#This file is written by Sayali Jadhav

**Day19: Docker for DevOps Engineers (Part-3)**

**Docker-Volume:**

Docker allows you to create something called volumes. Volumes are like separate storage areas that can be accessed by containers. They allow you to store data, like a database, outside the container, so it doesn't get deleted when the container is deleted. You can also mount from the same volume and create more containers having same data.

**Docker Network:**

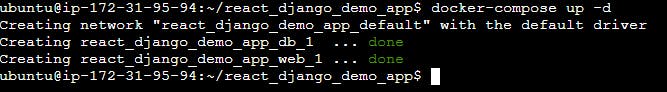
Docker allows you to create virtual spaces called networks, where you can connect multiple containers (small packages that hold all the necessary files for a specific application to run) together. This way, the containers can communicate with each other and with the host machine (the computer on which the Docker is installed). When we run a container, it has its own storage space that is only accessible by that specific container. If we want to share that storage space with other containers, we can't do that.

**Task-1**

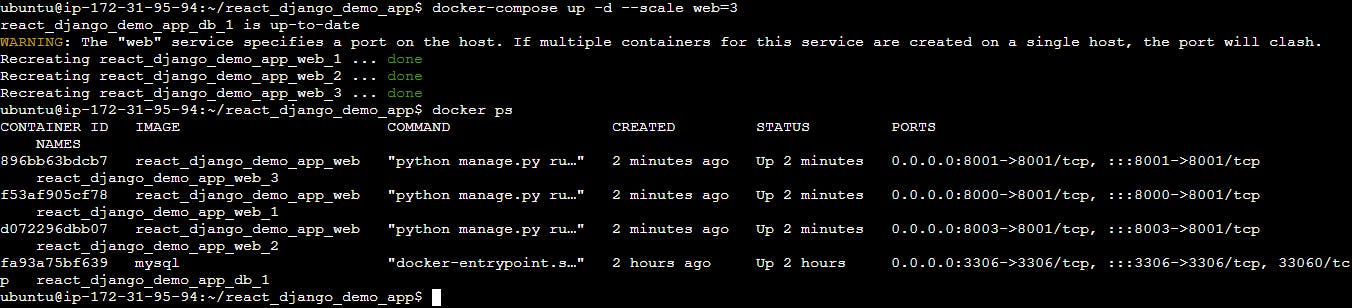
**Create a multi-container docker-compose file which will bring *UP* and bring *DOWN* containers in a single shot ( Example - Create application and database container )**



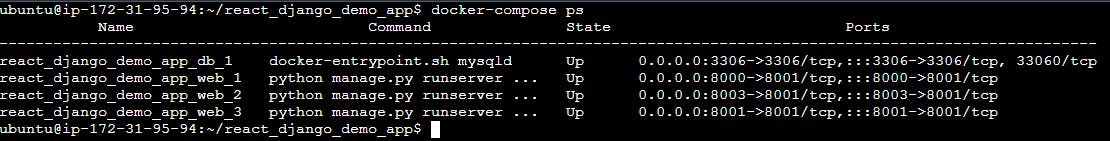
**Use the docker-compose up command with the -d flag to start a multi-container application in detached mode.**



