TRAINING CONTENT

Cloud Computing

YOUR NEXT DESTINATION

OF SOFTWARE OUTSOURCING

Lecture Outline

- Docker
- Container
- Containers vs Virtual Machines
- Docker Architecture Diagram
- Docker Installation
- Dockerfile
- Push Image into Registry
- Docker Networking
- Docker Compose
- Docker Compose Example
- Kubernetes Overview

Docker terminology(2/2)

Base and child images:

- **Base images** are images that have no parent image – they don't build on or derive from another image, usually images that represent an operating system (e.g. Ubuntu, busybox).

- **Child images** are images that build on base images and add additional functionality, most images you're likely to make will be child images.

Official and user images

- **Official images** are images that are officially maintained and supported by the people at Docker. These are typically one word long. Examples include python, ubuntu, and hello-world.

- **User images** are images created and shared by people who use Docker. They usually build on base images and add functionality. Typically these are formatted as user/image-name.

Uploading the images in Docker Registry

Create one docker hub account or any other platform (like AWS,Azure,GCP)
Create a repository on Docker Hub [ex: dockertrainig]

Build your image locally:
docker build -t username/repository_name .

Then Login through CLI:
docker login

For push execute below command:
docker push username/reponame

Go to your docker hub account you can see your uploaded image.

Run container

Run a container using docker Hub Image:
docker run -it nginx
docker run --publish 80:80 nginx

Run Multiple container using docker Hub Image :
docker run --name MyContainer1 <same image id>
docker run --name MyContainer2 <same image id>
docker run --name MyContainer3 <same image id>

Copy Docker image from one machine to another?

Step 0 : List docker image

Step 1 : Get the docker repository and tag name in machine 1

Step 2 : Save the Docker image as a tar file in machine 1
docker save -o <generated tar file name> <repository_name:tag_name>1
docker save -o app_build_001.tar repository_name:tag_name 1

Step 3 : Copy the tar file to machine 2
scp app_build_001.tar remote_username@IP:/remote/directory

Step 4 : Load the image into Docker (in the machine 2)
docker load -i <path to image tar file>
docker load -i app_build_001.tar

Prerequisite Install Python for demo

Install Python 3 Using apt:
python –version
Step 1: Update and Refresh Repository Lists:
sudo apt update
Step 2: Install Supporting Software:
sudo apt install software-properties-common
Step 3: Add Deadsnakes PPA:
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
Step 4: Install Python 3:
sudo apt install python3.8
python3 --version

Write Code

create a Simple Python Application using the Flask Framework –user root:

```
mkdir python-docker
cd python-docker/
cd /python-docker
python3 -m venv .venv
source .venv/bin/activate
(.venv) $ python3 -m pip install Flask
(.venv) $ python3 -m pip freeze > requirements.txt
(.venv) $ touch app.py
```

Now Open app.py file using any editor and write and Save:

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def  hello_world():
      return 'Hello, Docker!'
@app.route('/bjit')
def  bjit():
      return 'Hello From, BJIT!'
```

Test the application:

```
cd python-docker
source .venv/bin/activate
(.venv) $ python3 -m flask run
```

Now Open Another Terminal and write and check output: curl localhost:5000 or localhost:5000/bjit

Write Dockerfile

```
# syntax=docker/dockerfile:1
FROM python:3.8-slim-buster
WORKDIR /app
COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt
COPY . .
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Write Dockerfile

FROM - The base image or the root image.
MAINTAINER - Defines the author of that particular image. It can take a first name, last name, or email.
RUN - It is a command that carries the set of instructions for executing the Dockerfile while building the image.
CMD - The command provides a revert for a Dockerfile that’s executing.

Build an image

```
docker build --tag python-docker .
docker images
docker tag python-docker:latest python-docker:v1.0.0
```

Run

Run a container using docker Hub Image:
docker run -it nginx

Run a container using your own Image:
docker run python-docker
curl localhost:5000
Output: Failed to connect to localhost port 5000: Connection refused

docker run --publish 8000:5000 python-docker
curl localhost:8000
Output: Hello, Docker!

Push Image into AWS Registry

Push Image to AWS ECR:
Prerequisite:
1.Configure AWS CLI
2.Install Docker

Command for ecr , Image push:
#aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com

#docker tag e9ae3c220b23 aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:tag

#docker push aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:tag

Thank You

Get in touch with us:

www.bjitgroup.com