

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

The screenshot shows an AWS CloudShell session on an Amazon Linux 2023 EC2 instance. The terminal output is as follows:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-51-83 ~]$ sudo yum update
Last metadata expiration check: 0:02:59 ago on Sun May 26 18:05:01 2024.
Dependencies resolved.
Nothing to do.
Complete!

[ec2-user@ip-172-31-51-83 ~]$ sudo su
[root@ip-172-31-51-83 ec2-user]# yum install docker
Last metadata expiration check: 0:03:23 ago on Sun May 26 18:05:01 2024.
Dependencies resolved.

Package Architecture Version Repository Size
Installing:
docker x86_64 25.0.3-1.amzn2023.0.1 amazonlinux 44 M
Installing dependencies:
containerd x86_64 1.7.11-1.amzn2023.0.1 amazonlinux 35 M
iptables-libs x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 401 k
```

Package	Architecture	Version	Repository	Size
Installing: docker	x86_64	25.0.3-1.amzn2023.0.1	amazonlinux	44 M
Installing dependencies: containerd	x86_64	1.7.11-1.amzn2023.0.1	amazonlinux	35 M
iptables-libs	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux	401 k

i-0f04034185b571de1 (skillsmatch-docker)  
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

```
Complete!
[root@ip-172-31-51-83 ec2-user]# systemctl start docker
[root@ip-172-31-51-83 ec2-user]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Sun 2024-05-26 18:11:41 UTC; 13s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Process: 27122 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
    Process: 27125 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Main PID: 27126 (dockerd)
      Tasks: 9
     Memory: 31.5M
        CPU: 358ms
    CGroup: /system.slice/docker.service
            └─27126 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

May 26 18:11:40 ip-172-31-51-83.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
May 26 18:11:40 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:40.626018568Z" level=info msg="Starting up"
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:40.699764584Z" level=info msg="Loading containers: start."
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:41.128181084Z" level=info msg="Loading containers: done."
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:41.157956442Z" level=info msg="Docker daemon" commit=f417435 containerd-snapshots=
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:41.158154070Z" level=info msg="Daemon has completed initialization"
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:41.205270136Z" level=info msg="API listen on /run/docker.sock"
May 26 18:11:41 ip-172-31-51-83.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.

[root@ip-172-31-51-83 ec2-user]#
```

i-0f04034185b571de1 (skillsmatch-docker)  
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

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

The screenshot shows an AWS CloudShell terminal session. The terminal output is as follows:

```
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:41.158154070Z" level=info msg="Daemon has completed initialization"
May 26 18:11:41 ip-172-31-51-83.ec2.internal dockerd[27126]: time="2024-05-26T18:11:41.205270136Z" level=info msg="API listen on /run/docker.sock"
May 26 18:11:41 ip-172-31-51-83.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.

[root@ip-172-31-51-83 ec2-user]# yum install git
Last metadata expiration check: 0:36:54 ago on Sun May 26 18:05:01 2024.
Dependencies resolved.

Package Architecture Version Repository Size
Installing:
git x86_64 2.40.1-1.amzn2023.0.2 amazonlinux 54 k
Installing dependencies:
git-core x86_64 2.40.1-1.amzn2023.0.2 amazonlinux 4.3 M
git-core-doc noarch 2.40.1-1.amzn2023.0.2 amazonlinux 2.6 M
perl-Error noarch 1:0.17029-5.amzn2023.0.2 amazonlinux 41 k
perl-File-Find noarch 1.37-477.amzn2023.0.6 amazonlinux 26 k
perl-Git noarch 2.40.1-1.amzn2023.0.2 amazonlinux 42 k
perl-TermReadKey x86_64 2.38-9.amzn2023.0.2 amazonlinux 36 k
perl-lib x86_64 0.65-477.amzn2023.0.6 amazonlinux 15 k

Transaction Summary
Install 8 Packages
Total download size: 7.1 M
Installed size: 34 M

i-0f04034185b571de1 (skillsmatch-docker)
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83
```

The terminal output continues with the verification and installation of the packages:

```
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64
Verifying : perl-lib-0.65-477.amzn2023.0.6.x86_64

Installed:
git-2.40.1-1.amzn2023.0.2.x86_64 git-core-2.40.1-1.amzn2023.0.2.x86_64 git-core-doc-2.40.1-1.amzn2023.0.2.noarch
perl-Error-1:0.17029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.6.noarch perl-Git-2.40.1-1.amzn2023.0.2.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.6.x86_64

Complete!
[root@ip-172-31-51-83 ec2-user]# git clone https://github.com/K200265-Insia-Farhan/skillsmatch.git
Cloning into 'skillsmatch'...
remote: Enumerating objects: 1172, done.
remote: Counting objects: 100% (1172/1172), done.
remote: Compressing objects: 100% (746/746), done.
remote: Total 1172 (delta 418), reused 1094 (delta 343), pack-reused 0
Receiving objects: 100% (1172/1172), 3.38 MiB | 22.76 MiB/s, done.
Resolving deltas: 100% (418/418), done.
[root@ip-172-31-51-83 ec2-user]# ls
skillsmatch
[root@ip-172-31-51-83 ec2-user]# cd skillsmatch
[root@ip-172-31-51-83 skillsmatch]# ls
Candidate Courses Job README.md backend data-transfers frontend
[root@ip-172-31-51-83 skillsmatch]# cd Job
[root@ip-172-31-51-83 Job]# ls
job-flaskapi.py
[root@ip-172-31-51-83 Job]# nano job-flaskapi.py
[root@ip-172-31-51-83 Job]#
```

The terminal output ends with the same information as the first screenshot:

```
i-0f04034185b571de1 (skillsmatch-docker)
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83
```

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

The screenshot shows the AWS CloudShell interface with a nano editor open to the file `job-flaskapi.py`. The code defines a `connect_to_database` function that uses SQLAlchemy to connect to a PostgreSQL database. The configuration for the "development" environment is updated with the following details:

- `username`: "postgres"
- `password`: "skillsmatch"
- `database`: "skillsmatch"
- `host`: "skillsmatch.cjkwoye4kdui.us-east-1.rds.amazonaws.com"
- `port`: 5432
- `sslmode`: "require"

The function is then called with this configuration. The terminal output at the bottom shows the instance ID `i-0f04034185b571de1` and its public/private IP addresses.

5. (For recommender system) Create your Dockerfile.

The screenshot shows the AWS CloudShell interface where dependencies are being installed and the repository is cloned. The terminal output includes:

- Verification of `perl-lib-0.65-477.amzn2023.0.6.x86_64`.
- Installation of various packages including `git`, `perl`, and `perl-core`.
- Completion of the installation process.
- Cloning the repository: `git clone https://github.com/K200265-Insia-Farhan/skillsmatch.git`.
- Navigation into the `skillsmatch` directory and then into the `job` subdirectory.
- Opening `job-flaskapi.py` and `Dockerfile` in the nano editor.

The terminal output at the bottom shows the instance ID `i-0f04034185b571de1` and its public/private IP addresses.

