

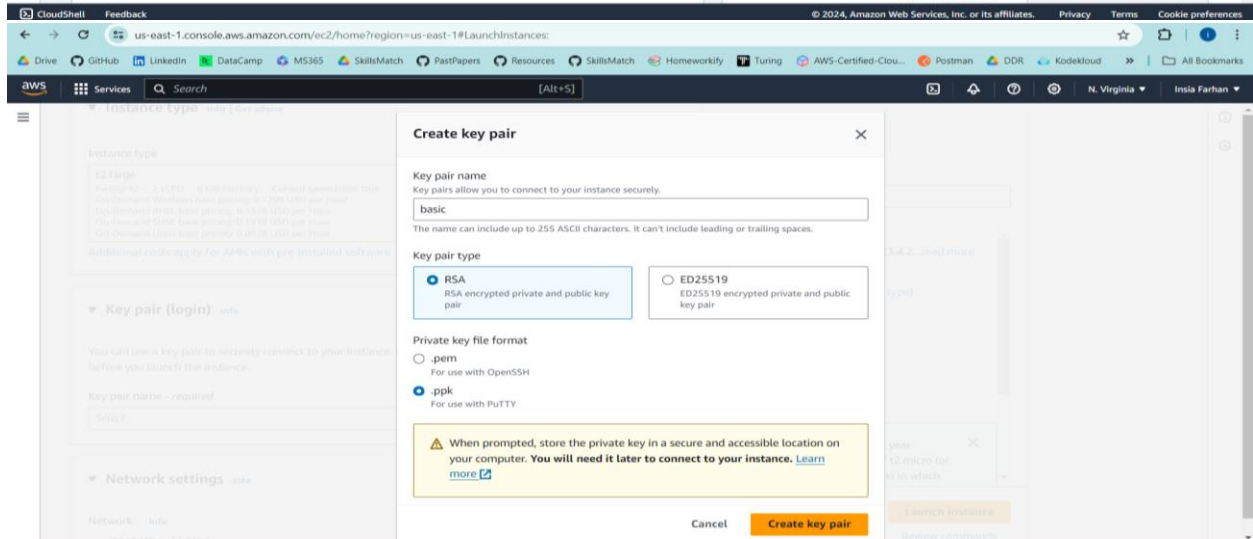
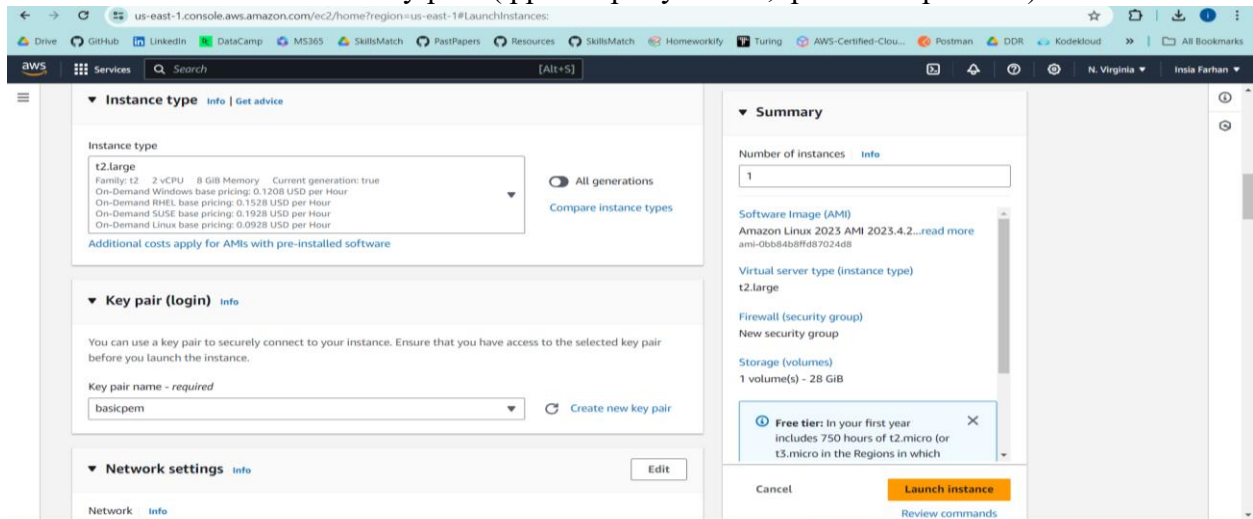
1. Log in to your AWS console and open EC2 via search bar. Click on Launch an instance.

The screenshot shows the AWS Management Console for the 'us-east-1' region. The left sidebar contains the 'EC2 Dashboard' and a list of services. The main content area is divided into several sections: 'Resources' (showing counts for various EC2 resources), 'Launch instance' (with a prominent orange 'Launch instance' button), 'Service health' (showing the status of the EC2 service), and 'EC2 Free Tier' (showing usage and offers). The 'Launch instance' section also includes a note about the default region (US East (N. Virginia)).

