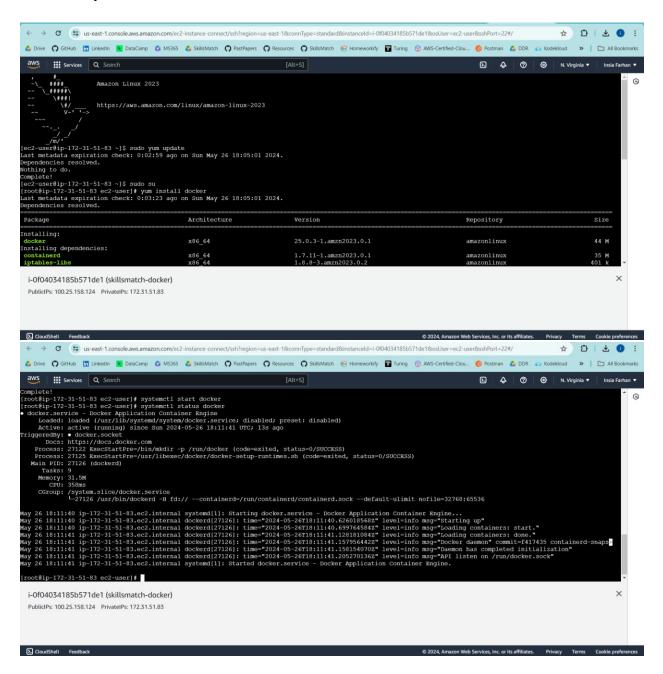
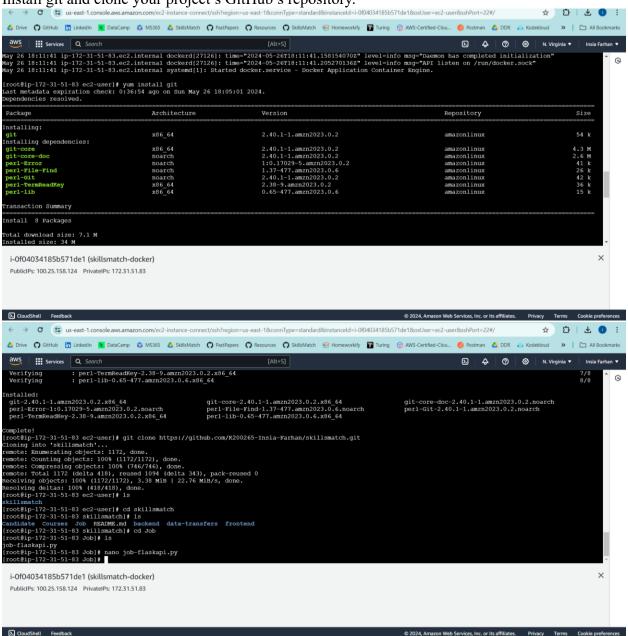
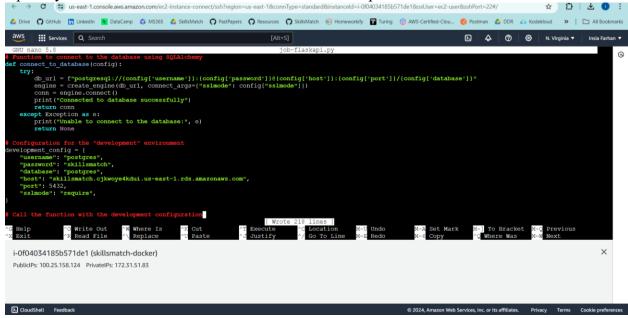
- 1. Creation of RDS. (Refer the document)
- 2. Connect to your EC2 instance. Install docker and start it.



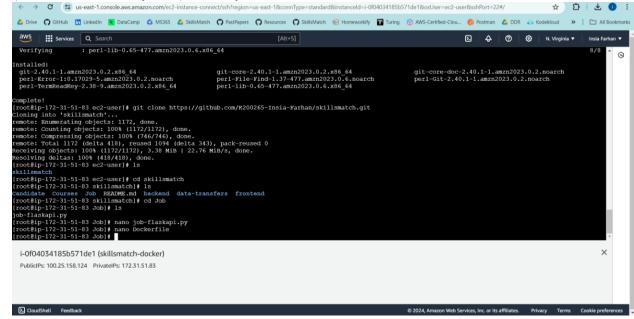
3. Install git and clone your project's GitHub's repository.



4. Update your code with RDS link in database connection places.



5. (For recommender system) Create your Dockerfile.



