CLOUD COMPUTING

Lab 08

Name: Abiha Nadeem

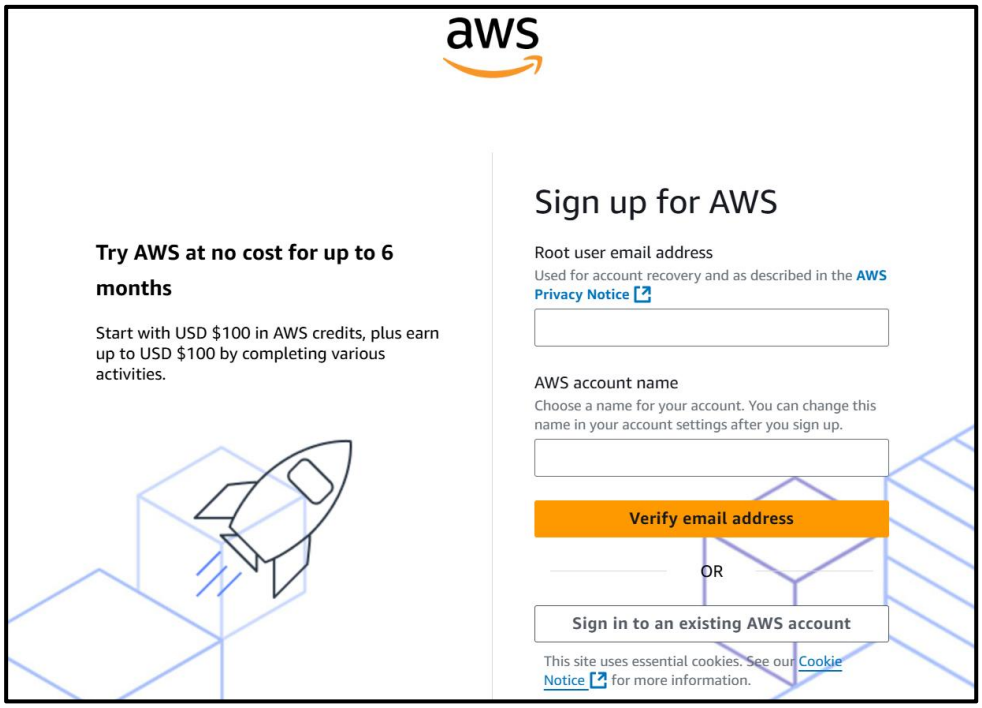
Roll no: 2023-BSE-001

Submitted to: Engr. Muhammad Shoaib

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

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

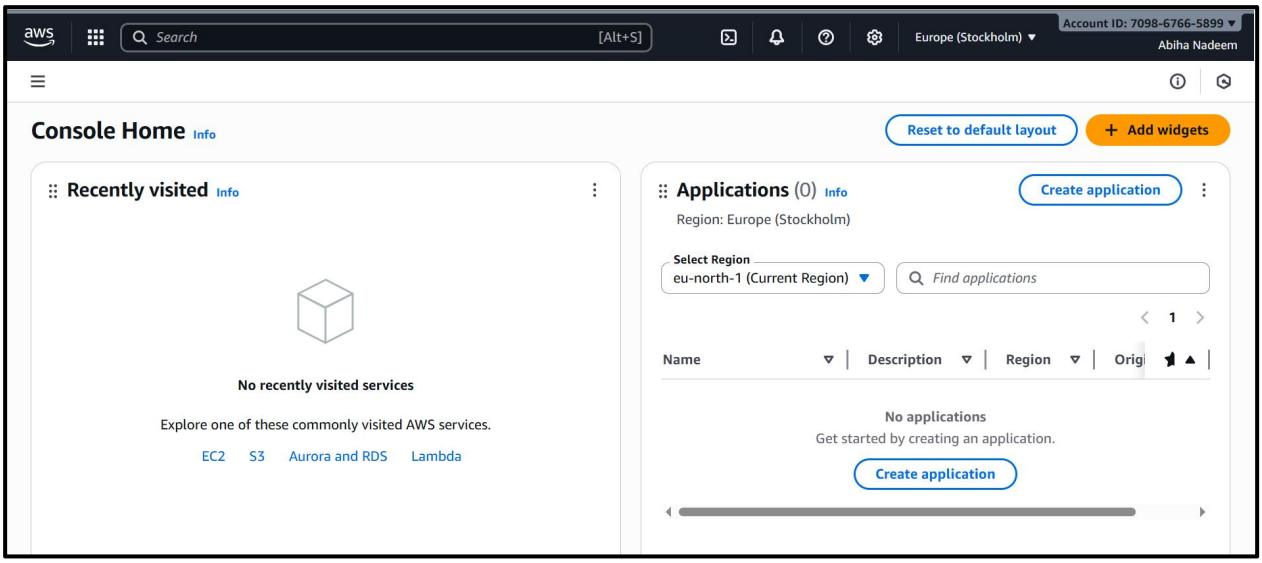
1. **Open your browser and go to: [AWS Signup]**



Save screenshot as: task1_open_signup_page.png — browser showing the signup page.

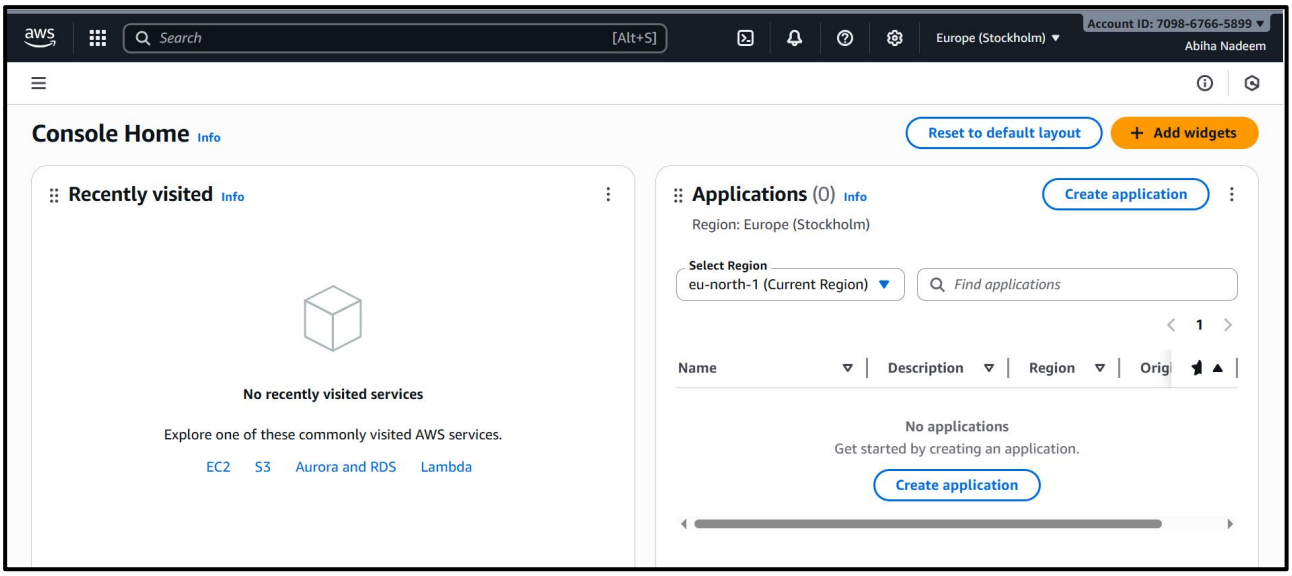
2. **Complete registration (Account type: Personal, Plan: AWS Paid Plan), fill contact, billing (credit card) and phone details, complete verification. After successful registration capture:**

3.



Save screenshot as: task1_signed_up_confirmation.png — registration success/confirmation page or payment confirmation (do NOT include credit card full details).

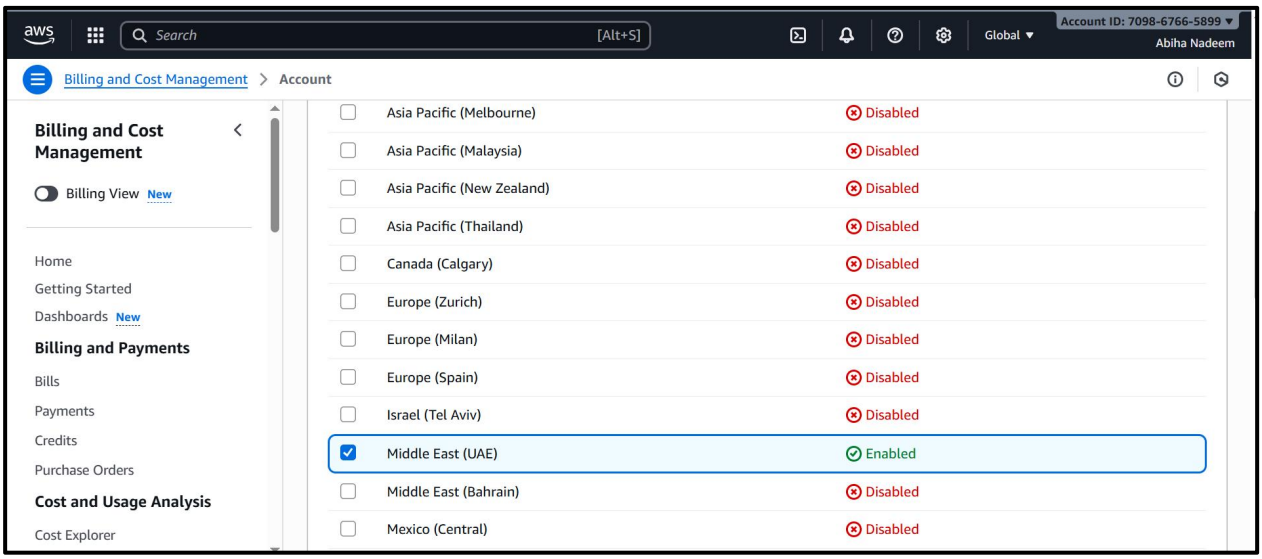
4. **Sign in as the root user (root email). Immediately capture:**



Save screenshot as: task1_root_signed_in.png — AWS Console Home after root login (top bar with root email/account alias visible).

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

Save screenshot as: task1_enable_region_me-central-1.png — region selector showing me-central-1 selected.



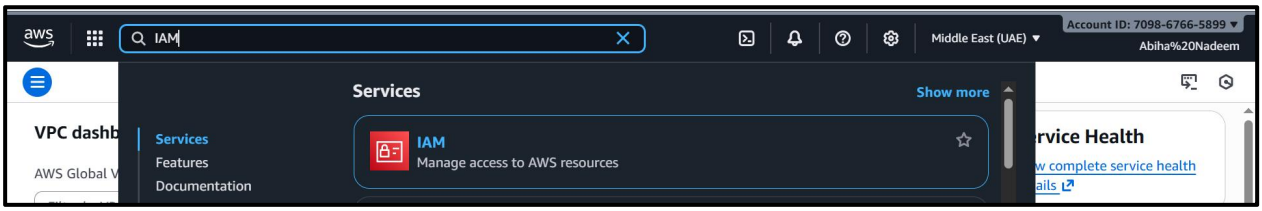
6. **Task 1 summary screenshot (combine evidence):**

Save screenshot as: task1_summary.png — single screenshot showing root console header (root email/account alias) and region set to me-central-1.

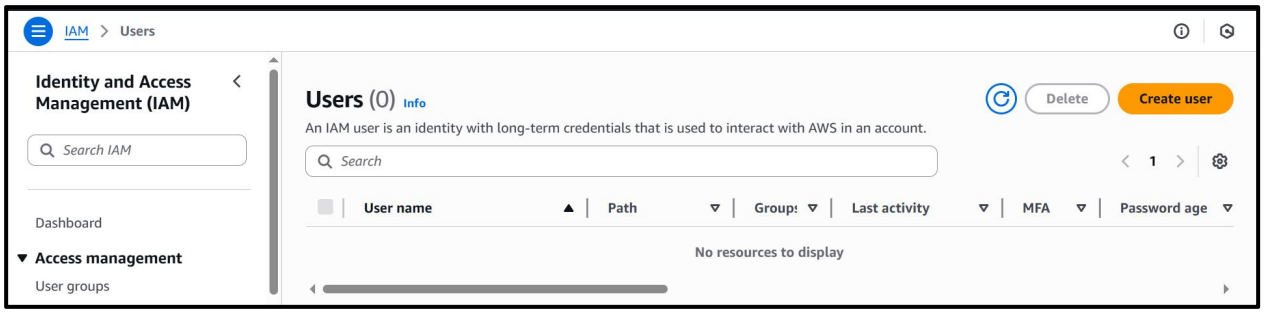
Task 2 — Create IAM Admin and Lab8User with console access

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

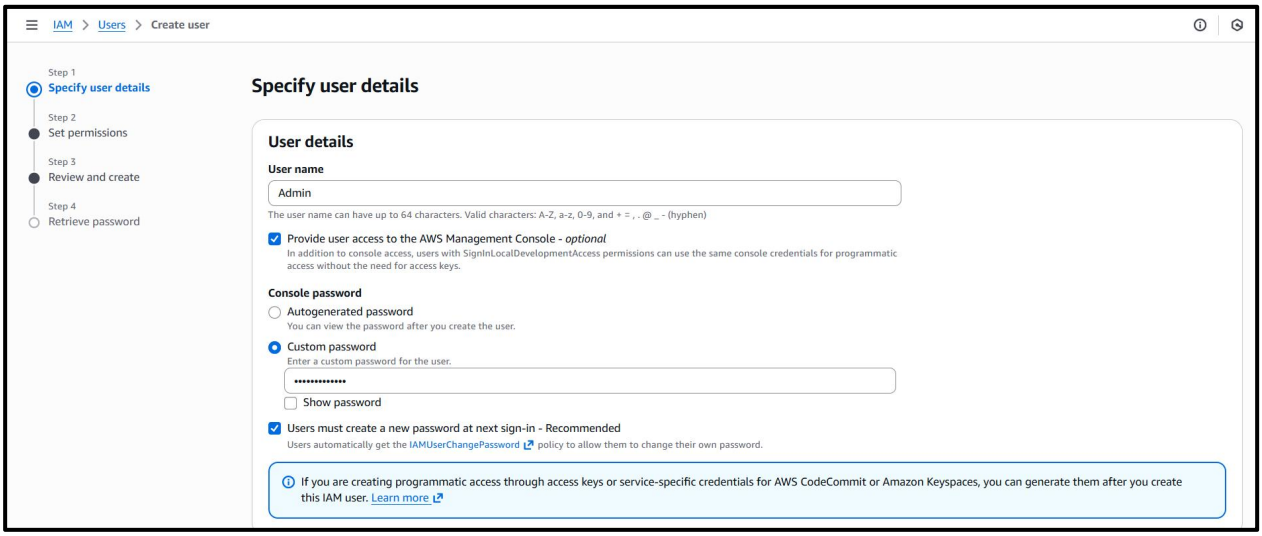
Save screenshot as: task2_open_iam_console.png — IAM console landing page (region me-central-1 visible).



