

# CLOUD COMPUTING

## Lab 08

**Name:** Abiha Nadeem

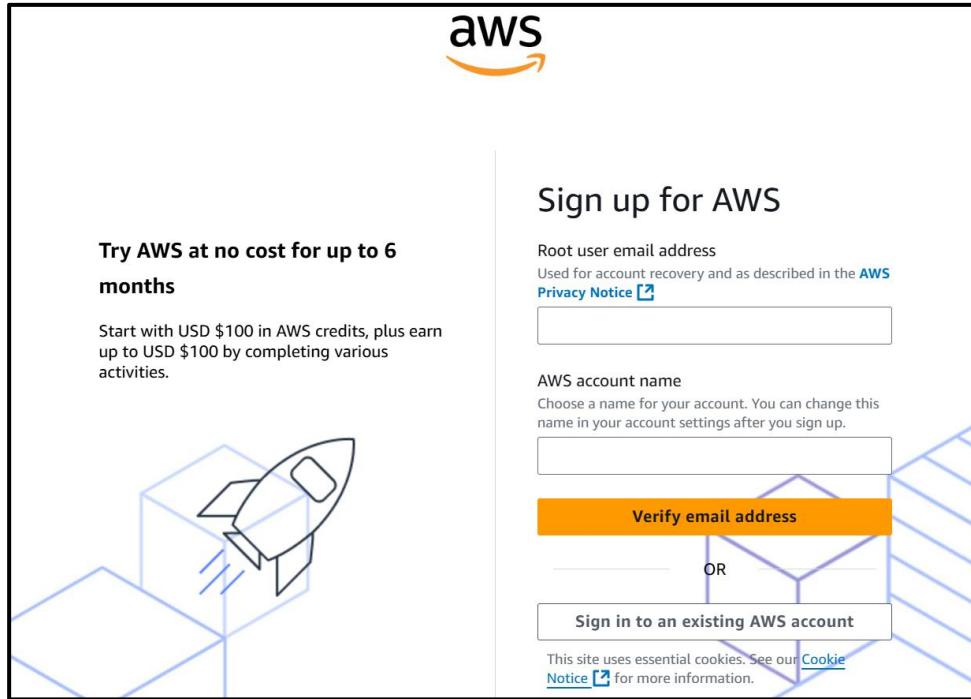
**Roll no:** 2023-BSE-001

**Submitted to:** Engr. Muhammad Shoaib

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

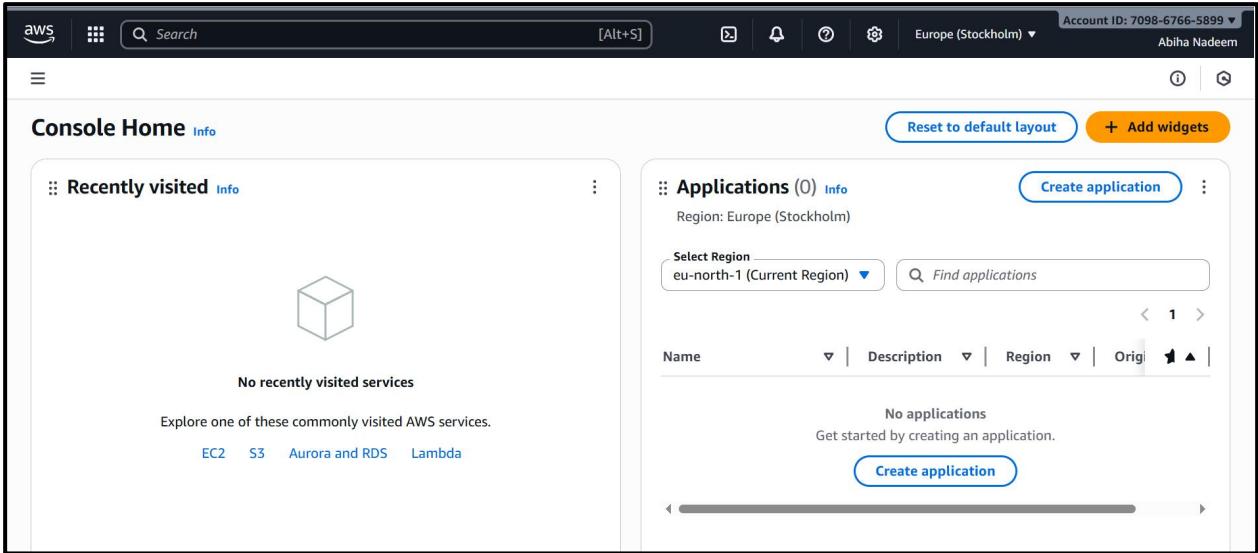
### Task 1 — Create an AWS account and enable UAE (me-central-1)

1. Open your browser and go to: [AWS Signup](#)



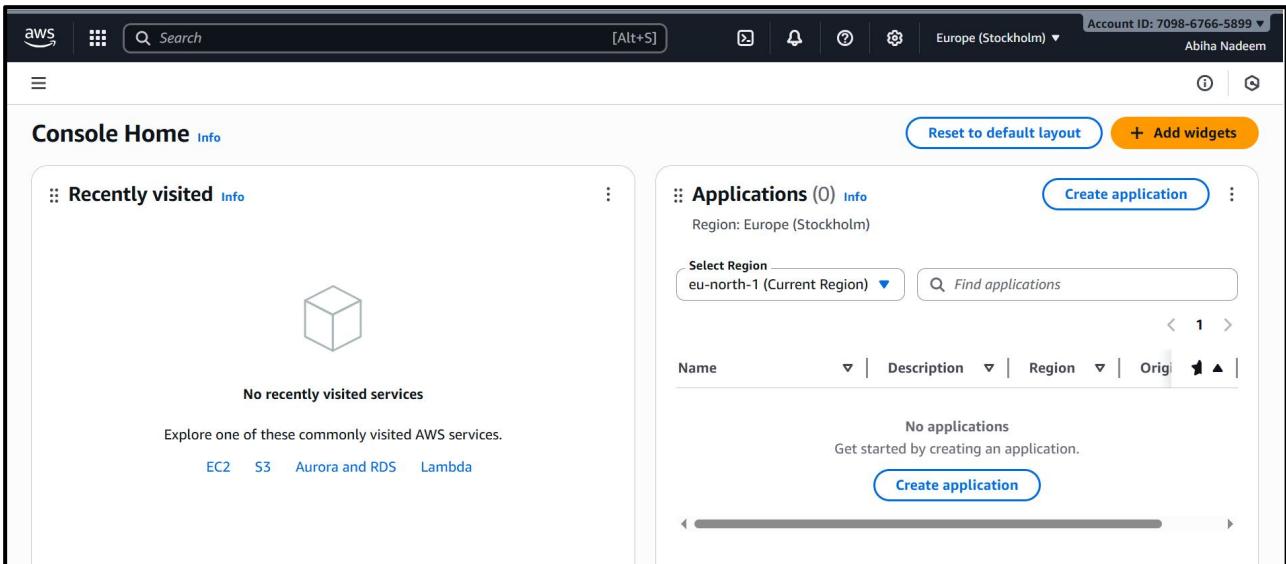
Save screenshot as: task1\_open\_signup\_page.png — browser showing the signup page.

