############
Docker
############

1) What is Docker ?

2) Why we need to go for Docker

3) Docker Architecture (Dockerfile + Image + Registry + Container)

4) Docker Setup in Linux VM (Ubuntu)

5) Docker Commands

6) Dockerizing Java web app (war)

7) Dockerizing Spring Boot app (jar)

8) Storing Docker image to docker hub

-> Docker is a containerization software

-> Docker is used to simplify our application deployment process

-> Using Docker we can run our application in any platform.

1) Dockerfile

2) Docker Image

3) Docker Hub (Docker registry)

4) Docker Container

=> Dockerfile contains instructions to build docker image

=> Using Dockerfile we can create Docker Image

=> Docker Image contains application code + required dependencies

=> Docker Images we can store in Docker Hub (Docker Registry)

Note: In realtime , we will use AWS ECR to store docker images

(ECR -> Elastic Container Registry)

=> Docker Images are used to create Docker Containers

=> Our application will execute inside Docker Container.

Note: To work with Docker we need to install Docker Software (Docker Engine)

==============
Docker Setup
===============

```
## install docker
$ curl -fsSL get.docker.com | /bin/bash

## Verify docker installation
$ sudo docker -v


## pull docker image from docker hub
$ sudo docker pull ashokit/spring-boot-rest-api

## Display docker image we have in our machine
$ sudo docker images


## Run Docker Image
$ sudo docker run -p 9090:9090  ashokit/spring-boot-rest-api
```

Note: Enable 9090 port number in security group

## Access application using URL like below

http://public-ip:9090/welcome/Ashok

==
Assignment : Dockerize Spring Boot Application
===

###### Spring Boot with Docker : https://youtu.be/iGz0cFwt5vI

```
# display docker images available in our machine
sudo docker images

# Remove docker image
sudo docker rmi <image-name/image-id>

# Display Running Docker containers
sudo docker ps

# Display stopped docker containers
sudo docker ps -a

# Remove container
sudo docker rm <container-id>

# Download docker image
```

sudo docker pull <image-name>

# It is used to clean up docker content
sudo docker system prune -a

# Run docker container in detached mode
sudo docker run -d -p 9090:9090 ashokit/spring-boot-rest-api

=============
Dockerfile
=============

-> Dockerfile contains set of instructions to build docker image

-> In Dockerfile we will use DSL keywords

FROM
MAINTAINER
COPY
ADD
WORKDIR
RUN
CMD
EXPOSE
ENTRYPOINT
VOLUME
ARG

=> FROM is used to specify base image to create our image

FROM open-jdk:11

FROM python:3.3

FROM tomcat:9.5

FROM node:15.1

=> MAINTAINER is used to specify Dockerfile Author

MAINTAINER Ashok<ashok@gmail.com>

=> COPY is used to copy the files from source to destination

COPY target/sb-api.jar /usr/app/sb-api.jar

COPY target/app.war /usr/tomcat/webapps/app.war

=> ADD is used to copy the files from source to destination

```
ADD   <URL>  /usr/tomcat/webapps/app.war
```

=> RUN is used to execute instructions while creating docker image

```
RUN   apt install git
RUN   apt install maven
RUN   git clone <repo>
```

=> CMD is used to execute instructions while container creation

```
CMD  java  -jar  jarfile
```

=> WORKDIR is used to specify working directory

=> ENTRYPOINT is used to execute instructions while creating container

Ex : ENTRYPOINT ["java" "-jar" "sb-rest-api.jar"]

=> EXPOSE is used to specify container running port number

Ex : EXPOSE 9090

=> Create Dockerfile with below content

file name : Dockerfile

```
FROM ubuntu

RUN echo ' hi'
RUN echo 'hello'
CMD echo 'how are you'
```

=> Create Docker image using above docker file

```
# Create Docker Image
$ sudo docker build -t first-image .

# Tag Docker Image
$ sudo docker tag <image-name> <tag-name>
Ex : $ sudo docker tag first-image ashokit/first-image

# Login into Docker Hub Account
$ sudo docker login

# Push Docker Image to Docker Hub
```

```
$ sudo docker push <tag-name>
```

==
How to Dockerize Java Web Application
==

-> Java Web Application will be packaged as war file

-> war file we will deploy in tomcat server webapps folder

### GIT repo : https://github.com/ashokitschool/maven-web-app.git

-> Below is the Dockerfile to dockerize java web application

```
FROM tomcat:8.0.20-jre8

COPY target/01-maven-web-app*.war /usr/local/tomcat/webapps/maven-web-app.war

EXPOSE 8080
```

===
How to Dockerize Spring Boot Application
===

-> Spring Boot application will be packaged as jar file

-> To run Spring Boot application we need to run jar file

-> Spring Boot provides embedded server for web applications

## GIT Hub repo : https://github.com/ashokitschool/spring-boot-docker-app.git

-> Below is the Dockerfile for Spring Boot Application

```
FROM open-jdk:11

COPY target/app.jar /usr/app/

WORKDIR /usr/app/

EXPOSE 8080

ENTRYPOINT [ "java", "-jar", "app.jar" ]
```

==================
Docker Compose
==================

=> Docker Compose is a tool which is used to manage multi-container based application.

=> We can create multiple containers at a time by using Docker Compose.

=> Using Docker Compose we can define dependencies among the containers

Ex: App Container Depends On DB Container

=> We will provide containers information to Docker Compose tool using docker-compose.yml

=> Docker Compose yml file contains below elements

version: It represents version number

services: It represents containers information

volumes: It represents storage

network: It represents connectivity

# Create Docker Container using Docker Compose YML

```
$ docker-compose up

$ docker-compose down

$ docker-compose stop

$ docker-compose start

$ docker-compose ps
```

========================
Docker Compose Setup
========================

# Download docker compose
```
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

# Give permissions
```
$ sudo chmod +x /usr/local/bin/docker-compose
```

# How to check docker compose is installed or not
```
$ docker-compose –version
```

===
Execute Spring Boot App with MySQL DB using Docker Compose
===

# clone git repo
```
git clone https://github.com/ashokitschool/spring-boot-mysql-docker-compose.git
```

```
# package our application
mvn clean package

# Create Docker Image for Spring Boot Application
sudo docker build -t spring-boot-mysql-app .

# Create Containers using Docker Compose in detached mode
sudo docker-compose up -d
```