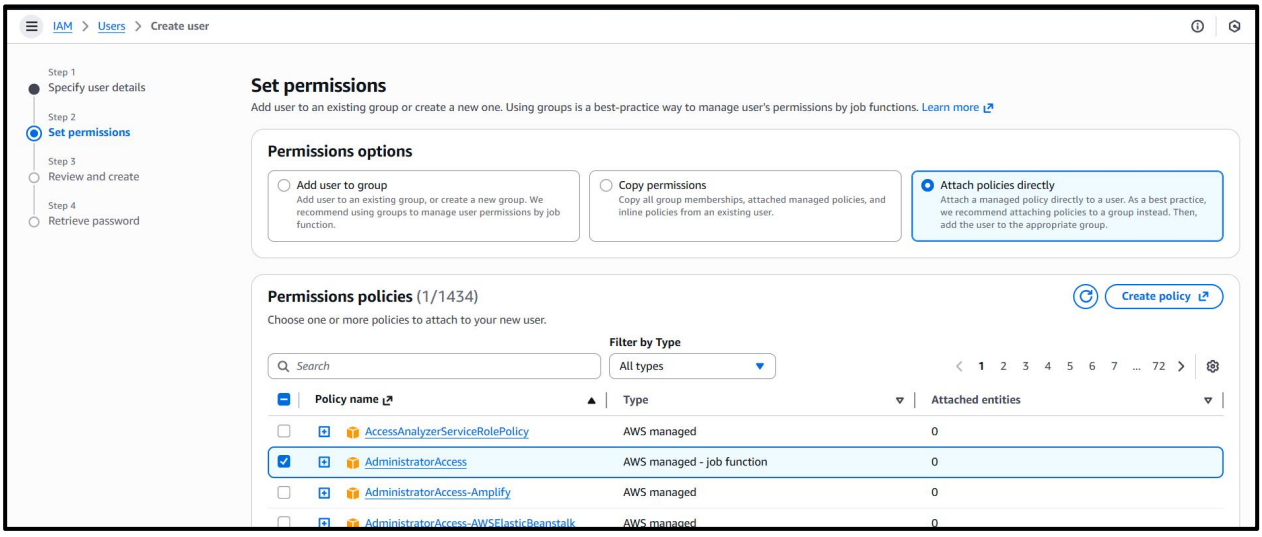
2. **Create the Admin user: IAM → Users → Create user. Fill:**

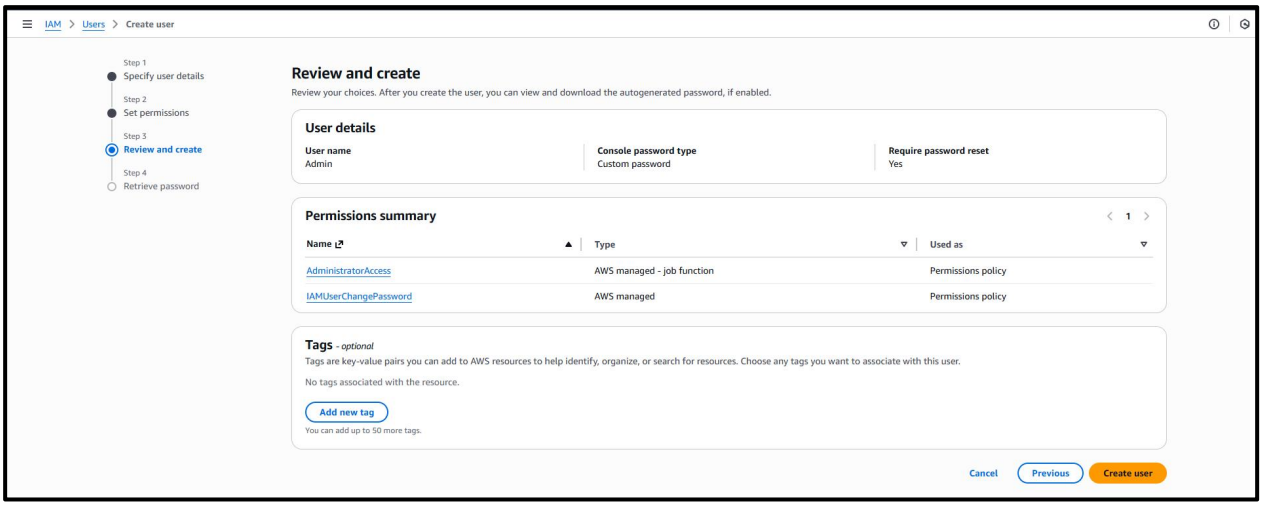


- Username: Admin
- Provide user access to the AWS Management Console
- Set console password (autogenerate or set)

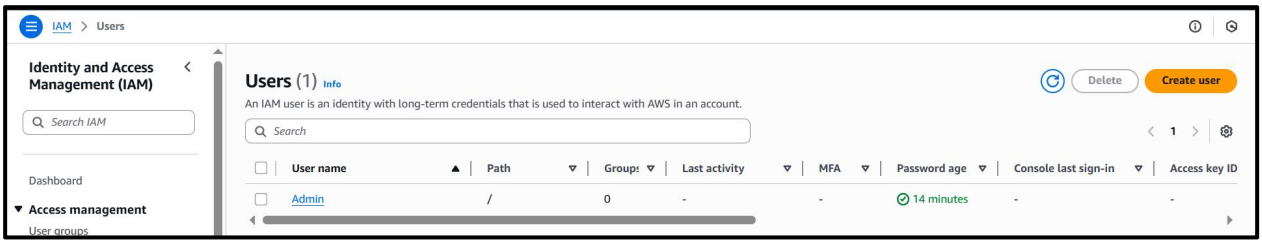


- Attach policies directly → AdministratorAccess





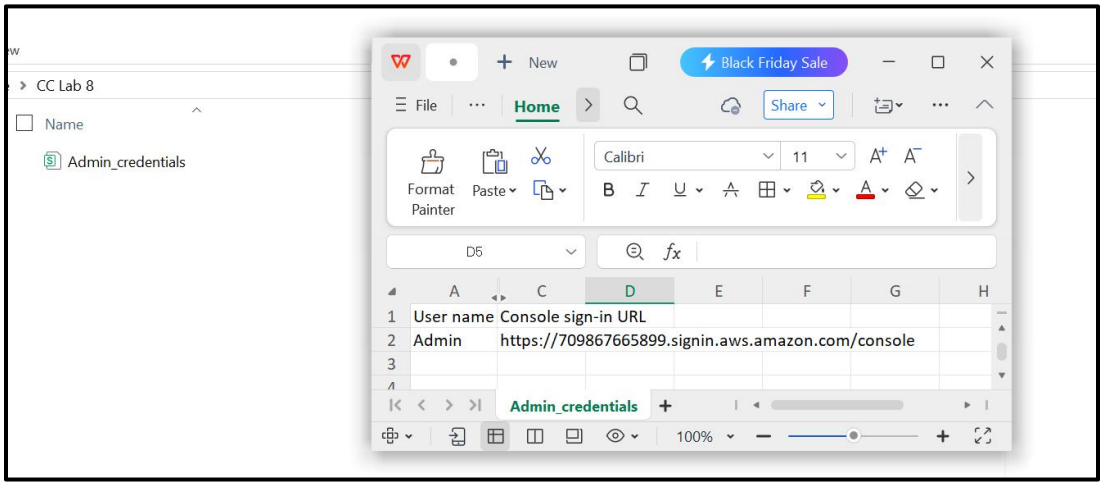
- Capture the completion screen when user is created:



Save screenshot as: task2_admin_create_confirmation.png — IAM "Create user" success screen showing Admin (do NOT include password).

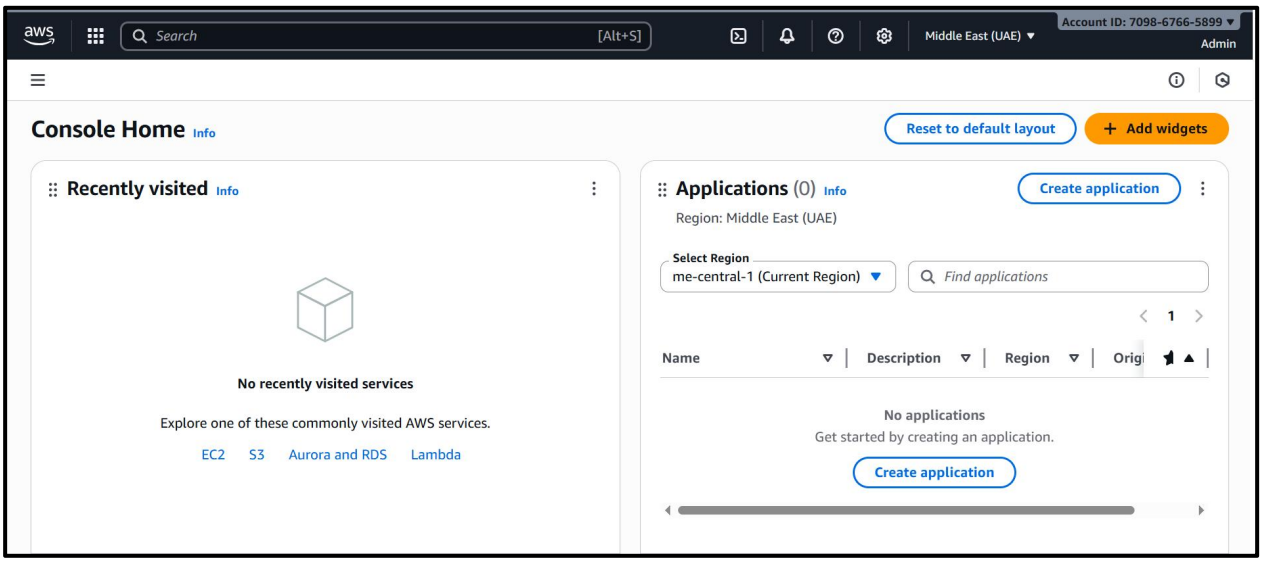
3. **Download the Admin .csv and show its presence on your Windows host (do not display the password text):**

Save screenshot as: task2_admin_csv_and_signin_url.png — Windows File Explorer showing the downloaded CSV filename and/or a cropped view of the CSV showing only the Sign-in URL and username (redact the password if visible).

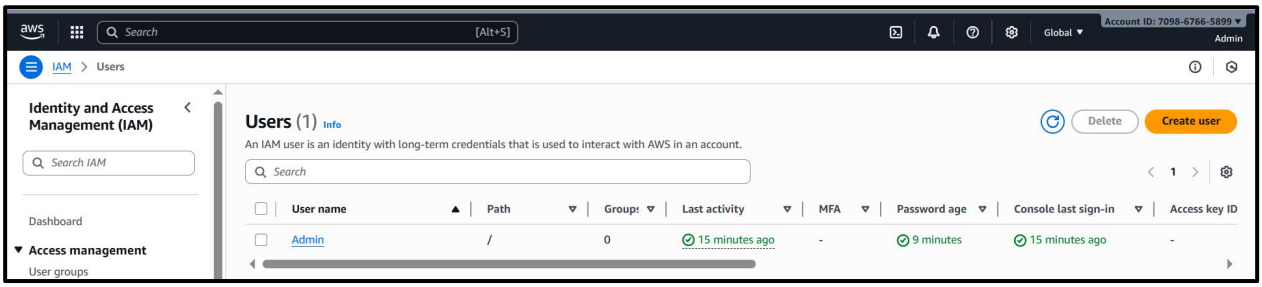


4. **Sign out of root, then sign in using the Admin account (use the signin URL from the .csv). Capture after successful Admin login:**

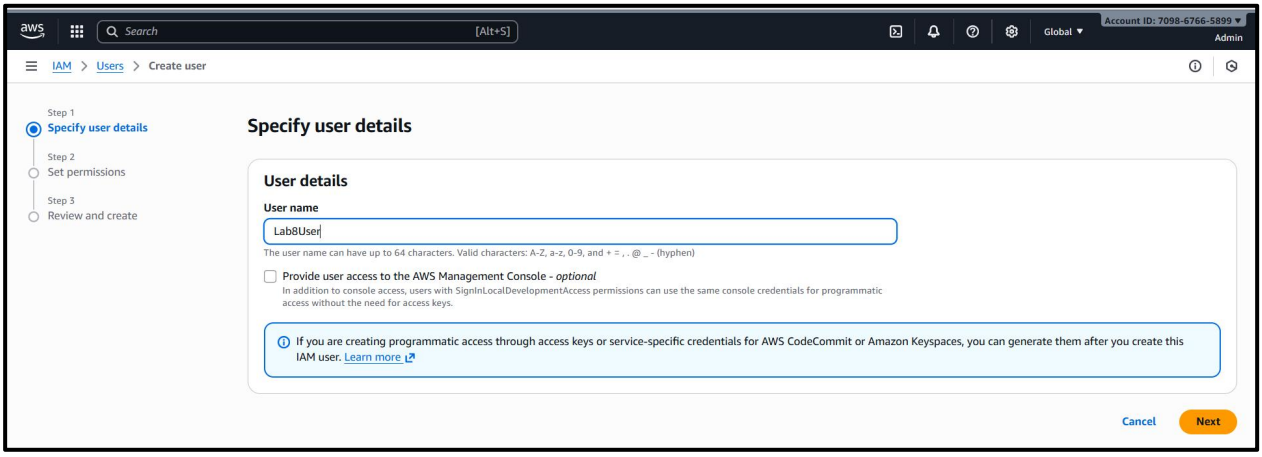
Save screenshot as: task2_admin_console_after_login.png — Admin user console home.



