Lab 8

Lab Title: AWS: Account Setup, IAM, VPC Inventory, EC2, Docker & Gitea

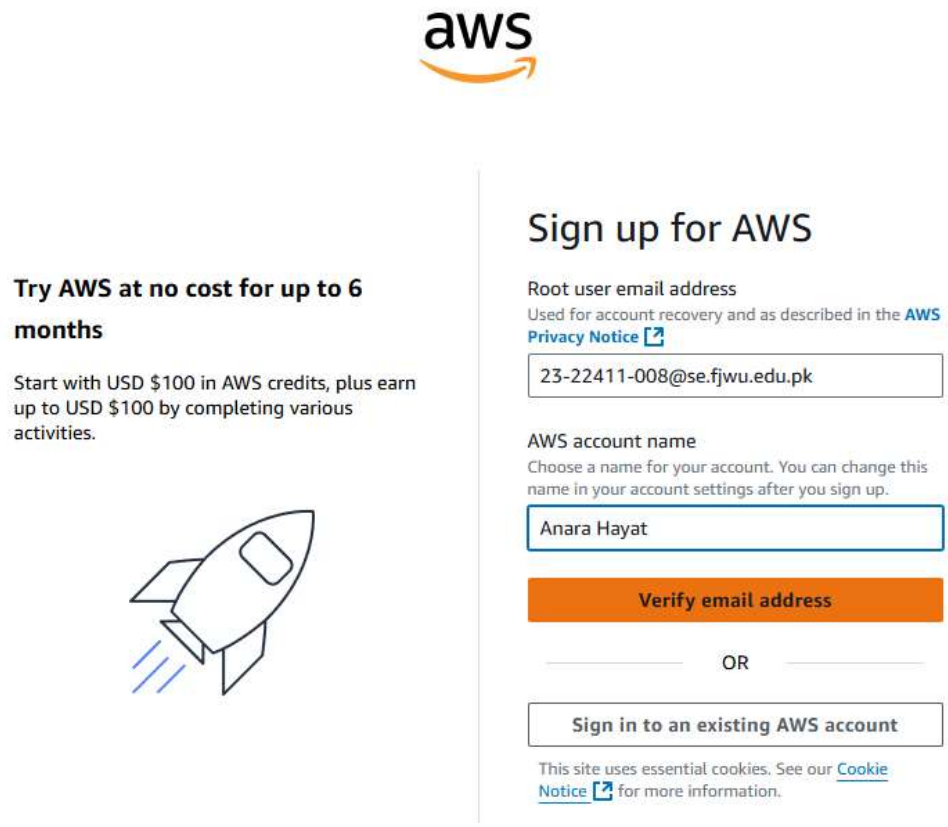
Submitted to: Engr. Muhammad Shoaib

Submitted by: Anara Hayat

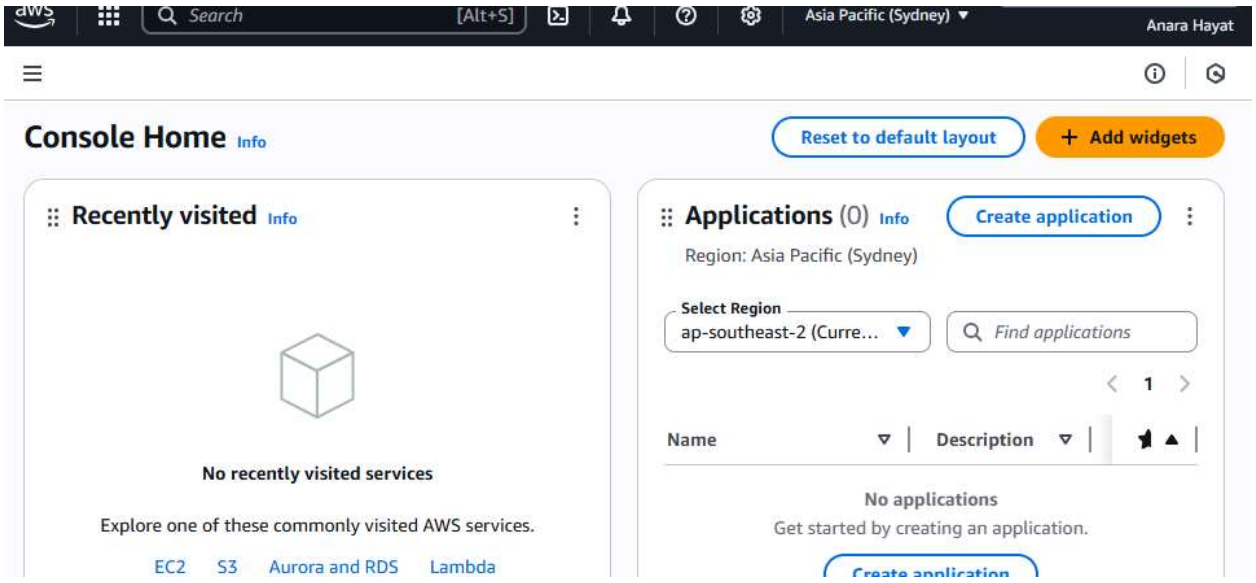
Reg#No: 2023-BSE-008

Task 1 — Create an AWS account and enable UAE (me-central-1)

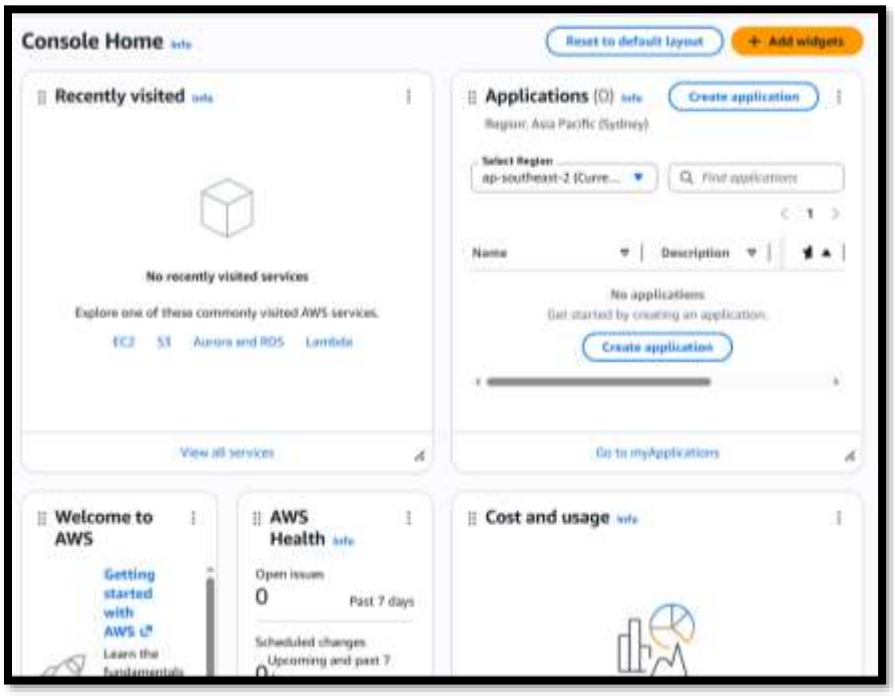
1. **Open your browser and go to: [AWS Signup]**



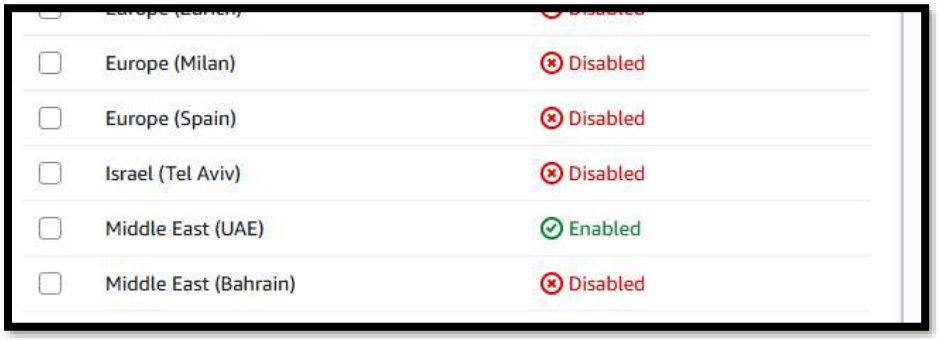
2. **Complete registration (Account type: Personal, Plan: AWS Paid Plan), fill contact, billing (credit card) and phone details, complete verification. After successful registration capture:**