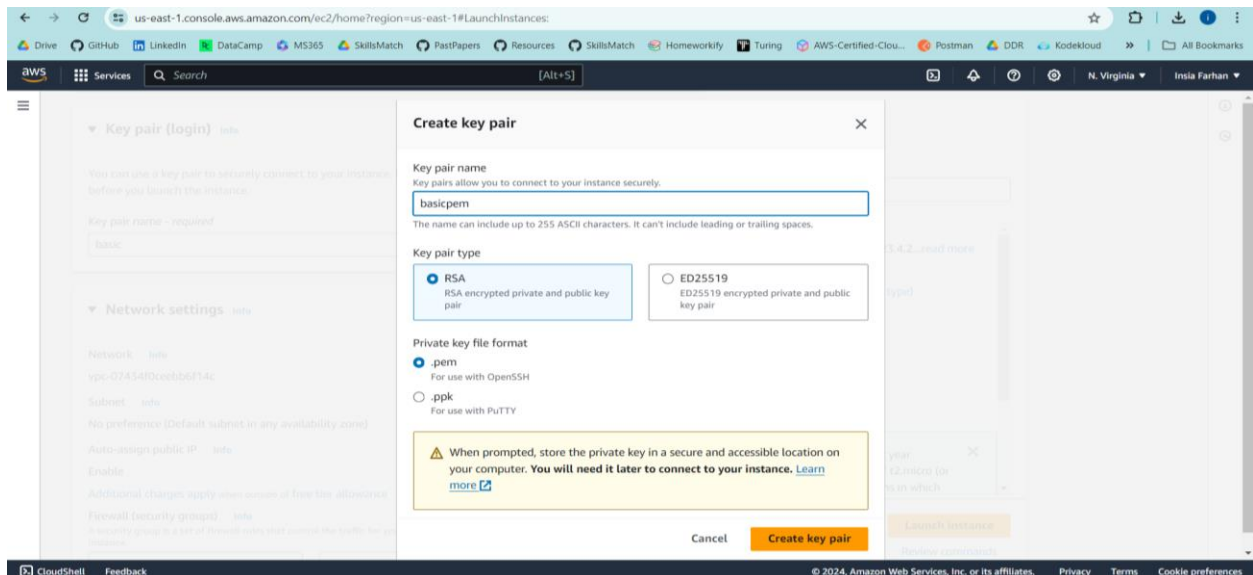
2. Choose EC2 configurations according to your project's requirements.

The screenshot shows the 'Launch an instance' configuration page in the AWS Management Console. The 'Name and tags' section has 'skillsmatch' entered as the name. The 'Application and OS Images (Amazon Machine Image)' section shows the 'Amazon Linux 2023 AMI' selected. The 'Summary' section on the right displays the configuration details: 1 instance, Amazon Linux 2023 AMI, t2.large instance type, new security group, and 1 volume of 28 GiB. A 'Free tier' notification indicates that the instance is eligible for the free tier in the first year. The 'Launch instance' button is highlighted in orange.

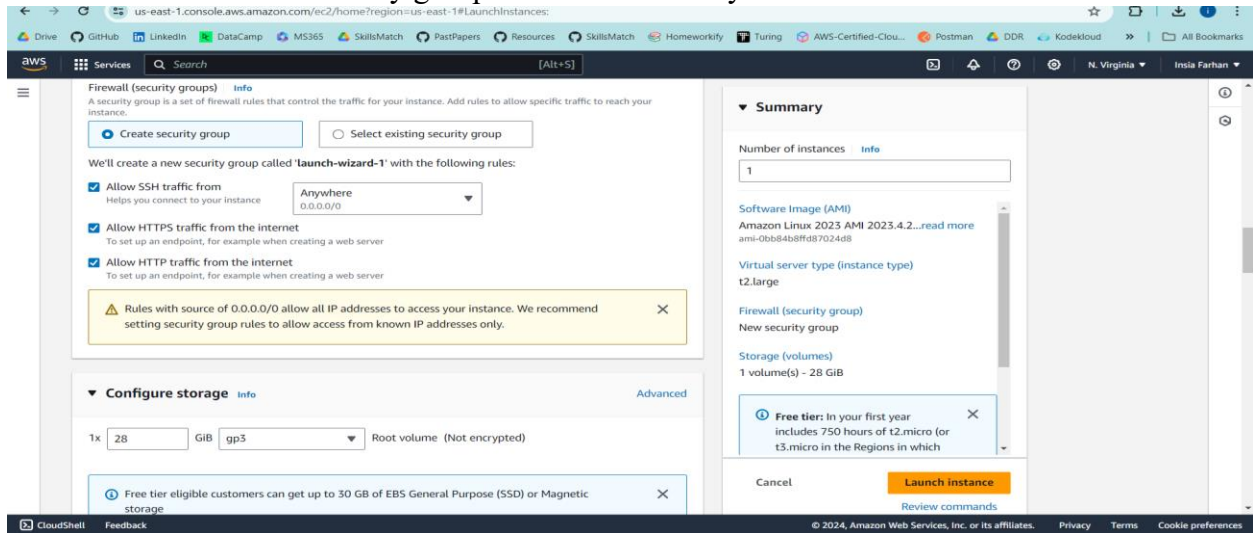
3. Click on Create new key pair. (.ppk for putty access, .pem for OpenSSH)