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.

The screenshot shows the AWS Billing and Cost Management console. On the left, there's a sidebar with navigation links like Home, Getting Started, Dashboards, Billing and Payments, Cost and Usage Analysis, and Cost Explorer. The main area is titled "Billing and Cost Management" and shows a list of regions. The "Middle East (UAE)" region is highlighted with a blue border and has a green "Enabled" status indicator next to it. Other regions listed include Asia Pacific (Melbourne), Asia Pacific (Malaysia), Asia Pacific (New Zealand), Asia Pacific (Thailand), Canada (Calgary), Europe (Zurich), Europe (Milan), Europe (Spain), Israel (Tel Aviv), Middle East (Bahrain), and Mexico (Central), all marked as "Disabled".

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

The screenshot shows the AWS root console header. It includes the AWS logo, a search bar with the word "Search", a "[Alt+S]" keyboard shortcut, and various notification and account information icons. The region "Middle East (UAE)" is explicitly mentioned in the header.

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).

The screenshot shows the AWS IAM console landing page. The top navigation bar includes the AWS logo, a search bar with "IAM", a "[Alt+S]" keyboard shortcut, and account information. Below the header, there's a "Services" section with a "Show more" link. A prominent "IAM" service card is displayed, featuring the IAM logo and the text "Manage access to AWS resources". To the right, there's a "Service Health" panel with a link to "View complete service health details". On the left, there's a sidebar with links to "VPC dashboard", "AWS Global VPC", "Services", "Features", and "Documentation".

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

The screenshot shows the AWS Identity and Access Management (IAM) service, specifically the 'Users' section. On the left, there's a sidebar with 'Identity and Access Management (IAM)' at the top, followed by 'Dashboard', 'Access management', and 'User groups'. The main area is titled 'Users (0) Info' and contains a search bar. Below it is a table header with columns: 'User name', 'Path', 'Group:', 'Last activity', 'MFA', and 'Password age'. A message 'No resources to display' is centered below the table.

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

This screenshot shows the 'Specify user details' step of the 'Create user' wizard. It includes a sidebar with steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), Step 4 (Retrieve password). The main form has a 'User details' section with a 'User name' field containing 'Admin'. There are two checked checkboxes: 'Provide user access to the AWS Management Console - optional' and 'Users must create a new password at next sign-in - Recommended'. A note says 'In addition to console access, users with SiginnLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.' A note at the bottom says 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user.' A 'Console password' section is also present.

- Attach policies directly → AdministratorAccess

This screenshot shows the 'Set permissions' step of the 'Create user' wizard. It includes a sidebar with steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), Step 4 (Retrieve password). The main form has a 'Permissions options' section with three choices: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected. Below is a 'Permissions policies (1/1434)' section with a table. The table has a header: 'Policy name', 'Type', and 'Attached entities'. It lists four policies: 'AccessAnalyzerServiceRolePolicy' (AWS managed, 0 attached entities), 'AdministratorAccess' (AWS managed - job function, 0 attached entities), 'AdministratorAccess-Amplify' (AWS managed, 0 attached entities), and 'AdministratorAccess-AWSLambdaElasticBeanstalk' (AWS managed, 0 attached entities). A 'Create policy' button is at the top right of the policy list.

**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	Custom password
		Require password reset Yes	

**Permissions summary**

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

**Tags - optional**

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

**Buttons:** Cancel, Previous, Create user

- Capture the completion screen when user is created:

**Users (1) Info**

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
Admin	/	0	-	-	14 minutes	-	-

**Buttons:** Delete, Create user

Save screenshot as: task2\_admin\_create\_confirmation.png — IAM "Create user" success screen showing Admin (do NOT include password).

- 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).

User name	Console sign-in URL
Admin	https://709867665899.signin.aws.amazon.com/console

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.

The screenshot shows the AWS Console Home page for an Admin user. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 7098-6766-5899, Middle East (UAE)). Below the navigation is a header for 'Console Home' with a 'Reset to default layout' button and an 'Add widgets' button. On the left, there's a 'Recently visited' section with a large cube icon and a message 'No recently visited services'. It lists commonly visited services: EC2, S3, Aurora and RDS, and Lambda. On the right, there's a 'Applications' section with a 'Create application' button, a 'Select Region' dropdown set to 'me-central-1 (Current Region)', and a 'Find applications' search bar. The main content area has columns for Name, Description, Region, and Origin.

## 5. While logged in as Admin, create Lab8User:

- IAM → Users → Create user

The screenshot shows the IAM Users page. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 7098-6766-5899, Global). The left sidebar shows 'Identity and Access Management (IAM)' and 'Access management' with 'User groups'. The main content area shows a table for 'Users (1)'. The table has columns for User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, and Access key ID. One user, 'Admin', is listed with a status of '15 minutes ago' for all metrics.