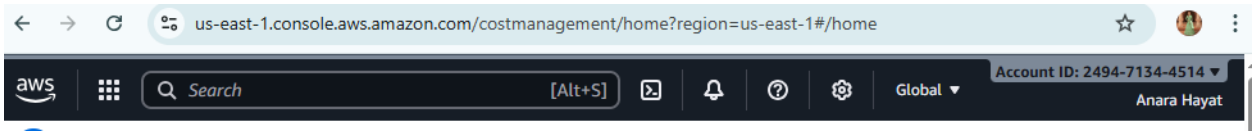
3. **Sign in as the root user (root email). Immediately capture:**



4. **From the Console, open the region selector and enable UAE (me-central-1), then switch to me-central-1. Capture the change**

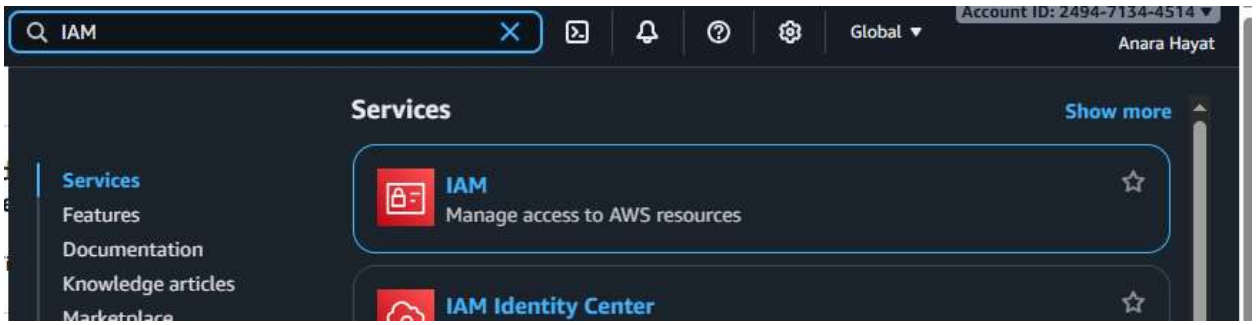


5. **Task 1 summary screenshot (combine evidence):**

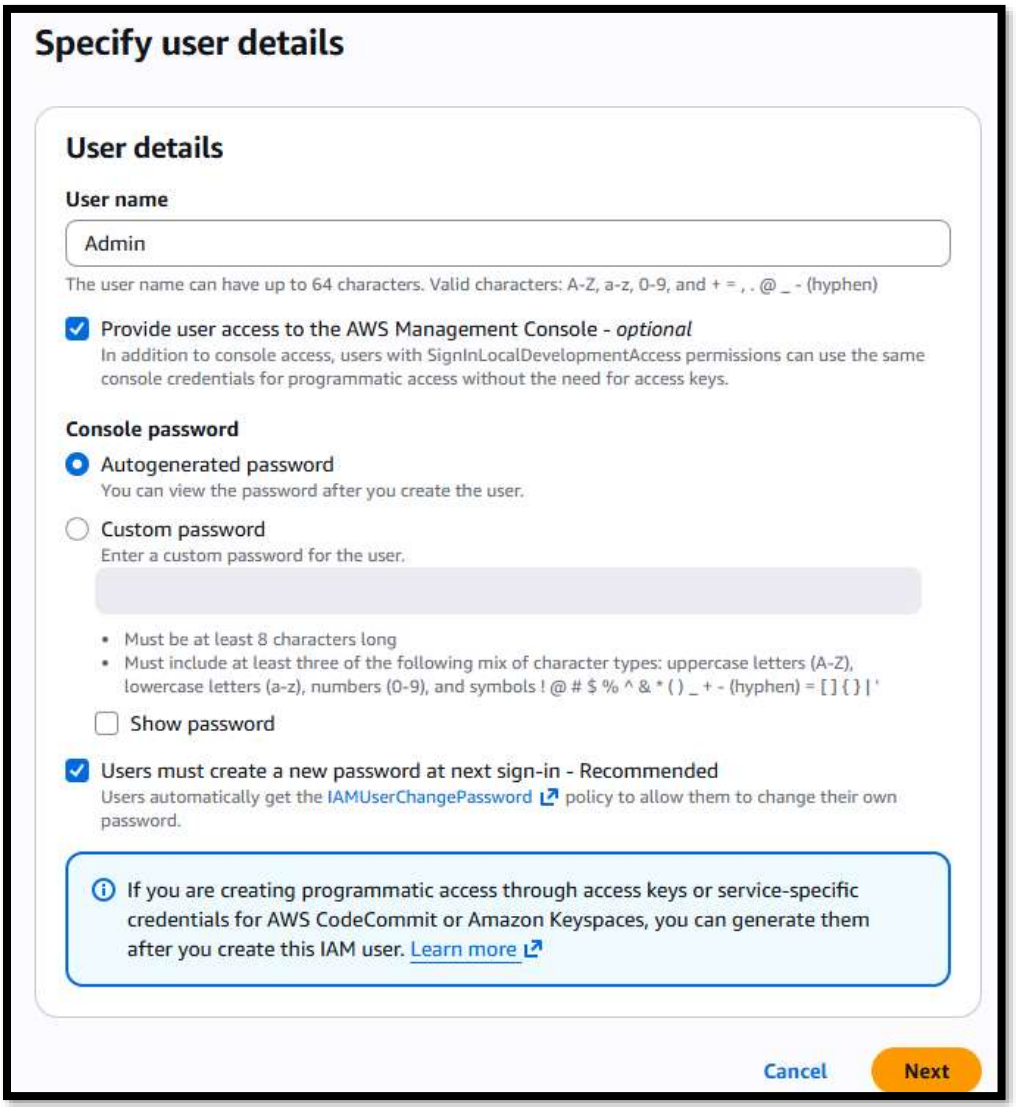


Task 2 — Create IAM Admin and Lab8User with console access

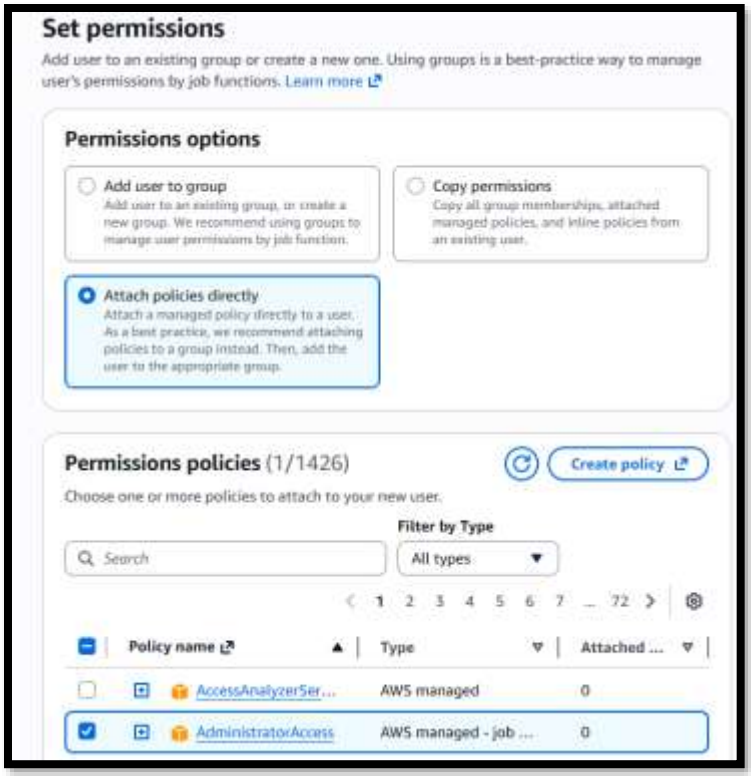
1. **Open IAM via Console search (Alt+S → "IAM").**