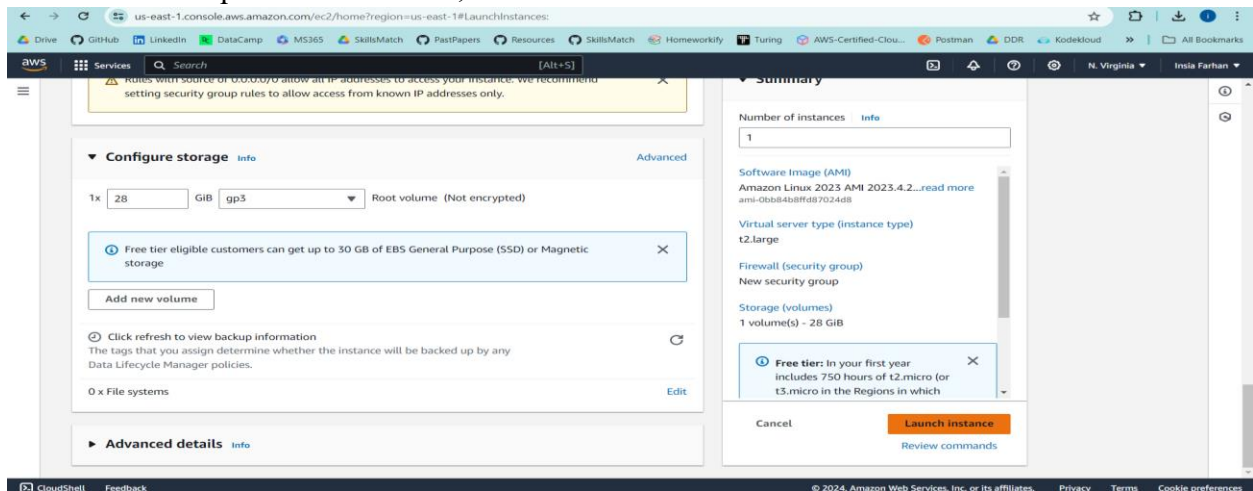
Or:



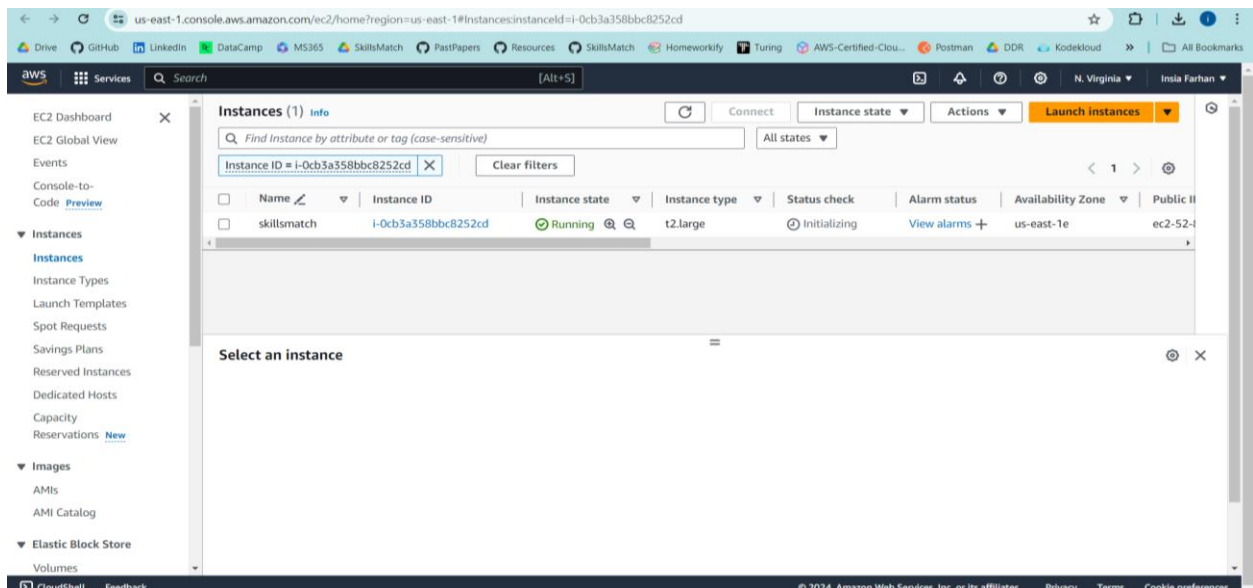
4. Click on create security group for custom security rules.



