Day 18 Task: Docker for DevOps Engineers:

Docker Compose:

Docker Compose is a tool for defining and running multi-container Docker applications. It allows users to configure multiple containers and their settings in a single "docker-compose.yml" file, and then start and stop all of the containers with a single command. This makes it easier to manage and deploy complex applications that consist of multiple components. Additionally, Compose also allows to define the network, volumes and environment variables for the containers.

What is YAML?

YAML (short for "YAML Ain't Markup Language") is a human-readable data serialization format. It is often used for configuration files, data exchange between systems, and other uses where a plain-text format is required. YAML is designed to be easy for humans to read and write, and it is often used in place of JSON or XML for these types of applications.

YAML files use indentation and simple punctuation to indicate the structure of the data. For example, a list of items might be represented like this in YAML:

```
- item1

- item2

- item3
```

YAML also supports more complex data structures, such as nested lists and dictionaries, and it allows for the use of special characters, such as the `#` symbol for commenting.

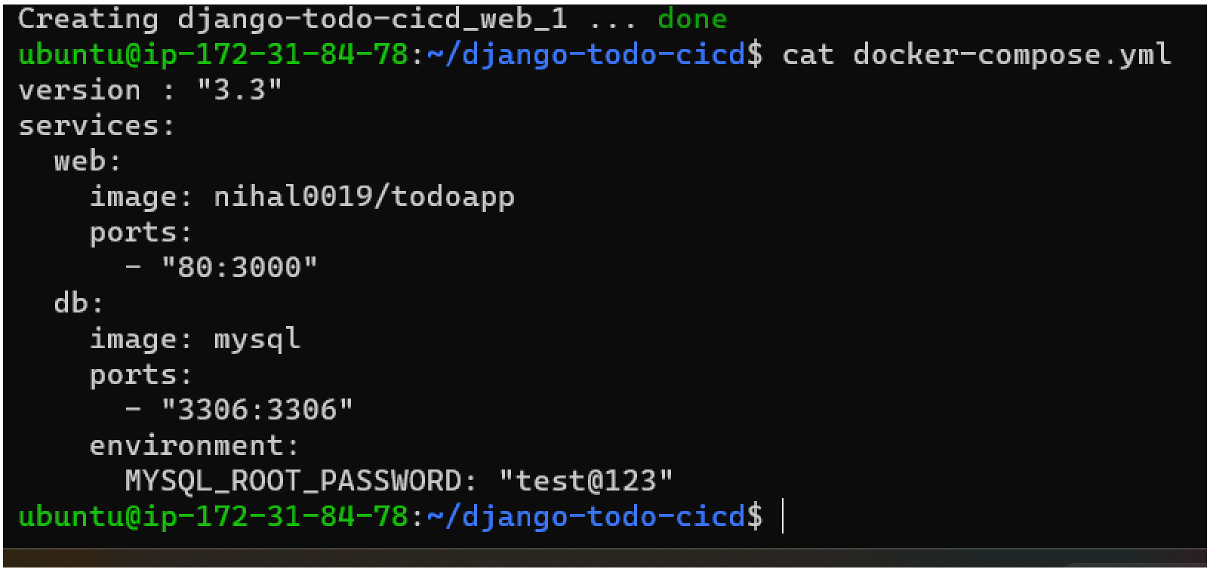
YAML files use the file extension **.yml** or **.yaml**

It is also used in many applications, like ansible, kubernetes, docker-compose and more.

Task-1:

Learn how to use the docker-compose.yml file, to set up the environment, configure the services and links between different containers, and also to use environment variables in the docker-compose.yml file.

