcp
0.0.0.0:8080->80/tcp, :::8080-
                      "docker-entrypoint.s.."
mydb1
                                                                        running
                      "docker-entrypoint.s.."
                                                                        running
```

- Word press application is dependent on MY-SQL database for storing data
- Word press has given some environmental variables to automate the login process

```
wp:
  image: wordpress:latest
  container_name: mywp1
  networks:
  - lwnet
  ports:
  - 8080:80
  environment:
  - WORDPRESS_DB_HOST=mydb1
  - WORDPRESS_DB_USER=vima1
  - WORDPRESS_DB_PASSWORD=redhat
  - WORDPRESS_DB_NAME=bilogdb
```

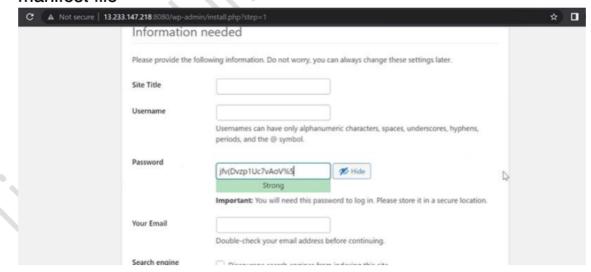
 Depends\_on is a keyword in docker-compose which is used as a control keyword to launch word press container after the database container



docker-compose down command is used to stop & remove the container



 Now the word press application will not ask to connect the database because of the environmental keywords we write in a manifest file



## Important Link

 Docker installation on Centos https://docs.docker.com/engine/install/centos/