- Username: Lab8User

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. The left sidebar shows 'Step 1: Specify user details', 'Step 2: Set permissions', 'Step 3: Review and create', and 'Cancel' and 'Next' buttons. The main content area has a 'User details' section with a 'User name' field containing 'Lab8User'. Below it is a note about character restrictions and a checkbox for 'Provide user access to the AWS Management Console - optional'. A note at the bottom explains how to generate programmatic access keys. The 'Next' button is highlighted.

c. Provide user access to the AWS Management Console

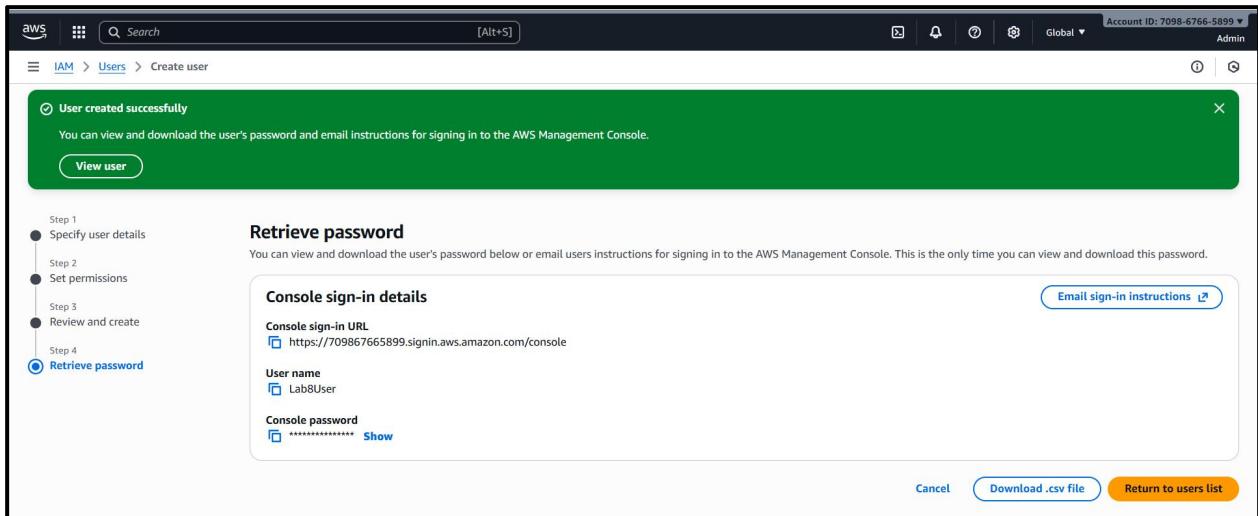
The screenshot shows the 'Specify user details' step of the 'Create user' wizard. On the left, a sidebar lists steps: Step 1 (Specify user details, selected), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area is titled 'Specify user details' and contains a 'User details' section. It shows a 'User name' input field with 'Lab8User' typed in. Below it is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)'. A checked checkbox labeled 'Provide user access to the AWS Management Console - optional' has a note below it: 'In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.'

d. Attach AdministratorAccess policy

The screenshot shows the 'Set permissions' step of the 'Create user' wizard. The sidebar shows Step 1 (Specify user details) is completed, and Step 2 (Set permissions) is selected. The main area is titled 'Set permissions' and contains a 'Permissions options' section with three choices: 'Add user to group', 'Copy permissions', and 'Attach policies directly' (which is selected). Below is a 'Permissions policies' list titled '(1/1434)'. It shows a table with one row selected: 'AdministratorAccess' (AWS managed - job function). There is a 'Create policy' button at the top right of the list.

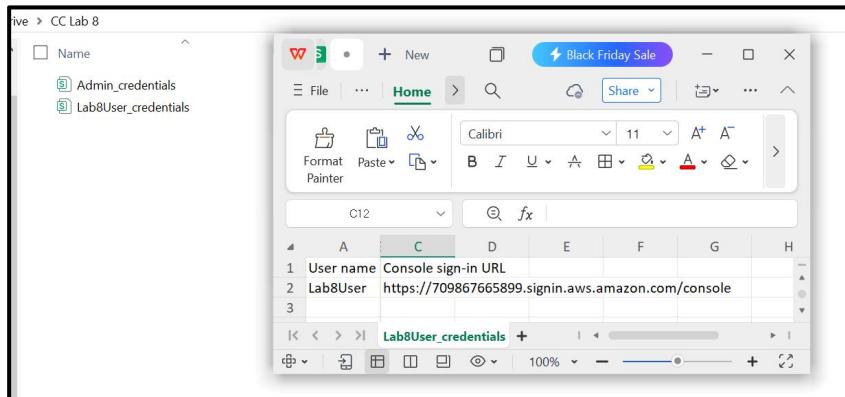
Capture the create-user success screen:

**Save screenshot as: task2\_create\_lab8user\_and\_csv.png — Lab8User create confirmation and CSV download prompt (do NOT include password).**



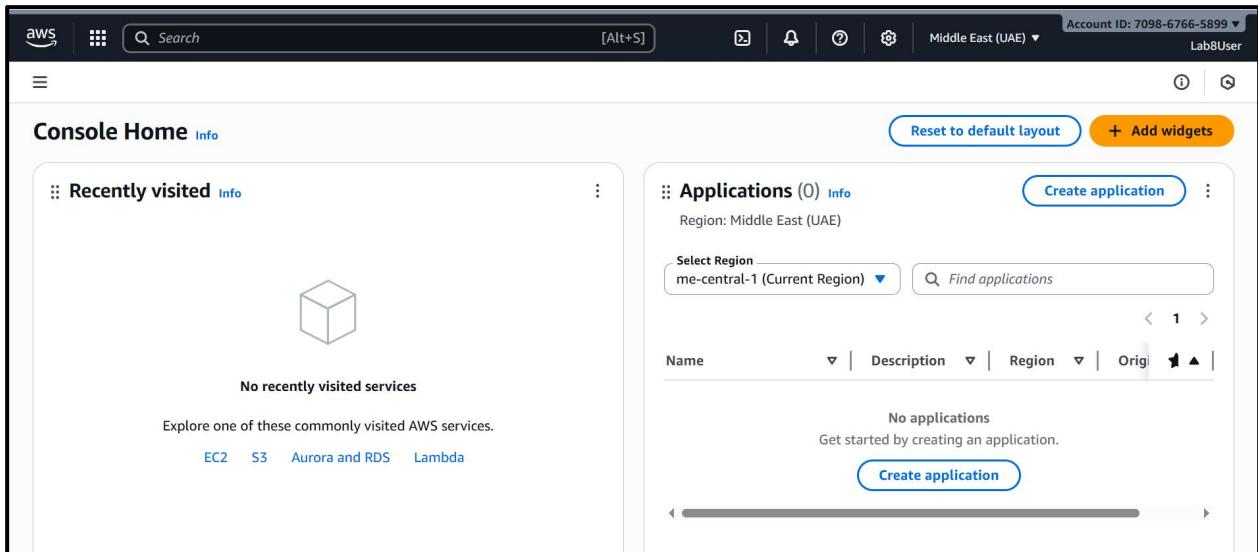
## 6. Download/save the Lab8User CSV on your Windows host (do not show password).

