Docker Compose

# Installation

sudo curl -L https://github.com/docker/compose/releases/download/1.24.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

docker-compose help

# Build Images

mkdir docker-compose

cd docker-compose

mkdir build

cd build

vi Dockerfile

FROM nginx:alpine

RUN echo "Welcome to Docker Workshop!" >/usr/share/nginx/html/index.html

CMD ["nginx", "-g", "daemon off;"]

*vi docker-compose.yml*

version: "3.7"

services:

webapp:

build:

context: .

dockerfile: Dockerfile

image: webapp:v1

docker-compose build

docker images

# 2-Tier Applications

cd ..

mkdir app01

cd app01

*vi docker-compose.yml*

version: '2'  
services:  
 #Nginx Service  
 webserver:  
 image: nginx:alpine  
 container\_name: webserver  
 restart: unless-stopped  
 ports:  
 - "80:80"  
 - "443:443"

depends\_on:

- db  
 #Mysql DB  
 db:  
 image: mysql:5.7  
 container\_name: Mysqldb  
 restart: unless-stopped  
 ports:  
 - "3306:3306"  
 environment:  
 MYSQL\_ROOT\_PASSWORD: "rootroot"

cd ..

mkdir app02

cd app02

*vi docker-compose.yml*

#Alternate

version: '3.7'

services:

#Nginx Service

webserver:

image: nginx:alpine

container\_name: Nginx

restart: unless-stopped

ports:

- "80:80"

- "443:443"

depends\_on:

- dbserver

dbserver:

image: mysql:5.7

container\_name: Mysqldb

restart: unless-stopped

ports:

- "3306:3306"

environment:

MYSQL\_ROOT\_PASSWORD: "rootroot"

MYSQL\_PASSWORD: "rootroot"

MYSQL\_DATABASE: test

volumes:

- db\_data:/var/lib/mysql

volumes:

db\_data:

# Docker Compose commands

docker-compose up -d

# To rebuild images

docker-compose up -d --build

docker-compose ps

curl <http://localhost>

docker exec -it Mysqldb mysql -uroot -prootroot

docker-compose logs

docker-compose logs -f webserver

docker-compose logs --tail="2" webserver

docker-compose stop

docker-compose start

docker-compose restart

#Suspends all processes in the specified containers. On Linux, this uses the freezer cgroup.

docker-compose pause

docker-compose unpause

#Remove stopped containers.

docker-compose rm

#Stop and Remove - containers and Networks

docker-compose down

#Removes images as well

docker-compose down --rmi all