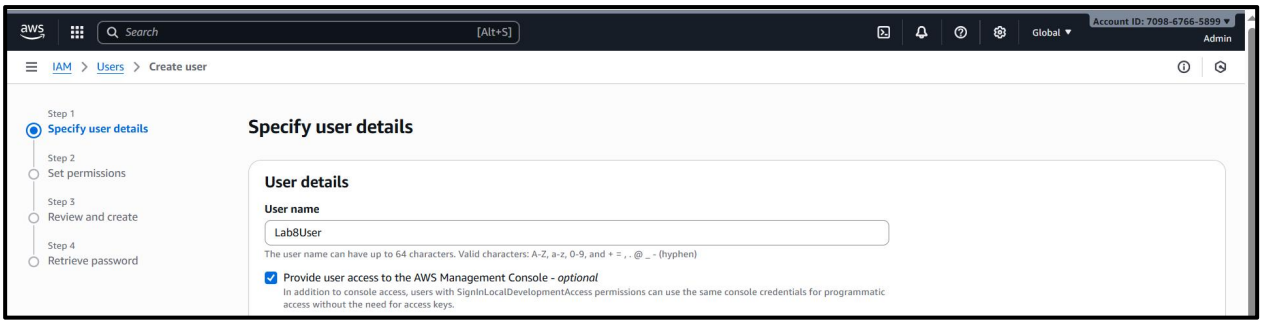
5. **While logged in as Admin, create Lab8User:**
   a. IAM → Users → Create user



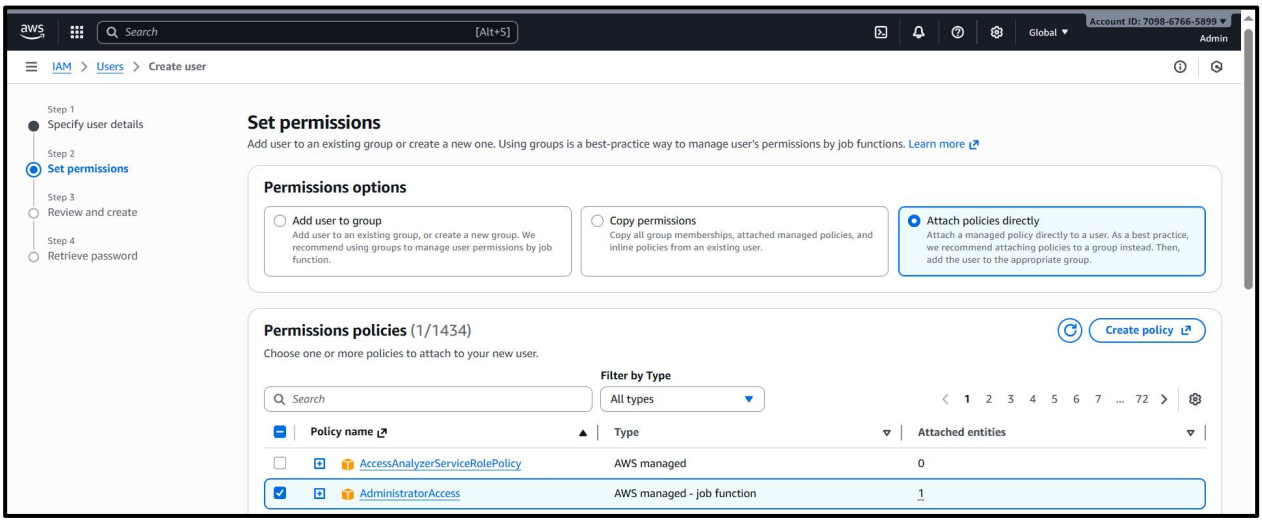
   b. Username: Lab8User



c. Provide user access to the AWS Management Console

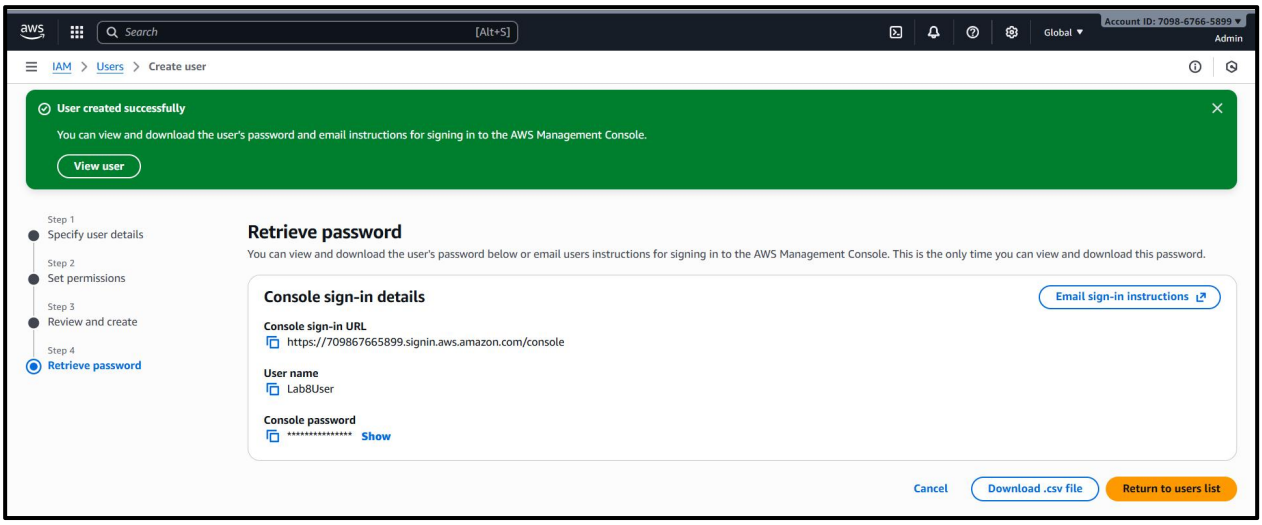


d. Attach AdministratorAccess policy



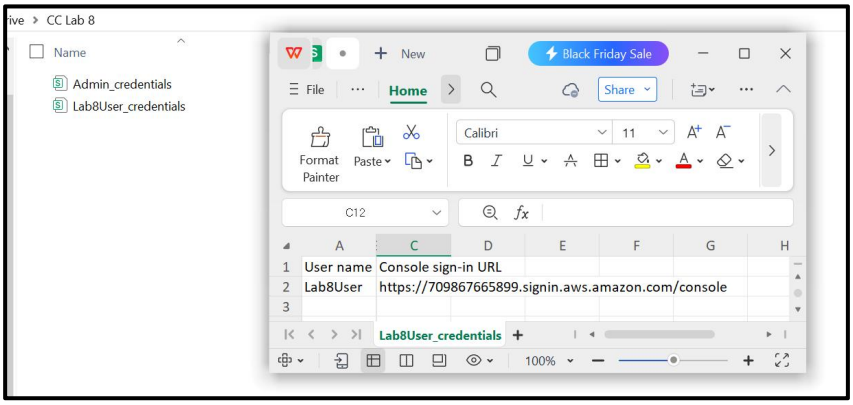
Capture the create-user success screen:

Save screenshot as: task2_create_lab8user_and_csv.png — Lab8User create confirmation and CSV download prompt (do NOT include password).



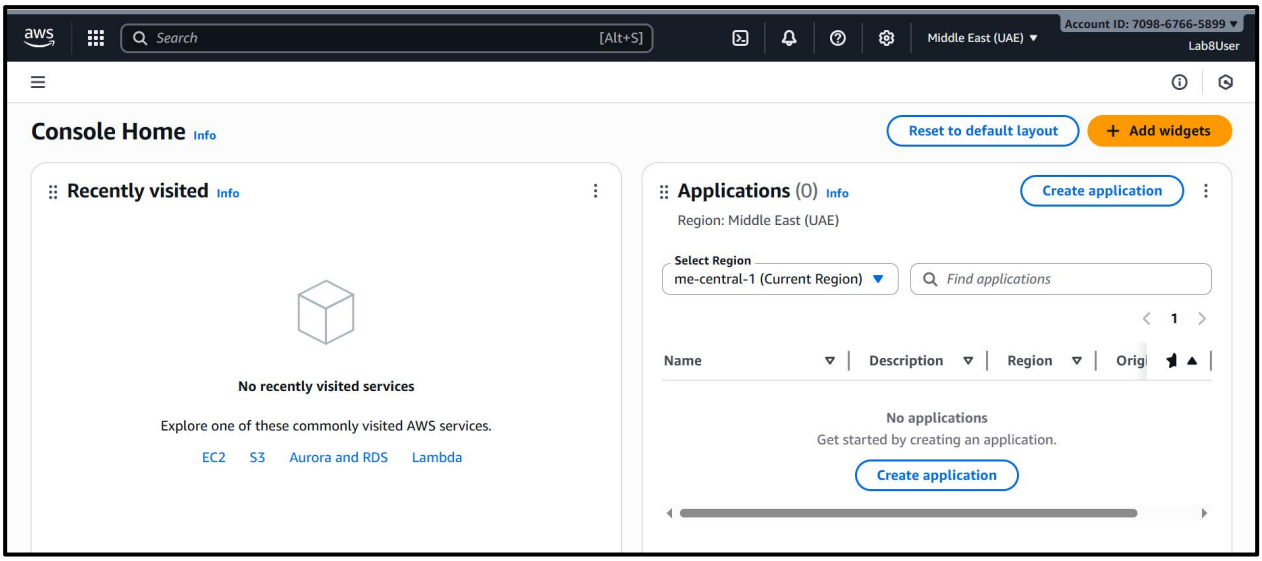
6. **Download/save the Lab8User CSV on your Windows host (do not show password).**

Save screenshot as: task2_lab8user_csv_saved.png — File Explorer showing the Lab8User CSV filename (cropped to exclude sensitive content).



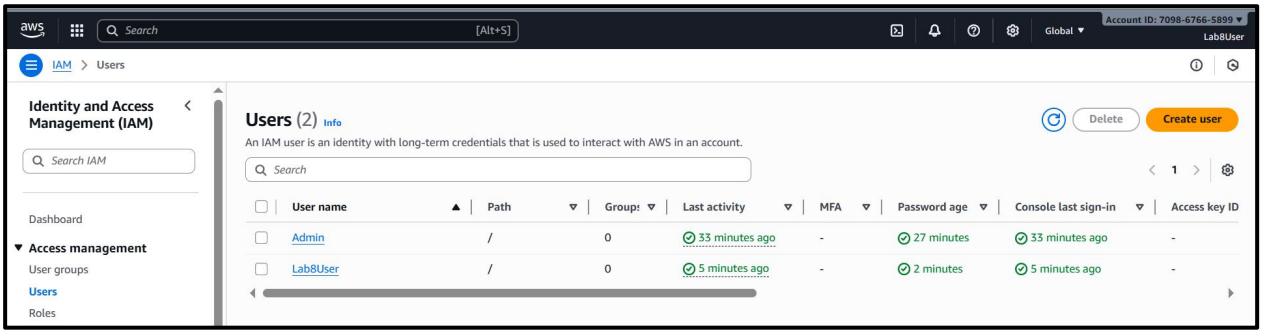
7. **Logout Admin and login as Lab8User (use the Lab8User signin URL and credentials). Capture after login:**

Save screenshot as: task2_lab8user_logged_in.png — Lab8User console home.



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

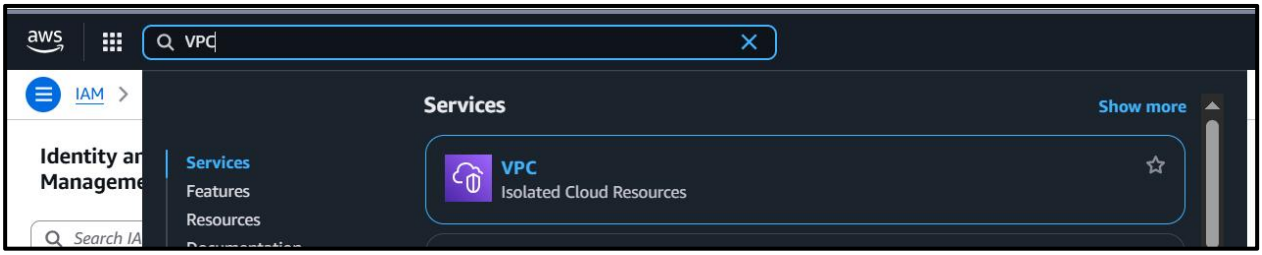
Save screenshot as: task2_summary.png — IAM Users list showing both Admin and Lab8User present (region me-central-1 visible).

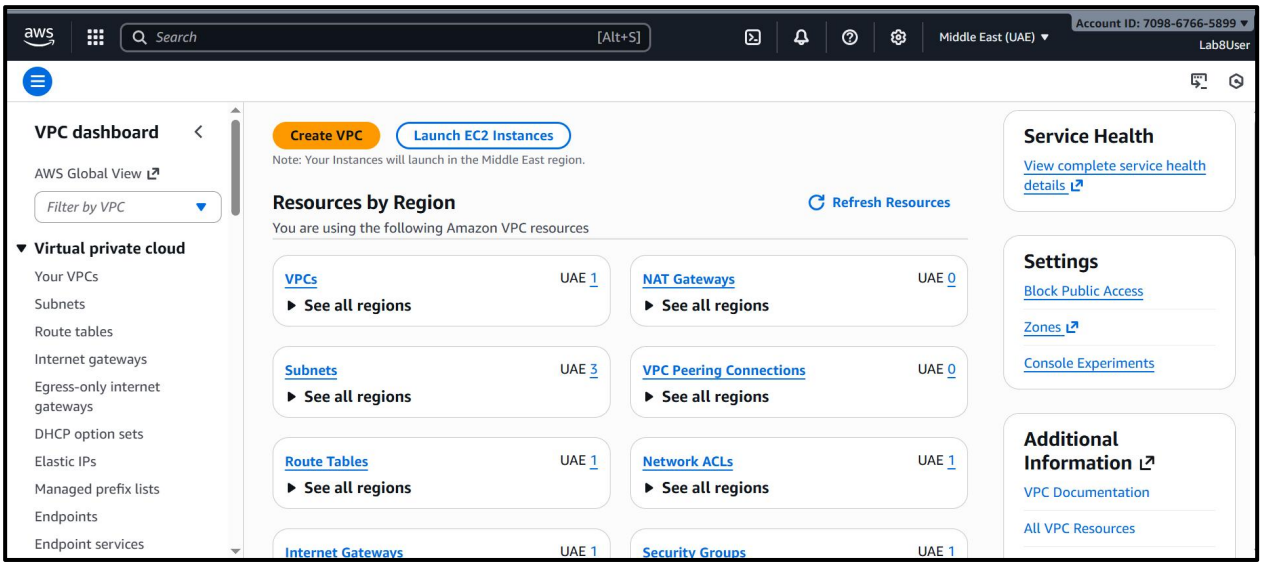


Task 3 — Inspect VPC resources (in UAE me-central-1)

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

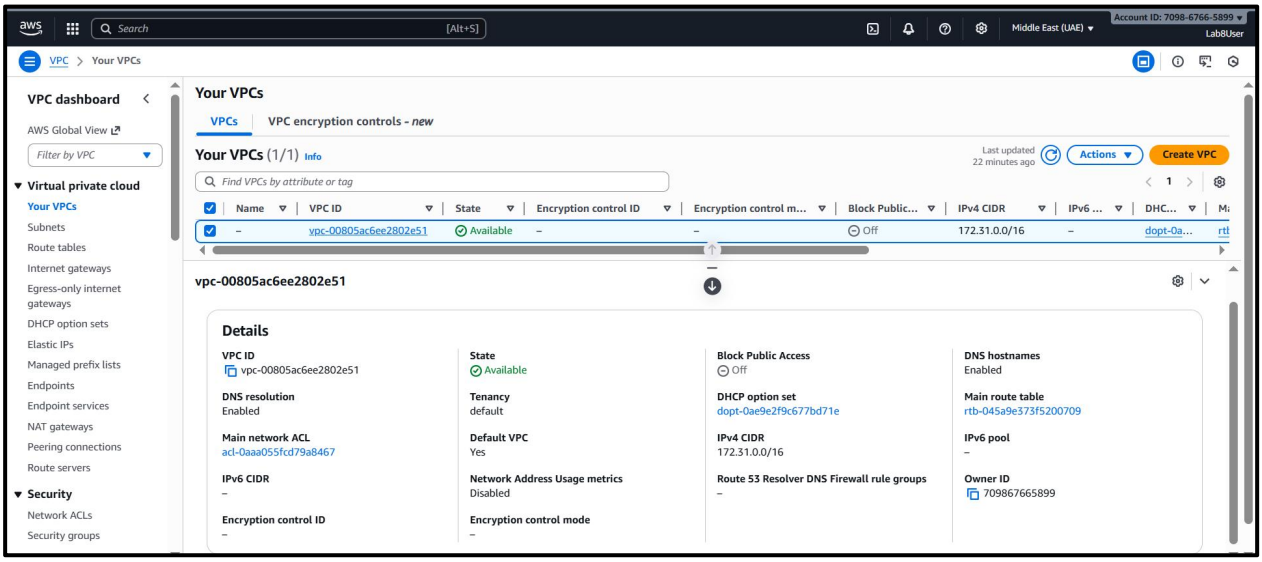
Save screenshot as: task3_open_vpc_console.png — VPC console landing page (region visible).