```
# 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 \
    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 \
scikit-learn \
Flask \
flask-cors \
sqlalchemy \
tensorflow \
psycpg2-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.

```

[aws] [Services] [Search] [Alt+S] [N. Virginia] [Insia Farhan]
[root@ip-172-31-51-83 skillsmatch]# ls
candidate Courses Job README.md backend data-transfers frontend
[root@ip-172-31-51-83 skillsmatch]# cd Job
[root@ip-172-31-51-83 Job]# ls
job-flaskapi.py
[root@ip-172-31-51-83 Job]# nano job-flaskapi.py
[root@ip-172-31-51-83 Job]# nano Dockerfile
[root@ip-172-31-51-83 Job]# docker build -t jobimage .
[+] Building 19.8s (9/16)
=> [internal] load metadata for docker.io/library/python:3.8-slim 0.0s
=> [internal] load .dockerignore 0.0s
=> transferring context: 2B 0.0s
=> [1/12] FROM docker.io/library/python:3.8-slim@sha256:2189174fda5dead11e719d74f5edd7bffc16be40101fb09523e918eb6b0024791 3.8s
=> resolve docker.io/library/python:3.8-slim@sha256:2189174fda5dead11e719d74f5edd7bffc16be40101fb09523e918eb6b0024791 0.0s
=> sha256:2189174fda5dead11e719d74f5edd7bffc16be40101fb09523e918eb6b0024791 1.86kB / 1.86kB 0.0s
=> sha256:1304b99f813582301d177e47a9123a1dddf605f44f54fd7a5fbf338379b3eeedd 1.37kB / 1.37kB 0.0s
=> sha256:4b496152a4e5e025f3c931e78084c0ec2faecd723de0bee739f9ed342234605 6.95kB / 6.95kB 0.0s
=> sha256:09cf76eb19021c8a490470e71bec7b2dfe11d6b2f098492b40dcd3f1d8ee 25.15MB / 25.15MB 0.0s
=> sha256:276709cbedc1f168290ee408fca2af2aacfeb4f922ddca125e9e8047f9641479 3.51MB / 3.51MB 0.0s
=> sha256:ebc3cad8c0d89af0d46285ecf0fa3de6531fe9c5224bce8ebbd5932554c30 11.68MB / 11.68MB 0.0s
=> sha256:a86c5f373d6612d177701b2afbd4ababf19e1da0dd101e5fbd3c6653511b691 245B / 245B 0.0s
=> sha256:15a9244356561d0217125ad9f890c6eed5e97235eed50376336b6ae028e74c13 3.14MB / 3.14MB 0.0s
=> extracting sha256:09f376eb130216b0459f470e71bec7b2dfe11d6b2f098492b40dcd3f1d8ee 1.7s
=> extracting sha256:276709cbedc1f168290ee408fca2af2aacfeb4f922ddca125e9e8047f9641479 0.2s
=> extracting sha256:4b496152a4e5e025f3c931e78084c0ec2faecd723de0bee739f9ed342234605 0.6s
=> extracting sha256:a86c5f373d6612d177701b2afbd4ababf19e1da0dd101e5fbd3c6653511b691 0.0s
=> extracting sha256:15a9244356561d0217125ad9f890c6eed5e97235eed50376336b6ae028e74c13 0.3s
i-Of04034185b571de1 (skillsmatch-docker)
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

```

## 7. Verify your docker images creation.

```

[aws] [Services] [Search] [Alt+S] [N. Virginia] [Insia Farhan]
[root@ip-172-31-51-83 Job]# nano Dockerfile
[root@ip-172-31-51-83 Job]# rm -rf Dockerfile
[root@ip-172-31-51-83 Job]# nano Dockerfile
[root@ip-172-31-51-83 Job]# docker build -t jobimage .
[+] Building 114.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> transferring Dockerfile: 1.04kB 0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim 0.1s
=> [internal] load .dockerignore 0.0s
=> transferring context: 2B 0.0s
=> [1/5] FROM docker.io/library/python:3.8-slim@sha256:2189174fda5dead11e719d74f5edd7bffc16be40101fb09523e918eb6b0024791 0.0s
=> [internal] load build context 0.0s
=> transferring context: 1.14kB 0.0s
=> CACHED [2/5] WORKDIR /app 0.0s
=> [3/5] COPY . /app 0.0s
=> [4/5] RUN apt-get update && apt-get install -y --no-install-recommends build-essential libgomp1 && apt-get cle 14.1s
=> [5/5] RUN pip install --no-cache-dir pandas numpy scikit-learn Flask flask-cors sqlalchemy tensorflow 86.6s
=> exporting to image 13.0s
=> exporting layers 13.0s
=> writing image sha256:d339cb7b9831e7d2052e9d8e7e488070c24d325c3f8482a10913c85b944bf585 0.0s
=> naming image docker.io/library/jobimage 0.0s
[root@ip-172-31-51-83 Job]# image ls
bash: image: command not found
[root@ip-172-31-51-83 Job]# docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
jobimage latest d339cb7b9831 About a minute ago 2.39GB
[root@ip-172-31-51-83 Job]#
i-Of04034185b571de1 (skillsmatch-docker)
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

