**SWE645 – HW3 – Installation and Setup Instruction**

**Team Members:**

Preeti Bhattacharya G01302375,

Uday Kumar Kamalapuram G01340201,

Paras Kanth G01333828.

1. **Download the required resources**
2. First, we downloaded the Node.js for the windows.
3. Node.js basically provides asynchronous event-driven JavaScript runtime which helps us to use JavaScript at the backend.
4. Node.js provides node package manager which helps us to install to package which helps us to build the backend.
5. Using NPM we installed express.js framework which provided an upper hand over the vanilla JavaScript like starting the server and defining different routes for different request like get, post, delete etc.
6. **Creating the RDS(AWS) database**
7. **Create Database:** Search RDS in AWS console and after that you see the option to create database. Click on that button.
8. After landing on that page choose standard create and after that select Mysql in the engine options.

Graphical user interface, application

Description automatically generated

1. Under the template we want to use free tier.

Graphical user interface, text, application

Description automatically generated

1. After that type the Unique Identifier for your database and also configure your credentials which consist of username and password.

Graphical user interface, text, application, email

Description automatically generated

1. In connectivity tab select yes to for the public access or you can configure according to your choice.
2. For database authentication it is preselected to be password authentication. So, keep it like that

Graphical user interface, text, application, email

Description automatically generated

1. Now click on the create database button to create your database.

Graphical user interface, text, application, email

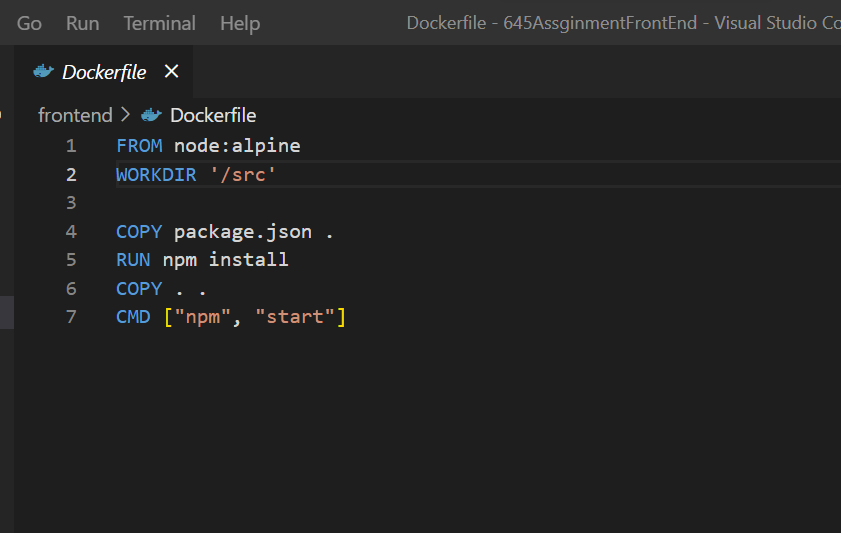
Description automatically generated

1. Once database is created it will look like below image.

Graphical user interface, text, application, Teams

Description automatically generated

1. **Creating Docker Image and Pushing it in Docker hub for front-end**
2. Install “docker desktop” on your desktop.
3. Create your account using “<https://hub.docker.com/>”.
4. In Visual Studio, create a file called “Dockerfile”. And write the below code in your file.



1. Open Cmd and use this command “docker build -f Dockerfile -t finalfrontend:0.1”.
2. We are verifying by running “docker run -it -p 3000:8080 finalfrontend:0.1”

to check whether the image is running properly.

1. To check, open the browser at “<http://localhost:3000>”.
2. In cmd login docker using “docker login -u <your username>”.
3. Change the name of your image to be <your username on dockerhub>/<name of the app>:<image tag> using the docker tag command.
4. In our case: “docker tag finalfrontend:0.1 paras108/finalfrontend:0.1”
5. Verify your image on dockerhub and it should be accessible through the internet.
6. **Creating Docker Image and Pushing it in Docker hub for Back-end**
7. Install “docker desktop” on your desktop.
8. Create your account using “<https://hub.docker.com/>”.
9. In Visual Studio, create a file called “Dockerfile”. And write the below code in your file.