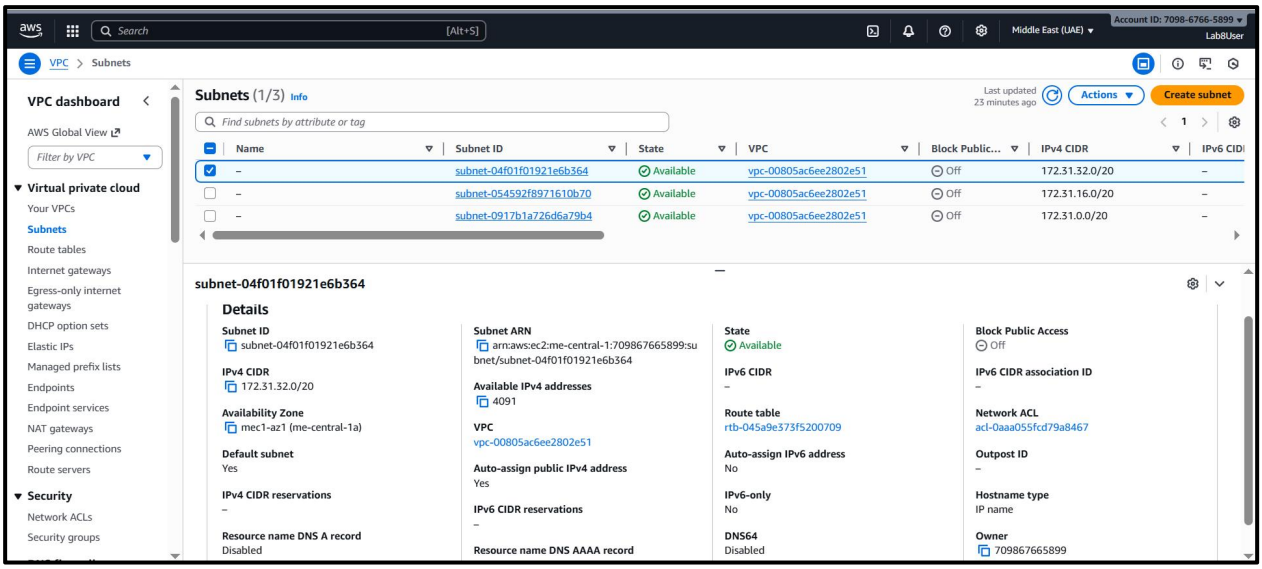
2. **View VPCs list. Capture:**

Save screenshot as: task3_vpcs_list.png — VPCs list view (show default VPC if present).



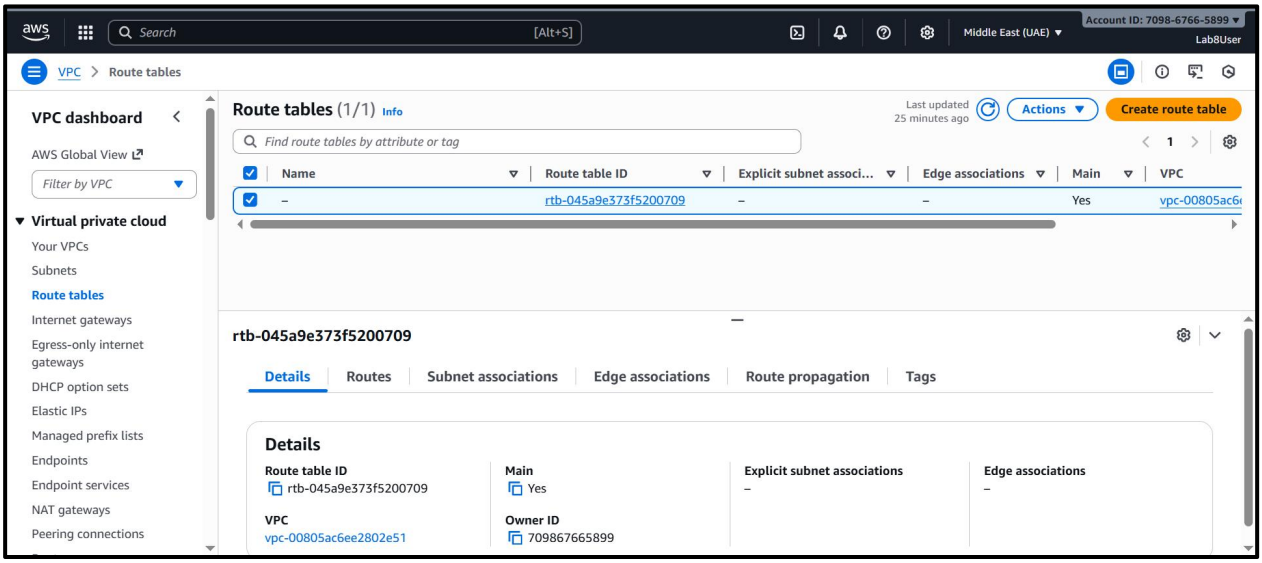
3. **View Subnets list. Capture:**

Save screenshot as: task3_subnets_list.png — Subnets list view (show at least 3 default subnets if present).



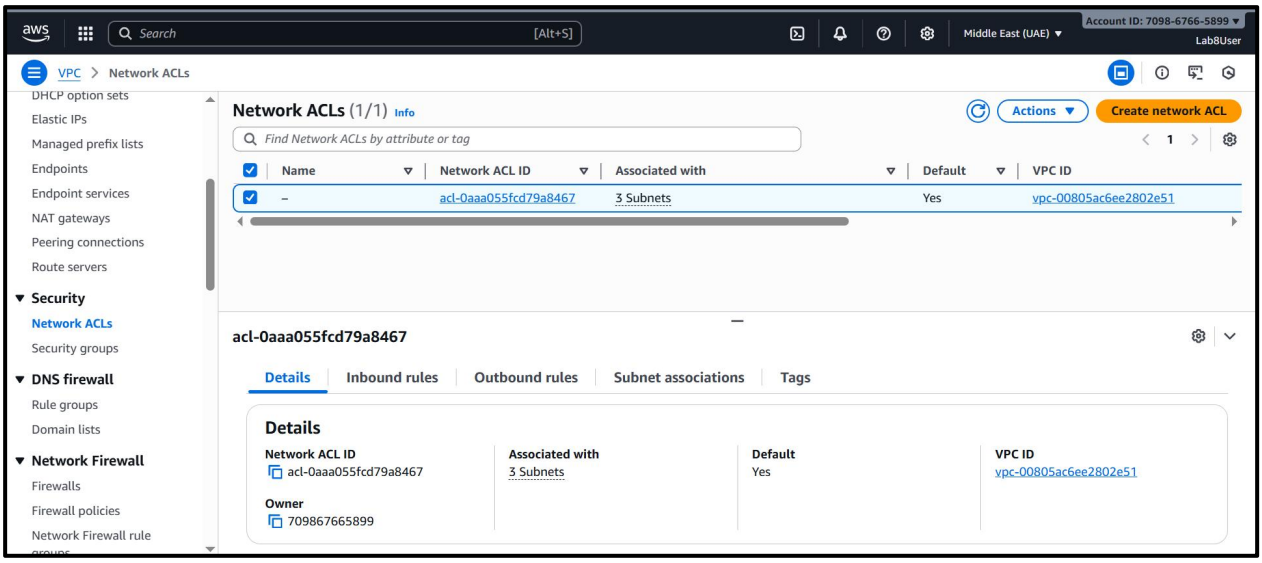
4. **View Route Tables list. Capture:**

Save screenshot as: task3_route_tables_list.png — Route Tables list view.



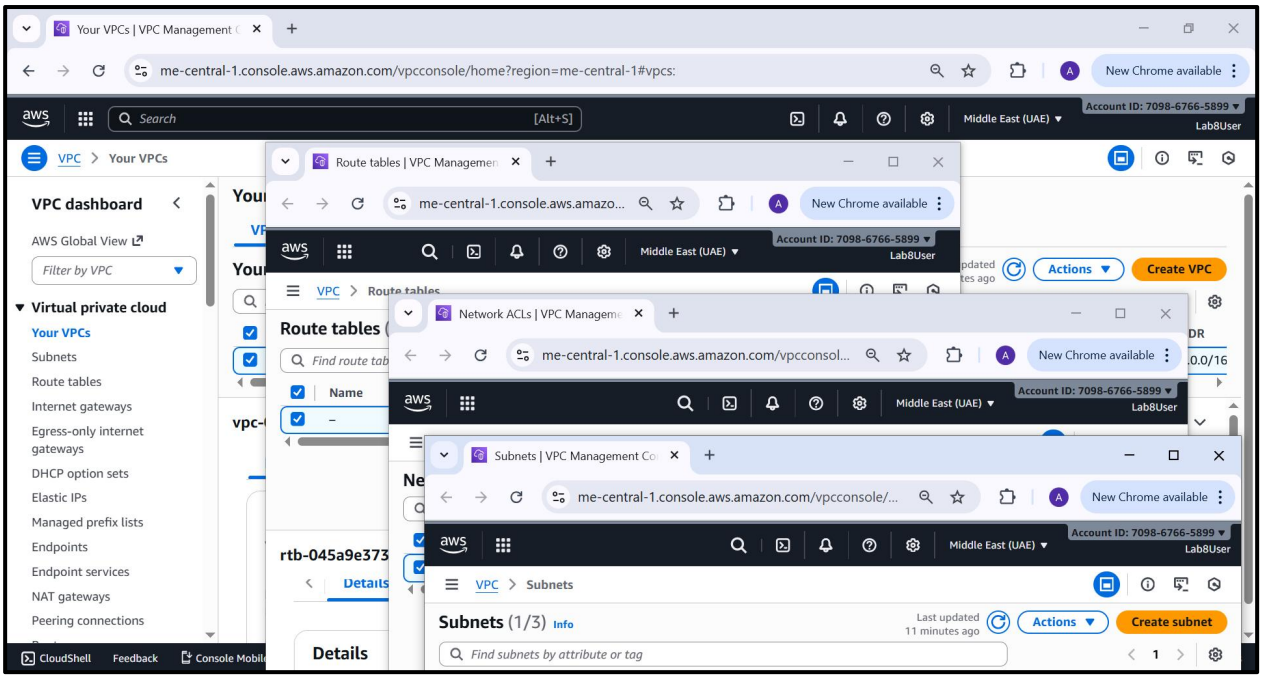
5. **View Network ACLs list. Capture:**

Save screenshot as: task3_network_acls_list.png — Network ACLs list view.



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

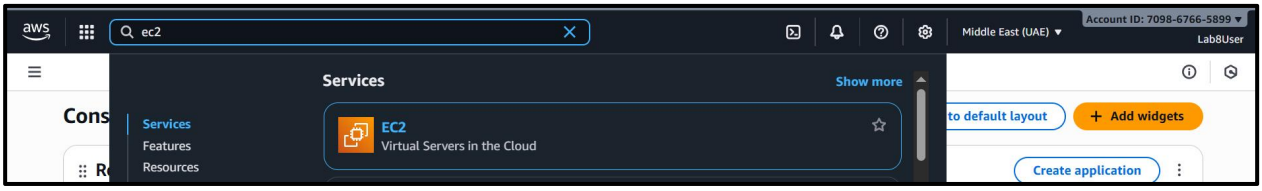
Save screenshot as: task3_summary.png — a single screenshot showing the VPC console left navigation and counts or multiple open tabs/windows tiled to show each resource's list (region me-central-1 visible).

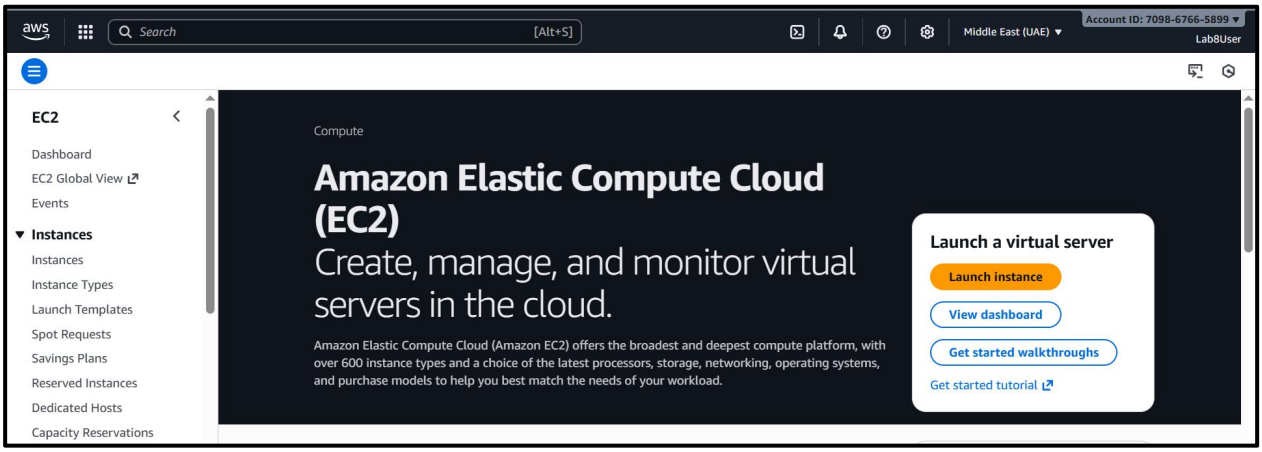


Task 4 — Launch EC2, SSH, install Docker & Docker Compose, deploy Gitea

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

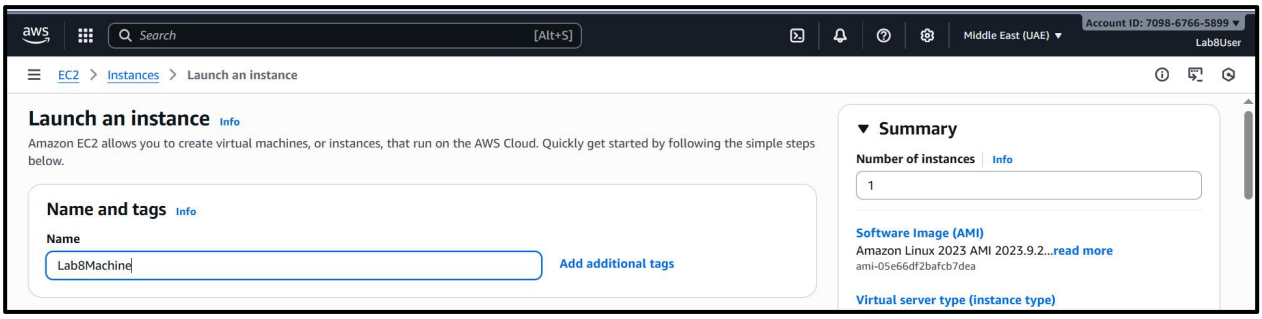
Save screenshot as: task4_open_ec2_console.png — EC2 console landing page with region visible.