```

## 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"]
```

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0f04034185b571de1&osUser=ec2-user&sshPort=22#/
AWS CloudShell
Last login: Sun May 26 19:20:34 2024 from 18.206.107.27
[ec2-user@ip-172-31-51-83 ~]$ cd skillsmatch
[ec2-user@ip-172-31-51-83 skillsmatch]$ cd frontend
[ec2-user@ip-172-31-51-83 frontend]$ ls
Dockerfile README.md next-env.d.ts next.config.js package-lock.json package.json public src tsconfig.json
[ec2-user@ip-172-31-51-83 frontend]$ sudo su
[root@ip-172-31-51-83 frontend]# nano Dockerfile
[root@ip-172-31-51-83 frontend]# docker build -t frontendimage .
(+) Building 0.3s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 474B
=> [internal] load metadata for docker.io/library/node:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:latest@sha256:a8ba58f54e770a0f910ec36d25f8a4f1670e741a58c2e6358b2c30b575c84263
=> [internal] load build context
=> => transferring context: 155.25kB
=> CACHED [2/5] RUN npm install -g next
=> CACHED [3/5] WORKDIR /myapp
=> CACHED [4/5] COPY . .
=> CACHED [5/5] RUN npm install && npm install nodemon && npm install pg
=> exporting to image
=> => exporting layers
=> => writing image sha256:df9c17ae33d37f641ca37982fc2301d18395924dfc546e195377750776ad4f91
=> => naming to docker.io/library/frontendimage
[ec2-user@ip-172-31-51-83 frontend]#

i-0f04034185b571de1 (skillsmatch-docker)
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83
```

## 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
CMD ["npm", "start"]

```

The screenshot shows an AWS CloudShell terminal session. The user is in a directory named 'backend' and runs 'ls', showing files like 'Dockerfile', 'README.md', and various JSON files. They then run 'nano Dockerfile' to edit the file. After saving, they run 'docker build -t backendimage .' to build the image. The build process is shown in progress, with steps like 'load build definition from Dockerfile', 'transferring dockerfile', 'load metadata for docker.io/library/node:latest', 'load .dockerignore', 'transferring context', 'FROM docker.io/library/node:latest', 'load build context', 'transferring context', 'WORKDIR /app', 'COPY package\*.json ./', 'RUN npm install', 'RUN npm install -g sequelize-cli', 'RUN npm install -g nodemon', 'COPY . .', 'RUN npx sequelize-cli db:migrate', 'exporting to image', 'exporting layers', 'writing image sha256:1a0907d3a01e0b0eac1fa72ced51701aa9b89df6a8a09591bc88c9a39d910e73', and 'naming to docker.io/library/backendimage'. The build completes successfully, and the user runs 'ls' again, showing the new 'backendimage' directory. Below the terminal, a summary box shows the instance ID 'i-0f04034185b571de1 (skillsmatch-docker)' and its public/private IP addresses.

## 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"