Text

Description automatically generated

1. Open Cmd and use this command “docker build -f Dockerfile -t paras108/backend:0.2”.
2. We are verifying by running “docker run -it -p 3000:8080 paras108/backend:0.2”

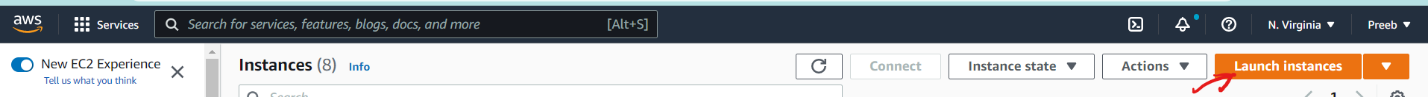
to check whether the image is running properly.

1. To check, open the browser at “[http://localhost:8182”](http://localhost:8182)
2. In cmd login docker using “docker login -u <your username>”.
3. Change the name of your image to be <your username on dockerhub>/<name of the app>:<image tag> using the docker tag command.
4. In our case: “docker tag finalfrontend:0.1 paras108/ paras108/backend:0.2”
5. Verify your image on dockerhub and it should be accessible through the internet.

Graphical user interface, text, application, email

Description automatically generated

1. **Creating AWS EC2 instance and installing Rancher**
2. Log in to the AWS console https://aws.amazon.com/ and create an account.
3. Navigate to EC2.
4. Launch a new instance.



1. Search Ubuntu AMI.
2. Search for Ubuntu AMI and select the Ubuntu Server 20.04 LTS (HVM), SSD Volume Type image and click Select.
3. Select t2.medium to support Kubernetes.
4. Then click at Next: Add Storage.
5. Then click at Next: Add Tags.And then we have given Key as “Name” and Value as “rancher0304”.
6. Then click at Next: Configure Security Group.
7. Then create SSH,HTTP,HTTPS security groups with “anywhere” accessible and also add Custom TCP port 8080 for Jenkins to work with and “anywhere” accessible.
8. Then click Review and Launch.
9. Review your configure and Launch.
10. Create a new key pair to access EC2 machine and download it for further use.
11. Then click Launch Instances.
12. Then click View Instance after that you can see your instance up and running after some time.
13. To see its configuration like public IPv4 address, Public IPv4 DNS click on instance ID.

C. **To create an IAM user**

1. Navigate to the IAM user dashboard.
2. Click on users on the left panel.
3. Click Add users.
4. We have given the username as “[ClusterUser2](https://us-east-1.console.aws.amazon.com/iam/home#/users/ClusterUser2)”.
5. And under AWS access type, select Access Key- Programmatic access.
6. Then click Next:Permissions.
7. Then click on Attach existing policies directly and click next
8. Then review configuration and click create User.
9. Click on Download .csv and download the Access Key and Secret Key required for cluster configuration.

Graphical user interface, text, application

Description automatically generated

D. **Setting up Rancher to create Kubernetes Cluster**

1. Connect to your EC2 machine using the Public IPv4 DNS and Private Key file using the command: ssh -i .pem ubuntu@<publicIPv4DNS>
2. Run command after login: sudo apt-get update.
3. Install docker using sudo apt install docker.io
4. Verify docker using sudo docker-v
5. Go to rancher.io and the on right click on get started and scroll down.
6. You will see command: sudo docker run --privileged=true -d -- restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher.

Graphical user interface, text, application

Description automatically generated

1. Verify rancher container is running using command sudo docker ps.
2. Copy the AMES ID and save it for later.
3. And go to your instance and click Public IPv4DNS.
4. After starting up rancher.It will take a couple of minutes for UI to come up.
5. To run Use command sudo docker logs <container Id>. Container Id is the AMES ID which we saved earlier.
6. Once the UI is up it will show a command for getting password sudo docker logs<container Id> 2>&1 | grep “Bootstrap Password:” .
7. Run the command as sudo user as we have installed docker as sudo user and it should output the password.
8. Paste the password in the UI and then click: Set the specific password to use.Save and generate the new password as you desire.
9. Click continue.
10. Navigate to cluster management on the left panel.
11. Click on Cloud Credentials and Click Create.
12. Give your Access Key and Secret Key from the earlier saved file.
13. Then select your “Amazon” as cloud provider.
14. Enter Name,Access Key, Secret key and select region as us-east-1.
15. Click create.
16. You should now see your credentials on the Cloud Credential Dashboard.
17. Then we will now navigate to Cluster Management and click on create cluster
18. Select Amazon EC2 and click next.
19. Then give cluster name and create node template with previous cloud credentials created.
20. Click create and dont change the VPC or security group.
21. In “IAM Profile Name” given AWS role name created.
22. To create IAM role:
23. Login to AWS console using created user.
24. Click Create Role
25. Select EC2.
26. Click on 'Generate Policy'
27. Give a policy name and paste the following in the JSON,

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "\*",

"Resource": "\*"

}

]

}

1. Complete creating Policy and Role.
2. We created “adminerole” role

Graphical user interface, text, application, email

Description automatically generated

1. We gave “adminerole” as "IAM Instance Profile Name"
2. Then given name to template and click on create.
3. Given Name Prefix for cluster and selected etcd, Control Plane, Worker checkbox and click on create button.
4. In some time, the cluster will be in an active state from the provision state.

E. **Deploying Docker Image on Cluster. (Create different clusters for backend and frontend)**

1. Once the created cluster in Active, click on created cluster on the left panel.
2. Click on Projects/Namespaces.
3. Create a Project with a unique name and in that project create a unique namespace.
4. After that expand ‘Workload’ on the left panel, under that we will get the “Deployment” option.
5. In deployments click on the create button.
6. Select the namespace we have created previously and give Name for deployment.
7. We selected replicas 3.
8. In container image given the Docker hub container image we pushed previously.

Graphical user interface, application

Description automatically generated

1. In ports added nodeport with any name “np” and Given Private container port- 8080, Protocol - TCP, Listening- 30103 as shown in above image.
2. Click on Create.
3. Once deployment is done, we will be able to see the pods on the left panel.
4. In the left panel, under Service Discovery.

Graphical user interface, text, application

Description automatically generated

1. We will be able to see our deployment with Active status. Click on the np hyperlink. It will open a page and give endpoint route you will get the application.
2. To access the URL externally we need to expose the URL we have used node port for that, and we need to add that port to our cluster running EC2 instance.
3. **Following are the steps to** **add port to security group of EC2 instance:**
4. Add the port 30103 in the cluster which is created via. Rancher.
5. Click on the cluster running EC2 instance name which is created by rancher in our case it is “Cluster03” and then go to its security group.
6. Then open the security group link and click on edit inbound rules.

Graphical user interface, text, application

Description automatically generated

1. Then click on “Add rule” Select Custom TCP and then add the port in the inbound rules in port range section and in source section make it available to “anywhere”.
2. Click on Save rules.
3. Then open instance using <DNS>:<node port number>/<endpoint routes>. Here in our case nodeport is 30103 and the endpoint route URL is “feed/getAllpost”.
4. We were able to see our backend details in the browser as below.

Text

Description automatically generated with medium confidence

1. And for the frontend click on the EC2 instance, which is created via rancher, click on its Public Ipv4 and then in its URL add the port 30500 as shown in **15th point** and don’t forget to remove the “s” from https and hit enter.
2. You can see the application running as shown in below image and this is the frontend of the application.
3. This is the Survey Page.

Graphical user interface, application

Description automatically generated

1. This is See Survey page.

Graphical user interface, website

Description automatically generated