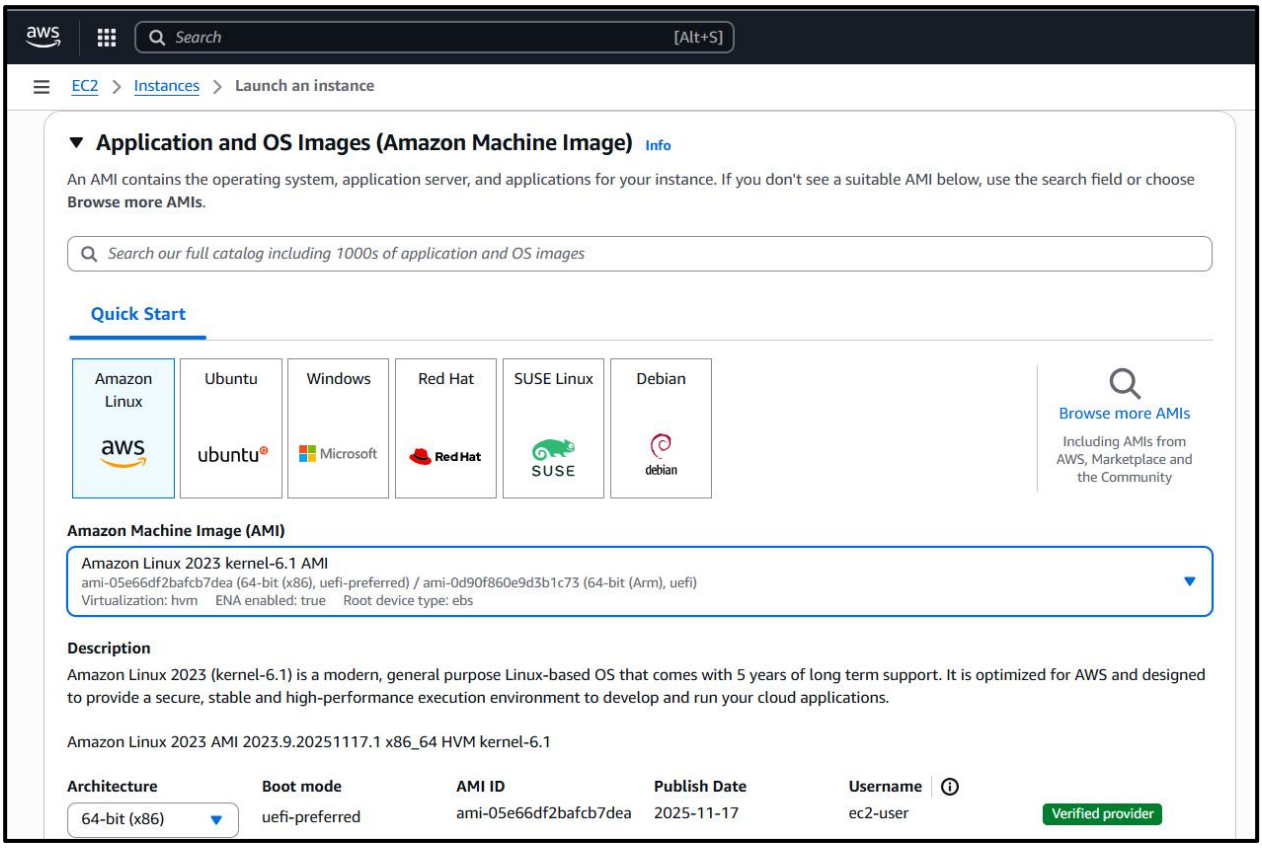


2. **Instance Launch configuration (during review before launching). Configure:**

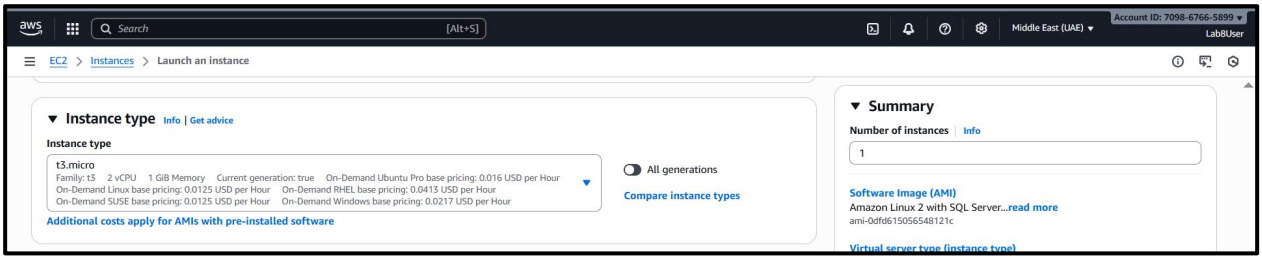
- **Name: Lab8Machine**



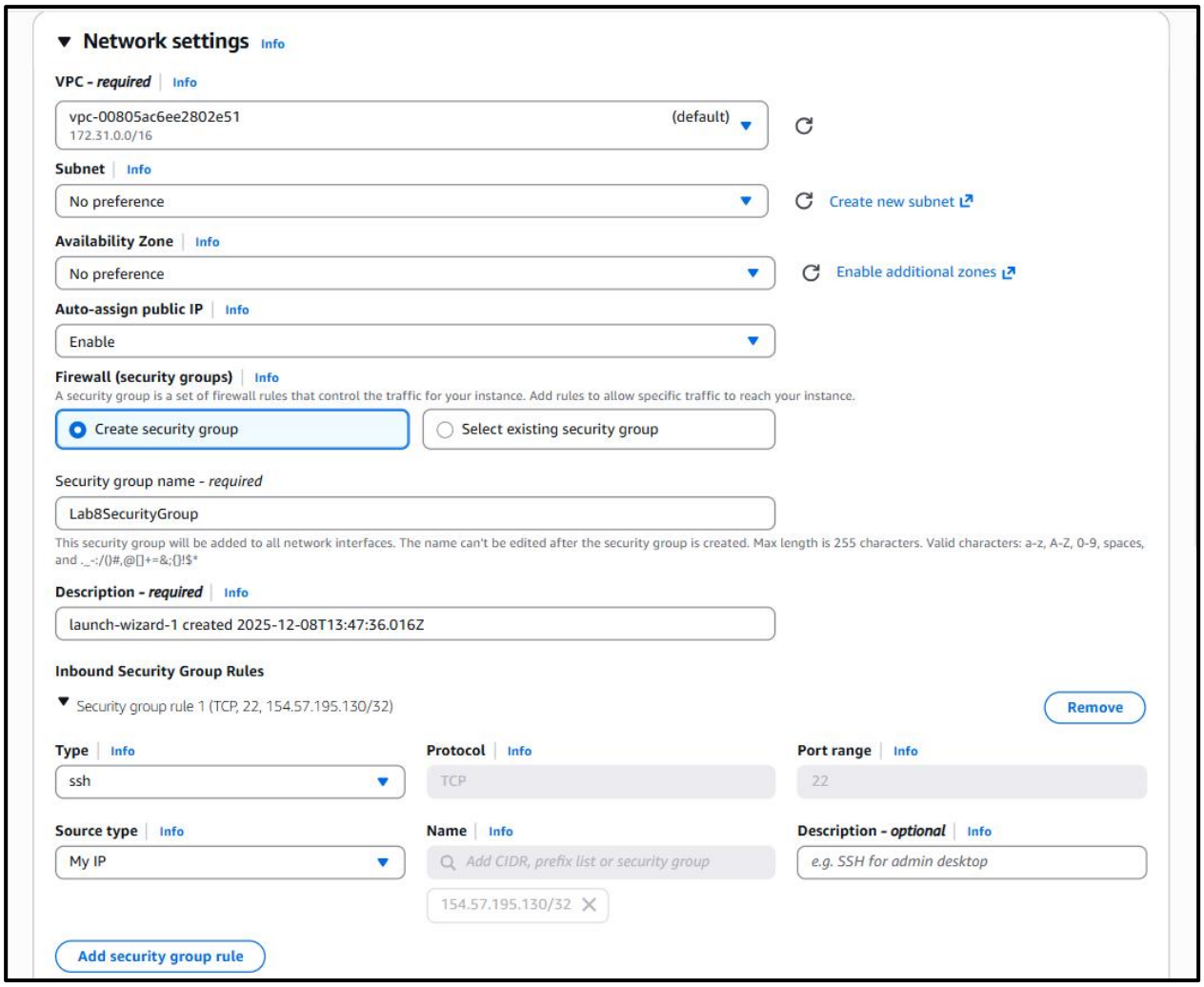
- **AMI: Amazon Linux 2**



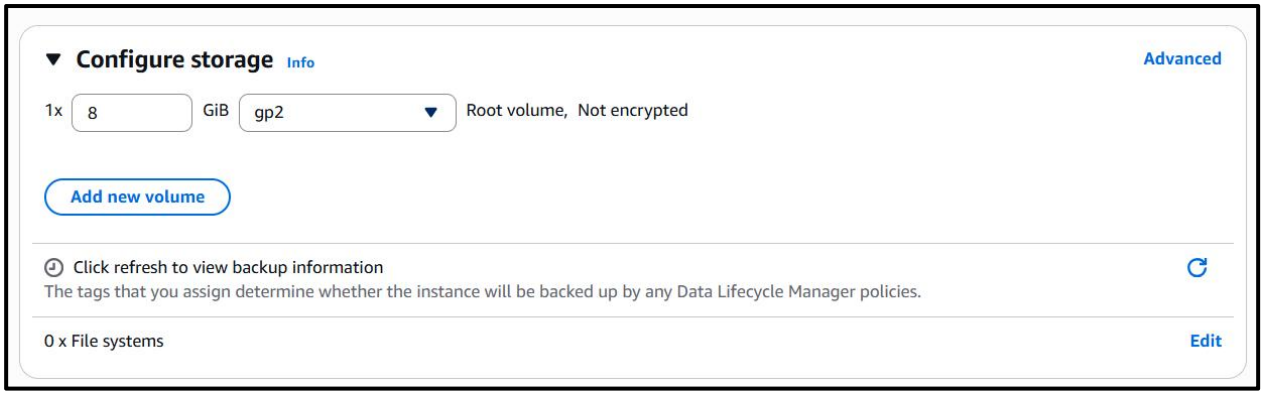
- **Instance type: t2.micro**



- **Security group: Create Lab8SecurityGroup with SSH from My IP**

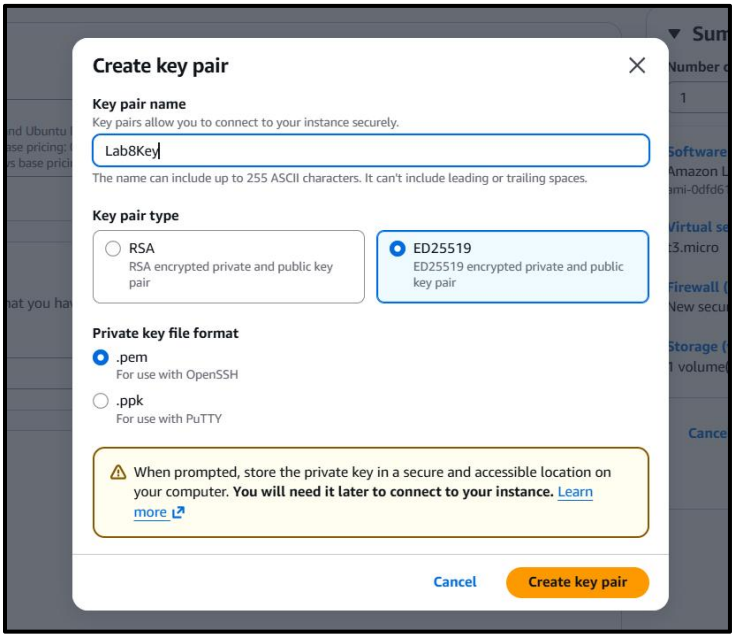


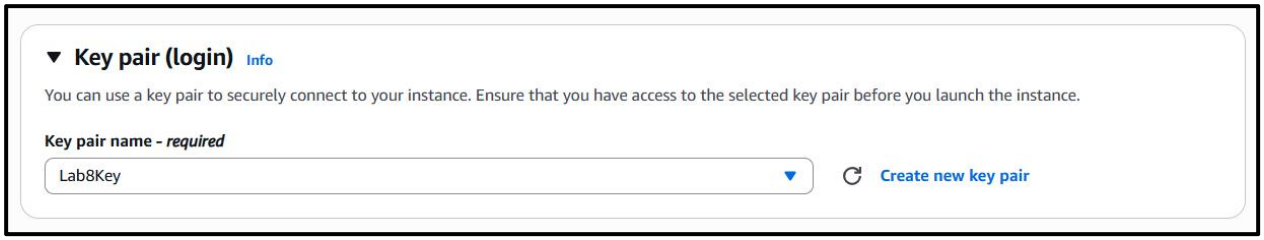
- **Storage: default**



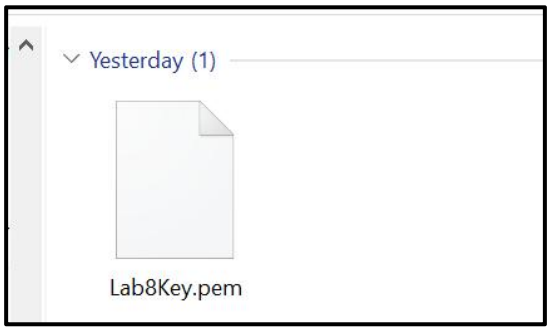
- **Key pair: Create Lab8Key (ED25519, .pem) and download the .pem file to your Windows host**
- **Capture the final review page and the key download prompt:**

- **Save screenshot as: task4_launch_instance_config.png — final review page showing instance name, AMI, type, security group, key pair.**



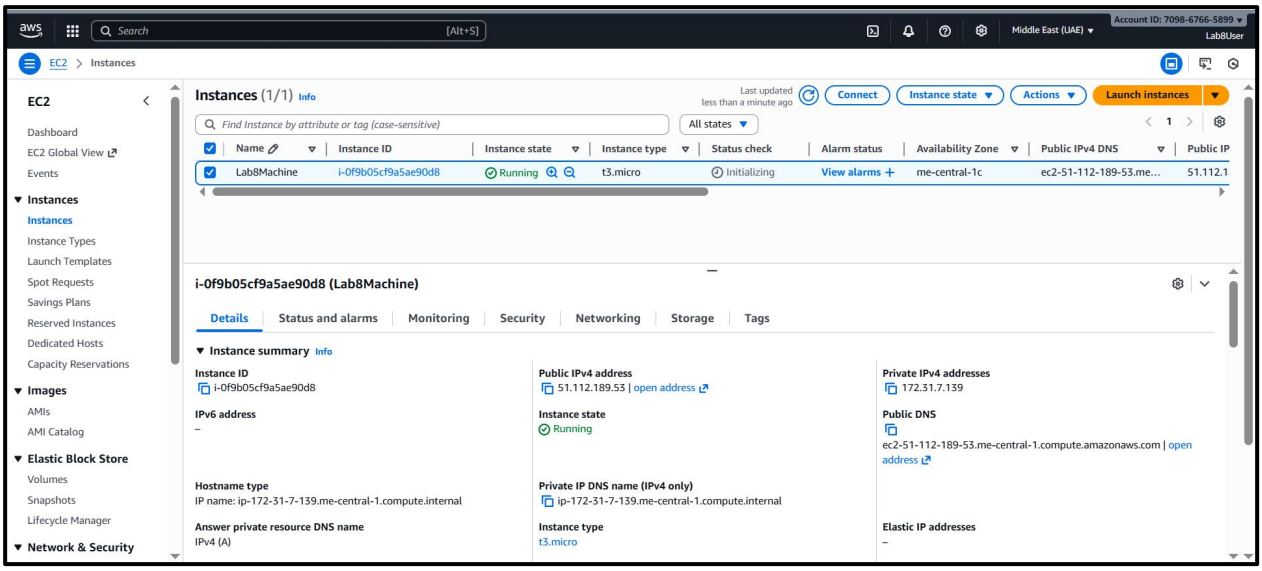


- **Save screenshot as: task4_keypair_download.png — Windows File Explorer showing Lab8Key.pem downloaded (do NOT open .pem contents).**



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

Save screenshot as: task4_instance_running_console.png — Instances table with Lab8Machine running and Public IPv4.