5. When requirements are met, click on Launch instance.



6. Click on the Instance ID for instance details.



7. Through security groups, edit the inbound rules, according to your applications ports.

The screenshot displays the AWS Management Console interface for editing the inbound rules of a security group. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:securityGroupId=sg-06072feea016ae5c6`.

The console shows the details for the security group `sg-06072feea016ae5c6 - launch-wizard-1`. The details section includes:

- Security group name: `launch-wizard-1`
- Security group ID: `sg-06072feea016ae5c6`
- Description: `launch-wizard-1 created 2024-05-26T10:11:11.814Z`
- VPC ID: `vpc-07434f0ceebb6f14c`
- Owner: `211125493694`
- Inbound rules count: `3 Permission entries`
- Outbound rules count: `1 Permission entry`

The **Inbound rules** tab is selected, showing a list of 3 rules. The rules are:

| Name | Security group rule... | IP version | Type | Protocol | Port range |
|------------|------------------------|------------|---------|-----------|------------|
| PostgreSQL | TCP | 5432 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 2003 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 2004 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |

The **Outbound rules** tab is also visible, showing a list of 1 rule. The rule is:

| Name | Security group rule... | IP version | Type | Protocol | Port range |
|------------|------------------------|------------|---------|-----------|------------|
| Custom TCP | TCP | 3000 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |

The **Tags** tab is also visible, showing a list of 0 tags.

A green notification banner at the top of the console indicates: **Inbound security group rules successfully modified on security group (sg-06072feea016ae5c6 | launch-wizard-1)**. The details section now shows:

- Inbound rules count: `9 Permission entries`
- Outbound rules count: `1 Permission entry`

The **Inbound rules** tab is selected, showing a list of 9 rules. The rules are:

| Name | Security group rule... | IP version | Type | Protocol | Port range |
|------------|------------------------|------------|---------|-----------|------------|
| PostgreSQL | TCP | 5432 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 2003 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 2004 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 2005 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 3000 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 5000 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 5000 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 5000 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |
| Custom TCP | TCP | 5000 | Anyw... | 0.0.0.0/0 | 0.0.0.0/0 |

8. Optional: To access the instance from windows via ssh and putty

The first screenshot shows the AWS Management Console with the 'Putty Key Generator' dialog box open. The dialog is configured to generate an SSH-1 (RSA) key pair with a key length of 2048 bits. The 'Public key for pasting into OpenSSH authorized_keys file' is displayed. The 'Generate' button is clicked.

The second screenshot shows the 'Putty Key Generator' dialog box with a warning message: 'Are you sure you want to save this key without a passphrase to protect it?'. The 'Yes' button is clicked.

The third screenshot shows the 'Connect to instance' page in the AWS Management Console. The 'SSH client' tab is selected, and the 'Public DNS' is copied. The 'Putty Configuration' dialog box is open, showing the 'Basic options for your PuTTY session' tab. The 'Host name (or IP address)' is set to 'ec2-52-87-220-206.compute-1.amazonaws.com' and the 'Port' is set to '22'. The 'Connection type' is set to 'SSH'. The 'Open' button is clicked.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-0cb3a358bbc8252cd

Services Search [Alt+S]

Connect to instance

Connect to your instance i-0cb3a358bbc8252cd (skillsmatch) using an SSH client.

Instance ID: i-0cb3a358bbc8252cd (skillsmatch)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is not public.
3. Run this command, if necessary, to ensure your key is not public: `ssh-keygen -f "/dev/null" -C "basicpem.pem"`

Public DNS copied: ec2-52-87-220-206.compute-1.amazonaws.com

Example: `ssh -i "basicpem.pem" ec2-user@ec2-52-87-220-206.compute-1.amazonaws.com`

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Putty Configuration

Category: Session

Options controlling SSH authentication:

- ☒ Display pre-authentication banner (SSH+2 only)
- ☐ Bypass authentication entirely (SSH+2 only)
- ☐ Disconnect if authentication succeeds trivially

Authentication methods:

- ☒ Attempt authentication using Pageant
- ☐ Attempt TIS or CryptoCard auth (SSH+1)
- ☒ Attempt "keyboard interactive" auth (SSH+2)

Authentication parameters:

- ☐ Allow agent forwarding
- ☐ Allow attempted changes of username in SSH+2

Private key file for authentication: C:\Users\Debi\Downloads\pem-basic.ppk

Buttons: About, Help, Open, Cancel

CloudShell Feedback

ec2-52-87-220-206.compute-1.amazonaws.com - PuTTY

GET GENUINE OFFICE

Putty Security Alert

The host key is not cached for this server: ec2-52-87-220-206.compute-1.amazonaws.com (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is: ssh-ed25519 255:54:42:66:41:0B:17:4D:05:29:42:79:46:90:0D:19B9:5e4E

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host press "Cancel" to abandon the connection.

