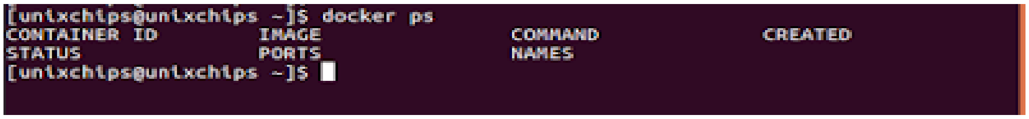
Docker management

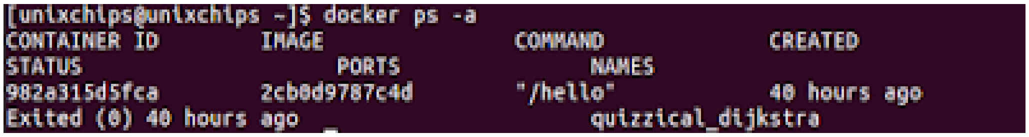
1. To check any containers running inside the docker use docker ps command



```
[unixchips@unixchips ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
[unixchips@unixchips ~]$ █
```

# docker ps

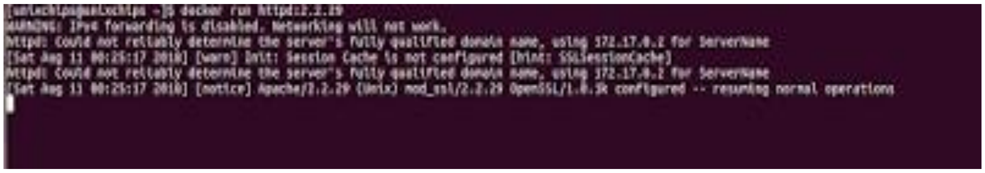
2. To get the currently running and stopped containers



```
[unixchips@unixchips ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                    PORTS               NAMES
982a315d5fca        2cb0d9787c4d        "/hello"            40 hours ago
Exited (0) 40 hours ago                       quizzical_dijkstra
```

# docker ps -a

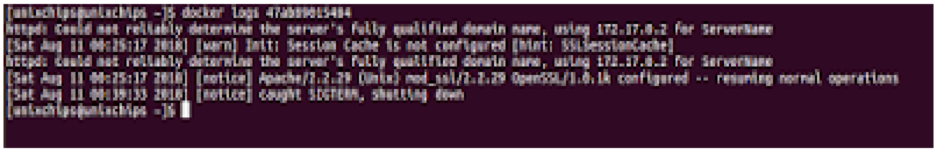
3. To run a container image



```
[unixchips@unixchips ~]$ docker run httpd:2.2.29
WARNING: IPv4 forwarding is disabled. Networking will not work.
httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2 for ServerName
[Sat Aug 11 00:25:17 2018] [warn] Init: Session Cache is not configured [hint: SSLSessionCache]
httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2 for ServerName
[Sat Aug 11 00:25:17 2018] [notice] Apache/2.2.29 (Unix) mod_ssl/2.2.29 OpenSSL/1.0.1k configured -- resuming normal operations
```

# docker run httpd:2.2.29

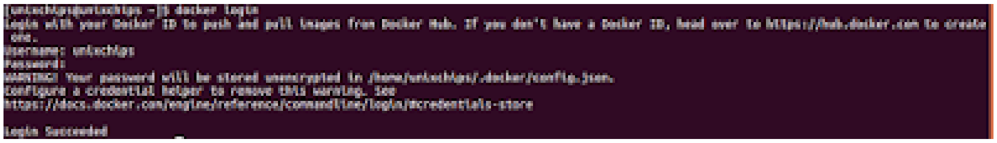
4. To get the logs of a container with container ID



```
[unixchips@unixchips ~]$ docker logs 47ab09015484
httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2 for ServerName
[Sat Aug 11 00:25:17 2018] [warn] Init: Session Cache is not configured [hint: SSLSessionCache]
httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2 for ServerName
[Sat Aug 11 00:25:17 2018] [notice] Apache/2.2.29 (Unix) mod_ssl/2.2.29 OpenSSL/1.0.1k configured -- resuming normal operations
[Sat Aug 11 00:39:33 2018] [notice] caught SIGTERM, shutting down
[unixchips@unixchips ~]$
```