Save screenshot as: task2\_lab8user\_csv\_saved.png — File Explorer showing the Lab8User CSV filename (cropped to exclude sensitive content).



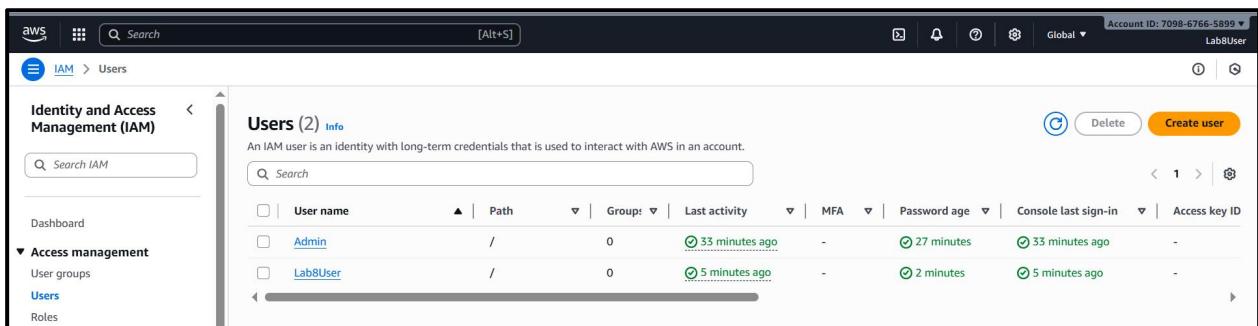
## 7. Logout Admin and login as Lab8User (use the Lab8User signin URL and credentials). Capture after login:

Save screenshot as: task2\_lab8user\_logged\_in.png — Lab8User console home.



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

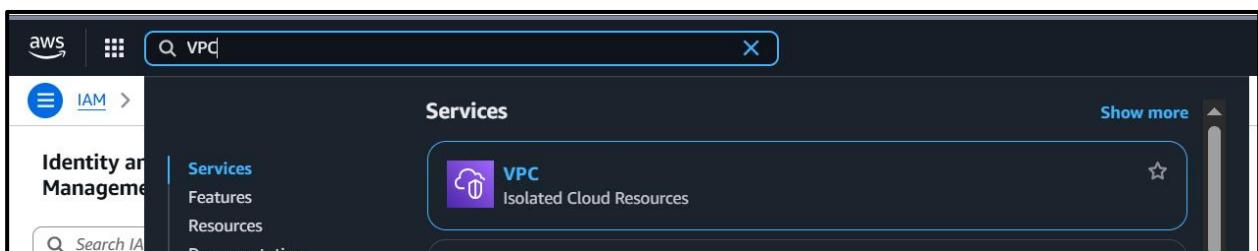
Save screenshot as: task2\_summary.png — IAM Users list showing both Admin and Lab8User present (region me-central-1 visible).



## 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).



The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like 'Your VPCs', 'Subnets', 'Route tables', etc. The main area displays 'Resources by Region' with sections for VPCs, Subnets, Route Tables, Internet Gateways, NAT Gateways, VPC Peering Connections, Network ACLs, and Security Groups. Each section shows the count of resources in the Middle East (UAE) region. There are also buttons for 'Create VPC' and 'Launch EC2 Instances'.

## 2. View VPCs list. Capture:

Save screenshot as: task3\_vpcs\_list.png — VPCs list view (show default VPC if present).

The screenshot shows the 'Your VPCs' list in the AWS VPC dashboard. It lists one VPC named 'vpc-00805ac6ee2802e51'. The table includes columns for Name, VPC ID, State, Encryption control ID, Block Public Access, IPv4 CIDR, IPv6 CIDR, and DHCP option set. Below the table, there's a detailed view of the selected VPC with sections for Details, State, Tenancy, Default VPC, Network Address Usage metrics, Block Public Access, DHCP option set, IPv4 CIDR, IPv6 pool, Main route table, DNS hostnames, and Owner ID.

## 3. View Subnets list. Capture:

Save screenshot as: task3\_subnets\_list.png — Subnets list view (show at least 3 default subnets if present).

**Subnets (1/3) Info**

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR	IPv6 CIDR
subnet-04f01f01921e6b364	subnet-04f01f01921e6b364	Available	vpc-00805ac6ee2802e51	Off	172.31.32.0/20	-
subnet-054592f8971610b70	subnet-054592f8971610b70	Available	vpc-00805ac6ee2802e51	Off	172.31.16.0/20	-
subnet-0917b1a726d6a79b4	subnet-0917b1a726d6a79b4	Available	vpc-00805ac6ee2802e51	Off	172.31.0.0/20	-

**subnet-04f01f01921e6b364**

**Details**

Subnet ID	subnet-04f01f01921e6b364	Subnet ARN	arn:aws:ec2:me-central-1:709867665899:su:bnet/subnet-04f01f01921e6b364
IPv4 CIDR	172.31.32.0/20	State	Available
Availability Zone	mec1-a21 (me-central-1a)	IPv6 CIDR	-
VPC	vpc-00805ac6ee2802e51	Route table	rtb-045a9e373f5200709
Default subnet	Yes	Auto-assign IPv6 address	No
IPv4 CIDR reservations	-	IPv6-only	No
Resource name DNS A record	Disabled	DNS64	Disabled
Resource name DNS AAAA record		Owner	709867665899

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

Save screenshot as: task3\_route\_tables\_list.png — Route Tables list view.