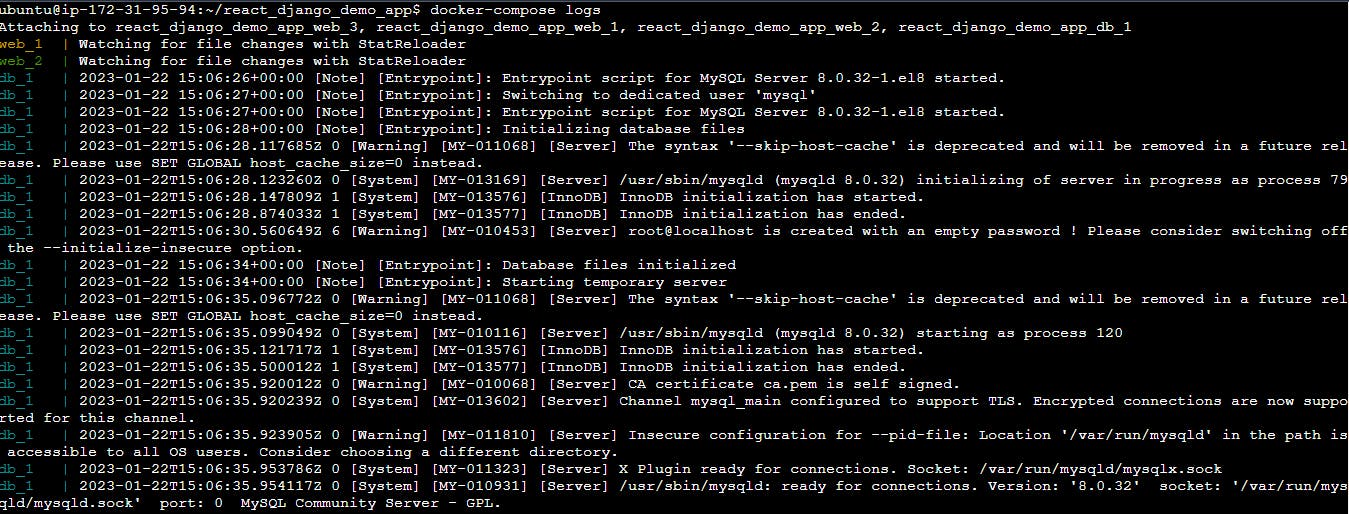
**Use the docker-compose scale command to increase or decrease the number of replicas for a specific service. You can also add** [**replicas**](https://stackoverflow.com/questions/63408708/how-to-scale-from-within-docker-compose-file) **in deployment file for *auto-scaling*.**



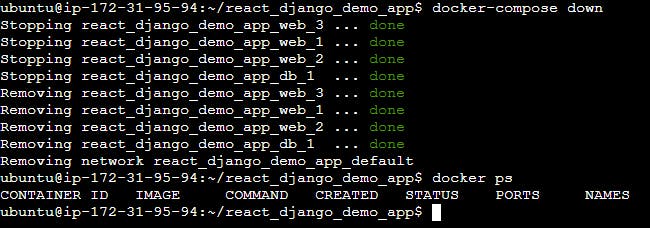
**Use the docker-compose ps command to view the status of all containers**