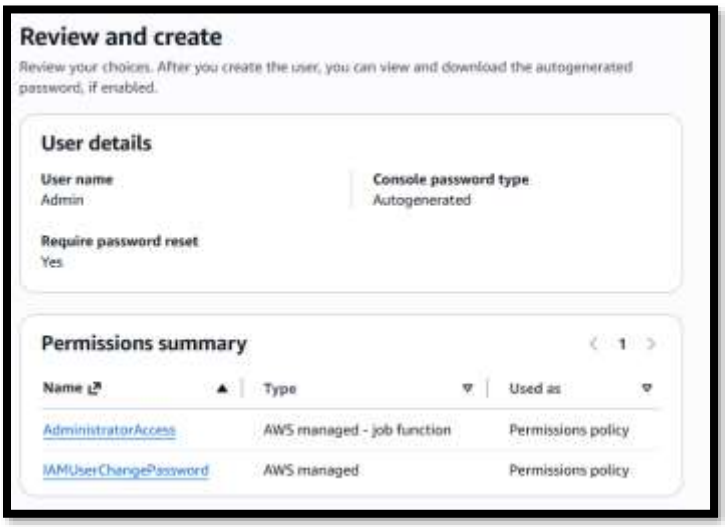


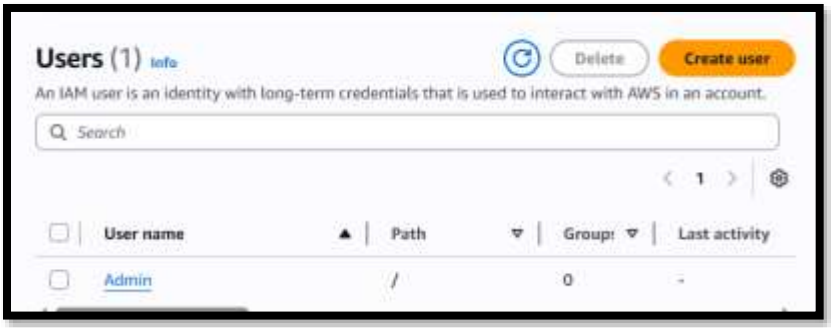
2. Create the Admin user: IAM → Users → Create user. Fill:



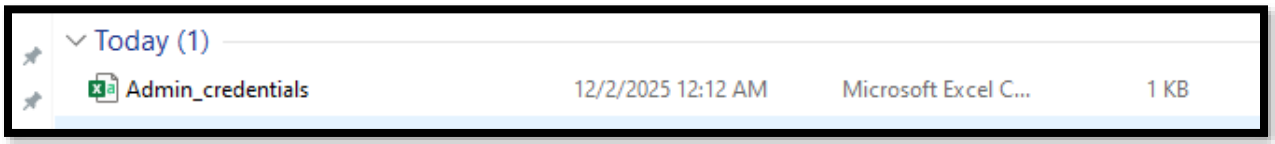
Attach policies directly → AdministratorAccess


Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (1/1426)
Create policy
Choose one or more policies to attach to your new user.
Filter by Type
Search
All types
1 2 3 4 5 6 7 ... 72
Policy name
Type
Attached ...
AccessAnalyzerSer...
AWS managed
0
AdministratorAccess
AWS managed - job ...
0


Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
Admin
Console password type
Autogenerated
Require password reset
Yes
Permissions summary
1
Name
Type
Used as
AdministratorAccess
AWS managed - job function
Permissions policy
IAMUserChangePassword
AWS managed
Permissions policy

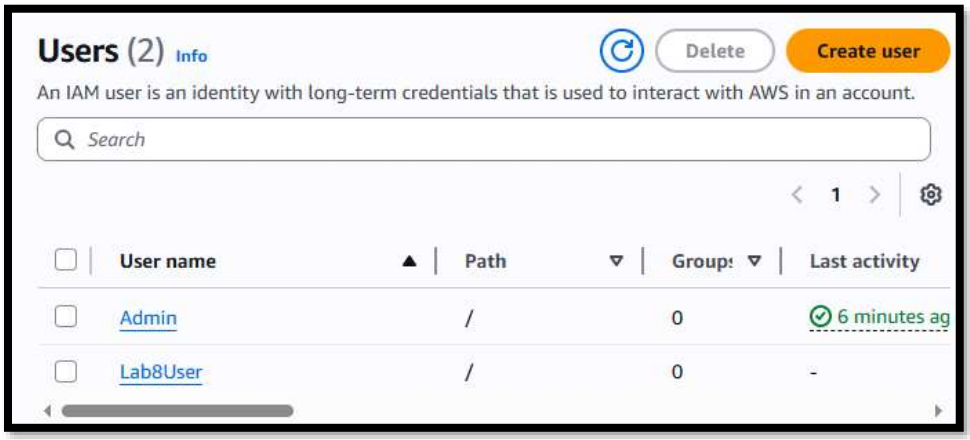

Users (1) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Group
Last activity
Admin
/
0
-

3. **Download the Admin .csv and show its presence on your Windows host**

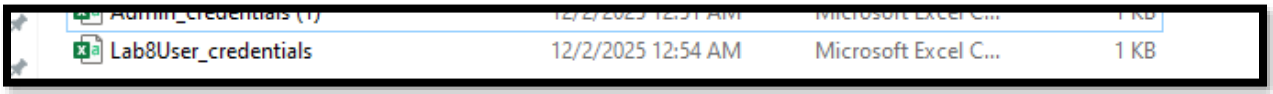


4. **Sign out of root, then sign in using the Admin account (use the signin URL from the .csv). Capture after successful Admin login:A**

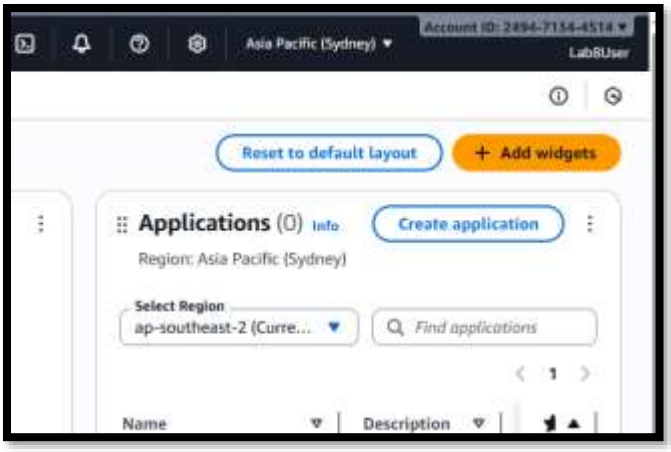
5. **While logged in as Admin, create Lab8User**



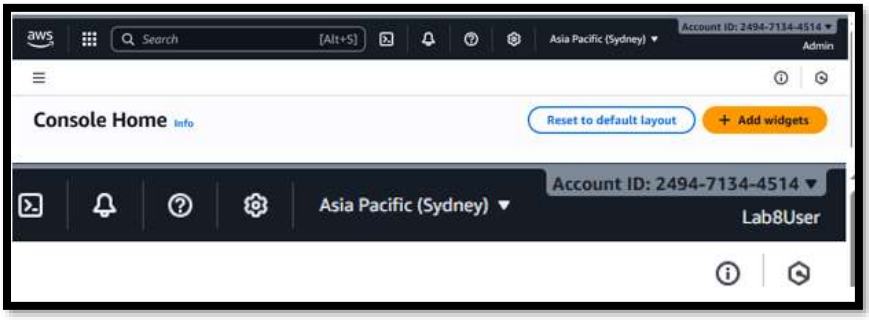
6. **Download/save the Lab8User CSV on your Windows host**



7. **Logout Admin and login as Lab8User (use the Lab8User signin URL and credentials). Capture after login:**

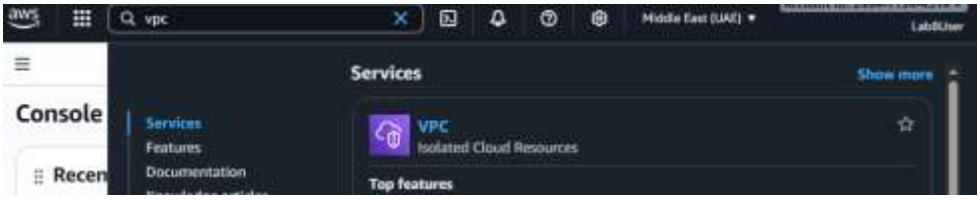


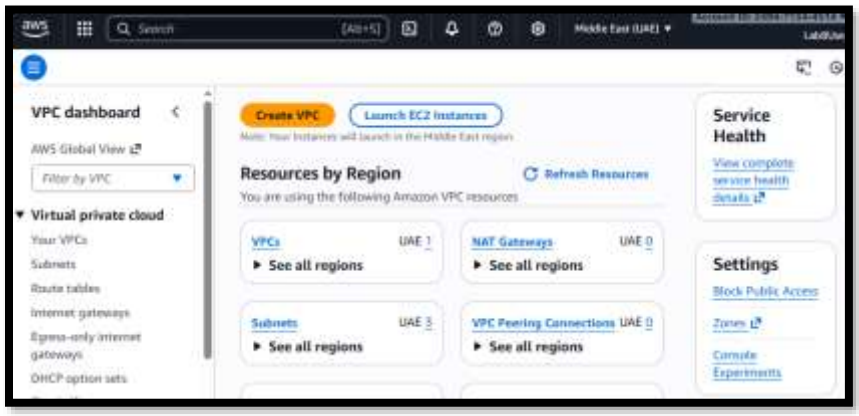
8. **Task 2 summary (combine evidence):**