```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0f04034185b571de1&osUser=ec2-user&sshPort=22#/  
AWS Services Search [Alt+S]  
[root@ip-172-31-51-83 backend]# cd ..  
[root@ip-172-31-51-83 skillsmatch]# ls  
Candidate Courses Job README.md backend data-transfers docker-compose.yml frontend  
[root@ip-172-31-51-83 skillsmatch]# nano docker-compose.yml  
[root@ip-172-31-51-83 skillsmatch]# docker-compose.yml up  
bash: docker-compose.yml: command not found  
[root@ip-172-31-51-83 skillsmatch]# docker-compose up  
(+) Running 4/1  
# Network skillsmatch_default Created 0.1s  
# Container jobapicontainer Created 0.1s  
# Container frontendapicontainer Created 0.1s  
# Container backendapicontainer Created 0.1s  
Attaching to backendapicontainer, frontendapicontainer, jobapicontainer  
backendapicontainer |> start  
backendapicontainer |> nodemon server.js  
backendapicontainer |  
jobapicontainer | [2024-05-26 19:48:21 +0000] [1] [INFO] Starting gunicorn 22.0.0  
jobapicontainer | [2024-05-26 19:48:21 +0000] [1] [INFO] Listening at: http://0.0.0.0:2003 (1)  
jobapicontainer | [2024-05-26 19:48:21 +0000] [1] [INFO] Using worker: sync  
jobapicontainer | [2024-05-26 19:48:21 +0000] [8] [INFO] Booting worker with pid: 8  
frontendapicontainer |> jobi@0.1.0 dev  
frontendapicontainer |> next dev  
frontendapicontainer |  
backendapicontainer | [nodemon] 3.1.1  
backendapicontainer | [nodemon] to restart at any time, enter `rs`  
  
i-0f04034185b571de1 (skillsmatch-docker)  
PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

## 11. Pushing docker images on dockerhub.

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

hub.docker.com/repository/create?namespace=insiafarhan

Repositories / Create Using 0 of 1 private repositories. [Get more](#)

### Create repository

Namespace: **insiafarhan** Repository Name: **skillsmatch1**

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

### Pushing images

You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname  
docker push new-repo:tagname
```

Make sure to replace **tagname** with your desired image repository tag.

### Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ Public Appears in Docker Hub search results

☐ Private Only visible to you

[Cancel](#) [Create](#)

```
AWS Services Search [Alt+S]  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
[root@ip-172-31-51-83 skillsmatch]# cd ..  
[root@ip-172-31-51-83 ec2-user]# cd skillsmatch  
[root@ip-172-31-51-83 skillsmatch]# cd frontend  
[root@ip-172-31-51-83 frontend]# docker login  
log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.  
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO.  
Learn more at https://docs.docker.com/go/access-tokens/  
  
Username: insiafarhan  
Password:  
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.  
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credentials-store  
  
Login Succeeded  
[root@ip-172-31-51-83 frontend]# docker build -t insiafarhan/skillsmatch1:frontendimage .  
(+) Building 0.3s (11/11) FINISHED  
=> [internal] load build definition from Dockerfile docker:default 0.0s  
=> => transferring dockerfile: 474B 0.0s  
=> [internal] load metadata for docker.io/library/node:latest 0.2s  
=> [auth] library/node:pull token for registry-1.docker.io 0.0s  
=> [internal] load .dockerignore 0.0s  
=> => transferring context: 2B 0.0s  
=> [1/5] FROM docker.io/library/node:latest@sha256:a8ba58f54e770a0f910ec36d25f8a4f1670e741a58c2e6358b2c30b575c84263 0.0s  
=> [internal] load build context 0.1s
```





```
aws Services Search [Alt+S]
=> CACHED (5/8) RUN npm install -g sequelize-cli 0.0s
=> CACHED (6/8) RUN npm install -g nodemon 0.0s
=> CACHED (7/8) COPY . . 0.0s
=> CACHED (8/8) RUN npm run sequelize-cli db:migrate 0.0s
=> exporting to image 0.0s
=> exporting layers 0.0s
=> writing image sha256:1a0907d3a01e0b8eac1fa72c6d51701aa9b89df6a8a09591bc8c9a39d910e73 0.0s
=> naming to docker.io/insiafarhan/skillsmatch1:backendimage 0.0s
[root@ip-172-31-51-83 backend]# docker push insiafarhan/skillsmatch1:backendimage
The push refers to repository [docker.io/insiafarhan/skillsmatch1]
c94ce086445d: Pushed
c566eccebf9c: Pushed
ea94567d0c07: Pushed
32e37ba47bb9: Pushed
b1175b343adf: Pushed
5be66988dd3: Pushed
a9e334bf2bf1: Pushed
3e01707c75b6: Layer already exists
f73e67c3d351: Layer already exists
cfc6fad4d981: Layer already exists
f2d41d232990: Layer already exists
734c0f0b65c2: Layer already exists
8845ab072c1c: Layer already exists
d7d4c2f9d26b: Layer already exists
bbela212f7e9: Layer already exists
backendimage: digest: sha256:dead8765a47afe57a3cb15f9bf1f3d257527552f571c382d606692000c6486cb size: 3469
[root@ip-172-31-51-83 backend]#
```

i-Of04034185b571de1 (skillsmatch-docker)

PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