**Use the docker-compose logs to view the logs of a specific service.**

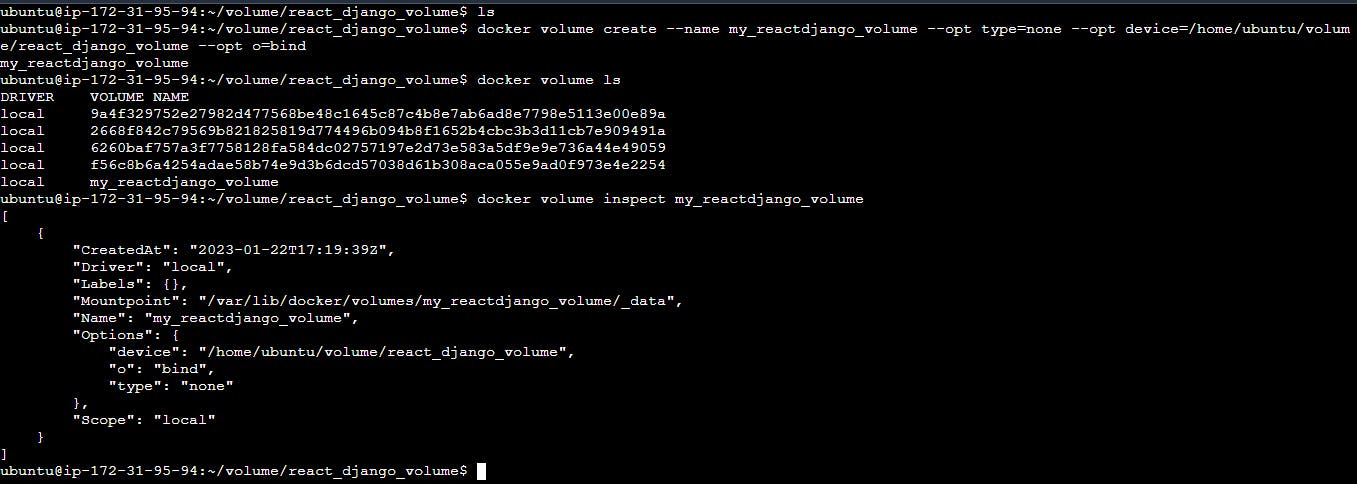


**Use the docker-compose down command to stop and remove all containers, networks, and volumes associated with the application**

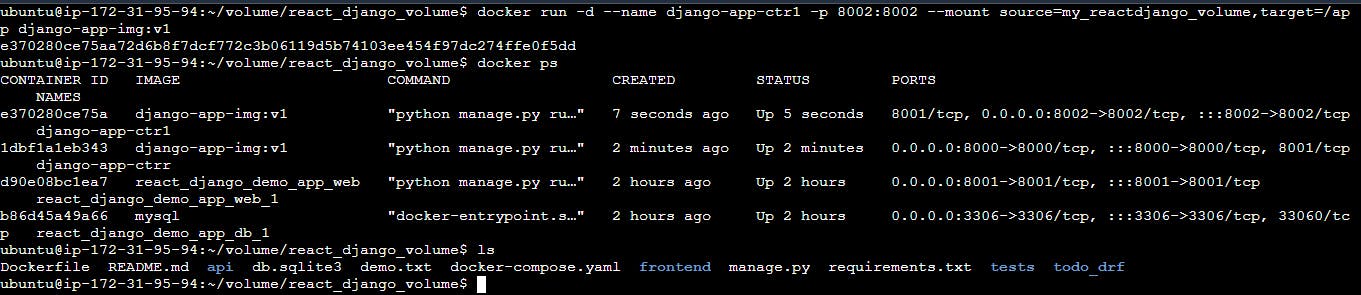


**Task-2**

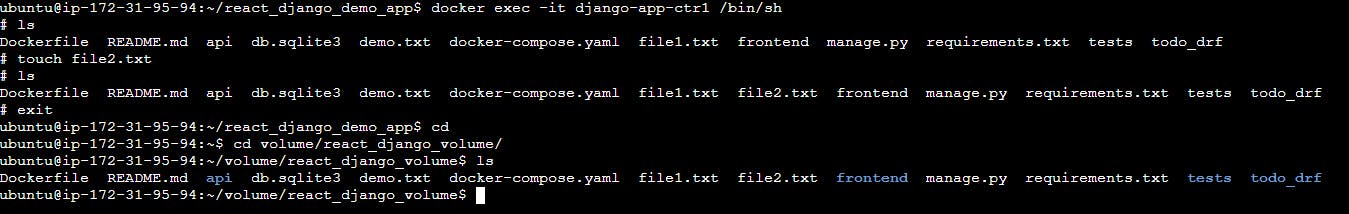
**Learn how to use Docker Volumes and Named Volumes to share files and directories between multiple containers.**



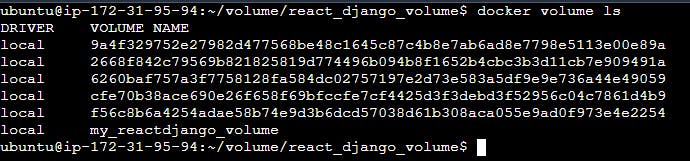
**Create two or more containers that read and write data to the same volume using the docker run --mount command.**



**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**



**Use the docker volume rm command to remove the volume when you're done**

