Docker basic commands

Follow these commands to get basic clarity on docker.

Install Docker and Docker Compose on Ubuntu:

- sudo su
- apt-get update
- apt-get install -y docker.io
- curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
- chmod +x /usr/local/bin/docker-compose

Check for installation:

- docker - -version
- docker info

Basic docker commands:

- docker images {shows all the images}
- docker ps {shows all the running containers]
- docker ps -all {shows all the containers}

Pull docker images:

- Docker pull <image name>
- Ex: Docker pull ubuntu

Run a docker image to create a container:

- Docker run <image name>
- Ex: docker run ubuntu
- Docker run -it -d ubuntu { -it : interactive -d : run as a server }
- Docker run --rm ubuntu { --rm : remove container after exit }
- Docker run -p <host port>:<docker container port> ubuntu {-p : expose port <host port> externally and map to port <docker container port>}
- Ex: docker run -p 82:80 ubuntu

Execute commands in docker container:

- Docker exec -it <container id> bash {opens bash of the container}

Docker stop/kill/delete commands for containers:

- Docker stop <container id>
- Docker kill <container id>
- Docker rm <container id>
- Docker rm -f <container id>
- Docker rm -f $(docker ps -a -q)

Docker delete command for image:

- Docker rmi <image name>

Docker Hub

- Docker commit <container id> <name you want to give to new image> {To create an image from contained with needed changes according to user}
- Docker login
- Extension: {enter your username and password}
- Docker push <image name you want to push>

Docker file

Docker can build images by reading set of instructions in side a docker file.

Sample file with name dockerfile

```
FROM ubuntu

RUN apt-get update

RUN apt-get -y install apache2

ADD . /var/www/html
```

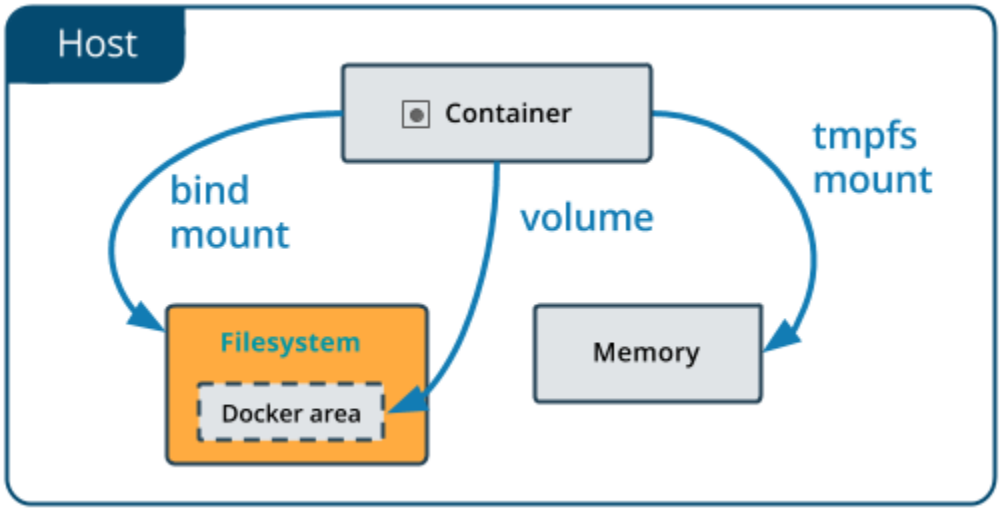
After creating this docker file – build the container

- Docker build . -t dockerfile

Docker Bind Mount

Bind mount has limited functionality compared to volumes. When we use this, a file or directory on the host machine is mounted into a container.

- Docker run -it -v <path local>:<path in docker> <inage name>
- Docker run -it -v /home/user/desktop/dockerfile:/var/www/html ubuntu



Docker volumes

Unlike Bind Mount a new directory is created within docker storage directory on the host machine, and docker will manage that contents.

- Docker volume create <name>
- Ex: Docker volume create data
- Docker run -it - -mount source=<name of volume>,target=<path> <image name>
- Docker run -it - -mount source=data,target=/var/www/html ubuntu

Docker Compose

Docker compose is used to run multiple containers, by reading a config Docke-compose.yml file.

With a single command, you create and start all the services from your configuration.

- Docker-compose.yaml

```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
    db_data: {}
```

- Run docker compose
- Docker -compose up -d

Docker Swarm

- Docker swarm init - -advertise-addr=<private ip> {run this command in master}
- This will return a URL
- Paste this URL in slave instance
- Docker service create - -name <name for the service> - -replicas <no of replicas needed> <image name>
- Ex: docker service create - -name <webserver> - -replicas 20 -p 82:80 ubuntu