```
Last login: Sun May 26 20:02:22 2024 from 18.206.107.28
[ec2-user@ip-172-31-51-83 ~]$ sudo su
[root@ip-172-31-51-83 ec2-user]# cd skillsmatch
[root@ip-172-31-51-83 skillsmatch]# ls
Candidate Courses Job README.md backend data-transfers docker-compose.yml frontend
[root@ip-172-31-51-83 skillsmatch]# cd Job
[root@ip-172-31-51-83 Job]# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ip-172-31-51-83 Job]# docker build -t insiafarhan/skillsmatch1:jobimage .
(+) Building 0.3s (11/11) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring Dockerfile 1.04kB 0.0s
```

i-Of04034185b571de1 (skillsmatch-docker)

PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

```
aws Services Search [Alt+S]
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/5] FROM docker.io/library/python:3.8-slim@sha256:2189174fda5dead11e719d74f5edd7bfff16e40101fb09523e918eb6b0024791 0.0s
=> [internal] load build context 0.0s
=> transferring context: 185B 0.0s
=> CACHED [2/5] WORKDIR /app 0.0s
=> CACHED [3/5] COPY . /app 0.0s
=> CACHED [4/5] RUN apt-get update && apt-get install -y --no-install-recommends build-essential libgomp1 && apt-get 0.0s
=> CACHED [5/5] RUN pip install --no-cache-dir pandas numpy scikit-learn Flask flask-cors sqlalchemy ten 0.0s
=> exporting to image 0.0s
=> exporting layers 0.0s
=> writing image sha256:d339cb7b9831e7d2052e9d9e7e488070c24d325c3f8482a10913c85b944bf585 0.0s
=> naming to docker.io/insiafarhan/skillsmatch1:jobimage 0.0s
[root@ip-172-31-51-83 Job]# docker push insiafarhan/skillsmatch1:jobimage
The push refers to repository [docker.io/insiafarhan/skillsmatch1]
c4b1c6d263f9: Pushed
095f42487a5d: Pushed
329692f55f73: Pushed
67c675b5394f: Pushed
9efd78ff03ee: Mounted from library/python
485ab5a9ccae: Mounted from library/python
0ca9c54a3df3: Mounted from library/python
146826fa3ca0: Mounted from library/python
5d4427064ecc: Mounted from library/python
jobimage: digest: sha256:b773517e9f4ad78a1177b97e8d59a11da38102abd8f9b7f21909eb363e233a52 size: 2209
[root@ip-172-31-51-83 Job]#
```

i-Of04034185b571de1 (skillsmatch-docker)

PublicIPs: 100.25.158.124 PrivateIPs: 172.31.51.83

dockerhub

Explore Repositories Organizations

Search Docker Hub

ctrl+K

?

I

insiafarhan / Repositories / skillsmatch1 / General

Using 0 of 1 private repositories. [Get more](#)

General Tags Builds Collaborators Webhooks Settings

insiafarhan/skillsmatch1

Updated 2 minutes ago

This repository does not have a description [🔗](#) INCOMPLETE

This repository does not have a category [🔗](#) INCOMPLETE

Docker commands

To push a new tag to this repository:

```
docker push insiafarhan/skillsmatch1:tagname
```

[Public View](#)

Tags

This repository contains 3 tag(s).

Tag	OS	Type	Pulled	Pushed
<a href="#">jobimage</a>		Image	---	2 minutes ago
<a href="#">backendimage</a>		Image	---	3 minutes ago
<a href="#">frontendimage</a>		Image	---	6 minutes ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)