Buttons: Help, More info, Accept, Connect Once, Cancel

ec2-user@ip-172-31-53-171:~\$

login as: ec2-user

Authenticating with public key "imported-openssh-key"

Amazon Linux 2023

<https://aws.amazon.com/linux/amazon-linux-2023>

[ec2-user@ip-172-31-53-171 ~]\$ sudo yum install docker

Last metadata expiration check: 0:25:04 ago on Sun May 26 10:10:41 2024.

Dependencies resolved.

| Package | Arch | Version | Repository | Size |
|---------|--------|-----------------------|-------------|------|
| docker | x86_64 | 25.0.3-1.amzn2023.0.1 | amazonlinux | 44 M |

Installing:

Installing dependencies:

| Package | Arch | Version | Repository | Size |
|------------------------|--------|-----------------------|-------------|-------|
| 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 |
| iptables-nft | x86_64 | 1.8.8-3.amzn2023.0.2 | amazonlinux | 183 k |
| libcgroup | x86_64 | 3.0-1.amzn2023.0.1 | amazonlinux | 75 k |
| libnetfilter_conntrack | x86_64 | 1.0.8-2.amzn2023.0.2 | amazonlinux | 58 k |
| libnftnl | x86_64 | 1.0.1-19.amzn2023.0.2 | amazonlinux | 30 k |
| libnftnl | x86_64 | 1.2.2-2.amzn2023.0.2 | amazonlinux | 84 k |
| pkgconf | x86_64 | 2.5-1.amzn2023.0.3 | amazonlinux | 83 k |
| runC | x86_64 | 1.1.11-1.amzn2023.0.1 | amazonlinux | 3.0 M |

Transaction Summary

Install 10 Packages

Total download size: 83 M

Installed size: 313 M

is this ok [y/N]: y

Downloading Packages:

| Package | Size | Download Speed | Time |
|--|--------|----------------|-------|
| (1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm | 401 kB | 5.0 MB/s | 00:00 |
| (2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm | 183 kB | 3.5 MB/s | 00:00 |
| (3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm | 75 kB | 3.9 MB/s | 00:00 |
| (4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm | 58 kB | 1.5 MB/s | 00:00 |
| (5/10): libnftnl-1.0.1-19.amzn2023.0.2.x86_64.rpm | 30 kB | 1.6 MB/s | 00:00 |
| (6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm | 84 kB | 4.3 MB/s | 00:00 |
| (7/10): pkgconf-2.5-1.amzn2023.0.3.x86_64.rpm | 83 kB | 2.5 MB/s | 00:00 |
| (8/10): runC-1.1.11-1.amzn2023.0.1.x86_64.rpm | 3.0 MB | 22 MB/s | 00:00 |
| (9/10): containerd-1.7.11-1.amzn2023.0.1.x86_64.rpm | 35 MB | 35 MB/s | 00:00 |
| (10/10): docker-25.0.3-1.amzn2023.0.1.x86_64.rpm | 44 MB | 33 MB/s | 00:01 |

```
ec2-user@ip-172-31-53-171:~$ sudo yum install -y docker
Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 9/10
Installing : libgroup-3.0-1.amzn2023.0.1.x86_64 9/10
Running scriptlet: docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Installing : docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Running scriptlet: docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

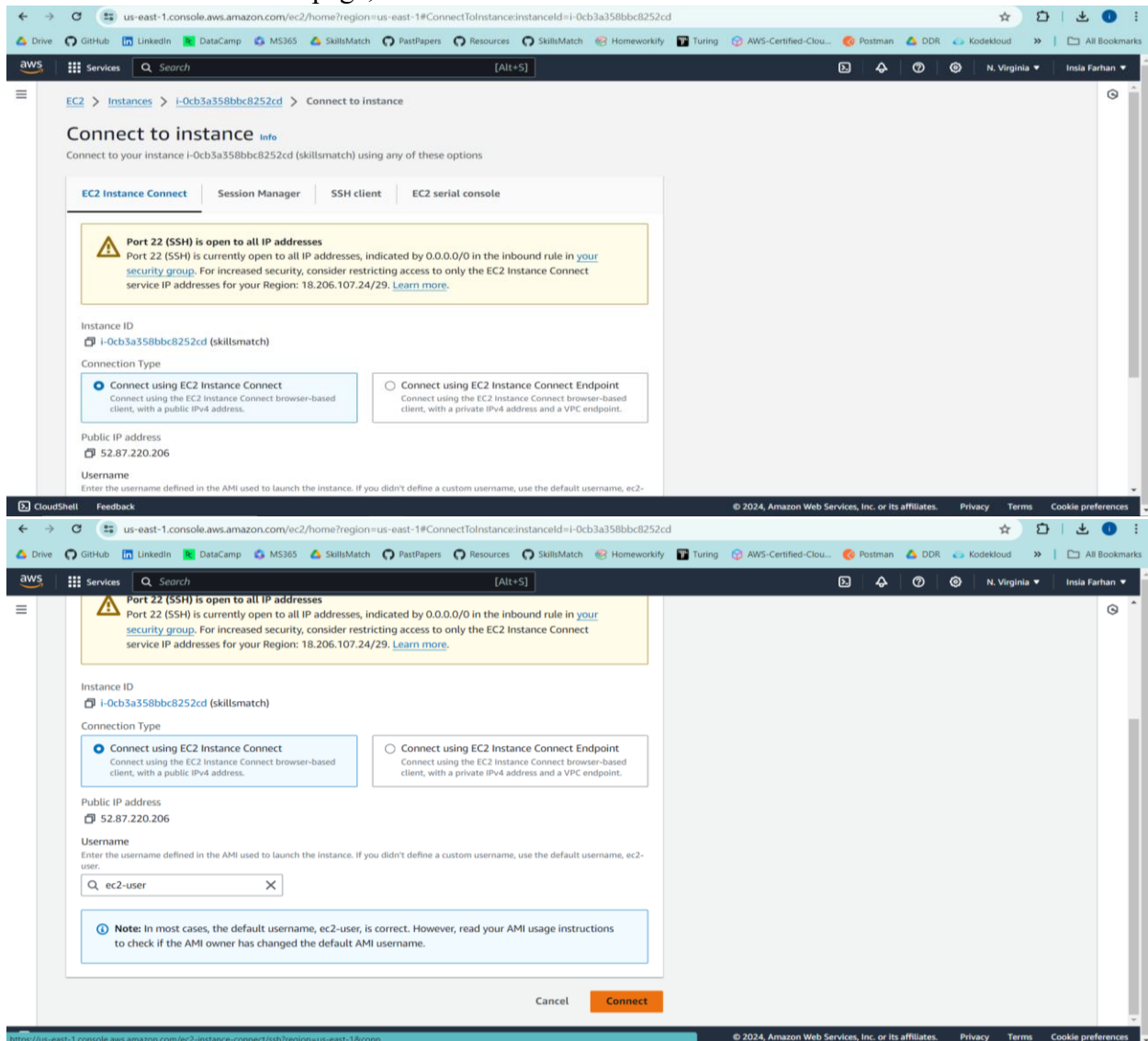
Verifying : containerd-1.7.11-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.3-1.amzn2023.0.1.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libgroup-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pipx-2.5.1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.11-1.amzn2023.0.1.x86_64 10/10

Installed:
  containerd-1.7.11-1.amzn2023.0.1.x86_64  docker-25.0.3-1.amzn2023.0.1.x86_64  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libgroup-3.0-1.amzn2023.0.1.x86_64  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnftnl-1.0.1-19.amzn2023.0.2.x86_64  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pipx-2.5.1.amzn2023.0.3.x86_64  runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-53-171 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-53-171 ~]$ sudo 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 10:44:38 UTC; 12s ago
     TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Process: 27258 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Process: 27259 ExecStartPre=/usr/libexec/docker/docker-setup-run-times.sh (code=exited, status=0/SUCCESS)
   Main PID: 27261 (dockerd)
      Tasks: 8
     Memory: 30.5M
        CPU: 341ms
    CGroup: /system.slice/docker.service
            └─2761 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

May 26 10:44:38 ip-172-31-53-171.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
May 26 10:44:38 ip-172-31-53-171.ec2.internal dockerd[27261]: time="2024-05-26T10:44:38.083510822Z" level=info msg="Starting up"
May 26 10:44:38 ip-172-31-53-171.ec2.internal dockerd[27261]: time="2024-05-26T10:44:38.143453422Z" level=info msg="Loading containers: start."
May 26 10:44:38 ip-172-31-53-171.ec2.internal dockerd[27261]: time="2024-05-26T10:44:38.749611659Z" level=info msg="Loading containers: done."
May 26 10:44:38 ip-172-31-53-171.ec2.internal dockerd[27261]: time="2024-05-26T10:44:38.774236702Z" level=info msg="Docker daemon" commit=f417435 containerd-snapshotter=false storage-driver=
May 26 10:44:38 ip-172-31-53-171.ec2.internal dockerd[27261]: time="2024-05-26T10:44:38.774353978Z" level=info msg="Daemon has completed initialization"
May 26 10:44:38 ip-172-31-53-171.ec2.internal dockerd[27261]: time="2024-05-26T10:44:38.812911089Z" level=info msg="API listen on /run/docker.sock"
May 26 10:44:38 ip-172-31-53-171.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
```

9. On EC2 instances page, click on the instance ID and click on Connect.



```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&region=us-east-1&sshPort=22#/

AWS Services Search [Alt+S]

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

Last login: Sun May 26 10:43:01 2024 from 101.53.224.70
[ec2-user@ip-172-31-53-171 ~]$ sudo 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 10:44:38 UTC; 2min 20s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Process: 27258 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
     Process: 27259 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Main PID: 27261 (dockerd)
      Tasks: 0
     Memory: 30.7M
        CPU: 358ms
      CGroup: /system.slice/docker.service
             └─27261 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

May 26 10:44:38 ip-172-31-53-171.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...

i-0cb3a358bbc8252cd (skillsmatch)
PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171
```