**Route tables (1/1) Info**

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
rtb-045a9e373f5200709	rtb-045a9e373f5200709	-	-	Yes	vpc-00805ac6ee2802e51

**rtb-045a9e373f5200709**

**Details**

Route table ID	rtb-045a9e373f5200709	Main	Yes
VPC	vpc-00805ac6ee2802e51	Owner ID	709867665899
Explicit subnet associations			
Edge associations			

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

Save screenshot as: task3\_network\_acls\_list.png — Network ACLs list view.

The screenshot shows the AWS VPC Network ACLs page. On the left, there's a navigation sidebar with options like DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections, Route servers, Security (Network ACLs), DNS firewall, Network Firewall, and Firewalls. The main area displays 'Network ACLs (1/1) Info' with a table showing one entry: Name 'acl-0aaa055fc79a8467', Network ACL ID 'acl-0aaa055fc79a8467', Associated with '3 Subnets', Default 'Yes', and VPC ID 'vpc-00805ac6ee2802e51'. Below this, a detailed view for 'acl-0aaa055fc79a8467' is shown with tabs for Details, Inbound rules, Outbound rules, Subnet associations, and Tags. The Details tab shows Network ACL ID, Associated with, Default, and VPC ID.

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

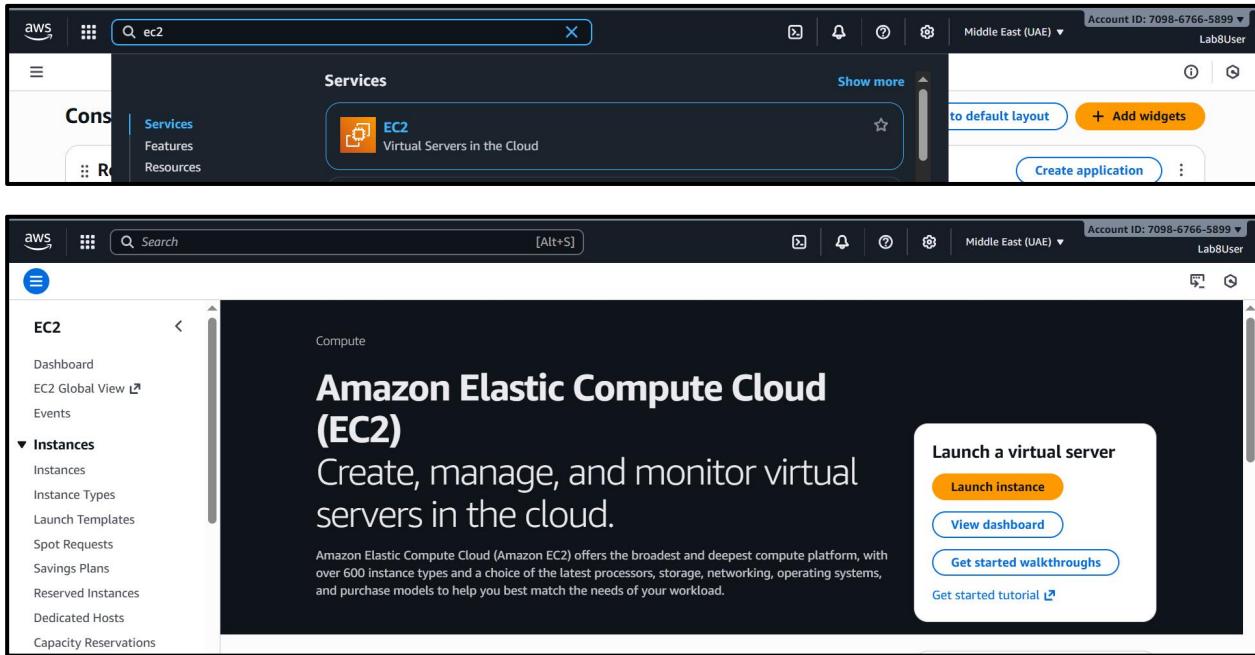
Save screenshot as: task3\_summary.png — a single screenshot showing the VPC console left navigation and counts or multiple open tabs/windows tiled to show each resource's list (region me-central-1 visible).

The screenshot shows the AWS VPC Management console with multiple tabs open simultaneously. The tabs include 'Your VPCs' (VPC dashboard), 'Route tables', 'Network ACLs', and 'Subnets'. The 'Your VPCs' tab shows a 'VPC dashboard' with an 'AWS Global View' and a 'Filter by VPC' dropdown. The 'Route tables' tab shows a list with one entry: 'rtb-045a9e373'. The 'Network ACLs' tab shows a list with one entry: 'acl-0aaa055fc79a8467'. The 'Subnets' tab shows a list with one entry: 'aws'. This demonstrates how multiple resources can be managed and viewed simultaneously within a single browser window.

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

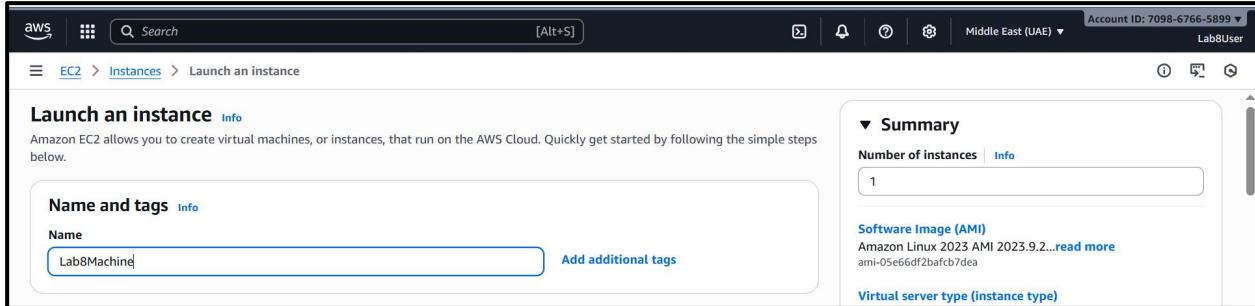
### 1. Open EC2 Console (Alt+S → "EC2") (me-central-1).

