**DOCKER-COMPOSE**

* It’s a Tool for defining & running multi-container docker applications
* One application consists of multiple containers
* One container depends on another
* We can define single container using docker-compose or multiple container using docker-compose
* use yaml/yml files to configure application services(docker-compose.yml)
* can start all services with a single command: ***docker compose up***
* can delete all services with a single command: ***docker compose down***
* can scale up selected services when required
* **dockerfile:** using dockerfile we can create a images
* **docker-compose:** its tool for define and deploy single container and multiple container
* default name for compose file is ***docker-compose.yml or docker-compse.yaml***
* **docker-compose installation:**
* sudo sudo curl -L https://github.com/docker/compose/releases/download/1.21.0/docker-compose-`uname -s`-`uname -m` | sudo tee /usr/local/bin/docker-compose > /dev/null
* sudo chmod +x /usr/local/bin/docker-compose
* ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
* docker-compose –version

**Ubuntu Linux:**

apt-get install docker-compose -y

**Compose file format:**

* **version:**
* **services:**
* **volumes:**
* **networks:**
* **volumes:**
* **image:**
* **build:**
* **restart:**
* **environment:**
* **depends\_on:**
* **port:**
* **links:**

*version: '3.1'*

*services:*

*springboot:*

*image: dockerhandson/spring-boot-mongo:latest*

*environment:*

*- MONGO\_DB\_HOSTNAME=mongo*

*- MONGO\_DB\_USERNAME=devdb*

*- MONGO\_DB\_PASSWORD=devdb@123*

*ports:*

*- 9090:8080*

***depends\_on:***

*- mongo*

*networks:*

*- springappnetwork*

*mongo:*

*image: mongo*

*environment:*

*- MONGO\_INITDB\_ROOT\_USERNAME=devdb*

*- MONGO\_INITDB\_ROOT\_PASSWORD=devdb@123*

*volumes:*

*- mongodbvol:/data/db*

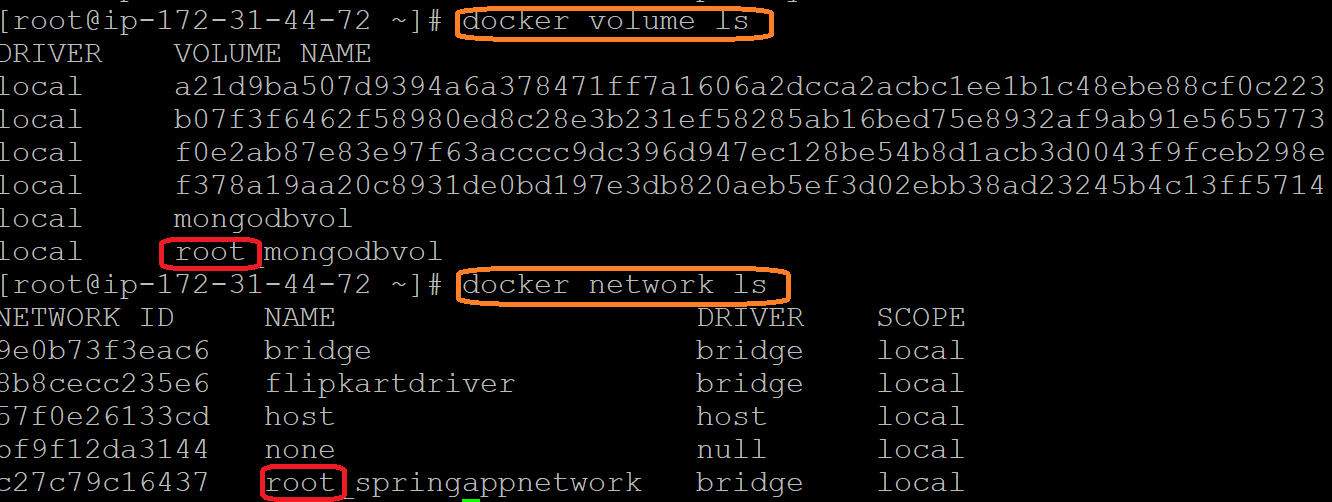
*networks:*

*- springappnetwork*

*volumes:*

*mongodbvol:*

*driver: local*



**Its taking current folder/project name**

* docker-compose -v
* docker-compose --version

**Create docker-compose file at any location at any location on your system**

* docker-compose.yml

**Check the valid of file by command (i.e default file format docker-compose.yml/yaml)**

* docker-compose config

**Run docker-compose.yml file by command**

* docker-compose up -d

**Bring down application by command**

* docker-compose down

**customer name**

docker-compose -f <custome-filename>.yml/yaml config

docker-compose -f <custom-filename>.ym/yaml up -d

**Scale docker-compose**

* + docker-compose up -d --scale <container-name>=<no-of-instance>
  + docker-compose up -d --scale mydb=3

**Note :**

**Before running all above command first docker should be running status**

* **docker-compose up -d**
* **docker-compose down**
* **docker-compose stop**
* **docker-compose start**
* **docker-compose restart**
* **docker-compose images**
* **docker-compose ps**
* **docker-compose logs**

**volumes:**

**mongovol:**

**external:true**

**networks:**

**springappbrige:**

**external:true**

**Note:**

**🡪docker-compose file won’t create volume and networks because *external:true***

**🡪 we have to create manually**