

DEVOPS

Date :

MON	TUE	WED	THU	FRI	SAT	SUN
<input type="checkbox"/>						

EXP-2

i) Install Docker

WSL --> Install

WSL --> Set - default - Version 2

ii) Verify Installation

docker --version

docker info

iii) Run pull and run docker image

docker pull Ubuntu

docker run -it Ubuntu

exit

iv) Manage docker container.

docker ps --for running

docker ps -a --for all

docker stop < containerid>

docker rm < containerid>

Date :

MON TUE WED THU FRI SAT SUN

v) Manage Images

docker images

docker rmi <image_id>

vi) Create a flask app

app.py

from flask import Flask

app = Flask(__name__)

@app.route("/")

def home():

return "Welcome to DockerHub!"

if __name__ == '__main__':

app.run(host='0.0.0.0', port=5000)

requirements.txt

Flask

Dockerfile

FROM python:3.8-slim

WORKDIR /app

COPY . .

Date :

MON TUE WED THU FRI SAT SUN

RUN pip install -r requirements.txt
CMD ["python", "app.py"]

A) Build and run application

docker build -t flask-app .

docker run -d -p 5000:5000 flask-app

You can verify me by going to

<http://localhost:5000>

You should see

Welcome to DockerHub!

Date:

MON TUE WED THU FRI SAT SUN

Chp-3

I) Create project Structure

mkdir microservice-app

cd microservice-app

mkdir app db.

1) Create file app.py in app folder

app.py

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def home():
```

```
    return "Welcome to Microservice!"
```

```
If __name__ == '__main__':
```

```
    app.run(host='0.0.0.0', port=5001)
```

requirements.txt

flask

Date :

MON TUE WED THU FRI SAT SUN

Dockerfile

```
FROM python:3.8-slim  
WORKDIR /app  
COPY requirement.txt  
RUN pip install -r requirement.txt  
COPY . .  
CMD ["python", "app.py"]
```

iii) Create a docker-compose.yml in root folder.

version: '3'

services:

web :

build : ./app

ports:

- "5000:5000"

depends_on:

- db

Date :

MON TUE WED THU FRI SAT SUN



db :

image: mysql:5.7

restart: always

environment:

MYSQL_DATABASE: sampledb

MYSQL_USER: user

MYSQL_PASSWORD: password

MYSQL_ROOT_PASSWORD: rootpass

ports:

- "3306:3306"

iv) Run the setup

docker-compose up --build

v) Check the application

vi) docker-compose down ~~stop~~

Date :

MON	TUE	WED	THU	FRI	SAT	SUN
<input type="checkbox"/>						

Exp-1

Initial Configuration

- I) Go to browser type git and install
- II) Check github version

git --version

- III) Configuration

git config --global user.name "yan"
user.email "yan@yan.com"

git config --global --list

- IV) git init

git add .

git status

git commit -m "first commit"

git branch -M main

Date :

MON TUE WED THU FRI SAT SUN

git remote add origin —
git push -u origin main

git branch feature
git checkout feature

git add feature.txt
git commit -m "Second commit"
git push origin feature

git checkout main
git pull origin main
git merge feature
git push origin main

Date :

MON TUE WED THU FRI SAT SUN



Exp-5

Steps

I) Install Java JDK 21 .msi

II) Install maven from <https://maven.apache.org> • ~~zip~~

Add maven to the system ~~variable path~~

Confirm ~~if~~ Maven works

`mvn -v`

III) Create maven project

`mvn archetype:generate -DgroupId=`
com.example -DarchetypeArtifactId=demo-project
-DarchetypeArtifactId=maven-archetype-
quickstart -DinteractiveMode=false

Date:

MON TUE WED THU FRI SAT SUN



iv) Go to folder

~~cd~~ ~~cp~~ cd demo-project

v) Go to the file and edit java program

Code

package com.example;

public class App {

 public static void main(String[] args)

{

 System.out.println("Hello, Maven
 project works!");

}

}

vi) Compile the project
mvn compile

Date:

MON	TUE	WED	THU	FRI	SAT	SUN
<input type="checkbox"/>						

vi) Test

mvn test

vii) package the project

mvn package

This will create a .jar file in the target folder

target / demo-project-1.0-SNAPSHOT.jar

viii) Run the program

java -cp target / demo-project-1.0-SNAPSHOT.jar com.example.App

Output

Hello, Maven project works!

Date :

MON TUE WED THU FRI SAT SUN

Exp-6

part-A - Repeat same as Exp-5

Part-B

Installation of Gradle

i) install gradle and set the path

and verify by

gradle -version

ii) go to the project directory and
create 2 files

build.gradle

plugins {

id 'java'

3

group = 'com.example'

version = '1.0-SNAPSHOT'

Date :

MON TUE WED THU FRI SAT SUN



repositories {

mavenCentral()

}

dependencies {

testImplementation 'junit:junit:4.13.2'

}

java {

sourceCompatibility = JavaVersion.VERSION_17

targetCompatibility = JavaVersion.VERSION_17

}

Setting.gradle

rootProject.name = 'demo-project'

(iii)

Build the project and test

gradle build

gradle test

Date :

MON	TUE	WED	THU	FRI	SAT	SUN
<input type="checkbox"/>						

iv) Run the Jar file created

java -cp ~~target~~ build \libs\demo-project-L-O-SNAPSHOT.jar com.example.App

It will give the output

Welcome to Gradle