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

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

**Quick Start**

Search our full catalog including 1000s of application and OS images

Amazon Linux Ubuntu Windows Red Hat SUSE Linux Debian

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Amazon Linux 2023 kernel-6.1 AMI  
ami-05e66df2bafc7dea (64-bit (x86), uefi-preferred) / ami-0d90f860e9d3b1c73 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.9.20251117.1 x86\_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username	Verified provider
64-bit (x86)	uefi-preferred	ami-05e66df2bafc7dea	2025-11-17	ec2-user	Verified provider

- **Instance type: t2.micro**

**Instance type** [Info](#) | [Get advice](#)

t3.micro  
Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand Ubuntu Pro base pricing: 0.016 USD per Hour  
On-Demand Linux base pricing: 0.0125 USD per Hour On-Demand RHEL base pricing: 0.0413 USD per Hour  
On-Demand SUSE base pricing: 0.0125 USD per Hour On-Demand Windows base pricing: 0.0217 USD per Hour

**Additional costs apply for AMIs with pre-installed software**

**Summary**  
Number of instances: 1

All generations  
[Compare instance types](#)

**Software Image (AMI)**  
Amazon Linux 2 with SQL Server...[read more](#)  
ami-0dfdd15056548121c

**Virtual server type (instance type)**

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

**▼ Network settings** [Info](#)

**VPC - required** | [Info](#)

vpc-00805ac6ee2802e51 (default) [▼](#) [Create new VPC](#)

**Subnet** | [Info](#)

No preference [▼](#) [Create new subnet](#) [Edit](#)

**Availability Zone** | [Info](#)

No preference [▼](#) [Enable additional zones](#) [Edit](#)

**Auto-assign public IP** | [Info](#)

Enable [▼](#)

**Firewall (security groups)** | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

**Security group name - required**

Lab8SecurityGroup

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_~-./()#,@[]+=&;!\$\*

**Description - required** | [Info](#)

launch-wizard-1 created 2025-12-08T13:47:36.016Z

**Inbound Security Group Rules**

▼ Security group rule 1 (TCP, 22, 154.57.195.130/32) [Remove](#)

Type	Protocol	Port range
ssh	TCP	22
Source type	Name	Description - optional
My IP	Add CIDR, prefix list or security group	e.g. SSH for admin desktop
	154.57.195.130/32 <a href="#">X</a>	

[Add security group rule](#)

- **Storage: default**

**▼ Configure storage** [Info](#) [Advanced](#)

1x  GiB  Root volume, Not encrypted

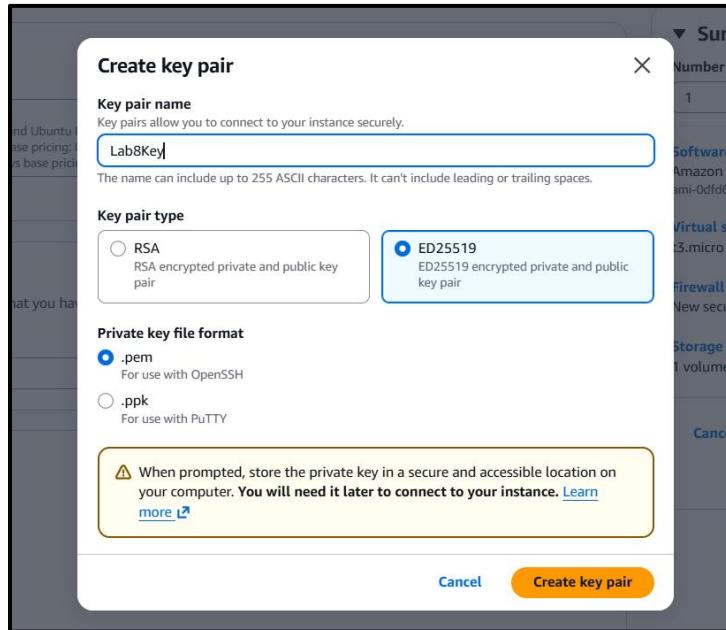
[Add new volume](#)

Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

- **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.



**▼ Key pair (login) Info**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

Create new 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).**

```
ec2-user@ip-172-31-7-139:~  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS C:\Users\MOHSIN> ssh -i "C:\Users\MOHSIN\Downloads\Lab8Key.pem" ec2-user@51.112.189.53  
, #_  
~\ ####_ Amazon Linux 2023  
~~ \#####\  
~~ \###|  
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023  
~~ V~'__->  
~~ /  
~~ . / /  
~/m/'  
[ec2-user@ip-172-31-7-139 ~]$
```

#### 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
Last metadata expiration check: 0:00:01 ago on Mon Dec 8 14:31:18 2025.          277 kB/s | 29 kB   00:00
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.

=====
Repository           Size
=====
dockers             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            491 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
=====
Architecture       Version
=====
amazonlinux          Installing
=====
Transaction Summary
=====
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): libnftnl-1.0.1-19.amzn2023.0.2.x86_64.rpm            1.3 MB/s | 36 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.amzn2023.0.3.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   : libnftnl-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
  docker-25.0.13-1.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.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 Current
          Dload Upload Total Spent Left Speed
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.

```
ec2-user@ip-172-31-7-139:~  
image  gitea/gitea:latest  
container_name  gitea  
environment  
  - DB_TYPE=postgres  
  - DB_HOST=db:5432  
  - DB_NAME=gitea  
  - DB_USER=gitea  
  - DB_PASSWORD=gitea  
restart  always  
volumes  
  - gitea:/data  
ports  
  - 3000:3000  
extra_hosts  
  - "www.jenkins.com:host-gateway"  
networks  
  - webnet  
db  
image  postgres:alpine  
container_name  gitea_db  
environment  
  - POSTGRES_USER=gitea  
  - POSTGRES_PASSWORD=gitea  
  - POSTGRES_DB=gitea  
restart  always  
volumes  
  - gitea_postgres:/var/lib/postgresql/data  
expose  
  - 5432  
networks  
  - webnet  
  
volumes  
  gitea_postgres  
    name  gitea_postgres  
gitea  
  name  gitea  
  
networks  
  webnet  
    name  webnet  
  external: true  
-  
-- INSERT --
```

**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)
```

```
ec2-user@ip-172-31-7-139:~$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-7-139 ~]$ sudo usermod -aG docker $USER
[ec2-user@ip-172-31-7-139 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-7-139 ~]$ exit
logout
Connection to 51.112.189.53 closed.
PS C:\Users\MOHSIN> ssh -i "C:\Users\MOHSIN\Downloads\Lab8Key.pem" ec2-user@51.112.189.53
,
#_
~\_ #####
~~ \####\ Amazon Linux 2023
~~ \###]
~~   \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~   V~ .->
~~   /
~~ .-
~~ / /
~/m/
Last login: Mon Dec  8 19:16:29 2025 from 154.57.195.130
[ec2-user@ip-172-31-7-139 ~]$ groups
ec2-user adm wheel systemd-journal docker
[ec2-user@ip-172-31-7-139 ~]$
```

**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.**

```

[ec2-user@ip-172-31-7-139:~]
[ec2-user@ip-172-31-7-139 ~]$ docker compose up -d
[+] up 23/23
  ⚡ Image postgres:alpine Pulled
  ⚡ Image gitea/gitea:latest Pulled
  ⚡ Network webnet Created
  ⚡ Volume gitea Created
  ⚡ Volume gitea_postgres Created
  ⚡ Container gitea_db Created
  ⚡ Container gitea Created
[ec2-user@ip-172-31-7-139 ~]$

```

**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.

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0450c2f515ecb02bc	IPv4	SSH	TCP	22	154.57.195.130/32
-	sgr-003d0eae92503e63	IPv4	Custom TCP	TCP	3000	0.0.0.0/0

**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.

**Initial Configuration**

If you run Gitea inside Docker, please read the [documentation](#) before changing any settings.

**Database Settings**

Gitea requires MySQL, PostgreSQL, MSSQL, SQLite3 or TiDB (MySQL protocol).

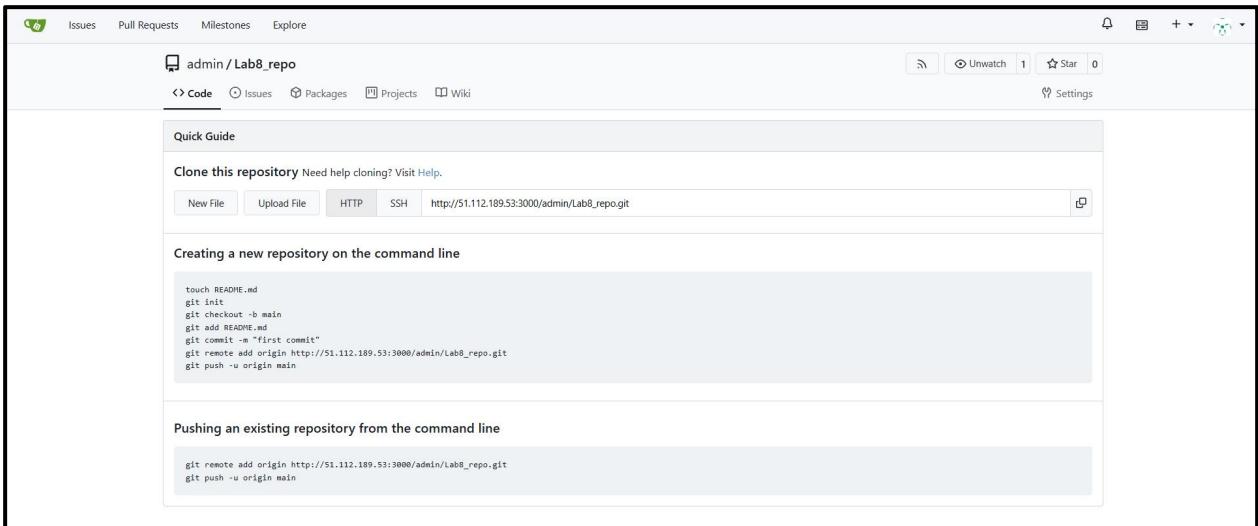
Database Type *	PostgreSQL
Host *	db:5432
Username *	gitea
Password *	*****
Database Name *	gitea
SSL *	Disable
Schema	Leave blank for database default ("public").

**General Settings**

Site Title *	Gitea: Git with a cup of tea
You can enter your company name here.	
Repository Root Path *	/data/git/repositories
Remote Git repositories will be saved to this directory.	

**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.

The tiled screenshot displays three windows side-by-side:

- Left Window (EC2 Instances):** Shows the 'Instances' list with one instance named 'Lab8Machine' (i-0f9b05cf9a5ae90d8) running. It provides details like Public IPv4 address (51.112.189.53), Private IPv4 address (172.31.7.139), and Public DNS name (ec2-51-112-189-53.me-central-1.compute.amazonaws.com).
- Middle Window (Security Groups):** Shows the 'Security Groups' list with two groups: 'default' and 'Lab8SecurityGroup'. The 'Lab8SecurityGroup' is selected, showing its details: Security group name (Lab8SecurityGroup), Security group ID (sg-0351271d21ec05145), Description (launch-wizard-1 created 2025-12-08T13:47:36.016Z), VPC ID (vpc-00805ac6ee2802e51), Owner (709867665899), Inbound rules count (2 Permission entries), and Outbound rules count (0).
- Right Window (Gitea UI):** Shows the Gitea repository 'admin / Lab8\_repo'. It includes sections for 'Quick Guide' (Clone this repository, Creating a new repository on the command line, Pushing an existing repository from the command line), a file browser, and a commit history.

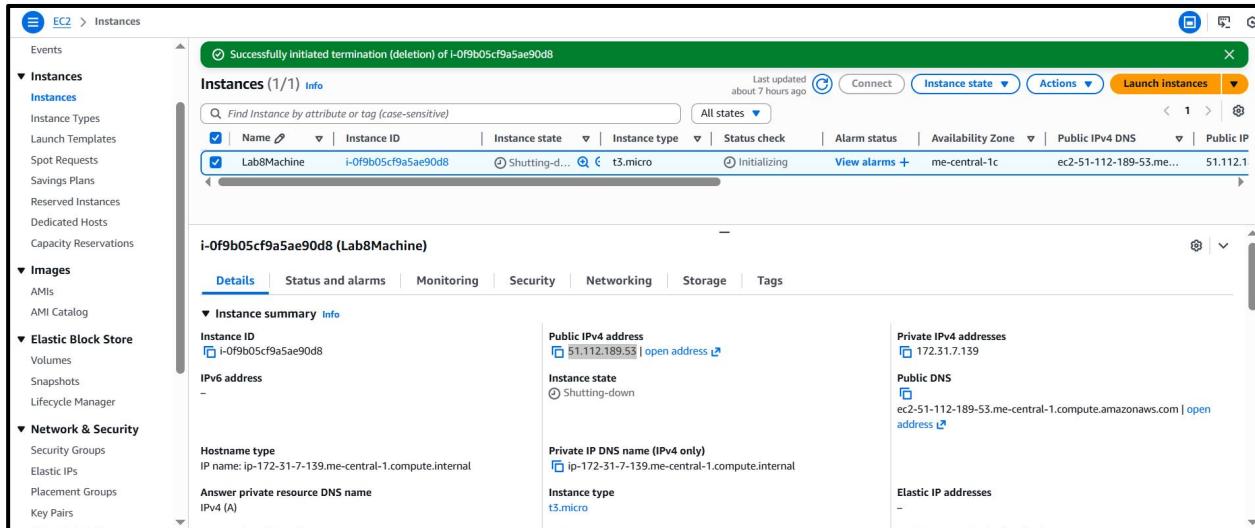
## Cleanup — Remove resources to avoid charges

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

Cleanup steps and required screenshots:

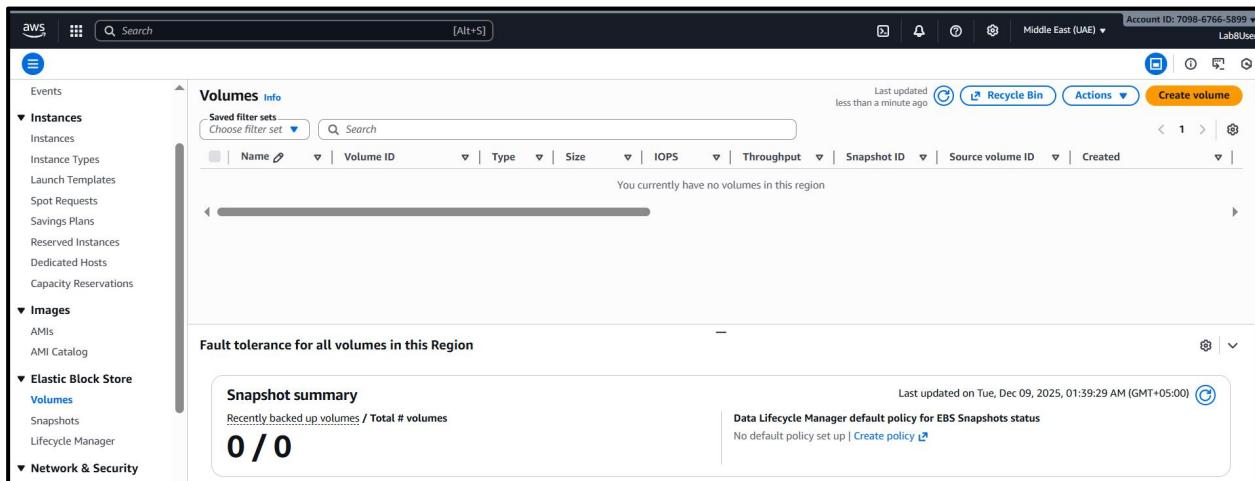
1. Terminate the EC2 instance Lab8Machine.

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).**

The screenshot shows two separate browser tabs. The top tab is titled 'Security groups | EC2 | me-cent' and displays the 'Security Groups' page with one entry: 'sg-0351271d21ec05145' (Name), 'default' (Security group name), and 'vpc-00805ac6ee2802e51' (VPC ID). A delete icon is visible next to the entry. The bottom tab is titled 'Key pairs | EC2 | me-cent' and displays the 'Key pairs' page with a green notification bar stating 'Successfully deleted 1 key pairs'. Both tabs have the URL 'me-central-1.console.aws.amazon.com/ec2/home?region=me-central-1#SecurityGroups:' and 'me-central-1.console.aws.amazon.com/ec2/home?region=me-central-1#KeyPairs:' respectively.

#### 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).**

The screenshot shows the 'Users' page under the 'Identity and Access Management (IAM)' section. The table header includes columns for 'User name', 'Path', 'Groups', 'Last activity', 'MFA', 'Password age', 'Console last sign-in', and 'Access key ID'. A red box highlights a message box with the text: 'Access denied to : You don't have permission to perform this action. The security token included in the request is invalid.' This indicates that the user account has been deleted.

#### 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).**

Screenshot of the AWS Billing and Cost Management Bills page.

**Bills** Info

Page refresh time: Tuesday, December 9, 2025 at 5:37:51 PM GMT+5

[Download all to CSV](#) [Print](#) **Billing period: December 2025**

**AWS estimated bill summary** Info

Total charges and payment information

Account ID <b>709867665899</b>	Billing period <small>Info</small> <b>December 1 - December 31, 2025</b>	Bill status <small>Info</small> <b>Pending</b>
Service provider <b>Amazon Web Services, Inc.</b>		Total in USD <b>USD 0.00</b>
<b>Estimated grand total:</b> <b>USD 0.00</b>		

**Payment information** Info

**Highest estimated cost by service provider** Info

Viewing Amazon Web Services, Inc.

Highest service spend <b>USD 0.00</b>	Trend compared to prior month No data to display.	Highest AWS Region spend <b>USD 0.00</b>	Trend compared to prior month No data to display.
--	--	---	--

\*\*\*\*\*