# Day 19 Task: Docker for DevOps Engineers: Part-3

## Docker-Volume:

A Docker volume is a way to store data outside of a Docker container's filesystem. This allows data to persist even if the container is deleted or recreated. Volumes can also be shared among multiple containers and can be backed by various storage drivers, such as local files or network storage. This allows for greater flexibility and control over data storage in a Docker environment.

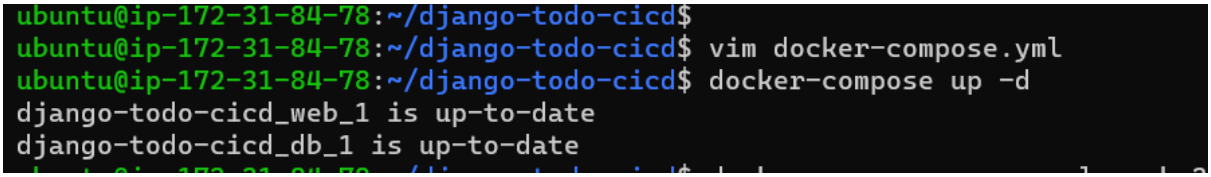
## Docker Network:

Docker network is a feature in Docker that allows containers to communicate with each other and with the host system. It provides a way to create and manage virtual networks for container communication. There are several types of networks that can be created in Docker, including bridge, host, and overlay networks. Each type of network provides different capabilities and can be used for different use cases. For example, a bridge network allows containers on the same host to communicate with each other, while an overlay network allows containers on different hosts to communicate. Additionally, Docker networks can be used to configure network settings such as IP addresses, ports, and DNS.

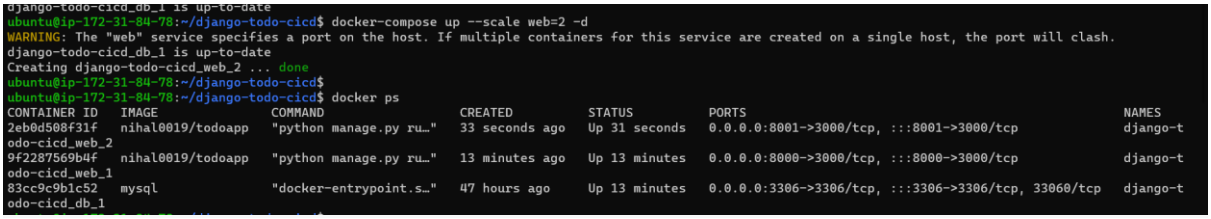
## Task-1:

**Create a multi-container docker-compose file which will bring UP and bring DOWN containers in a single shot (Example - Create application and database container)**

- The **docker-compose up** command with the **-d** flag is used to start and run a multi-container application defined in a **docker-compose.yml** file in detached mode. The **-d** flag stands for "detached" mode and it runs the container in the background.




- The **docker-compose scale** command is used to adjust the number of containers for a service defined in a **docker-compose.yml** file. This command allows you to easily scale the number of containers running for a particular service, which can be useful for handling changes in traffic or load.



- The **docker-compose ps** command is used to list the containers that are running for a multi-container application defined in a **docker-compose.yml** file. This command will display the status of each container, including the container name, service name, and the command that was used to start the container.

```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker-compose ps
         Name                       Command               State                          Ports
-------------------------------------------------------------------------------------------------------------------
django-todo-cicd_db_1    docker-entrypoint.sh mysqld      Up      0.0.0.0:3306->3306/tcp,:::3306->3306/tcp, 33060/tcp
django-todo-cicd_web_1   python manage.py runserver ...   Up      0.0.0.0:8000->3000/tcp,:::8000->3000/tcp
django-todo-cicd_web_2   python manage.py runserver ...   Up      0.0.0.0:8001->3000/tcp,:::8001->3000/tcp
ubuntu@ip-172-31-84-78:~/django-todo-cicd$
```

- The **docker-compose logs** command is used to view the logs for all the services defined in a **docker-compose.yml** file. This command will display the logs for all the running containers for the specified services, in real-time.