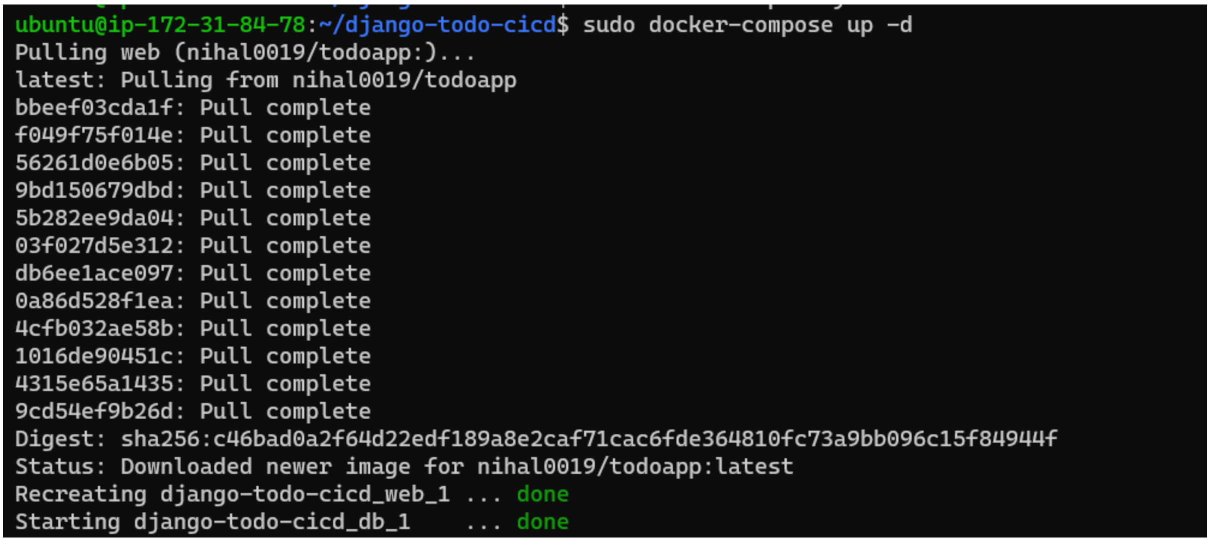
- We need to install docker-compose.
- After installation of docker-compose, create docker-compose.yml file.



```
Creating django-todo-cicd_web_1 ... done
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ cat docker-compose.yml
version : "3.3"
services:
  web:
    image: nihal0019/todoapp
    ports:
      - "80:3000"
  db:
    image: mysql
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: "test@123"
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ 
```

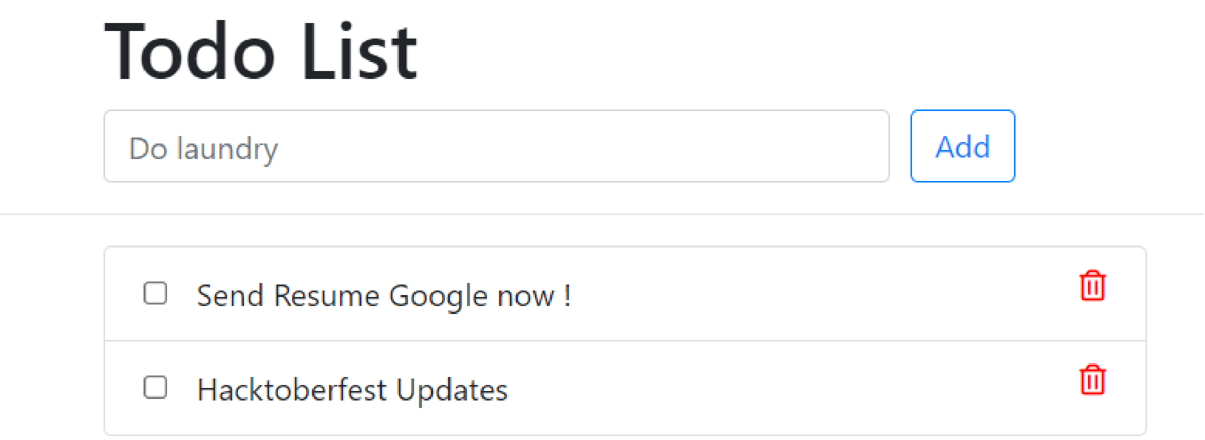
- Now run the docker-compose.yml file.

The **“docker-compose up”** command is used to start and run a Docker Compose application. It reads the configuration defined in the docker-compose.yml file and creates and starts the specified services (containers) and networks.