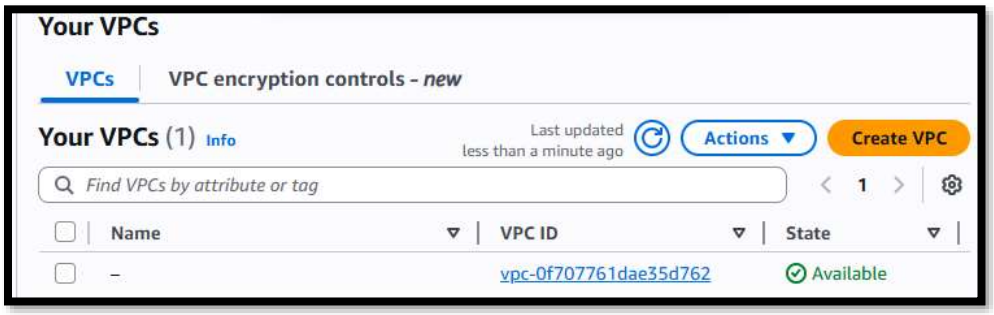
Task 3 — Inspect VPC resources (in UAE me-central-1)

1. **Open VPC console (Alt+S → "VPC") while region is me-central-1.**

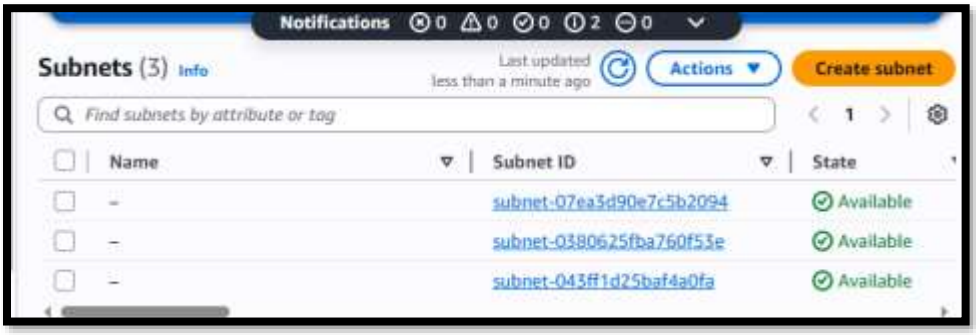




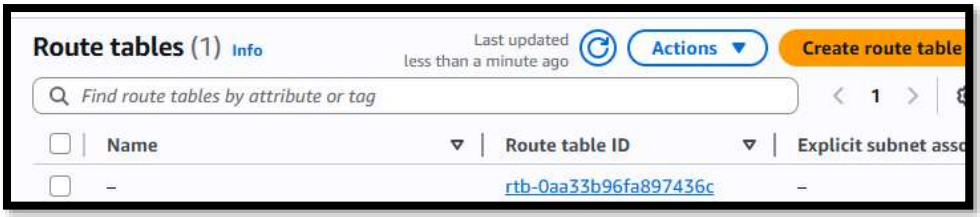
2. **View VPCs list. Capture:**



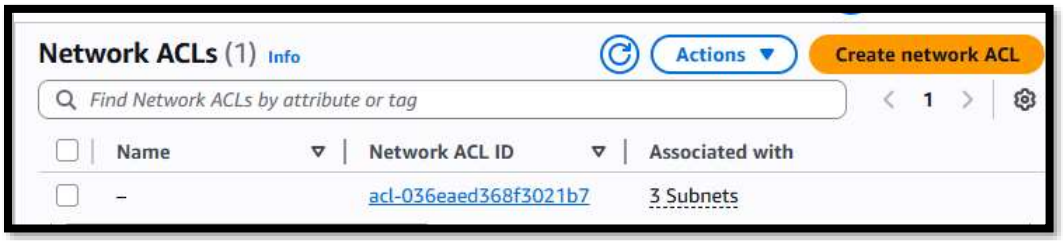
3. **View Subnets list.**



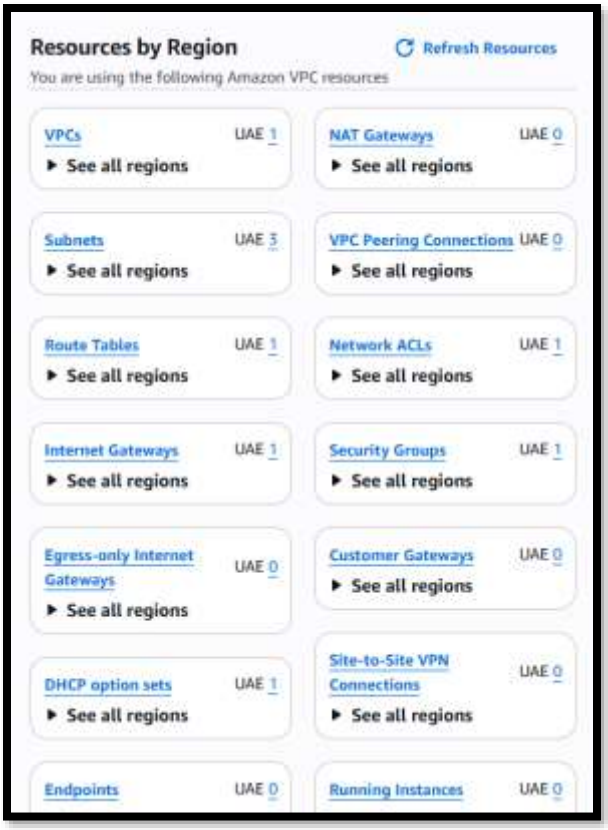
4. **View Route Tables list. Capture:**



5. **View Network ACLs list**

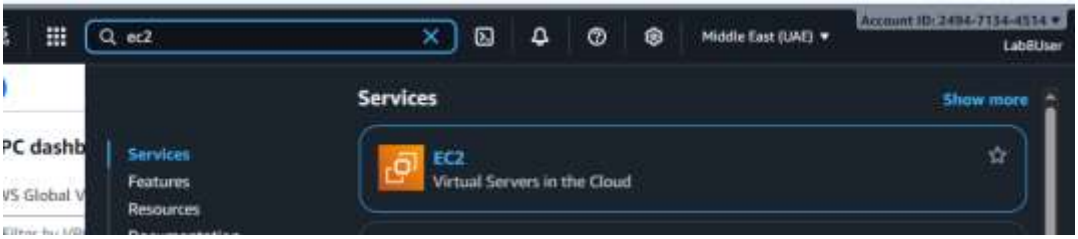


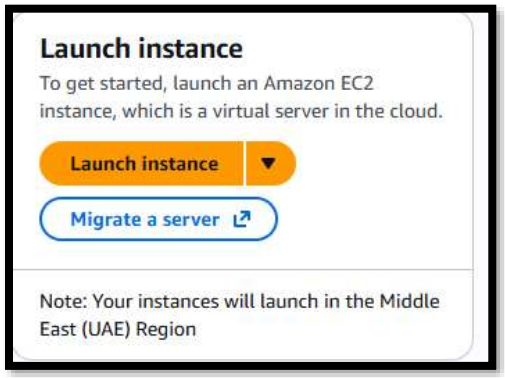
6. **Task 3 summary (combine evidence)**