Use the official Python image as base

FROM python:3.8-slim

Set environment variables for Flask

ENV FLASK_APP=job-flaskapi.py \

 $FLASK_RUN_HOST = 0.0.0.0 \setminus$

FLASK RUN PORT=2003

Set the working directory in the container

WORKDIR /app

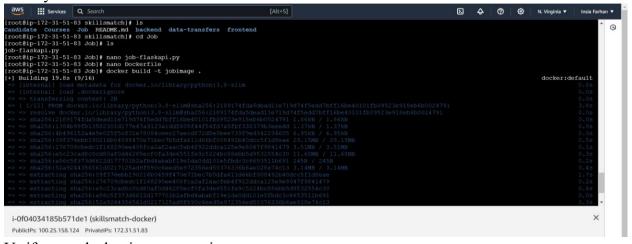
Copy the current directory contents into the container at /app

COPY . /app

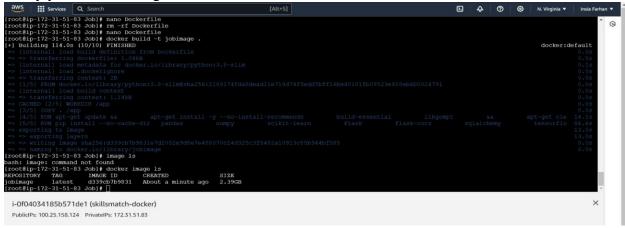
Install system dependencies

```
RUN apt-get update && \
apt-get install -y --no-install-recommends \
build-essential \
libgomp1 \
&&\
apt-get clean && \
rm -rf /var/lib/apt/lists/*
# Install Python dependencies
RUN pip install --no-cache-dir \
pandas \
numpy \setminus
scikit-learn \
Flask \
flask-cors \
sqlalchemy \
tensorflow \setminus \\
psycopg2-binary \
gunicorn
# Expose port 2003 to the outside world
EXPOSE 2003
# Command to run the Flask application
CMD ["gunicorn", "-b", "0.0.0.0:2003", "job-flaskapi:app"]
```

6. Build your dockerfile.



7. Verify your docker images creation.



8. For frontend:

Use the official Node.js image as a base

FROM node:latest

Install Next.js globally

RUN npm install -g next

Set the working directory

WORKDIR /myapp

Copy the application files

COPY..

Install dependencies

RUN npm install && \

npm install nodemon && \

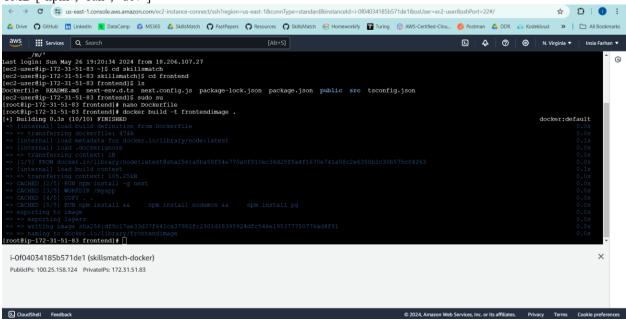
npm install pg

Expose port 3000

EXPOSE 3000

Run the development server

CMD ["npm", "run", "dev"]



9. For backend:

Use the official Node.js image as base

FROM node

Set the working directory inside the container

WORKDIR /app

Copy package.json and package-lock.json to the working directory

COPY package*.json ./

Install dependencies

RUN npm install

Install Sequelize CLI globally

RUN npm install -g sequelize-cli

RUN npm install -g nodemon

Copy the rest of the application code to the working directory

COPY..

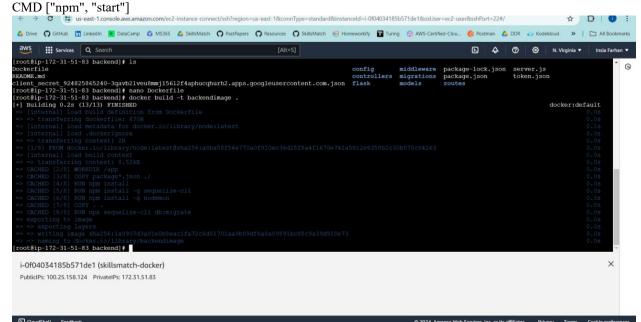
Run sequelize migrations

RUN npx sequelize-cli db:migrate

Expose any ports the app is expecting

EXPOSE 5000

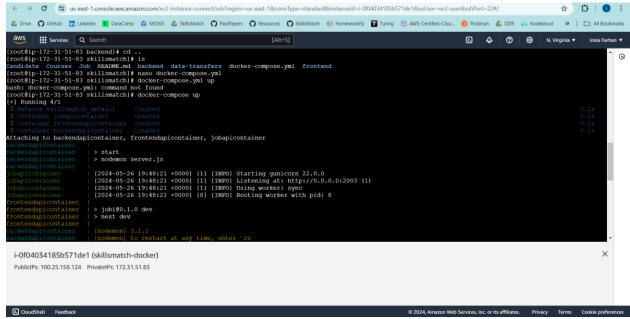
Start the application



10. Second method: By docker compose

Docker-compose file:

```
version: '3.8'
services:
 frontend:
  image: frontendimage:latest
  container_name: frontendapicontainer
  ports:
   - "3000:3000"
 backend:
  image: backendimage:latest
  container_name: backendapicontainer
  ports:
   - "5000:5000"
 job:
  image: jobimage:latest
  container name: jobapicontainer
  ports:
   - "2003:2003"
```



11. Pushing docker images on dockerhub.

(For dockerhub you need images names to start from docker hub's namespace/reponame:tagname)

