**SWE-645 HW4**

This document covers setup and usage steps for the Python-based reimplementation of Homework 3 (Student Survey microservice), up through submitting survey data via the API. This document provides detail explanation of the steps along with screenshots. It has URLs of each completed tasks for reference. The tasks given in assignment is given below:

1. Python microservices for api development

2. Create MySQL DB on Amazon RDS

3. Building an image for the application and pushing it to Docker Hub using Docker Desktop.

4. Use Rancher to set up AWS EC2 instances to deploy the application on a Kubernetes cluster.

5. Install and Set Up Rancher

6. Deploy the Kubernetes Cluster Using Rancher UI

7. Set Up AWS EC2 Instance for Jenkins

8. Set Up GitHub Repository for project

9. Create a CI/CD Pipeline with Jenkins

**Step 1: Python Development instead of java microservices**

1. We need to create an app.py the flask app

**A screenshot of a computer

Description automatically generated**

**Step 2: We need to create an amazon RDS database**

1. We need to launch AWS Academy and then in amazon RDS and choose MySQL 8x

version.

Give credentials as

Master: admin, password as: Surveydbs123

A screenshot of a computer

Description automatically generated

1. In the flask app i.e. in app.py we need to link our AWS RDS dbs that we created in above step. Below is the code to link

# Database configuration (update with your AWS RDS details)

app.config['SQLALCHEMY\_DATABASE\_URI'] = os.getenv('DATABASE\_URL', 'mysql+pymysql://admin:Surveydbs123@database-1.c1o2r9fxwliu.us-east-1.rds.amazonaws.com:3306/studentdb')

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db = SQLAlchemy(app)

**Step 3: Building an image for the application and pushing it to Docker Hub using Docker Desktop.**

1. First, create an account through Docker hub (https://hub.docker.com/). Then download

and install docker desktop through docker website (https://www.docker.com/). Verifying

the installation:

A screenshot of a computer

Description automatically generated

1. Login to the docker desktop through the app that was downloaded before and login to

docker hub through web browser.

A screenshot of a computer

Description automatically generated

1. Preparing the folder for microservices. In order to push into the Docker hub, First Create

Hw4\_project folder and push all the relevant files into the folder.

**A screenshot of a computer

Description automatically generated**

1. Now create docker file in the same directory as of Studentmicroservices.war file. To create docker file we use **nano Dockerfile** then add content as shown below

A screenshot of a computer

Description automatically generated

1. Create a docker image by opening the terminal at the directory created above. The command used to create docker image is as follows:

docker build -t student-survey4 .

A screenshot of a computer program

Description automatically generated

1. After that we can push our image to docker hub using following commands.

à docker login -u prasanna0307

A screenshot of a computer

Description automatically generated

🡪docker tag student-survey4:latest prasanna0307/student-survey4:latest

🡪docker push prasanna0307/student-survey4:latest

A screenshot of a computer

Description automatically generated

🡪 It can be visible in the following image which is accessible from docker hub

A screenshot of a computer

Description automatically generated

1. We need to run the image created using the following command.

🡪 docker run -d -p 5001:5000 --name student-survey4-container student-survey4

1. Launch the browser and navigate to localhost 8080 then add api/surveys which will return the records from the POSTMAN app that we inserted.

<http://localhost:5001/api/surveys>

A screenshot of a computer

Description automatically generated

**Step 4: Set up AWS EC2 instances in order to deploy the application on a Kubernetes cluster**

1. We need to install an AWS EC2 instance to do this step. Log in to the AWS console

https://aws.amazon.com/ and create an account.

1. After logging into AWS Academy, clicking on Launch AWS Academy Learner Lab, we

need to click on start lab then click on AWS green dot we can launch AWS console.

1. In AWS console, Click on Launch Instance option to create an instance with the following

specifications.

à AMI used: Ubuntu 24.04 HVM SSD

à t3.large

à 30 GiB Storage

1. We need two of those instances for this task. And we need to create an Elastic IP address and need to allocate them to the instances we created which will be useful for Rancher and Jenkins.
2. Now we need to assign inbound and outbound rules by clicking security tab under the

instances.