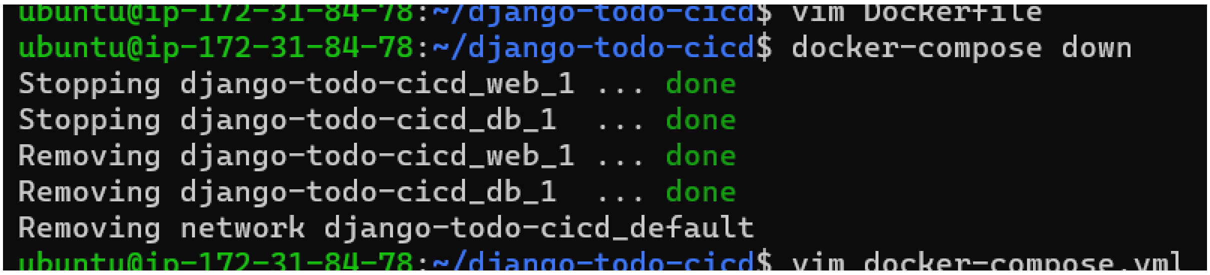


```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ sudo docker-compose up -d
Pulling web (nihal0019/todoapp:)...
latest: Pulling from nihal0019/todoapp
bbeef03cda1f: Pull complete
f049f75f014e: Pull complete
56261d0e6b05: Pull complete
9bd150679dbd: Pull complete
5b282ee9da04: Pull complete
03f027d5e312: Pull complete
db6ee1ace097: Pull complete
0a86d528f1ea: Pull complete
4cfb032ae58b: Pull complete
1016de90451c: Pull complete
4315e65a1435: Pull complete
9cd54ef9b26d: Pull complete
Digest: sha256:c46bad0a2f64d22edf189a8e2caf71cac6fde364810fc73a9bb096c15f84944f
Status: Downloaded newer image for nihal0019/todoapp:latest
Recreating django-todo-cicd_web_1 ... done
Starting django-todo-cicd_db_1    ... done
```

- Verify that application is working by accessing it in a web browser



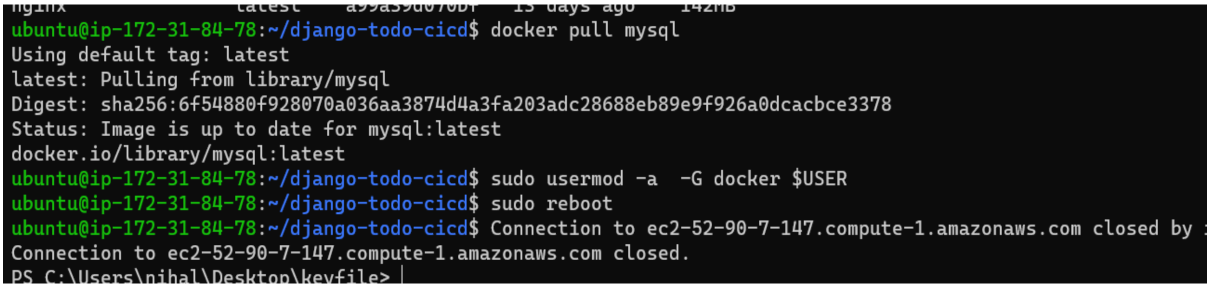
The “**docker-compose down**” command stops all the services and cleans up the containers, networks, and images.



```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ vim Dockerfile
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker-compose down
Stopping django-todo-cicd_web_1 ... done
Stopping django-todo-cicd_db_1  ... done
Removing django-todo-cicd_web_1 ... done
Removing django-todo-cicd_db_1  ... done
Removing network django-todo-cicd_default
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ vim docker-compose.yml
```

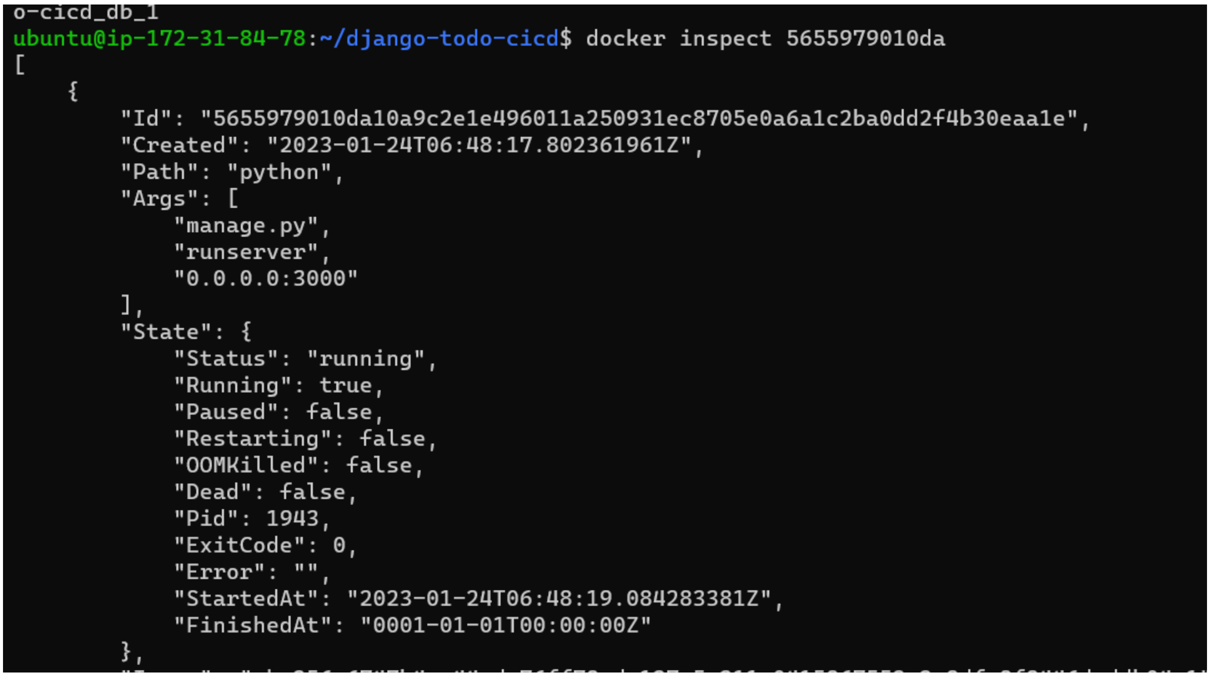
Task-2:

- **Pull a pre-existing Docker image from a public repository (e.g. Docker Hub) and run it on your local machine. Run the container as a non-root user (Hint- Use usermod command to give user permission to docker). Make sure you reboot instance after giving permission to user.**



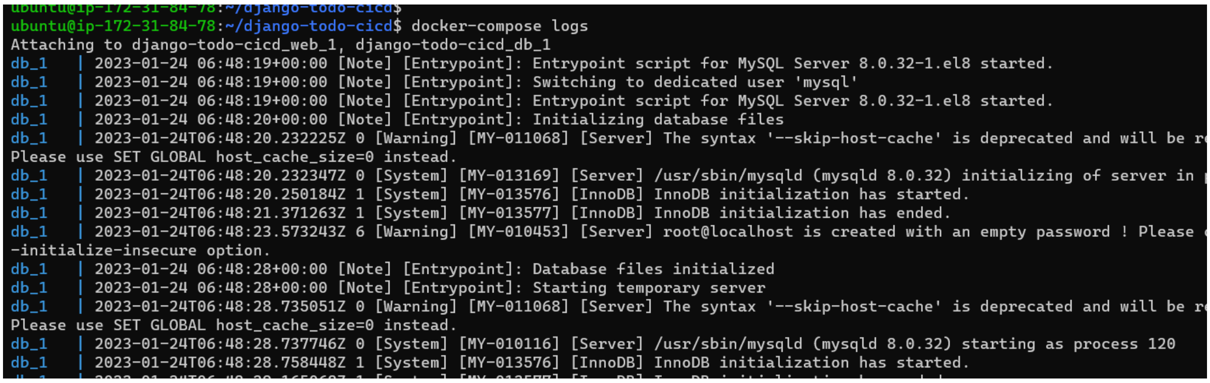
```
nginx             latest    a99a39d070bf   13 days ago    142MB
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
Digest: sha256:6f54880f928070a036aa3874d4a3fa203adc28688eb89e9f926a0dcacbce3378
Status: Image is up to date for mysql:latest
docker.io/library/mysql:latest
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ sudo usermod -a  -G docker $USER
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ sudo reboot
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ Connection to ec2-52-90-7-147.compute-1.amazonaws.com closed by
Connection to ec2-52-90-7-147.compute-1.amazonaws.com closed.
PS C:\Users\nihal\Desktop\keyfile>
```

- **Inspect the container's running processes and exposed ports using the docker inspect command.**



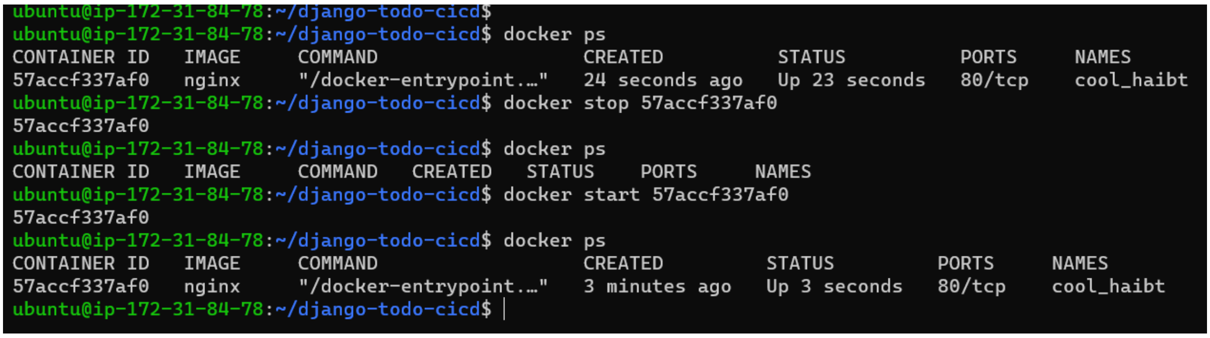
```
o-cicd_db_1
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker inspect 5655979010da
[
    {
        "Id": "5655979010da10a9c2e1e496011a250931ec8705e0a6a1c2ba0dd2f4b30eaa1e",
        "Created": "2023-01-24T06:48:17.802361961Z",
        "Path": "python",
        "Args": [
            "manage.py",
            "runserver",
            "0.0.0.0:3000"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 1943,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-01-24T06:48:19.084283381Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
```

- **Use the docker logs command to view the container's log output.**



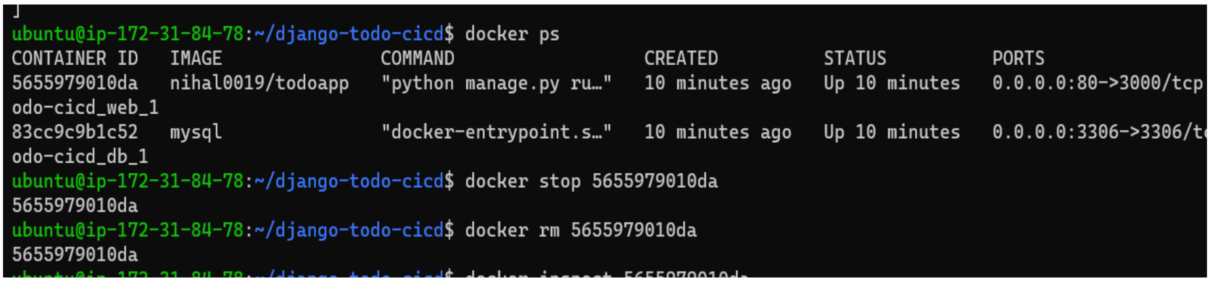
```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker-compose logs
Attaching to django-todo-cicd_web_1, django-todo-cicd_db_1
db_1    | 2023-01-24 06:48:19+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
db_1    | 2023-01-24 06:48:19+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db_1    | 2023-01-24 06:48:19+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
db_1    | 2023-01-24 06:48:20+00:00 [Note] [Entrypoint]: Initializing database files
db_1    | 2023-01-24T06:48:20.232225Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be re
Please use SET GLOBAL host_cache_size=0 instead.
db_1    | 2023-01-24T06:48:20.232347Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.32) initializing of server in 
db_1    | 2023-01-24T06:48:20.250184Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
db_1    | 2023-01-24T06:48:21.371263Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
db_1    | 2023-01-24T06:48:23.573243Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please 
-initialize-insecure option.
db_1    | 2023-01-24 06:48:28+00:00 [Note] [Entrypoint]: Database files initialized
db_1    | 2023-01-24 06:48:28+00:00 [Note] [Entrypoint]: Starting temporary server
db_1    | 2023-01-24T06:48:28.735051Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be re
Please use SET GLOBAL host_cache_size=0 instead.
db_1    | 2023-01-24T06:48:28.737746Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.32) starting as process 120
db_1    | 2023-01-24T06:48:28.758448Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
```

- **Use the docker stop and docker start commands to stop and start the container.**



```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
57accf337af0   nginx     "/docker-entrypoint.…"   24 seconds ago   Up 23 seconds   80/tcp    cool_haibt
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker stop 57accf337af0
57accf337af0
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker start 57accf337af0
57accf337af0
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
57accf337af0   nginx     "/docker-entrypoint.…"   3 minutes ago   Up 3 seconds   80/tcp    cool_haibt
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ |
```

- **Use the docker rm command to remove the container when you're done.**



```
]
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS
5655979010da   nihal0019/todoapp   "python manage.py ru…"   10 minutes ago   Up 10 minutes   0.0.0.0:80->3000/tcp
odo-cicd_web_1
83cc9c9b1c52   mysql              "docker-entrypoint.s…"   10 minutes ago   Up 10 minutes   0.0.0.0:3306->3306/t
odo-cicd_db_1
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker stop 5655979010da
5655979010da
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker rm 5655979010da
5655979010da
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker inspect 5655979010da
```

Thank you for reading! I hope you find this article helpful.

Happy Learning 😊