4. **On Windows host, run SSH using the downloaded .pem (PowerShell/Git Bash/Windows Terminal):**

ssh -i <path>/Lab8Key.pem ec2-user@<public-IP>

Capture the SSH command and successful shell prompt on the EC2 instance:

Save screenshot as: task4_ssh_from_windows_to_ec2.png — PowerShell showing ssh command and EC2 shell (do NOT show private key contents).

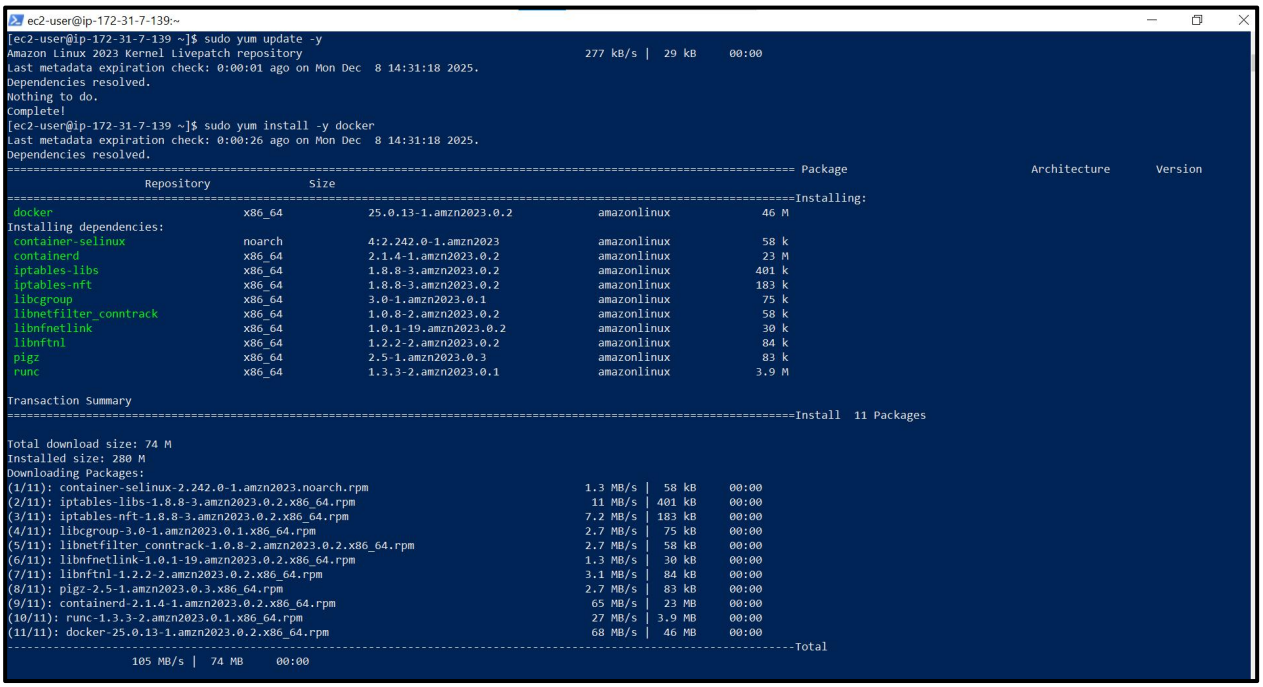
5. **Run the install commands on the EC2 shell:**

sudo yum update -y
sudo yum install -y docker
sudo mkdir -p /usr/local/lib/docker/cli-plugins

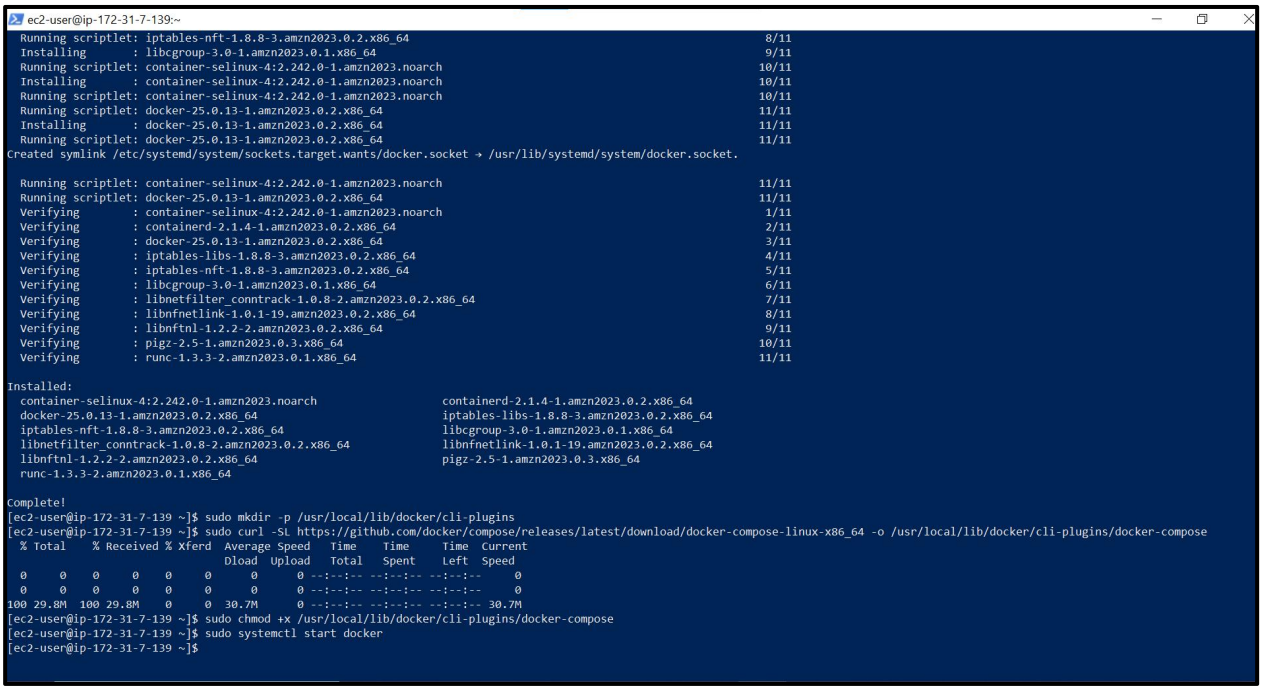
sudo curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86_64 -o /usr/local/lib/docker/cli-plugins/docker-compose
sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
sudo systemctl start docker

Capture the terminal showing these commands run and successful outputs:

Save screenshot as: task4_ec2_install_docker_compose_started.png — outputs of update/install and systemctl start.



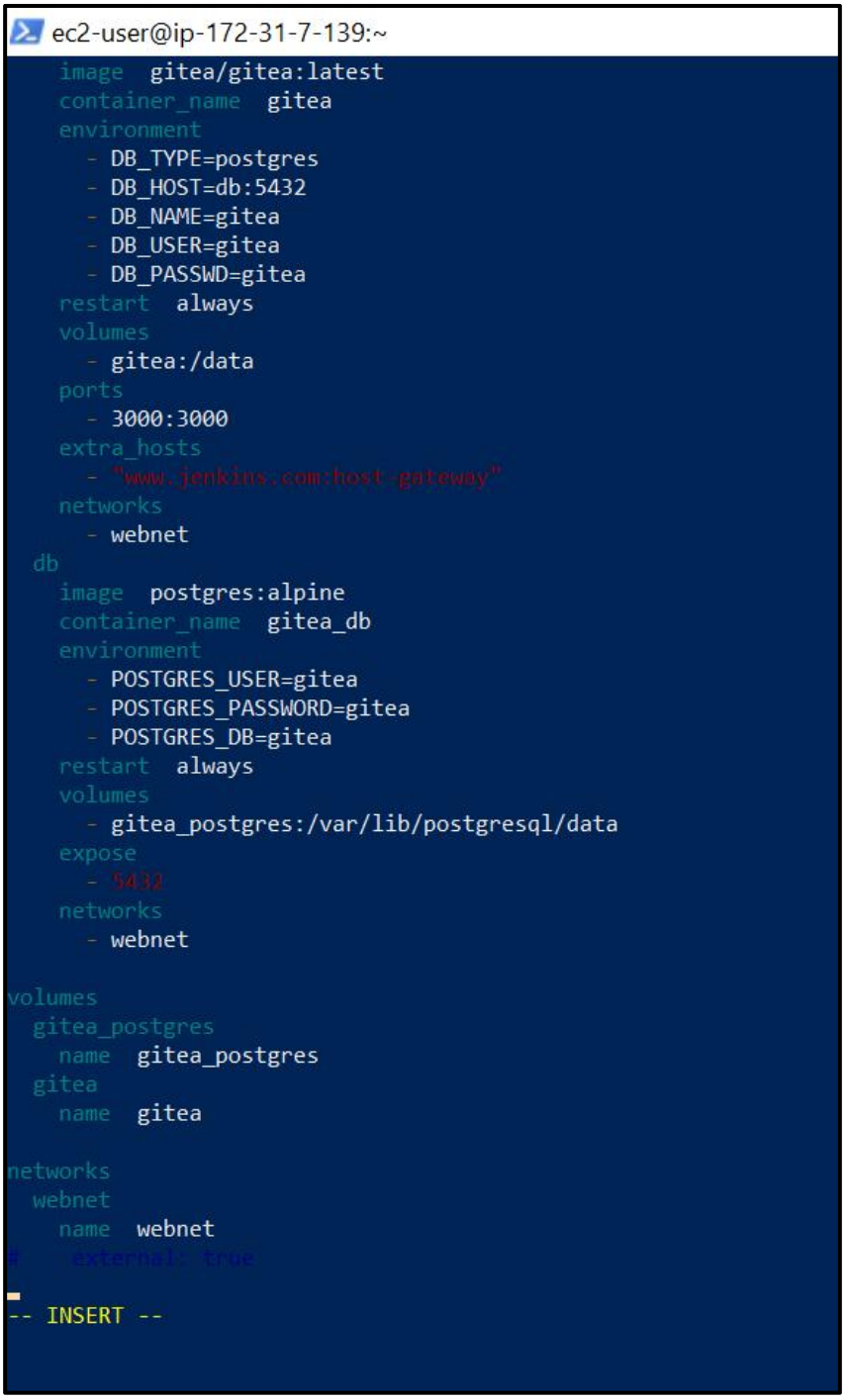
```
ec2-user@ip-172-31-7-139:~
[ec2-user@ip-172-31-7-139 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                    277 kB/s |  29 kB     00:00
Last metadata expiration check: 0:00:01 ago on Mon Dec  8 14:31:18 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-7-139 ~]$ sudo yum install -y docker
Last metadata expiration check: 0:00:26 ago on Mon Dec  8 14:31:18 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
 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
 libnfnetlink                    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
 pigz                            x86_64             2.5-1.amzn2023.0.3                 amazonlinux              83 k
 runc                            x86_64             1.3.3-2.amzn2023.0.1               amazonlinux             3.9 M

Transaction Summary
================================================================================Install  11 Packages

Total download size: 74 M
Installed size: 280 M
Downloading Packages:
(1/11): container-selinux-2.242.0-1.amzn2023.noarch.rpm          1.3 MB/s |  58 kB     00:00
(2/11): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm             11 MB/s | 401 kB     00:00
(3/11): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm             7.2 MB/s | 183 kB     00:00
(4/11): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm                  2.7 MB/s |  75 kB     00:00
(5/11): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm   2.7 MB/s |  58 kB     00:00
(6/11): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm            1.3 MB/s |  30 kB     00:00
(7/11): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm                 3.1 MB/s |  84 kB     00:00
(8/11): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                       2.7 MB/s |  83 kB     00:00
(9/11): containerd-2.1.4-1.amzn2023.0.2.x86_64.rpm                65 MB/s |  23 MB     00:00
(10/11): runc-1.3.3-2.amzn2023.0.1.x86_64.rpm                     27 MB/s | 3.9 MB     00:00
(11/11): docker-25.0.13-1.amzn2023.0.2.x86_64.rpm                 68 MB/s |  46 MB     00:00
--------------------------------------------------------------------------------Total
                  105 MB/s |  74 MB     00:00
```



```
ec2-user@ip-172-31-7-139:~
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                 8/11
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                                      9/11
  Running scriptlet: container-selinux-4:2.242.0-1.amzn2023.noarch                           10/11
  Installing       : container-selinux-4:2.242.0-1.amzn2023.noarch                           10/11
  Running scriptlet: container-selinux-4:2.242.0-1.amzn2023.noarch                           10/11
  Running scriptlet: docker-25.0.13-1.amzn2023.0.2.x86_64                                    11/11
  Installing       : docker-25.0.13-1.amzn2023.0.2.x86_64                                    11/11
  Running scriptlet: docker-25.0.13-1.amzn2023.0.2.x86_64                                    11/11
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Running scriptlet: container-selinux-4:2.242.0-1.amzn2023.noarch                           11/11
  Running scriptlet: docker-25.0.13-1.amzn2023.0.2.x86_64                                    11/11
  Verifying        : container-selinux-4:2.242.0-1.amzn2023.noarch                            1/11
  Verifying        : containerd-2.1.4-1.amzn2023.0.2.x86_64                                   2/11
  Verifying        : docker-25.0.13-1.amzn2023.0.2.x86_64                                     3/11
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                4/11
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                 5/11
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                      6/11
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                       7/11
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                8/11
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                     9/11
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                          10/11
  Verifying        : runc-1.3.3-2.amzn2023.0.1.x86_64                                        11/11

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-172-31-7-139 ~]$ sudo mkdir -p /usr/local/lib/docker/cli-plugins
[ec2-user@ip-172-31-7-139 ~]$ 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
100 29.8M  100 29.8M    0     0  30.7M      0 --:--:-- --:--:-- --:--:-- 30.7M
[ec2-user@ip-172-31-7-139 ~]$ sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
[ec2-user@ip-172-31-7-139 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-7-139 ~]$
```

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:

Save screenshot as: task4_vim_compose_yaml_paste.png — vim editor showing compose.yaml contents while pasted.