à Type: Custome TCP, Port : 8080, Source: 0.0.0.0/0

1. After setting up all the configurations we need to launch EC2 instance and needs to connect to it. We need to enter username as “root” and then click on connect we will the following in the new window

A screenshot of a computer

Description automatically generated

1. We need to run following Commands in EC2 instance.

à sudo apt update

à sudo apt install docker.io -y

After this docker should be successfully installed

A screenshot of a computer program

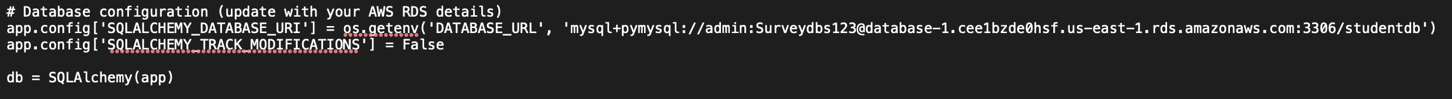
Description automatically generated

1. It indicates that docker is successfully installed. And the instance will look as follows:

A screenshot of a computer

Description automatically generated

1. We need to copy these credentials in python file



**Step 5: Install and Set Up Rancher**

1. Click https://www.rancher.com/quick-start and navigate to the link. Under Rancher will find Deploy Rancher we have some instructions as shown below.

A screenshot of a computer

Description automatically generated

1. After running the command docker ps, we will get the following output

A black and white rectangular object with white text

Description automatically generated

1. We need to go to the “Public IPV4 DNS” in the instance page. There it redirects to Rancher. It initially shows the warning but we can continue with that. It looks something like this:

A screenshot of a computer

Description automatically generated

1. We need to copy the command and replace the container id with (8139d473f71a) this is the one we got above after running docker ps command so that it will displays the password.
2. Copy the password and paste it in the rancher login screen that is displayed above. After that we will be able to create a new password(Surveydbs1234). After successfully login it will display the following screen.

A screenshot of a computer

Description automatically generated

**Step 6: Deploy the Kubernetes Cluster Using Rancher UI**

1. As the Rancher is ready, with this UI we can create a cluster and deploy the application on it.
2. In the dashboard shown above we can click on the create button. It will displays the following

A screenshot of a computer

Description automatically generated

1. Click on Custom, It will displays a form to fill in which we need to give all the details about the cluster like name as “swehw4” leaving remaining default values as it and then proceed clicking on create.
2. In step1, we need to check all the boxes. Now copy the command which is there in the step2 and paste it in the EC2 instance and run it. The cluster is in updating state for few minutes then it will become active.

A screenshot of a computer

Description automatically generated

1. Once the cluster is active, click on explore mode
2. After active state, click on the created cluster and on the left side display we have several options. Choose workloads and then deployment. Click create then we will get the form. Give details as below:

à Name : hw4deployed, Replicas(number of pods): 3, Container image:

prasanna0307/ prasanna0307/survey-flask-app:latest (which is available on docker hub), click on Add port or service" button under Networking, Provide the following information:

àprotocol: TCP, private container port: 8080, service type: "node Port," and name: "nodeport." Leave remaining details as is and click create. It will take few minutes to update and after that it will displays the active status as shown below.

A screenshot of a computer

Description automatically generated

1. Once it is in active status, go inside the deployment and go to services tab.
2. In services tab, under the target option beside the nodeport. It will redirect to the new window. In that for the URL add /api/surveys

à If I click on 8080 nodeport I got this URL:

<https://34.205.61.176/k8s/clusters/c-m-wnk6wcbv/api/v1/namespaces/default/services/http:hw4deployed:8080/proxy>

🡪 After adding api/surveys

<https://34.205.61.176/k8s/clusters/c-m-wnk6wcbv/api/v1/namespaces/default/services/http:hw4deployed:8080/proxy/api/surveys>

A screenshot of a computer

Description automatically generated

**Step 7: Set Up AWS EC2 Instance for Jenkins**

1. In this step, Jenkins is used which will be helpful for developing CI/CD pipeline that will automatically build and update our source code.
2. Download and keep aside the KubeConfig file from the Rancher UI i.e. In Cluster Management dashboard.
3. We need to install Jenkins for that we need an AWS EC2 instance on the AWS lab. We can use the 2nd instance that we created in Step2 which named as “swehw4jenkins”.

A screenshot of a computer

Description automatically generated