10. Install all the required dependencies of your project, create code's file, copy code in it, run it, and access through public URL. For example:
http://52.87.220.226:2003/recommend_jobs/11

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&region=us-east-1&sshPort=22#/

AWS Services Search [Alt+S]

Error loading data from the database. Exiting...
[root@ip-172-31-53-171 ec2-user]# pip3 install pyscpg2-binary
Collecting pyscpg2-binary
  Downloading pyscpg2_binary-2.9.9-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0 MB)
    [100%] 3.0 MB 5.5 MB/s
Installing collected packages: pyscpg2-binary
Successfully installed pyscpg2-binary-2.9.9
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[root@ip-172-31-53-171 ec2-user]# python3 jobs.py
2024-05-26 11:23:35.026243: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-05-26 11:23:35.030512: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-05-26 11:23:35.086609: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-05-26 11:23:36.173263: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Unable to connect to the database: (pyscpg2.OperationalError) could not translate host name "skillsmatch1.ci2ygcqcl5k.us-east-1.rds.amazonaws.com" to address: Name or service not known
(Background on this error at: https://sqlalche.me/e/20/e3q8)
Unable to connect to the database: (pyscpg2.OperationalError) could not translate host name "skillsmatch1.ci2ygcqcl5k.us-east-1.rds.amazonaws.com" to address: Name or service not known
(Background on this error at: https://sqlalche.me/e/20/e3q8)
Error loading data from the database. Exiting...
[root@ip-172-31-53-171 ec2-user]# nano jobs.py
[root@ip-172-31-53-171 ec2-user]# python3 jobs.py

i-0cb3a358bbc8252cd (skillsmatch)
PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171
```

11. Backend code copying from GitHub.

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&sshPort=22#/

AWS Services Search [Alt+S]

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

Last login: Sun May 26 13:14:49 2024 from 18.206.107.28
(ec2-user@ip-172-31-53-171 ~)$ sudo yum install git
Last metadata expiration check: 3:39:39 ago on Sun May 26 10:18:41 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-tib x86_64 0.65-477.amzn2023.0.6 amazonlinux 15 k

i-0cb3a358bbc8252cd (skillsmatch)
PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171
```



```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&sshPort=22#/  
AWS Services Search [Alt+S]  
N. Virginia Insia Farhan  
Complete!  
[ec2-user@ip-172-31-53-171 ~]$ git clone https://github.com/K200265-Insia-Farhan/skillsmatch.git  
Cloning into 'skillsmatch'...  
remote: Enumerating objects: 817, done.  
remote: Counting objects: 100% (817/817), done.  
remote: Compressing objects: 100% (609/609), done.  
remote: Total 817 (delta 127), reused 814 (delta 127), pack-reused 0  
Receiving objects: 100% (817/817), 3.28 MiB | 23.46 MiB/s, done.  
Resolving deltas: 100% (127/127), done.  
[ec2-user@ip-172-31-53-171 ~]$ git branch  
fatal: not a git repository (or any of the parent directories): .git  
[ec2-user@ip-172-31-53-171 ~]$ ls  
jobsrs.py skillsmatch  
[ec2-user@ip-172-31-53-171 ~]$ cd skillsmatch  
[ec2-user@ip-172-31-53-171 skillsmatch]$ cd backend  
[ec2-user@ip-172-31-53-171 backend]$ sudo yum update -y  
Last metadata expiration check: 3:43:59 ago on Sun May 26 10:18:41 2024.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-172-31-53-171 backend]$ sudo yum install -y curl  
Last metadata expiration check: 3:44:35 ago on Sun May 26 10:18:41 2024.  
Error!  
Problem: problem with installed package curl-minimal-8.5.0-1.amzn2023.0.4.x86_64  
- package curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 from @system conflicts with curl provided by curl-7.87.0-2.amzn2023.0.2.x86_64 from amazonlinux  
- package curl-minimal-7.87.0-2.amzn2023.0.2.x86_64 from amazonlinux conflicts with curl provided by curl-7.87.0-2.amzn2023.0.2.x86_64 from amazonlinux  
- package curl-minimal-7.88.0-1.amzn2023.0.1.x86_64 from amazonlinux conflicts with curl provided by curl-7.87.0-2.amzn2023.0.2.x86_64 from amazonlinux  
i-0cb3a358bbc8252cd (skillsmatch)  
PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

12. Backend code's dependencies installations.

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&sshPort=22#/  
AWS Services Search [Alt+S]  
N. Virginia Insia Farhan  
>  
>  
> ~C  
[ec2-user@ip-172-31-53-171 backend]$ curl -fsSL https://rpm.nodesource.com/setup_16.x | sudo bash -  
  
-----  
DEPRECATION WARNING  
  
_Node.js 16.x is no longer actively supported!  
  
You will not receive security or critical stability updates for this version.  
  
You should migrate to a supported version of Node.js as soon as possible.  
Use the installation script that corresponds to the version of Node.js you  
wish to install. e.g.  
  
* https://rpm.nodesource.com/setup_16.x - Node.js 16 "Gallium" (deprecated)  
* https://rpm.nodesource.com/setup_18.x - Node.js 18 "Hydrogen" (Maintenance)  
* https://rpm.nodesource.com/setup_19.x - Node.js 19 "Bineteen" (deprecated)  
* https://rpm.nodesource.com/setup_20.x - Node.js 20 LTS "Iron" (recommended)  
* https://rpm.nodesource.com/setup_21.x - Node.js 21 "Iron" (current)  
  
Please see https://github.com/nodejs/Release for details about which  
i-0cb3a358bbc8252cd (skillsmatch)  
PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&sshPort=22#/  
AWS Services Search [Alt+S]  
N. Virginia Insia Farhan  
Metadata cache created.  
2024-05-26 14:04:23 - Repository is configured and updated. Run 'dnf install nodejs -y' to complete the installation.  
[ec2-user@ip-172-31-53-171 backend]$ sudo yum install -y nodejs  
Last metadata expiration check: 0:00:25 ago on Sun May 26 14:04:23 2024.  
Dependencies resolved.  


| Package               | Architecture | Version               | Repository        | Size |
|-----------------------|--------------|-----------------------|-------------------|------|
| Installing:<br>nodejs | x86_64       | 2:16.20.2-1nodesource | nodesource-nodejs | 32 M |

  
Transaction Summary  
-----  
Install 1 Package  
  
Total download size: 32 M  
Installed size: 89 M  
Downloading Packages:  
nodejs-16.20.2-1nodesource.x86_64.rpm 44 MB/s | 32 MB 00:00  
-----  
Total  
Node.js Packages for Linux RPM based distros - x86_64  
Importing GPG key 0x9B18E0B4:  
Userid : "NSolid <nsolid-gpg@nodesource.com>"  
Fingerprint: 6F71 F525 2828 41EK DAF8 51B4 2F59 9B1B E0B4  
From : https://rpm.nodesource.com/gpgkey/NODESOURCE-GPG-SIGNING-KEY-EL  
Key imported successfully  
i-0cb3a358bbc8252cd (skillsmatch)  
PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&sshPort=22#

nodejs-2:16.20.2-1nodesource.x86_64