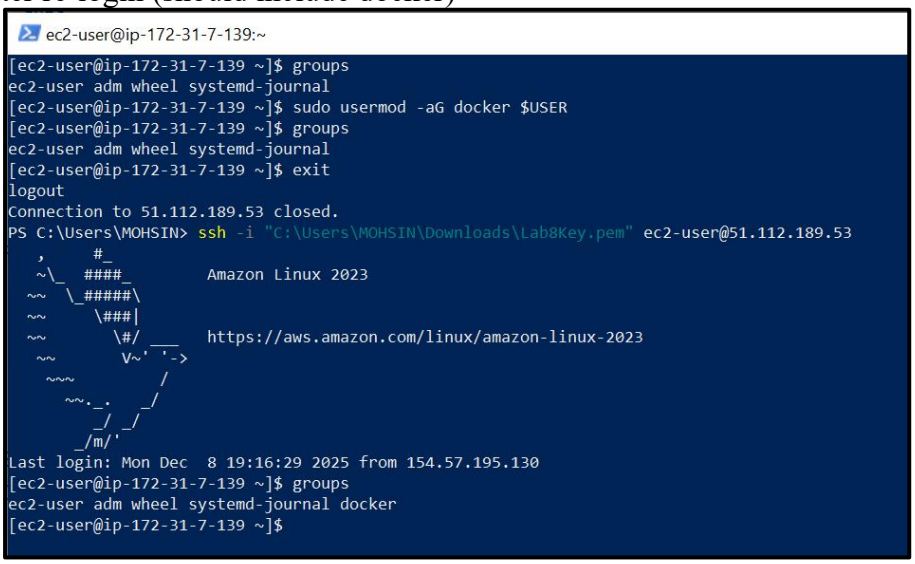
7. Save and verify file exists:

Save screenshot as: task4_compose_yaml_saved_ls.png — ls -l showing compose.yaml present.

```
[ec2-user@ip-172-31-7-139 ~]$ sudo vim compose.yaml
[ec2-user@ip-172-31-7-139 ~]$ ls -l compose.yaml
-rw-r--r--. 1 root root 5196 Dec  8 18:39 compose.yaml
[ec2-user@ip-172-31-7-139 ~]$
```

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

groups # user does not docker permission
sudo usermod -aG docker $USER
groups # before re-login
exit
Reconnect
ssh -i <path>/Lab8Key.pem ec2-user@<public-IP>
groups # after re-login (should include docker)

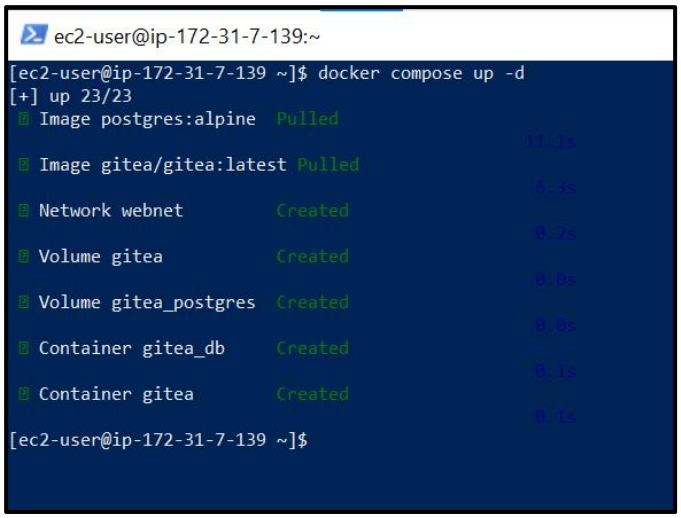


Save screenshot as: task4_usermod_and_groups_before_after.png — show usermod command, groups output before exit, reconnect sequence, and groups output after (docker included).

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

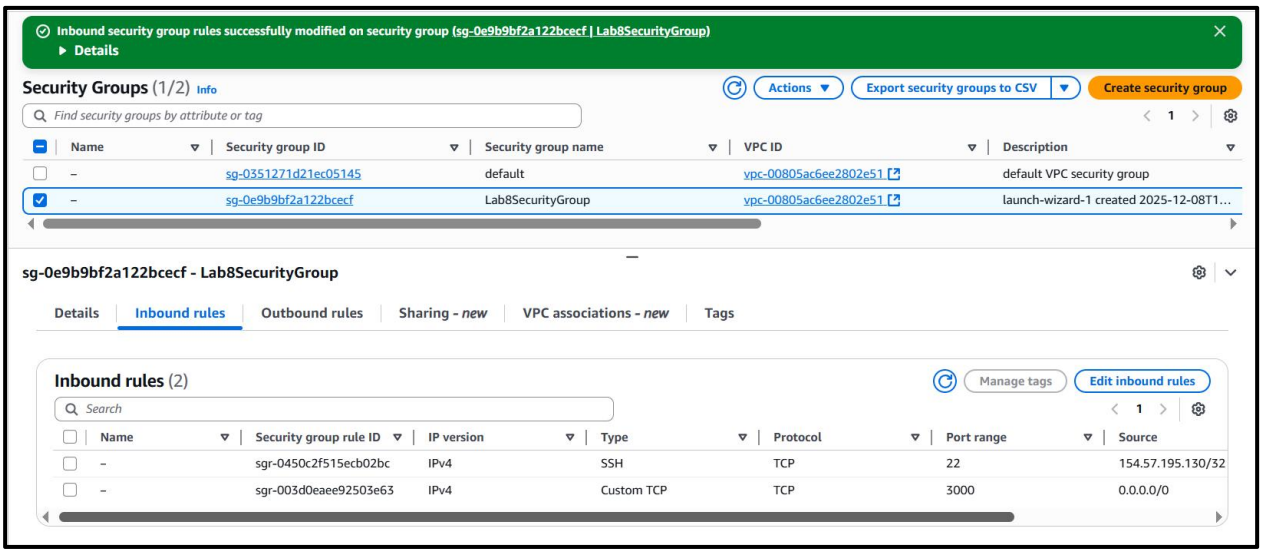
docker compose up -d

Save screenshot as: task4_docker_compose_up.png — output of docker compose up -d showing containers starting.



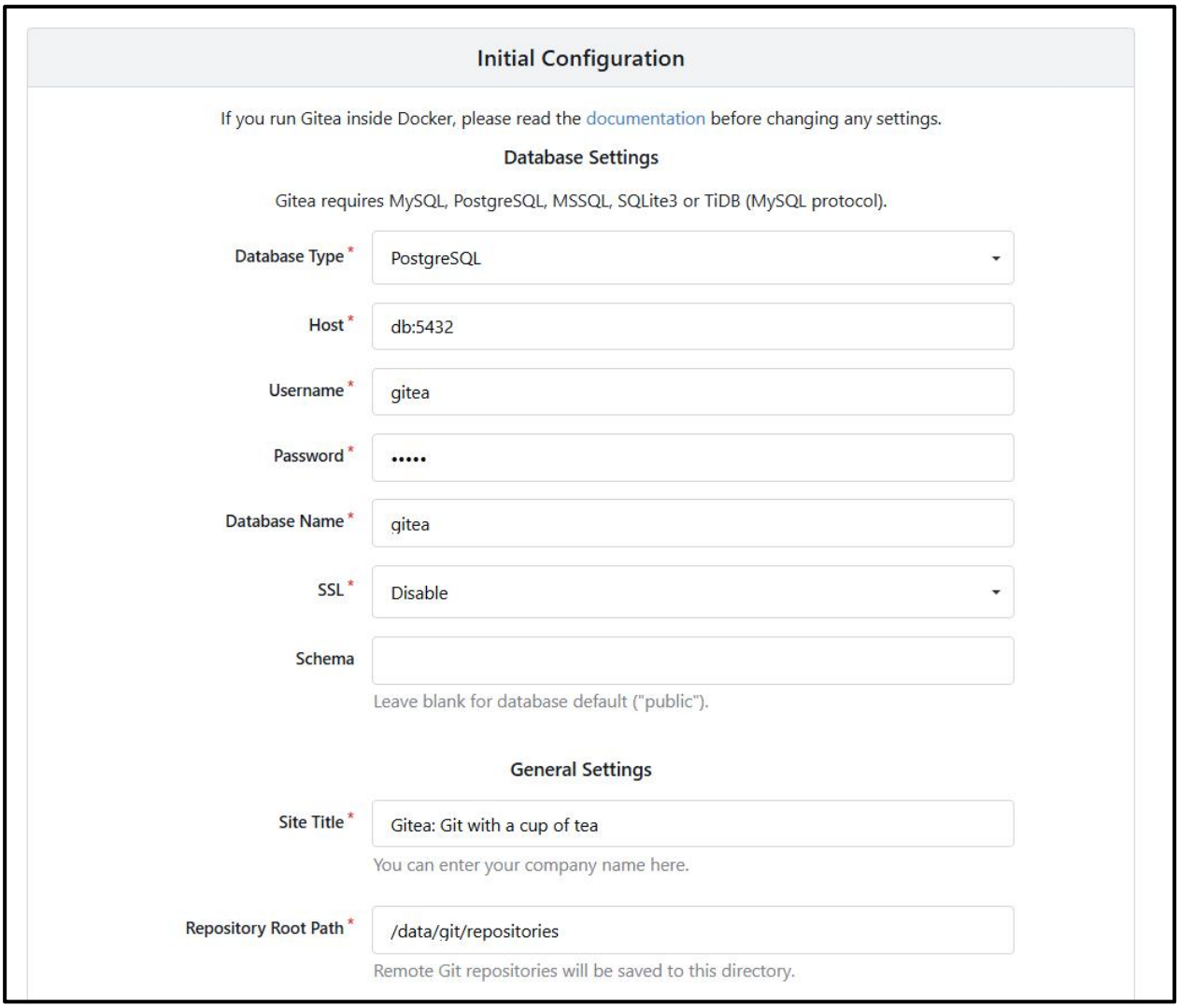
10. Edit the security group Lab8SecurityGroup inbound rules in the EC2 console: add Custom TCP rule port 3000 source 0.0.0.0/0 and save. Capture the inbound rules after saving:

Save screenshot as: task4_security_group_allow_3000.png — security group inbound rules list showing SSH from My IP and Custom TCP 3000 anywhere.



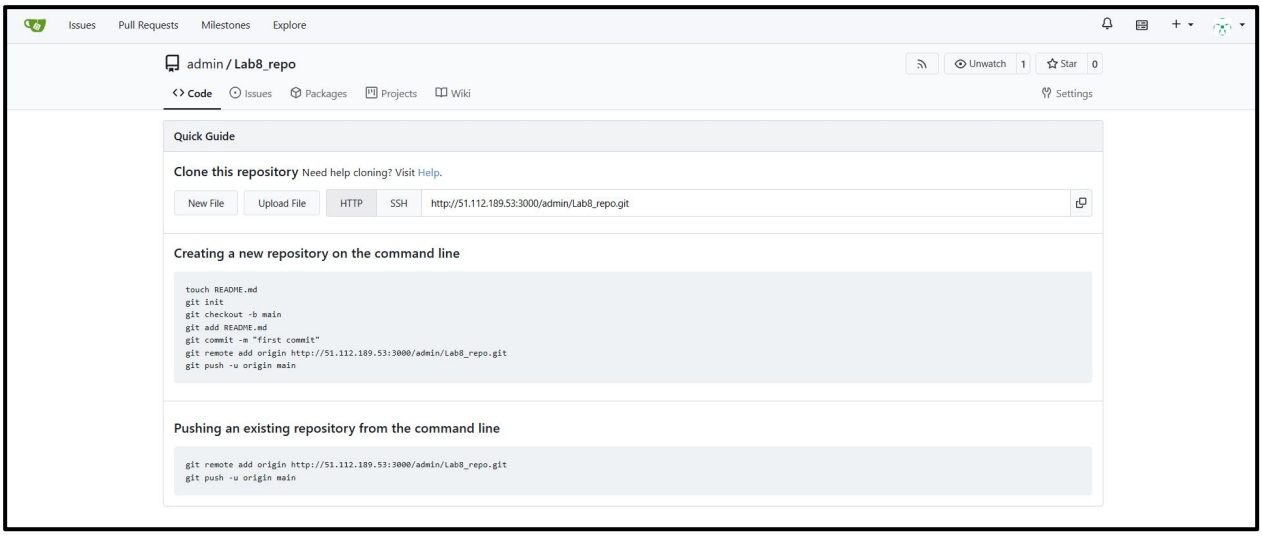
11. From your Windows browser navigate to: http://Public-IP:3000 — capture the Gitea setup/install page:

Save screenshot as: task4_gitea_install_page.png — Gitea installation page in browser.



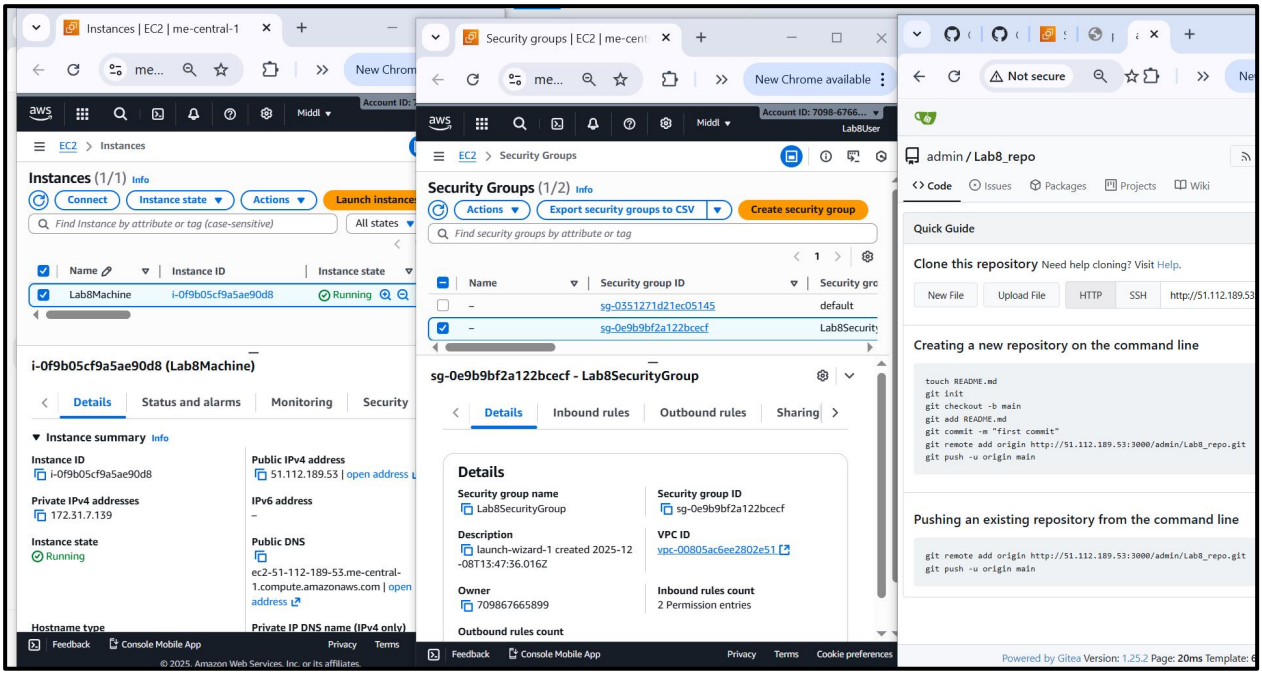
12. Complete initial Gitea setup (create admin user, create a repo) and capture Gitea showing the created repository:

Save screenshot as: task4_gitea_create_repo.png — Gitea UI showing the created repository.