Task 4 — Launch EC2, SSH, install Docker & Docker Compose, deploy Gitea

1. **Open EC2 Console (Alt+S → "EC2") (me-central-1).**

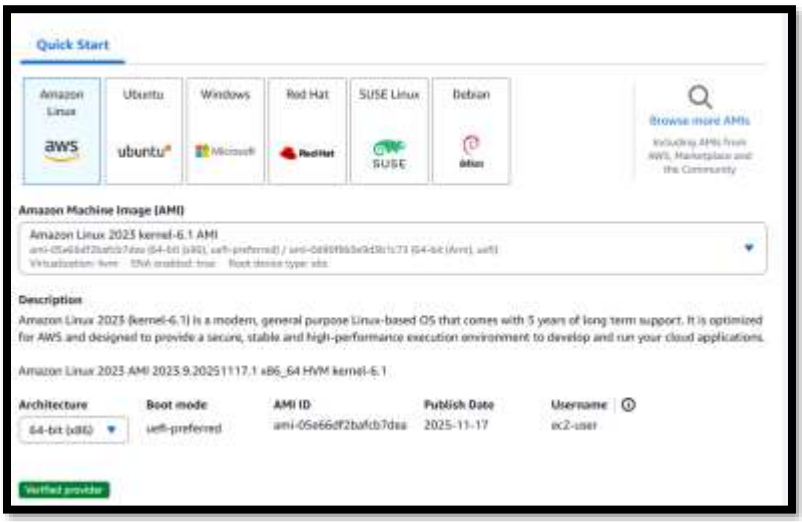




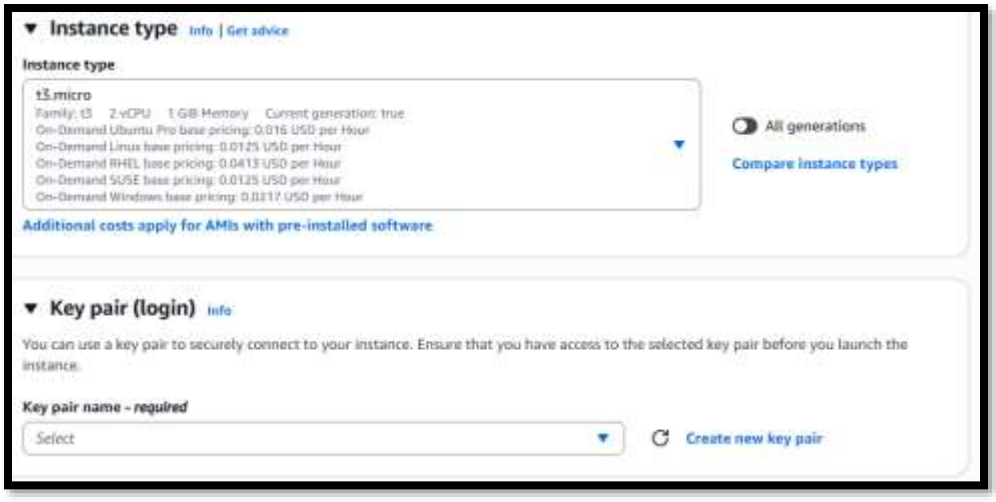
2. **Instance Launch configuration (during review before launching). Configure**

Name: Lab8Machine

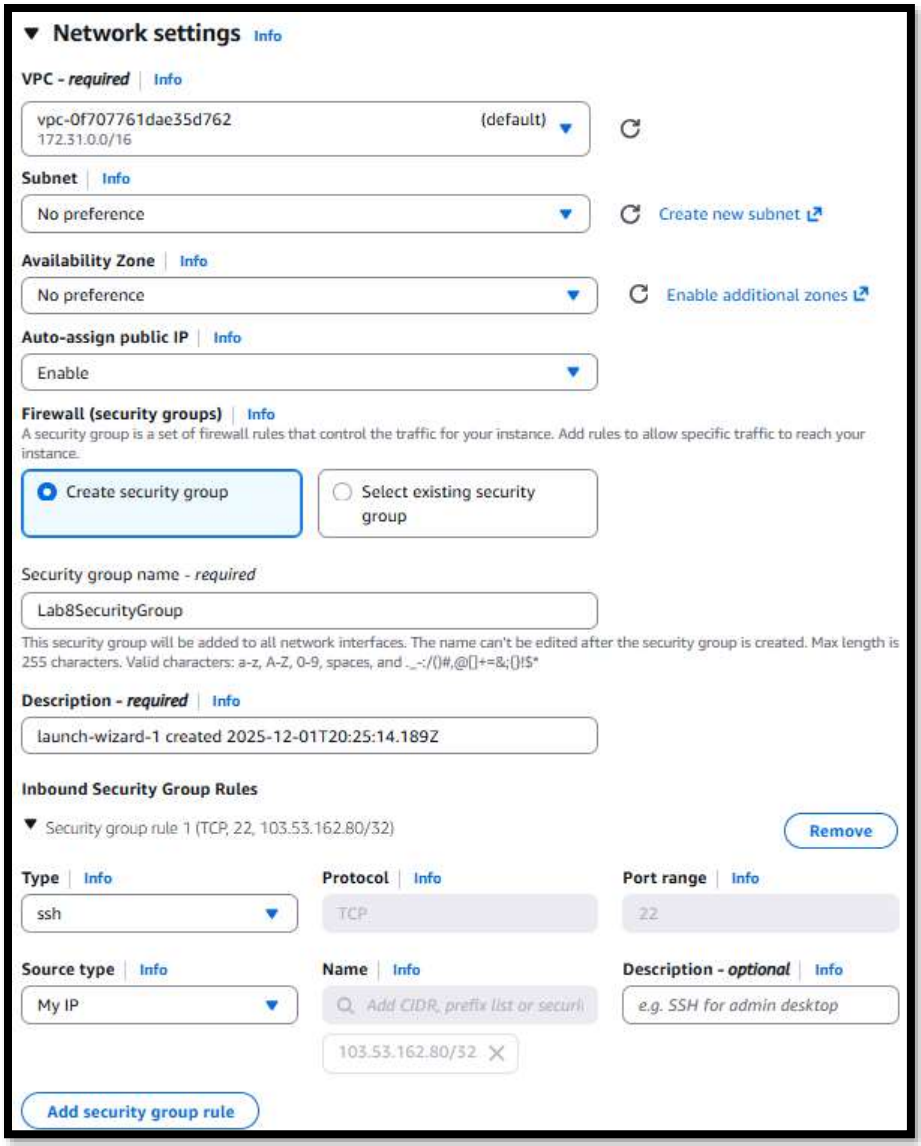
AMI: Amazon Linux 2



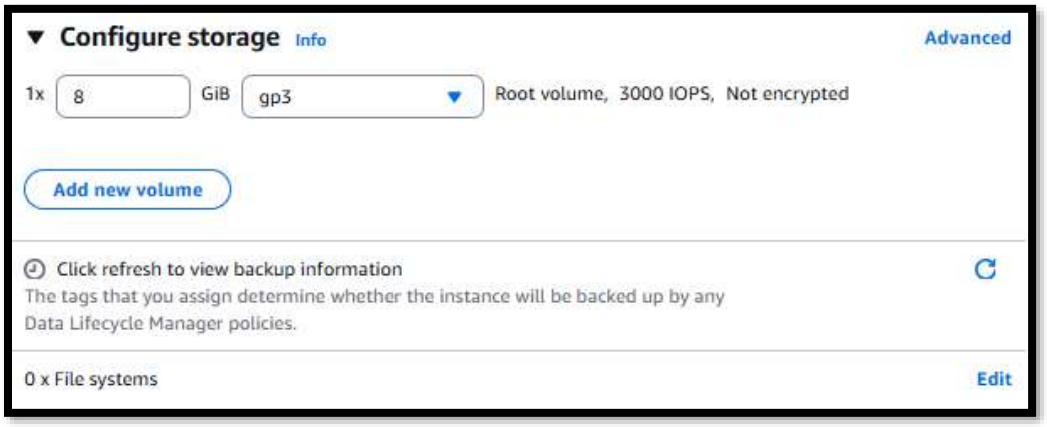
Instance type: t3.micro



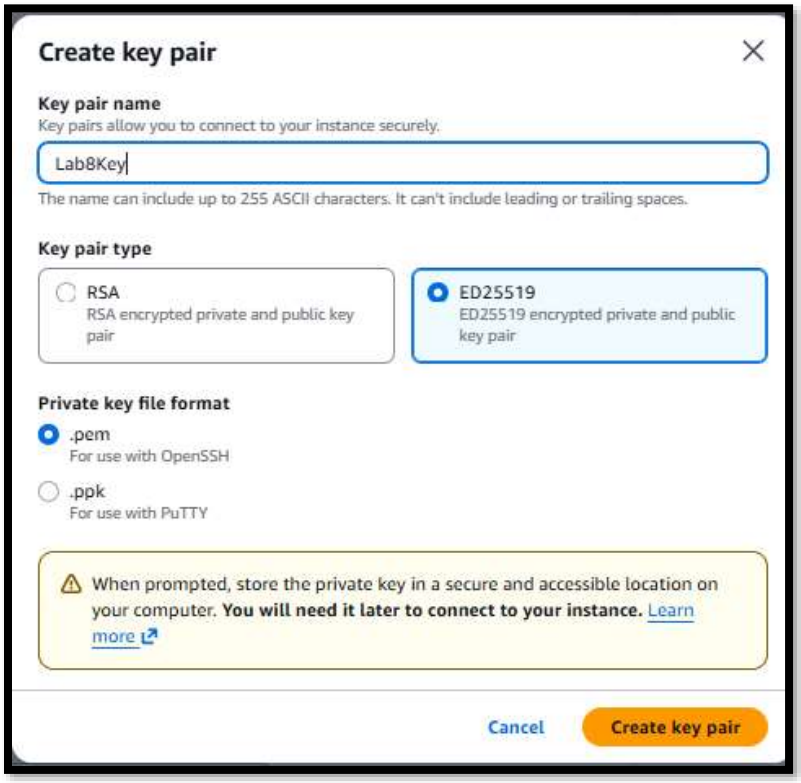
Security group: Create Lab8SecurityGroup with SSH from My IP