```
Complete!
[ec2-user@ip-172-31-53-171 backend]$ node --version
node: bad option: --version
[ec2-user@ip-172-31-53-171 backend]$ node -v
v16.20.2
[ec2-user@ip-172-31-53-171 backend]$ npm -v
8.19.4
[ec2-user@ip-172-31-53-171 backend]$ npm install
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /home/ec2-user/skillsmatch/backend/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/home/ec2-user/skillsmatch/backend/package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR!   /home/ec2-user/.npm/_logs/2024-05-26T14_05_40_410Z-debug-0.log
[ec2-user@ip-172-31-53-171 backend]$ ls
package-lock.json
[ec2-user@ip-172-31-53-171 backend]$ ls
package-lock.json
[ec2-user@ip-172-31-53-171 backend]$ cd ..
[ec2-user@ip-172-31-53-171 skillsmatch]$ cd ..
[ec2-user@ip-172-31-53-171 ~]$ sudo rm -rf skillsmatch
```

i-0cb3a358bbc8252cd (skillsmatch)

PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

13. Running backend code.

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0cb3a358bbc8252cd&osUser=ec2-user&sshPort=22#

aws Services Search [Alt+S]

README.md client_secret_924825865240-3qavb2lveu8mmj15612f4aphucqghurh2.apps.googleusercontent.com.json controllers flask middleware migrations models node_modules package-lock.json package.json routes server.js token.json

```
[ec2-user@ip-172-31-53-171 backend]$ mkdir config
[ec2-user@ip-172-31-53-171 backend]$ nano config.js
[ec2-user@ip-172-31-53-171 backend]$ nano config.json
[ec2-user@ip-172-31-53-171 backend]$ ls
README.md client_secret_924825865240-3qavb2lveu8mmj15612f4aphucqghurh2.apps.googleusercontent.com.json config.js controllers flask migrations models node_modules package-lock.json package.json routes server.js token.json
[ec2-user@ip-172-31-53-171 backend]$ sudo rm -rf config.js
[ec2-user@ip-172-31-53-171 backend]$ sudo rm -rf config.json
[ec2-user@ip-172-31-53-171 backend]$ cd config
[ec2-user@ip-172-31-53-171 config]$ nano config.json
[ec2-user@ip-172-31-53-171 config]$ nano config.js
[ec2-user@ip-172-31-53-171 config]$ cd ..
[ec2-user@ip-172-31-53-171 backend]$ npm start

> start
> nodemon server.js

[nodemon] 3.1.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Server is running on port 5000
```

i-0cb3a358bbc8252cd (skillsmatch)

PublicIPs: 52.87.220.206 PrivateIPs: 172.31.53.171

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences