



Summary

Session No 3

- **docker info** command is used to see all the information about the docker

```
Server:
Containers: 8
  Running: 1
  Paused: 0
  Stopped: 7
Images: 2
Server Version: 20.10.17
Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
userxattr: false
Logging Driver: json-file
Cgroup Driver: cgroupfs
Cgroup Version: 1
Plugins:
```

- **docker stats -a** command is used to see the live statistic of all the docker container

338a9254800e	webclient	0.00%	28.02MiB / 965.8MiB	2.90%	25.4MB / 53.8kB	13.5MB / 25MB	1
ab0e7e1d0236	condescending_gagarin	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0
b521fa23f4b5	wizardly_joliot	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0
95e21ccc22bb	sleepy_hermann	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0
a37162c1aeaa	naughty_dhawan	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0
ca15ebc88b5a	friendly_lamport	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0
474472007da0	intelligent_kilby	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0

- **docker ps -a -q** command is used to print all the IDs of the containers

```
[root@ip-172-31-40-68 ~]# docker ps -a -q
566669d0999d
338a9254800e
ab0e7e1d0236
95e21ccc22bb
a37162c1aeaa
ca15ebc88b5a
474472007da0
[root@ip-172-31-40-68 ~]#
```

- Trick to remove all the containers in one go
 - Command:- **docker rm -f \$(docker ps -a -q)**

```
[root@ip-172-31-40-68 ~]# docker rm -f $(docker ps -a -q)
566669d0999d
338a9254800e
ab0e7e1d0236
95e21ccc22bb
a37162c1aeaa
ca15ebc88b5a
474472007da0
[root@ip-172-31-40-68 ~]#
```

- Docker automatically gives the Ip address to each container

- By default, the container has connectivity with the base system

```
[root@ip-172-31-40-68 ~]# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.054 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.049 ms
^C
--- 172.17.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.049/0.051/0.054/0.007 ms
```

- By public IP anyone in the world can connect to the operating system
- Any server in the world that can connect with the outside world and they are running a program & every program they give a unique number known as a port number
- **netstat -tnlp** command is used to check the port number

```
[root@a266e89cd9d3 html]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      69/httpd
```

- By default, the container is isolated it cannot connect to the outside world
- Connecting container with the outside world
 - Giving the container a port number & volume

```
[root@ip-172-31-40-68 ~]# docker run -it --name web -p 1234:80 -v /code:/var/www/html centos:7
[root@1015889037dd /]#
[root@1015889037dd /]#
```

- Installing Apache web server in a container

```
[root@1015889037dd /]# yum install httpd -y
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
```

- Starting webservices
- Anyone in the world can access the website just have to type the protocol name, IP address, and port number



← → ↻ ⚠ Not secure | 65.1.132.46:1234/index.html

i m vimal from LW !!

- **-d** keyword in the run command is used to launch the container in detached mode

```
[root@ip-172-31-40-68 ~]# docker run -it -d --name w2 httpd
b913e94ae108ec078254d99fd291c42a283e42e90a64751c04269c551bd1800c
[root@ip-172-31-40-68 ~]#
```

- **docker inspect (container name or ID)** command is used to see the details about the container

```

"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "f36b891693265ccb50187a0283a9dcd63d3682a6fdb880c165efd0079fd29157",
    "EndpointID": "86b0c6769a80f88b6b331dcc2173cdc89a3564b921de76856ba24af6e9d3b61e",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "02:42:ac:11:00:02",
    "PortMappings": null
  }
}

```

- In the terminal, we can create variables that are called user-defined variables

```

[root@ip-172-31-40-68 ~]# x=5
[root@ip-172-31-40-68 ~]#
[root@ip-172-31-40-68 ~]# echo x
x
[root@ip-172-31-40-68 ~]#
[root@ip-172-31-40-68 ~]# echo $x
5
[root@ip-172-31-40-68 ~]#

```

- My-SQL database image requires environmental variables to start the services

```

[root@ip-172-31-40-68 ~]# docker run -it --name db1 mysql
2022-10-07 16:16:15+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.30-1.el8 started.
2022-10-07 16:16:15+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2022-10-07 16:16:15+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.30-1.el8 started.
2022-10-07 16:16:15+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
You need to specify one of the following:
- MYSQL_ROOT_PASSWORD
- MYSQL_ALLOW_EMPTY_PASSWORD
- MYSQL_RANDOM_ROOT_PASSWORD

```

- We can set the environmental variable in the container with the **-e** keyword in the run command

```

[root@ip-172-31-40-68 ~]# docker run -it -e x=5 centos:7
[root@714397d618a1 /]# echo $x
5
[root@714397d618a1 /]# exit
exit
[root@ip-172-31-40-68 ~]#
[root@ip-172-31-40-68 ~]#
[root@ip-172-31-40-68 ~]# docker run -it -e y=5 -e z=90 centos:7
[root@36fe0ec61cbd /]#
[root@36fe0ec61cbd /]# echo $y
5
[root@36fe0ec61cbd /]# echo $z
90

```

- Launching container with My-SQL database image

```

[root@ip-172-31-40-68 ~]# docker run -dit --name db1 -e MYSQL_ROOT_PASSWORD=redhat mysql
d76e698170615693eadc344505cd2a4086a3666f17d37f13574391b2487a5822
[root@ip-172-31-40-68 ~]#
[root@ip-172-31-40-68 ~]#

```

- Running bash shell in My-SQL container

```

[root@ip-172-31-40-68 ~]# docker exec -it db1 bash
bash-4.4#
bash-4.4#
bash-4.4#
bash-4.4# ps -aux
bash: ps: command not found
bash-4.4#

```

- Connecting MY-SQL server in a container

```
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- \ Symbol in Linux is used to cut the command into multiple parts

```
[root@ip-172-31-40-68 ~]# cal 2 \
> 2022
      February 2022
Su Mo Tu We Th Fr Sa
                1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28
[root@ip-172-31-40-68 ~]#
```

- Creating the general user and database with the help of environmental variables in a container

```
[root@ip-172-31-40-68 ~]# docker run -dit --name db123 -e MYSQL_ROOT_PASSWORD=redhat -e MYSQL
USER=vimal -e \
> MYSQL_PASSWORD=redhat -e \
> MYSQL_DATABASE=lwstudent mysql
9e627f4787d229294a1069d0dbdae8e2e64ffffae633083af350b99521d69755
[root@ip-172-31-40-68 ~]#
```

- Executing bash shell My-SQL container and connecting to a general user

```
[root@ip-172-31-40-68 ~]# docker exec -it db123 bash
bash-4.4# mysql -u vimal -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- Show databases is MY-SQL command to see the database

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| lwstudent |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql>
```