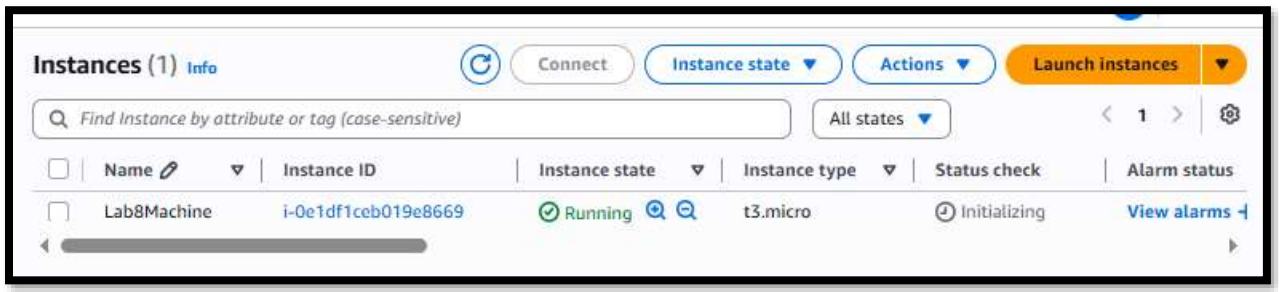
Storage: default

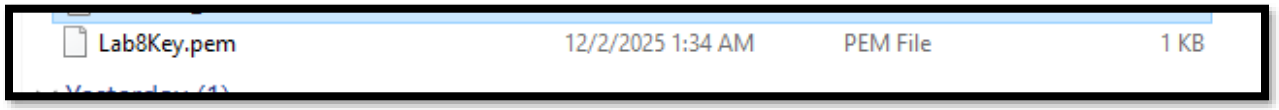


Key pair: Create Lab8Key (ED25519, .pem) and download the .pem file to your Windows host



Capture the final review page and key download





3. **After launch, EC2 Instances list showing Lab8Machine in "running" state and public IPv4 visible.**

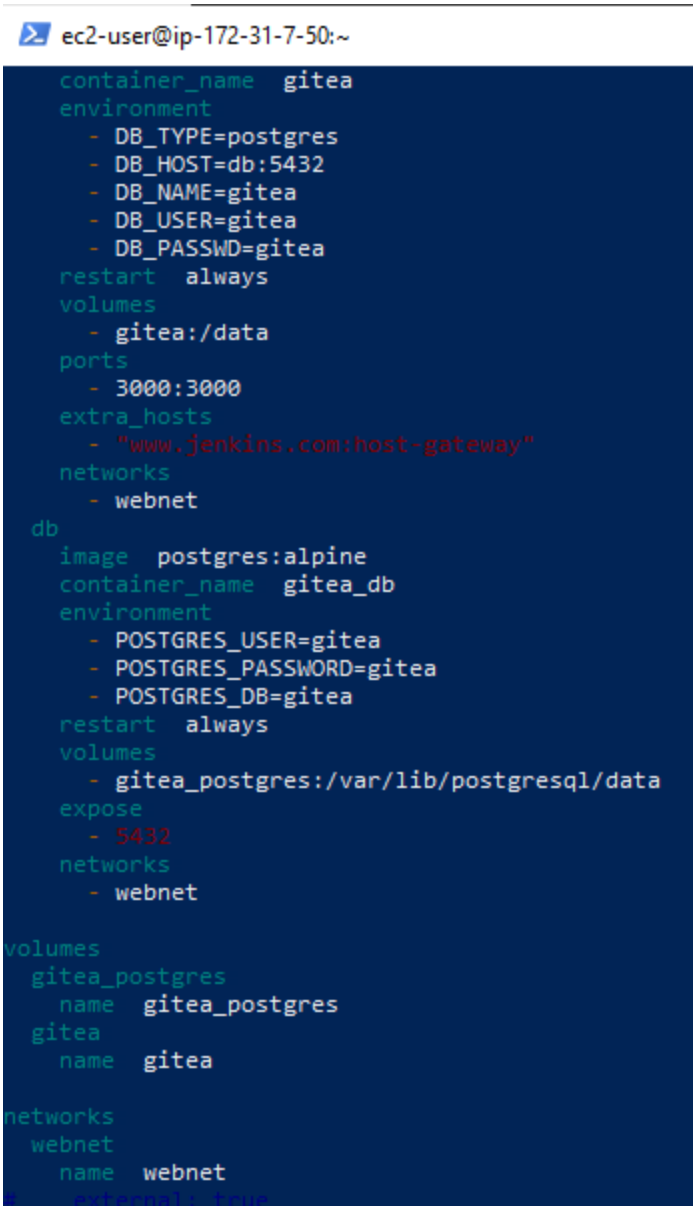
4. **On Windows host, run SSH using the downloaded .pem**

```
PS C:\Users\Anara Hayat> ssh -i ~/Downloads/Lab8Key.pem ec2-user@3.29.237.162
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-7-50 ~]$
[ec2-user@ip-172-31-7-50 ~]$
```

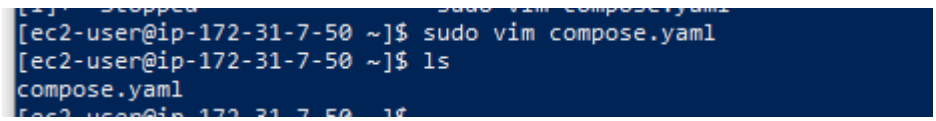
5. **Run the install commands on the EC2 shell:**

```
[ec2-user@ip-172-31-7-50 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                    278 kB/s |  29 kB     00:00
Last metadata expiration check: 0:00:01 ago on Tue Dec  2 09:03:39 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-7-50 ~]$ sudo yum install -y docker
Last metadata expiration check: 0:00:18 ago on Tue Dec  2 09:03:39 2025.
Dependencies resolved.
================================================================================
 Package                 Architecture    Version                 Repository    Size
================================================================================
Installing:
 docker                  x86_64          25.0.13-1.amzn2023.0.2  amazonlinux   46 M
Installing dependencies:
 container-selinux       noarch          4:2.242.0-1.amzn2023    amazonlinux   58 k
 containerd              x86_64          2.1.4-1.amzn2023.0.2    amazonlinux   23 M

Installed:
  container-selinux-4:2.242.0-1.amzn2023.noarch
  containerd-2.1.4-1.amzn2023.0.2.x86_64
  docker-25.0.13-1.amzn2023.0.2.x86_64
  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.3.3-2.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-7-50 ~]$ sudo mkdir -p /usr/local/docker/cli-plugins
[ec2-user@ip-172-31-7-50 ~]$ sudo curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86_64 -o /usr/local/lib/docker/cli-plugins/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
Warning: Failed to open the file /usr/local/lib/docker/cli-plugins/docker-compose: No such file or directory
  0 73.0M    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
curl: (23) client returned ERROR on write of 16375 bytes
[ec2-user@ip-172-31-7-50 ~]$ sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
chmod: cannot access '/usr/local/lib/docker/cli-plugins/docker-compose': No such file or directory
[ec2-user@ip-172-31-7-50 ~]$ sudo systemctl start docker
```

