31.08May.Adv Docker

8 May 2023 07:16 PM

Virtualization = Shared Resources
Cloud = On Rent Resources
Containerization = Enhance Technique, No Guest OS

Components of Docker Architecture:

1. Client(Terminal)
2. Host/Docker Engine
    a. Docker Daemon
    b. Local Images
    c. Container
3. Container Registry
    a. Open Source: hub.docker.com
    b. Cloud Vendors: ECR, ACR, GCR

How to Manage Docker based Application:

1. **Ad Hoc Commands**
    a. $ docker pull ImageName
    b. $ docker images(list all downloadable/local image)
    c. $ docker ps (list all Running container/Process)
    d. $ docker --version or $ docker version
    e. $ docker info
    f. $ docker **rmi** Image-ID
    g. $ docker **rm** Container-ID
    h. $ docker stop Container-ID
    i. $ docker ps -a (to list all stopped containers)
    j. $ docker kill Container-ID
    k. $ docker login
2. **Dockerfile Commands(Custom Image --> Run Container)**
    a. FROM --> Pull image from container registry
    b. WORKDIR --> Set current directory/folder
    c. COPY --> Copy from Docker Engine to Custom-Image
    d. RUN --> Executes commands in Custom-Image
3. **Docker Compose(Create Multi container Application)**
    a. Up
    b. Down
4. **Docker Swarm**(Manage multiple Containers= Orchestration tool= V1 K8S)

Installation of Docker

EC2 --> Instance --> Launch --> UserData
#include https://get.docker.com

apt install docker-ce

Docker Compose

- Tool for defining and sharing multi-container applications
- Containers:
    - Container1: Front End Application
    - Container2: Back End Database
- YAML file

- Using Up/Down Commands you can control whole project.
- Version1 is Docker Compose ----> V2 = Kubernetes

Project1 : MS.Net Application
C1: ASP.Net Web Server
C2: NGINX Proxy Server
C3: MySQL Database Server

Project2 : Python Application
C1: Flask Web Server(Python)
8000

1. Manual way: Python, PIP, Virtualenv, Install Flask --> Run
   a. 1 Hour
2. Dockerfile : build image --> run image as container
   a. 20 min
3. Docker Compose --> Create = up , Delete = down
   a. 5 min

POSIX copy paste:

COPY: ctrl+ Insert or Ins

PASTE: Shift+ Insert or Ins

Toggle Screen Size: Alt + Shift + Enter

Font Size: Increase(Ctrl + Shift+ Plus) , Decrease(Ctrl+ Minus)

$ git clone https://github.com/NubeEra-Samples/DockerCompose-FlaskApp.git

$ cd DockerCompose-FlaskApp

$ docker compose up -d

$ docker compose ps

$ curl http://localhost:8000

$ docker compose down

V2
Github $ Change in app/app.py file --> Add --> Commit--> Push

PWD $ git pull

PWD $ docker compose up -d

PWD $ docker compose rm

PWD $ docker rm -f **$(**docker ps -a -q**)**

$ docker ps -a -q

CONTAINER-ID

$ docker rm -f CONTAINER-ID

Docker Compose vs Docker Swarm:
C: It creates multiple containers on a single host
S: it manage multiple containers on Multiple hosts

C: it uses YAML file to manage different containers as a **single** service
S: it doesn't use any file but helps you to manage different docker hosts in a **cluster**.

C: used for Containers Creations
S: Orchestration tool

C: Creation
S: Management

Docker Swarm:

1. Decentralized access
2. Auto load Balancing
3. Roll-back a task
4. High scalability
5. High Security

Swarm Components:

1. Service
2. Tasks
3. Node
    a. Manager: # docker swarm init --advertise-addr 192.168.2.151
    b. Worker#

docker swarm join --token SWMTKN-1-21o….8 192.168.0.28:2377

List all nodes:

$ docker node ls

Create Service:
$ docker service create --name helloWorld alpine ping docker.com

List Service:
$ docker service ls