13. Task 4 summary (combine evidence)

Save screenshot as: task4_summary.png — single screenshot (or tiled screenshot) showing: EC2 Instances list with Lab8Machine running and public IP, security group inbound rules showing SSH and port 3000, and browser tab open to Gitea UI or repo list.



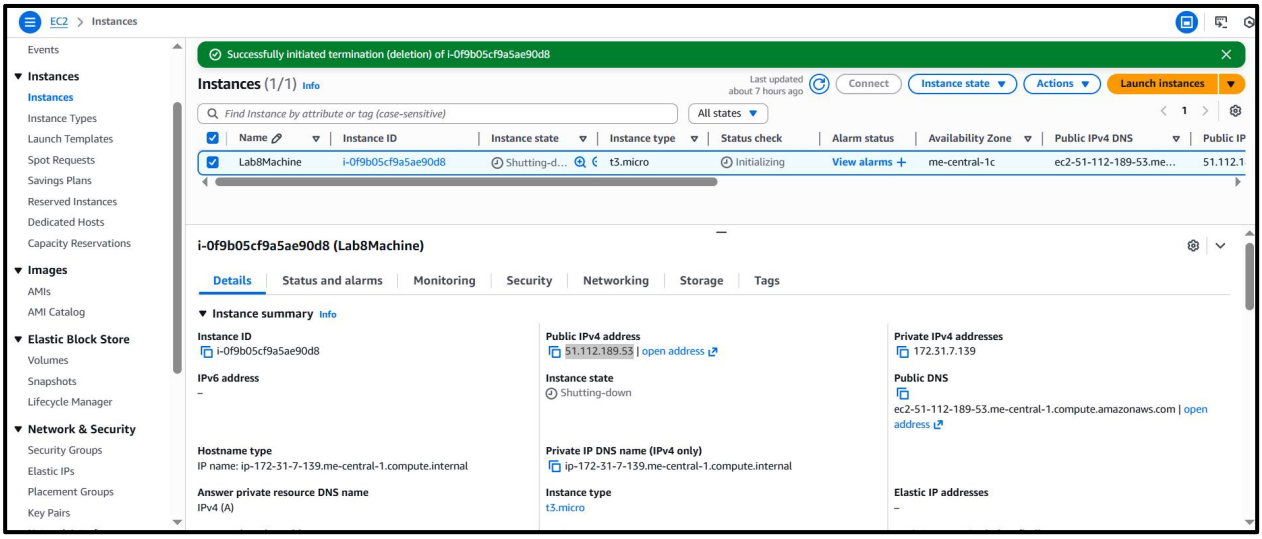
Cleanup — Remove resources to avoid charges

After verification, terminate and delete everything you created. Capture screenshots immediately after each cleanup step.

Cleanup steps and required screenshots:

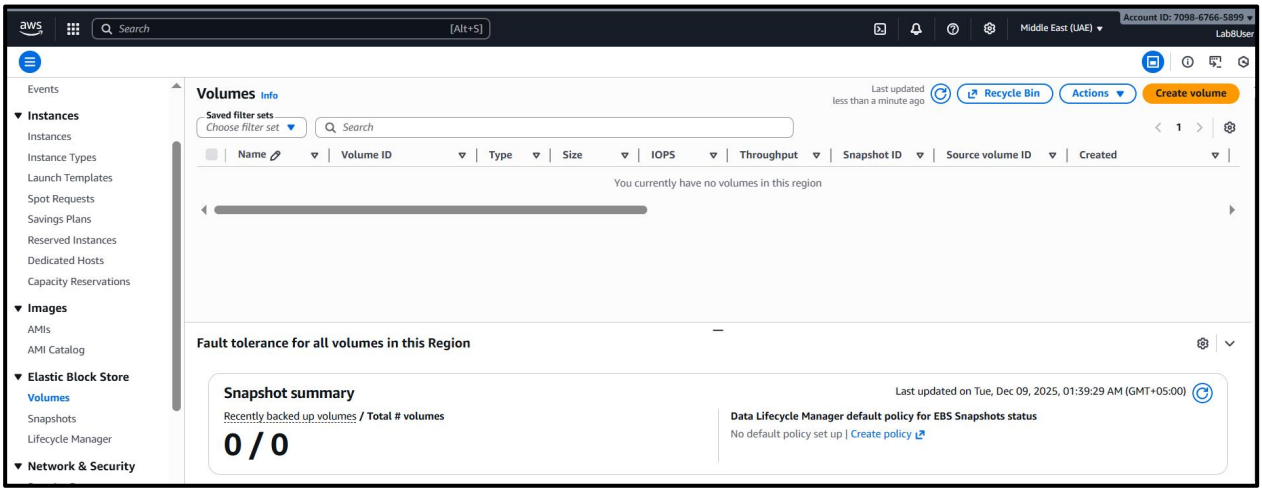
1. **Terminate the EC2 instance Lab8Machine.**

Save screenshot as: cleanup_terminate_instance.png — EC2 terminate instance confirmation.



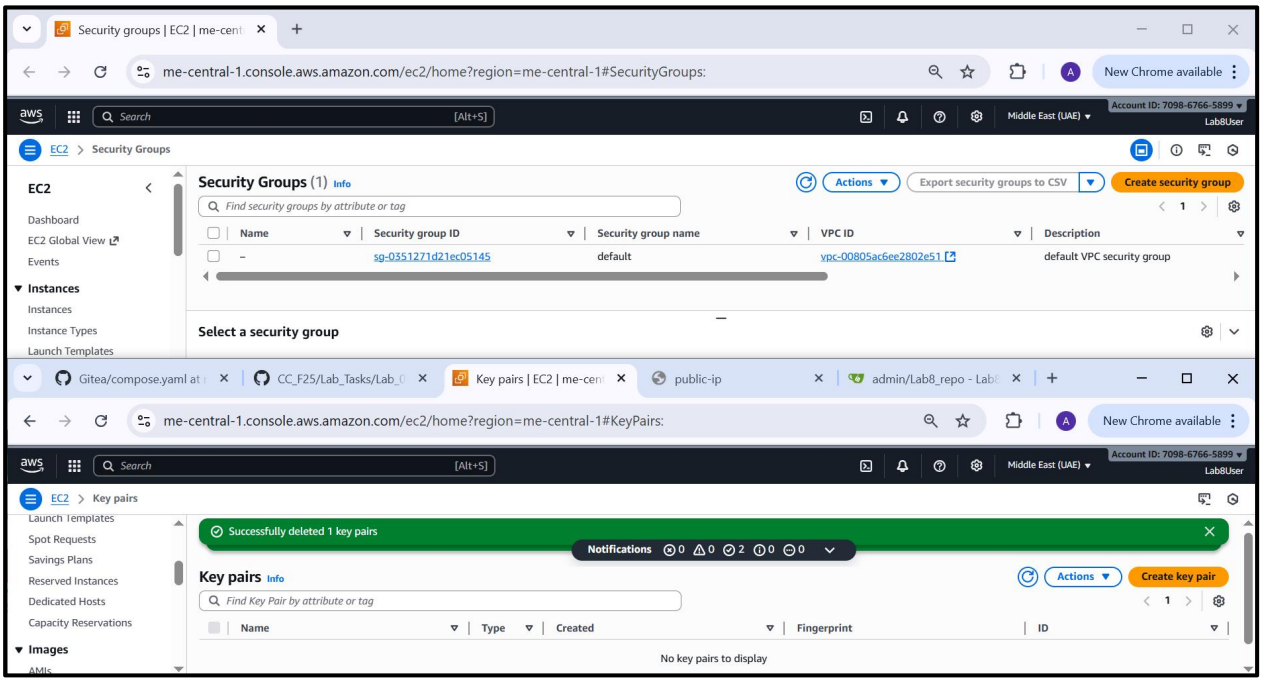
2. **Delete associated EBS volumes and snapshots (if any).**

Save screenshot as: cleanup_delete_volumes_snapshots.png — confirmation or list showing volumes/snapshots deleted.



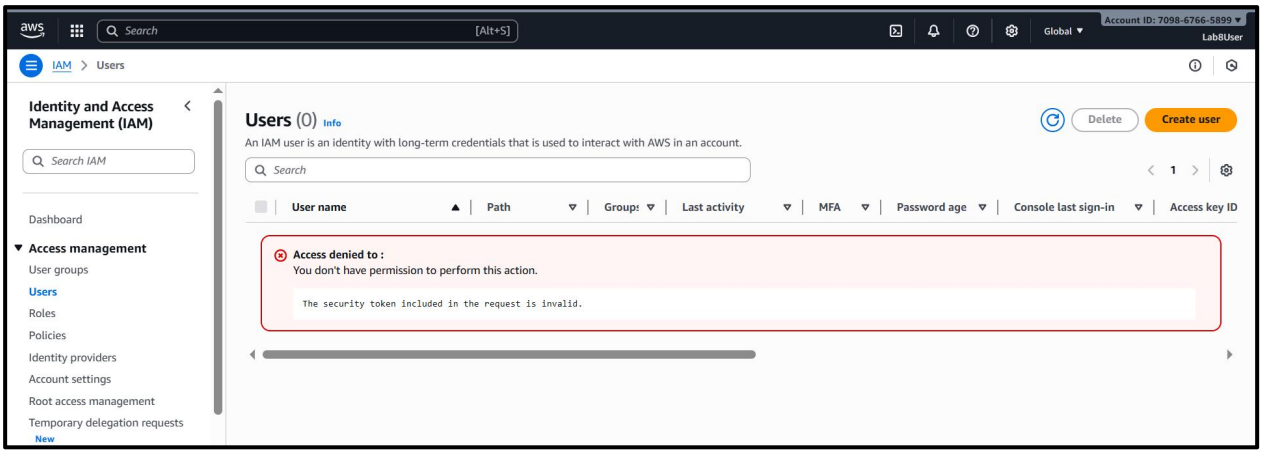
3. **Delete security group Lab8SecurityGroup and key pair Lab8Key from the EC2 console (after instances terminated).**

Save screenshot as: cleanup_delete_security_group_and_keypair.png — deletion confirmation(s) (show key pair list and security group list after deletion).



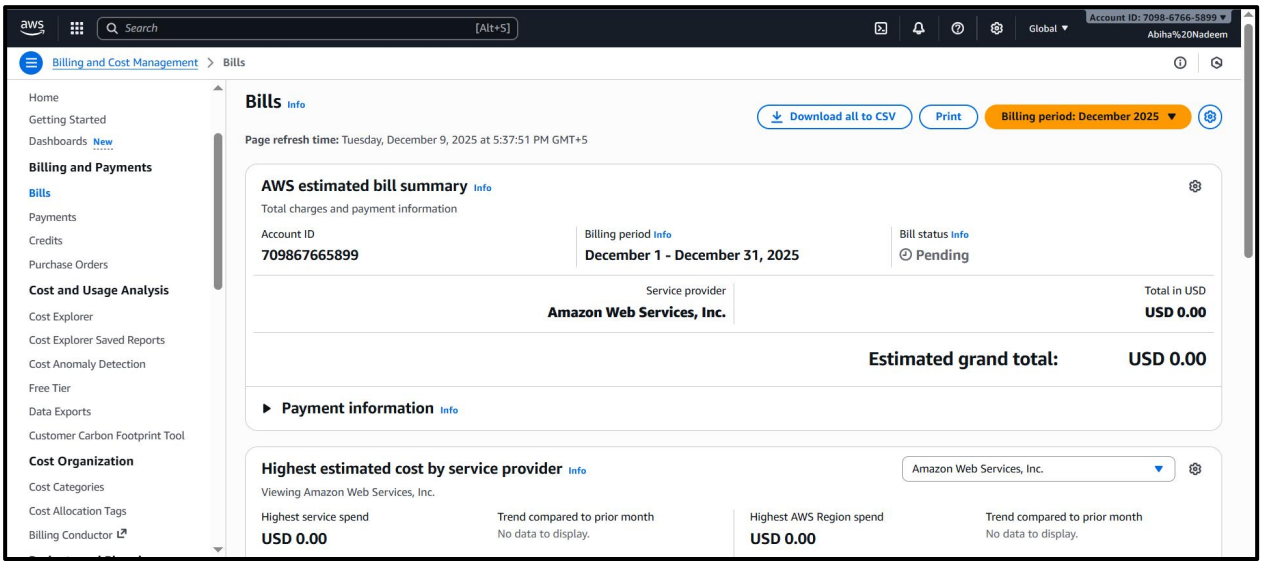
4. **Delete IAM users Lab8User and any access keys.**

Save screenshot as: cleanup_iam_users_deleted.png — IAM Users list showing Lab8User no longer present (or a deletion confirmation).



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

Save screenshot as: cleanup_summary.png — AWS console Billing/Resource Groups showing no active resources or no recent charges (if available).


aws
Search
[Alt+S]
Global
Account ID: 7098-6766-5899
Abiha%20Nadeem
Billing and Cost Management
Bills
Home
Getting Started
Dashboards New
Billing and Payments
Bills
Payments
Credits
Purchase Orders
Cost and Usage Analysis
Cost Explorer
Cost Explorer Saved Reports
Cost Anomaly Detection
Free Tier
Data Exports
Customer Carbon Footprint Tool
Cost Organization
Cost Categories
Cost Allocation Tags
Billing Conductor
Bills Info
Download all to CSV
Print
Billing period: December 2025
Page refresh time: Tuesday, December 9, 2025 at 5:37:51 PM GMT+5
AWS estimated bill summary Info
Total charges and payment information
Account ID
709867665899
Billing period Info
December 1 - December 31, 2025
Bill status Info
Pending
Service provider
Amazon Web Services, Inc.
Total in USD
USD 0.00
Estimated grand total:
USD 0.00
Payment information Info
Highest estimated cost by service provider Info
Amazon Web Services, Inc.
Viewing Amazon Web Services, Inc.
Highest service spend
USD 0.00
Trend compared to prior month
No data to display.
Highest AWS Region spend
USD 0.00
Trend compared to prior month
No data to display.

****