# docker logs

5. To login inside the docker use below command

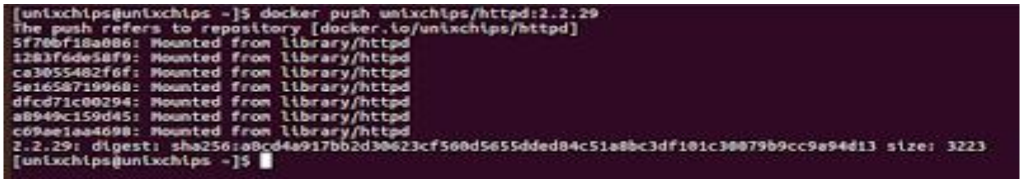


```
[unixchips@unixchips ~]$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create
 one.
Username: unixchips
Password:
WARNING! Your password will be stored unencrypted in /home/unixchips/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

# docker images

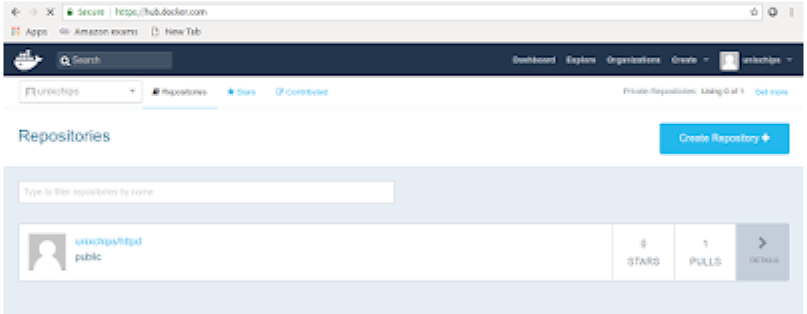
6. To push the images in custom path in your docker server use below command



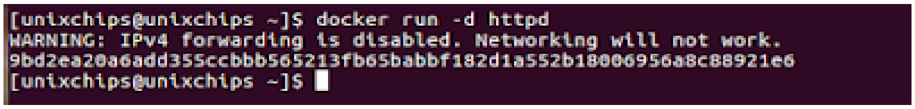
```
[unixchips@unixchips ~]$ docker push unixchips/httpd:2.2.29
The push refers to repository [docker.io/unixchips/httpd]
5f70bf18a086: Mounted from library/httpd
1283f6de58f9: Mounted from library/httpd
ca3055482f6f: Mounted from library/httpd
5e1658719968: Mounted from library/httpd
dfcd71c00294: Mounted from library/httpd
a8949c159d45: Mounted from library/httpd
c69ae1aa4698: Mounted from library/httpd
2.2.29: digest: sha256:a0cd4a917bb2d30623cf560d5655dded84c51a8bc3df101c30079b9cc9a94d13 size: 3223
[unixchips@unixchips ~]$
```

# docker push raushan/httpd:2.2.29

7 . If we login to the docker hub using web we can see the image as below



8. To run the docker image on background ,

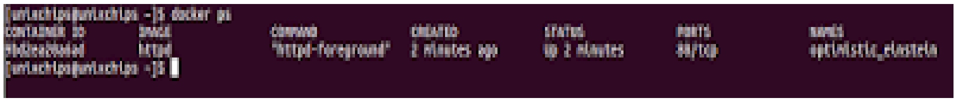


```
[unixchips@unixchips ~]$ docker run -d httpd
WARNING: IPv4 forwarding is disabled. Networking will not work.
9bd2ea20a6add355ccbbb565213fb65babbf182d1a552b18006956a8c88921e6
[unixchips@unixchips ~]$ 
```

# docker run –d httpd

This command will generate a random ID where the first 12 characters are the docker ID

9. Use the docker ps command to verify the docker ID and status

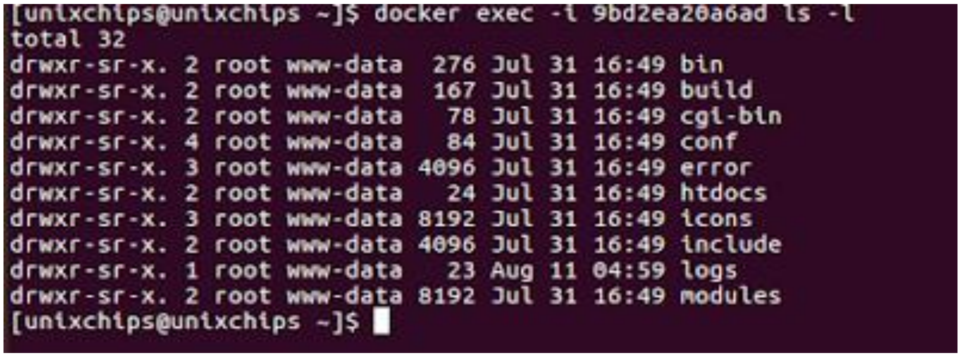


```
[unixchips@unixchips ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS              PORTS               NAMES
9bd2ea20a6ad        httpd               "httpd-foreground"   2 minutes ago       Up 2 minutes        80/tcp              optimistic_einstein
[unixchips@unixchips ~]$ 
```

# docker ps

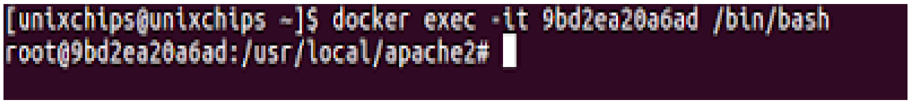
Now we can execute commands inside the docker

10 . To see the contents of a docker image use the below command



```
[unixchips@unixchips ~]$ docker exec -i 9bd2ea20a6ad ls -l
total 32
drwxr-sr-x. 2 root www-data  276 Jul 31 16:49 bin
drwxr-sr-x. 2 root www-data  167 Jul 31 16:49 build
drwxr-sr-x. 2 root www-data   78 Jul 31 16:49 cgi-bin
drwxr-sr-x. 4 root www-data   84 Jul 31 16:49 conf
drwxr-sr-x. 3 root www-data 4096 Jul 31 16:49 error
drwxr-sr-x. 2 root www-data   24 Jul 31 16:49 htdocs
drwxr-sr-x. 3 root www-data 8192 Jul 31 16:49 icons
drwxr-sr-x. 2 root www-data 4096 Jul 31 16:49 include
drwxr-sr-x. 1 root www-data   23 Aug 11 04:59 logs
drwxr-sr-x. 2 root www-data 8192 Jul 31 16:49 modules
[unixchips@unixchips ~]$ 
```

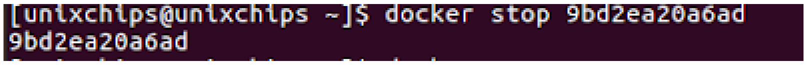
11. We can login to the container image using the below command



```
[unixchips@unixchips ~]$ docker exec -it 9bd2ea20a6ad /bin/bash
root@9bd2ea20a6ad:/usr/local/apache2#
```

# docker exec -it 9bd2ea20a6ad /bin/bash

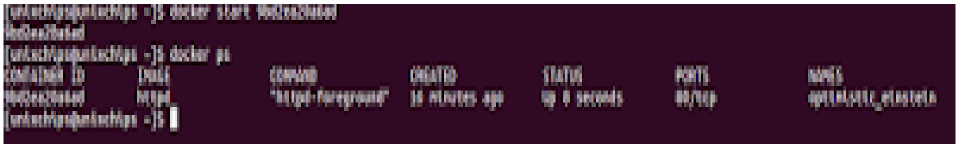
12. stop the container



```
[unixchips@unixchips ~]$ docker stop 9bd2ea20a6ad
9bd2ea20a6ad
```

# docker stop 9bd2ea20a6ad

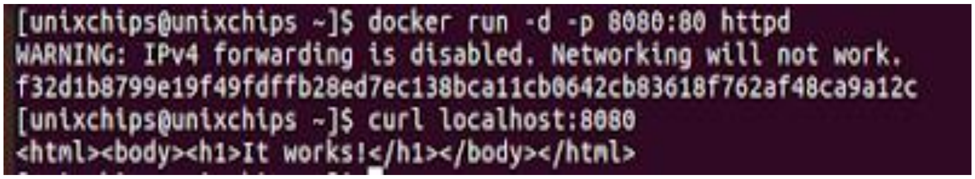
13 Start the container



```
[unixchips@unixchips ~]$ docker start 9bd2ea20a6ad
9bd2ea20a6ad
[unixchips@unixchips ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS              PORTS               NAMES
9bd2ea20a6ad        httpd               "httpd-foreground"   16 minutes ago      Up 8 seconds        80/tcp              optimistic_einstein
[unixchips@unixchips ~]$
```

# docker start 9bd2ea20a6ad

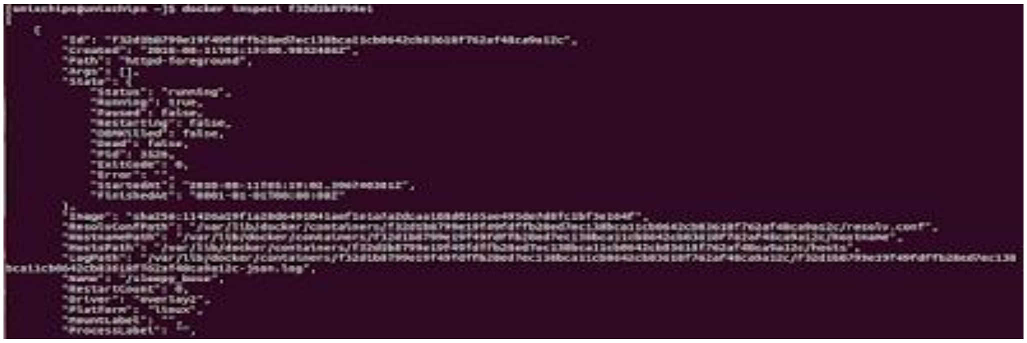
14 . To map the ports inside the docker use below commands



```
[unixchips@unixchips ~]$ docker run -d -p 8080:80 httpd
WARNING: IPv4 forwarding is disabled. Networking will not work.
f32d1b8799e19f49fdffb28ed7ec138bca11cb0642cb83618f762af48ca9a12c
[unixchips@unixchips ~]$ curl localhost:8080
<html><body><h1>It works!</h1></body></html>
```

# docker run -d -p 8080:80 httpd

15. Inspecting a docker container using docker inspect <docker id>



```
[unixchips@unixchips ~]$ docker inspect f32d1b8799e1
[
    {
        "Id": "f32d1b8799e19f49fdffb28ed7ec138bca11cb0642cb83618f762af48ca0a12c",
        "Created": "2020-08-11T05:19:00.[illegible]Z",
        "Path": "httpd-foreground",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 3520,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2020-08-11T05:19:02.[illegible]Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:[illegible]",
        "ResolvConfPath": "/var/lib/docker/containers/[illegible]/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/[illegible]/hostname",
        "HostsPath": "/var/lib/docker/containers/[illegible]/hosts",
        "LogPath": "/var/lib/docker/containers/[illegible]/[illegible]
[illegible]-json.log",
        "Name": "/sleepy_[illegible]",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
```

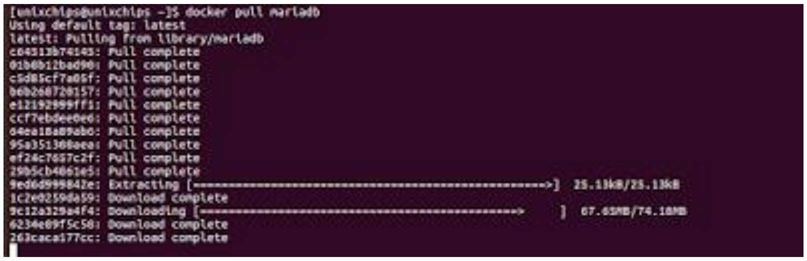
# docker inspect

16 Docker let's you to store data such as configuration settings encryption keys and external resource address in environment variables . At run time environment variables are exposed to the application inside the container. We can set the environment variables inside a services containers with environment key .

Below are the example of setting below parameters inside a maradb image

Parameter	Description
MYSQL_ROOT_PASSWORD	This variable is mandatory and specifies the password that will be set for the MariaDB root superuser account.
MYSQL_DATABASE	This variable is optional and allows you to specify the name of a database to be created on image startup. If a user/password was supplied (parameters in the row below) then that user will be granted superuser access (corresponding to GRANT ALL) to this database.
MYSQL_USER and MYSQL_PASSWORD	These variables are optional and used in conjunction to create a new user and to set that user's password. This user will be granted superuser permissions for the database specified by the MYSQL_DATABASE variable. Both variables are required for a user to be created.

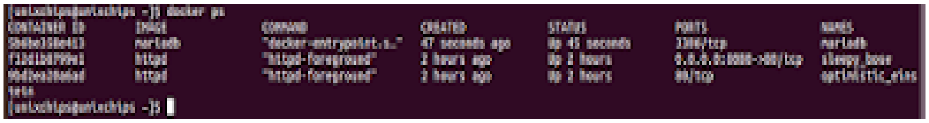
- First pull the image of mariadb



```
[unixchips@unixchips ~]$ docker pull mariadb
Using default tag: latest
latest: Pulling from library/mariadb
c04513b74145: Pull complete
01b0b12bad90: Pull complete
c5d85cf7a05f: Pull complete
b6b268720157: Pull complete
e12192999ff1: Pull complete
ccf7ebdee0e6: Pull complete
d4ea18a89ab0: Pull complete
95a351300aea: Pull complete
ef24c7657c2f: Pull complete
29b5cb4061e5: Pull complete
9ed6d999842e: Extracting [==================================================>]  25.13kB/25.13kB
1c2e0259da59: Download complete
9c12a329a4f4: Downloading [=============================================>     ]  67.65MB/74.18MB
6234e09f5c58: Download complete
263caca177cc: Download complete
```

# docker pull mariadb

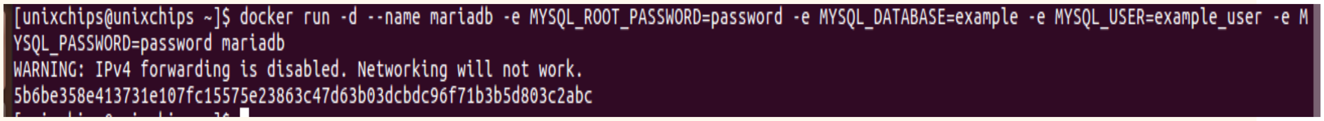
- We can check the status of the image using below command



```
[unixchips@unixchips ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                  NAMES
5b6be358e413        mariadb             "docker-entrypoint.s…"   47 seconds ago      Up 45 seconds       3306/tcp               mariadb
f12d1bd799e1        httpd               "httpd-foreground"       2 hours ago         Up 2 hours          0.0.0.0:8080->80/tcp   sleepy_bose
9bd2ea20a6ad        httpd               "httpd-foreground"       2 hours ago         Up 2 hours          80/tcp                 optimistic_elea
1e5a
[unixchips@unixchips ~]$
```

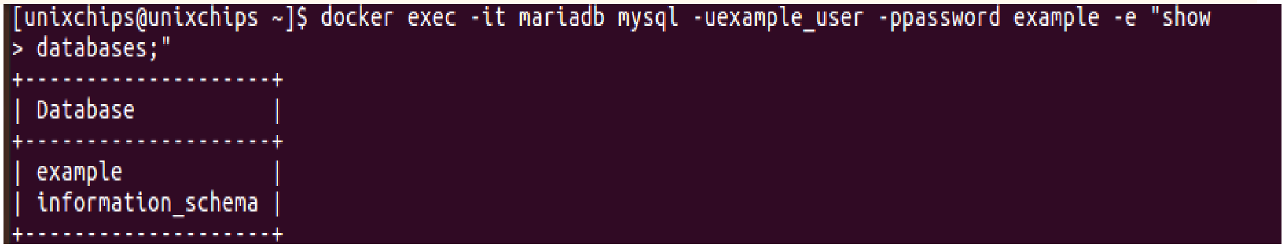
# docker ps

No let's run the command to start the database container by passing required variables mentioned above



```
[unixchips@unixchips ~]$ docker run -d --name mariadb -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=example -e MYSQL_USER=example_user -e M
YSQL_PASSWORD=password mariadb
WARNING: IPv4 forwarding is disabled. Networking will not work.
5b6be358e413731e107fc15575e23863c47d63b03dcbdc96f71b3b5d803c2abc
[unixchips@unixchips ~]$
```

- The container was created successfully and we need to verify whether example_user has the access to example database



```
[unixchips@unixchips ~]$ docker exec -it mariadb mysql -uexample_user -ppassword example -e "show
> databases;"
+--------------------+
| Database           |
+--------------------+
| example            |
| information_schema |
+--------------------+
```