6. **Create/edit compose.yaml on the EC2 instance (sudo vim compose.yaml) and paste content from the repo: Gitea . While pasting, capture the editor content:**



7. **Save and verify file exists:**



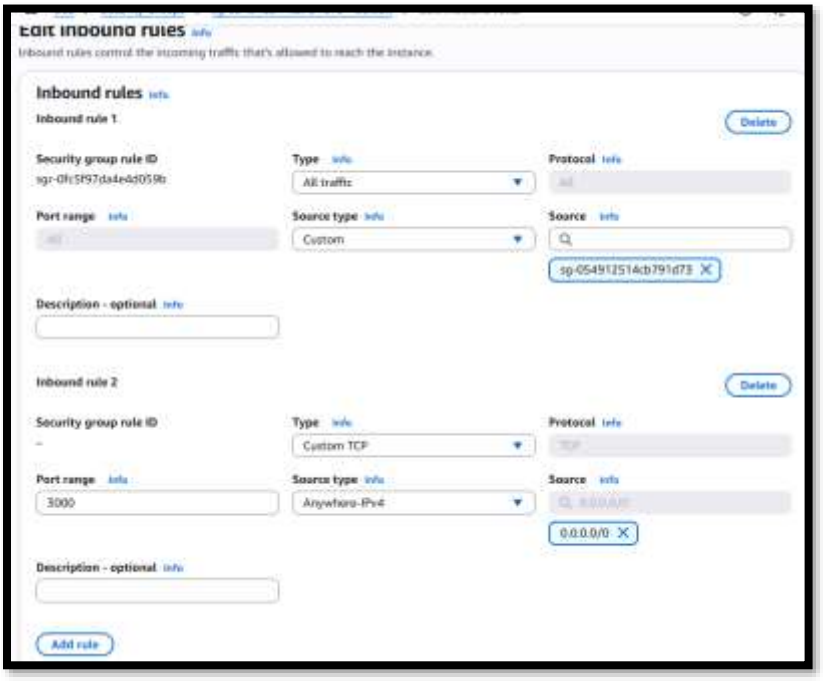
8. **Add ec2-user to docker group, show groups before re-login, exit and reconnect, show groups after reconnect:**

```
[ec2-user@ip-172-31-7-50 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-7-50 ~]$
[ec2-user@ip-172-31-7-50 ~]$
[ec2-user@ip-172-31-7-50 ~]$ sudo usermod -aG docker $USER
[ec2-user@ip-172-31-7-50 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-7-50 ~]$ exit
logout
There are stopped jobs.
Connection to 3.29.237.162 closed.
PS C:\Users\Anara Hayat> ssh -i ~/Downloads/Lab8Key.pem ec2-user@3.29.237.162
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Dec  2 09:01:48 2025 from 103.53.162.80
[ec2-user@ip-172-31-7-50 ~]$ groups
ec2-user adm wheel systemd-journal docker
```

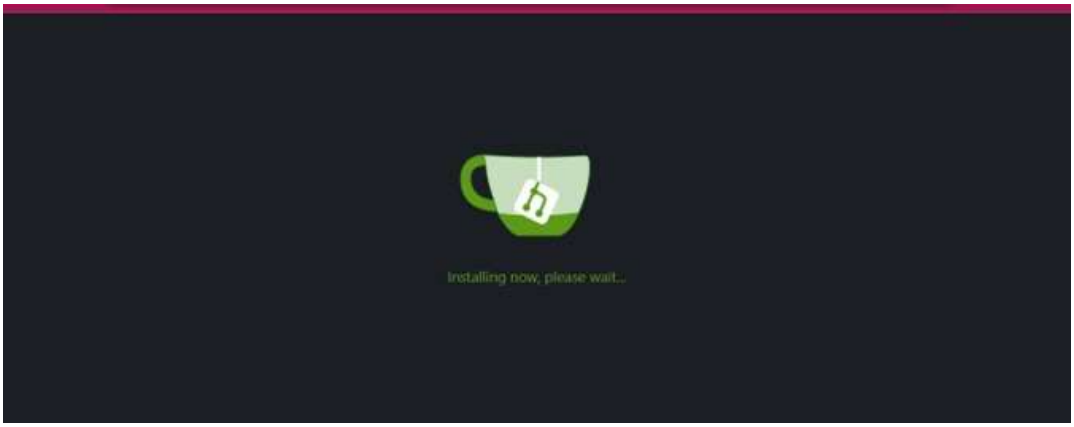
9. **Run docker compose up -d from the directory with compose.yaml:**

```
[ec2-user@ip-172-31-7-50 ~]$ sudo docker-compose up -d
[+] Running 2/2
 ⠿ Container gitea     Running
 ⠿ Container gitea_db  Running
[ec2-user@ip-172-31-7-50 ~]$ sudo docker-compose ps
NAME       IMAGE              COMMAND                  SERVICE   CREATED         STATUS         PORTS
gitea      gitea/gitea:latest "/usr/bin/entrypoint…"   gitea     3 minutes ago   Up 3 minutes   22/tcp, 0.0.0.0:3000-
3000/tcp, [::]:3000->3000/tcp
gitea_db   postgres:alpine     "docker-entrypoint.s…"   db        3 minutes ago   Up 3 minutes   5432/tcp
[ec2-user@ip-172-31-7-50 ~]$
```

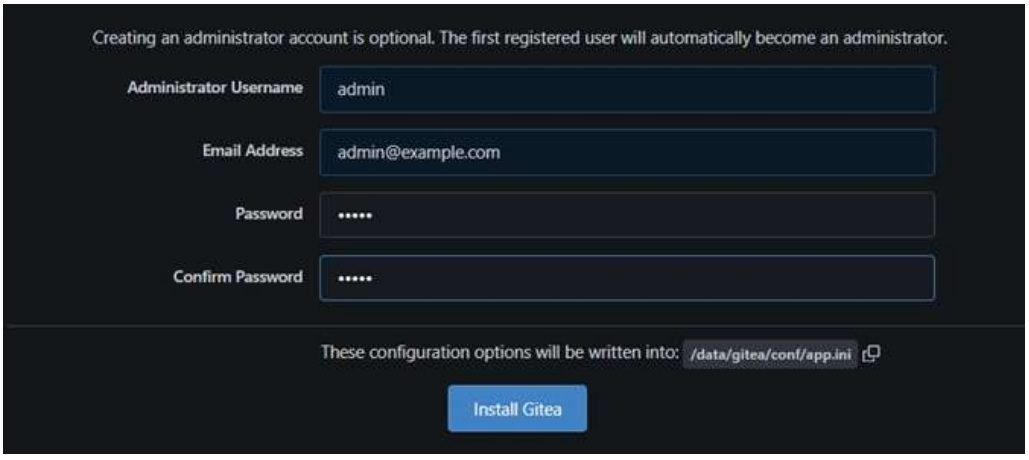
10. **Edit the security group Lab8SecurityGroup inbound rules in the EC2 console: add Custom TCP rule port 3000 source 0.0.0.0/0 and save. Capture the inbound rules after saving:**



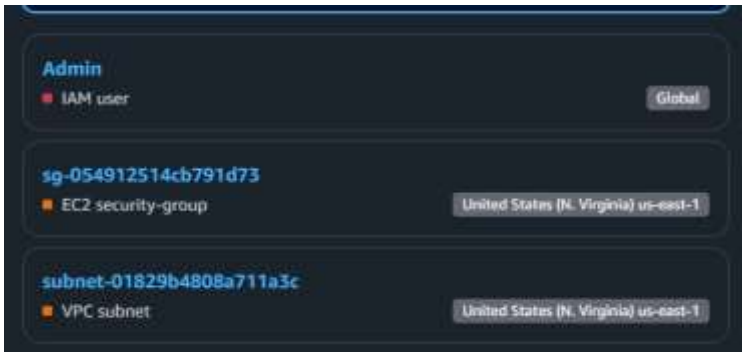
11. **From your Windows browser navigate to: http://Public-IP:3000 — capture the Gitea setup/install page:**