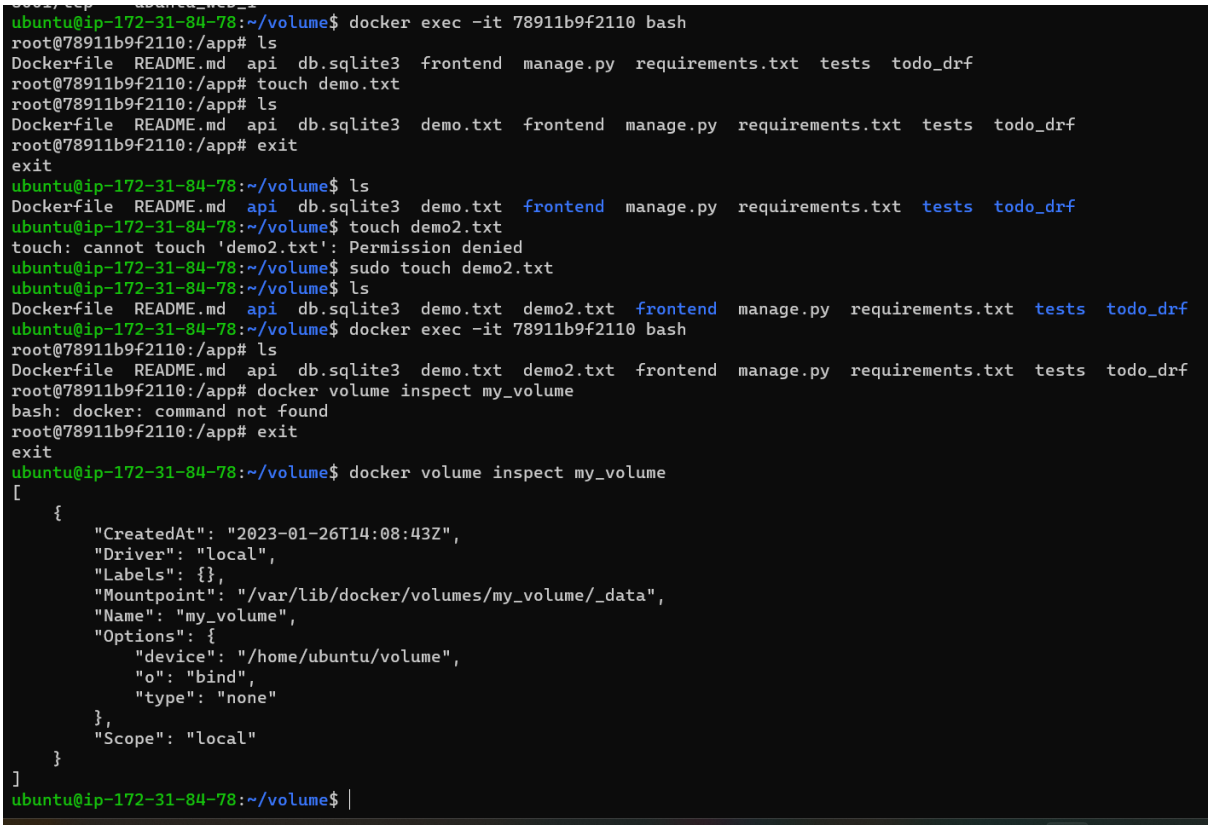
```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker-compose logs
Attaching to django-todo-cicd_web_2, django-todo-cicd_web_1, django-todo-cicd_db_1
db_1    | 2023-01-24 06:48:19+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
db_1    | 2023-01-24 06:48:19+00:00 [Note] [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
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 s
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
-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
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 proce
db_1    | 2023-01-24T06:48:28.758448Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
db_1    | 2023-01-24T06:48:29.165068Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
db_1    | 2023-01-24T06:48:29.650863Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
db_1    | 2023-01-24T06:48:29.651077Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encr
or this channel.
db_1    | 2023-01-24T06:48:29.654936Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '
sible to all OS users. Consider choosing a different directory.
db_1    | 2023-01-24T06:48:29.688197Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: /var/run/m
```

- The **docker-compose down** command is used to stop and remove all the containers, networks, and volumes defined in a **docker-compose.yml** file. This command will stop and remove all the containers that were created by the **docker-compose up** command, as well as any networks and volumes that were created for the application.

```
ubuntu@ip-172-31-84-78:~/django-todo-cicd$ docker-compose down
Stopping django-todo-cicd_web_2 ... done
Stopping django-todo-cicd_web_1 ... done
Stopping django-todo-cicd_db_1  ... done
Removing django-todo-cicd_web_2 ... 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$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-84-78:~/django-todo-cicd$
ubuntu@ip-172-31-84-78:~/django-todo-cicd$
```

## Task-2:

- **Learn how to use Docker Volumes and Named Volumes to share files and directories between multiple containers.**
- **Create two or more containers that read and write data to the same volume using the docker run --mount command.**
- **Verify that the data is the same in all containers by using the docker exec command to run commands inside each container.**
- **Use the docker volume ls command to list all volumes and docker volume rm command to remove the volume when you're done.**




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

**Happy Learning 😊**