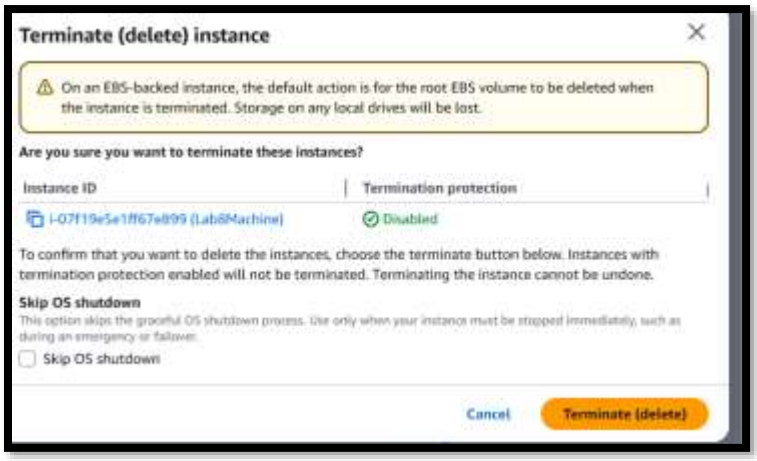
12. **Complete initial Gitea setup (create admin user, create a repo) and capture Gitea showing the created repository:**



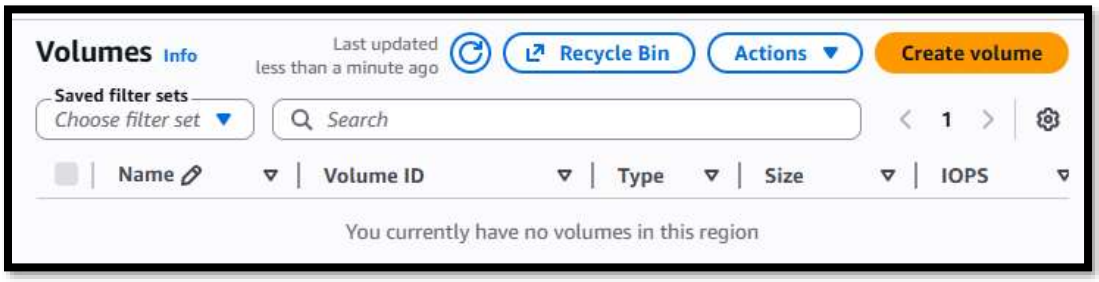
Cleanup — Remove resources to avoid charges



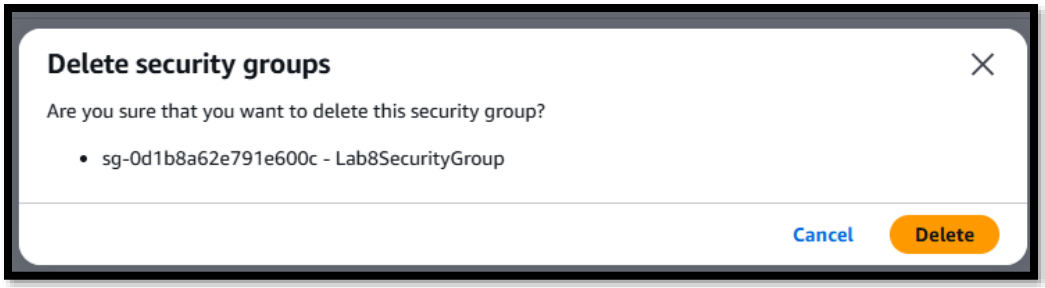
1. **Terminate the EC2 instance Lab8Machine:**

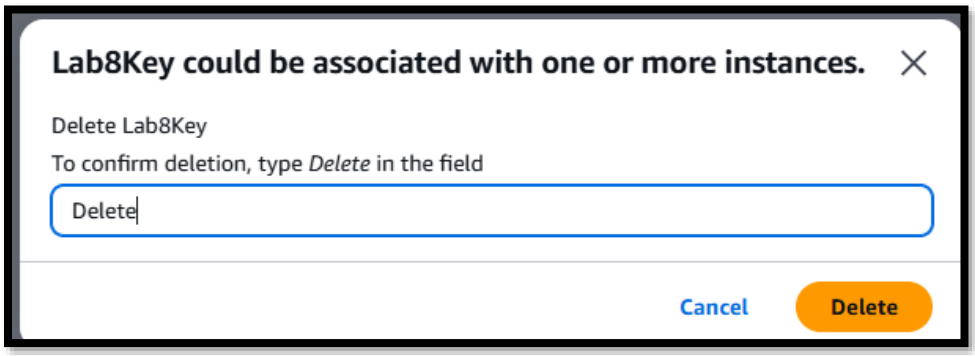


2. **Delete associated EBS volumes and snapshots (if any).**

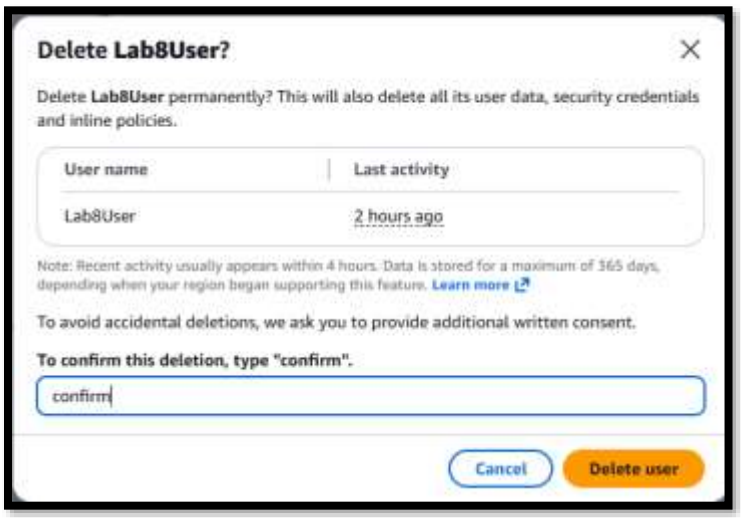


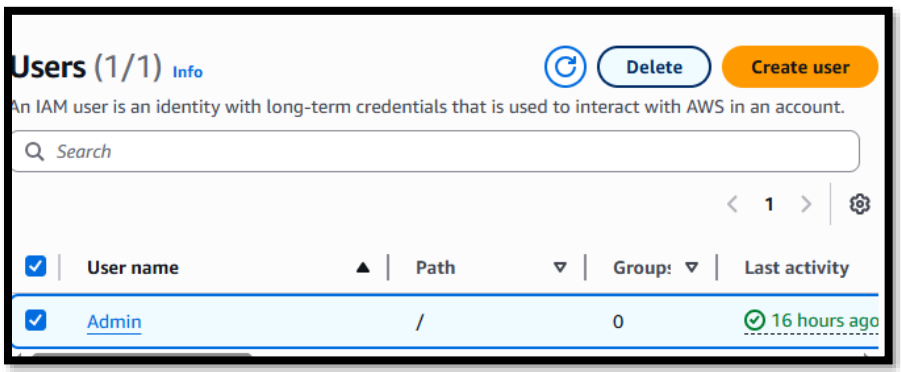
3. **Delete security group Lab8SecurityGroup and key pair Lab8Key from the EC2 console (after instances terminated)**



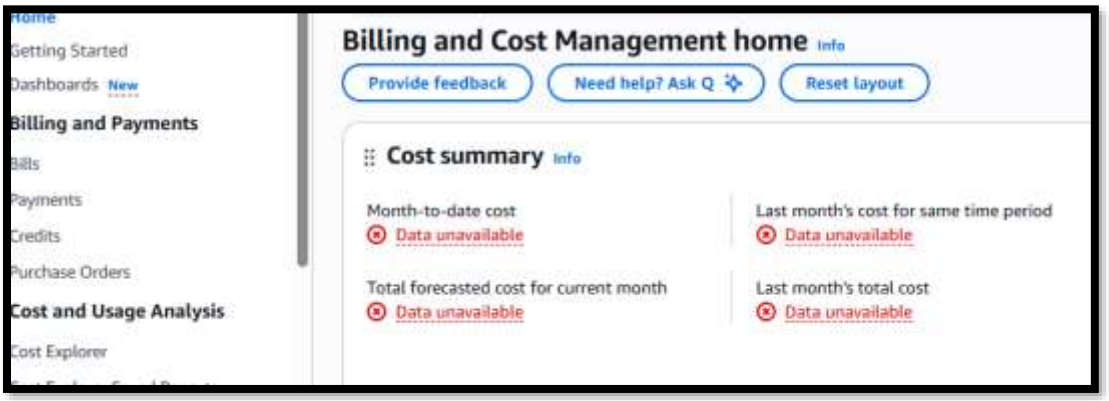


4. **Delete IAM users Lab8User and any access keys**





5. **Final cleanup summary (show billing or resource groups with no active resources if possible).**



****