

# **AWS Cloud & DevOps**

## **Ebubechukwu Azozie**

### **EC2 HOMEWORK :**

#### **Deadline: Saturday, July 16th by 10AM ET**

Prerequisite:

---

#### **SETUP YOUR BASE INFRASTRUCTURE AS FOLLOWS:**

Create a VPC named “awesome\_vpc” with the primary CIDR “10.0.0.0/16”

Create an Internet Gateway “awesome\_igw” and attach it to the VPC.

Create a Public subnet “awesome\_pub\_sub1” – “10.0.1.0/24” in US-EAST-1A

Create a Public subnet “awesome\_pub\_sub2” – “10.0.2.0/24” in US-EAST-1B

Create a Private subnet “awesome\_priv\_sub1” – “10.0.3.0/24” in US-EAST-1A

Create a Private subnet “awesome\_priv\_sub2” – “10.0.4.0/24” in US-EAST-1B

NOTE: We recommend to keep that base infrastructure in your account as you will be using it in the future to deploy resources

from other homeworks. YOU MAY NOT INCUR CHARGES FOR THE VPC AND SUBNETS.

#### **SETUP YOUR SSH CLIENT:**

Download and install a SSH client of your choice. We recommend MobaXterm for Windows users and iTerm2 for Mac users.

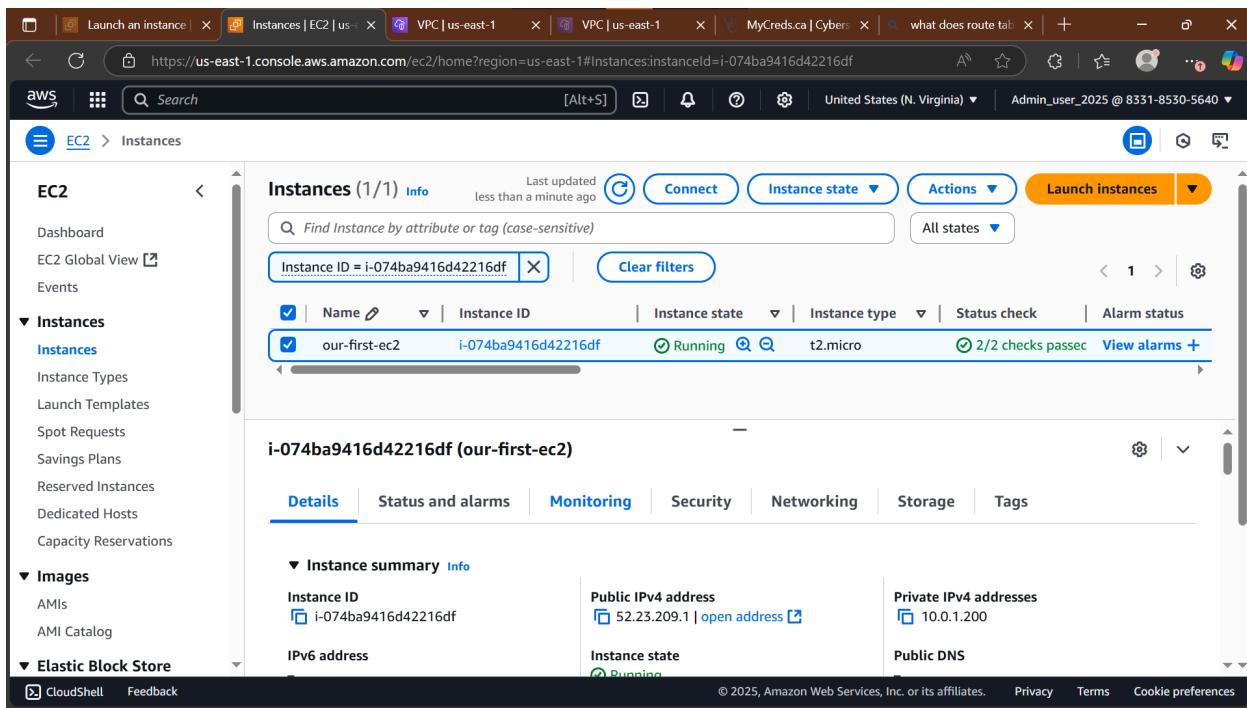
# Homework 1:

Launch an Amazon Linux instance in awesome\_pub\_sub1 subnet as follows:

- AMI: Amazon linux 2
- Specs: t2.micro
- Storage: 8GB
- Attach a Security Group call web-sg allowing SSH traffic to your IP address and HTTP traffic to the outside world.
- Tag the instance our-first-ec2
- KeyPair: awesome-key-east1

## SUBMISSION:

Attach a screenshot of your EC2 Dashboard with the running instance in a word/pdf document and share it in your cluster.



Attach a screenshot of your security group displaying all the rules in the same word/pdf document and share it in your cluster.

The screenshot shows the AWS VPC Security Groups console. A green success message at the top states: "Security group (sg-03bc38393a2ca5d1c | web-sg) was created successfully". Below this, the "Inbound rules" section displays two rules:

Security group rule ID	IP version	Type	Protocol	Port range
sgr-04974639729cb1e40	IPv4	SSH	TCP	22
sgr-089d9cd02b8d55bd6	IPv4	HTTP	TCP	80

## Homework 2:

Log into the EC2 (our-first-ec2) created in homework1 using SSH.

Run the command hostnamectl on your ec2 server.

SUBMISSION:  
Attach a screenshot of the output of the hostnamectl command and share it in your cluster.

The screenshot shows a MobaXterm window titled "52.23.209.1 (ec2-user)". The terminal session displays the output of the hostnamectl command. The output includes the static hostname (ip-10-0-1-200.ec2.internal), icon name (computer-vm), chassis (vm), machine ID, boot ID, virtualization (xen), operating system (Amazon Linux 2023.7.20250512), kernel version (Linux 6.1.134-152.225.amzn2023.x86\_64), architecture (x86-64), hardware vendor (Xen), hardware model (HVM domU), and firmware version (4.11.amazon). The terminal window also shows the user's path (~) and the prompt [ec2-user@ip-10-0-1-200 ~]\$.

```
[ec2-user@ip-10-0-1-200 ~]$ hostnamectl
Static hostname: ip-10-0-1-200.ec2.internal
Icon name: computer-vm
Chassis: vm
Machine ID: ec2ef281e4869be194c629e082e1e931
Boot ID: 9061d57d0be14a1896968e6333978150
Virtualization: xen
Operating System: Amazon Linux 2023.7.20250512
    CPE OS Name: cpe:2.3:o:amazon:amazon_linux:2023
                Kernel: Linux 6.1.134-152.225.amzn2023.x86_64
Architecture: x86-64
Hardware Vendor: Xen
Hardware Model: HVM domU
Firmware Version: 4.11.amazon
[ec2-user@ip-10-0-1-200 ~]$
```

## Homework 3:

Launch an EC2 Linux instance and connect to it via SSH.

Manually deploy Apache HTTP server on a Linux EC2 instance using the script below.

Display the message “webpage built by student name” on a webpage.

SCRIPT:

```
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
sudo echo "webpage built by Ebubechukwu Azozie" > /var/www/html/index.html
```

SUBMISSION: Replace <student name> in the message above by your name, and post a screenshot of the webpage with the message. Attach your screenshot(s) to a Word or pdf document.

The screenshot shows a MobaXterm window titled '13.220.209.31 (ec2-user)'. The terminal session displays the execution of a script to install Apache HTTP server and its configuration. The output shows the download of several packages from the Amazon Linux 2023 repository, including 'apr-1.7.5-1.amzn2023.0.4.x86\_64' and 'httpd-2.4.62-1.amzn2023.x86\_64'. It also shows the creation of symlinks for the service and the execution of commands to start and enable the service, followed by the insertion of a custom message into the index.html file.

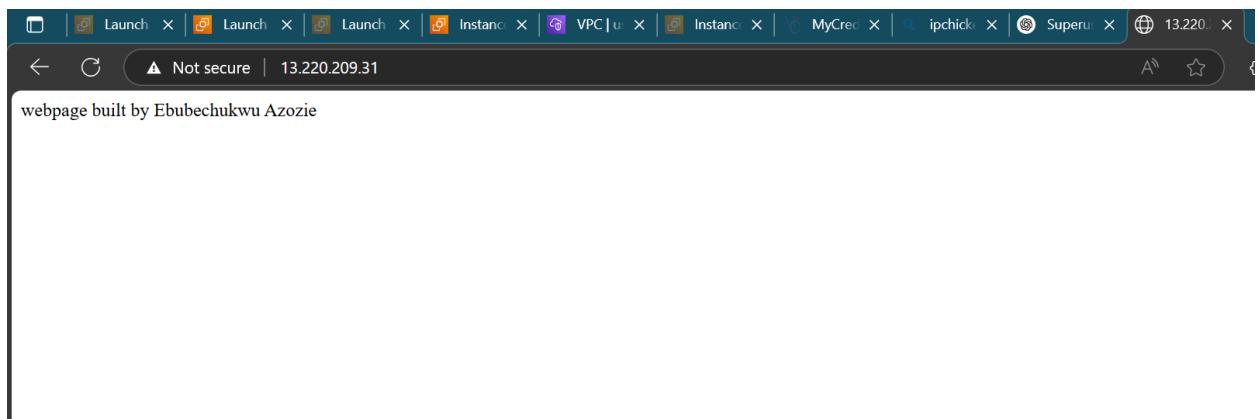
```
Running scriptlet: httpd-2.4.62-1.amzn2023.x86_64
Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
Verifying : httpd-2.4.62-1.amzn2023.x86_64
Verifying : httpd-core-2.4.62-1.amzn2023.x86_64
Verifying : httpd-filesystem-2.4.62-1.amzn2023.noarch
Verifying : httpd-tools-2.4.62-1.amzn2023.x86_64
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64
Verifying : mod_lua-2.4.62-1.amzn2023.x86_64

Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  httpd-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64
  mod_lua-2.4.62-1.amzn2023.x86_64

apr-util-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-core-2.4.62-1.amzn2023.x86_64
httpd-tools-2.4.62-1.amzn2023.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@ip-10-0-1-111 ec2-user]# yum install -y httpd
Last metadata expiration check: 0:01:47 ago on Mon Jun  2 12:11:23 2025.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-1-111 ec2-user]# sudo systemctl start httpd
[root@ip-10-0-1-111 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-1-111 ec2-user]# sudo echo "Webpage built by Ebubechukwu Azozie" >/var/www/html/index.html
[root@ip-10-0-1-111 ec2-user]#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



## Homework 4:

- Work with a cluster member

Create an AMI based on our previous EC2 Apache HTTP Server.

Tag AMI with value “homework4-apache-ami”.

Share the AMI with one member of your cluster.

Launch an EC2 instance using the AMI created in the previous step.

SUBMISSION: Post a screenshot of:

The Newly created AMI.

The AMI in the destination account (Work with your peer to get a screenshot from their account).

Attach your screenshot(s) to a Word or pdf document.

The screenshot shows the AWS EC2 console interface. The left sidebar is collapsed, and the main area displays the 'Amazon Machine Images (AMIs)' section. At the top, there are buttons for 'Recycle Bin', 'EC2 Image Builder', and 'Actions'. A search bar is present, and the table below lists the AMI details. One row is selected, showing the following information:

Name	AMI ID	Source	Owner
homework4-a...	ami-014e03092949c203b	833185305640/homework4-apache-ami	833185305640

Below the table, a detailed view for the selected AMI is shown with tabs for 'Details', 'Permissions', 'Storage', and 'Tags'. The 'Details' tab displays the following data:

AMI ID	Image type	Platform details	Root device type
ami-014e03092949c203b	machine	Linux/UNIX	EBS

Other tabs show 'Permissions', 'Storage', and 'Tags'.

Newly created AMI

The screenshot shows the AWS EC2 AMI permissions configuration page. It includes sections for 'AMI availability' (set to 'Private - (current setting)'), 'Shared accounts (1)', and 'Shared organizations/OUss (0)'. In the 'Shared accounts' section, there is one account listed: '317767385440'. In the 'Shared organizations/OUss' section, there are no entries.

Shared AMI with my peer, Fidelis Nwudo

The screenshot shows the AWS EC2 AMI details page for the AMI ID: ami-014e03092949c203b. The table lists two AMIs: 'homework4-apache-ami' and 'destinationacct\_ami'. The 'destinationacct\_ami' row is selected. The details page shows the following information for the selected AMI:

Details	Storage	Tags	
AMI ID: ami-014e03092949c203b AMI name: homework4-apache-ami Root device name: /dev/xvda Boot mode: uefi-preferred Description: homework AMI to be shared with a cluster member Source AMI ID: ami-0953476d60561c955	Image type: machine Owner account ID: 833185305640 Status: Available Block devices: /dev/xvda=snap-004b4ec567ef925a5:8:false:gp3	Platform details: Linux/UNIX Architecture: x86_64 Source: 833185305640/homework4-apache-ami Creation date: 2025-06-02T12:57:40.000Z RAM disk ID: - Deregistration protection: Disabled	Root device type: EBS Usage operation: RunInstances Virtualization type: hvm Kernel ID: - Deprecation time: - Allowed image: -

Shared AMI received by my peer, Fidelis Nwudo

NB: make sure you peer launch a new EC2 instance using the shared AMI

**Instances (1/1) Info**

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Name: FidelisNwudo- Instance ID: i-01cfdd11f04b9a814 Instance state: Running Instance type: t2.micro Status check: 2/2 checks passed View alarms +

**i-01cfdd11f04b9a814 (FidelisNwudo-shared-ami-instance)**

Details Status and alarms Monitoring Security Networking Storage Tags

**Instance summary**

Instance ID: i-01cfdd11f04b9a814	Public IPv4 address: 13.218.177.164   open address	Private IPv4 addresses: 10.0.1.13
IPv6 address: -	Instance state: Running	Public DNS: -

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Instance launched using AMI shared by my peer - Fidelis Nwudo

**Instances (1/1) Info**

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive)

All filters

Instance ID: i-08bec1f4385524788

**i-08bec1f4385524788 (sharedami\_ec2)**

Details Status and alarms Monitoring Security Networking Storage Tags

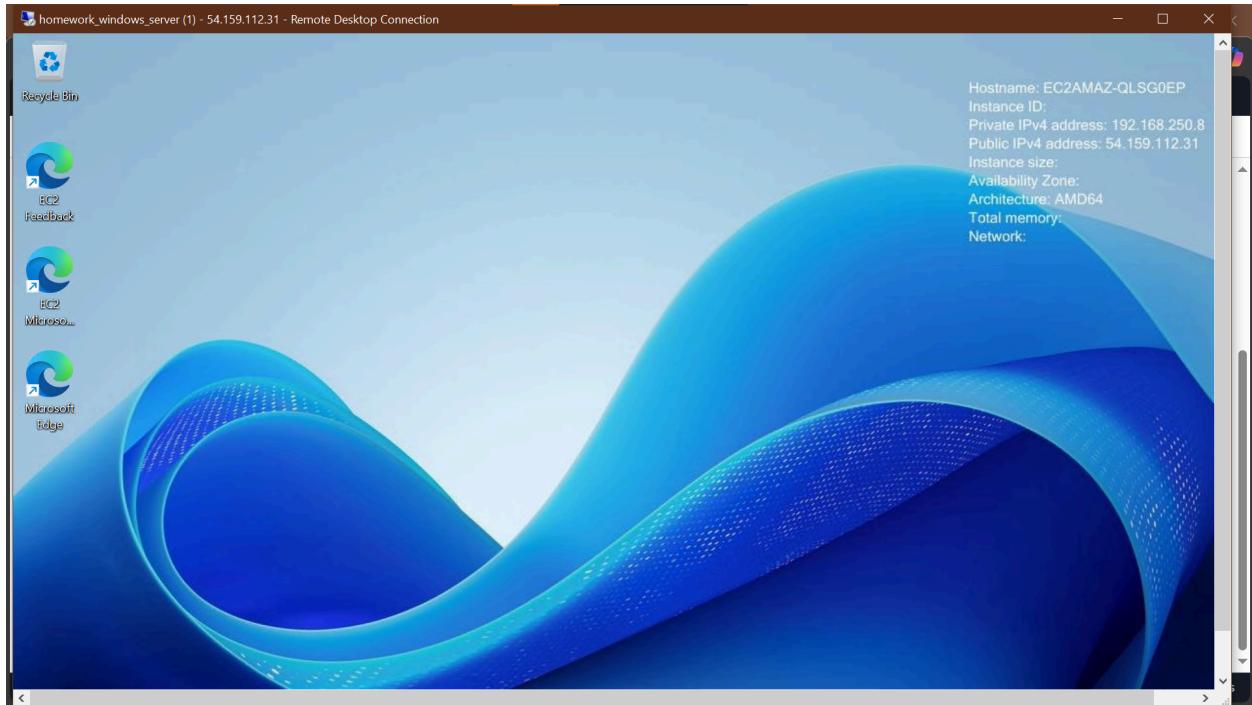
**Instance summary**

Instance ID: i-08bec1f4385524788	Public IPv4 address: 54.224.179.109   open address	Private IPv4 addresses: 10.0.1.69
IPv6 address: -	Instance state: Running	Public DNS: -
Hostname type: IP name: ip-10-0-1-69.ec2.internal	Private IP DNS name (IPv4 only): ip-10-0-1-69.ec2.internal	Elastic IP addresses: -
Answer private resource DNS name: -	Instance type: t2.micro	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.   Learn more
Auto-assigned IP address: 54.224.179.109 [Public IP]	VPC ID: vpc-01c19dd73679be2d0 (awesome_vpc)	Auto Scaling Group name: -
IAM Role: -	Subnet ID: subnet-07af715769b426119 (awesome_pub_sub1)	Managed: false
IMDSv2: Required	Instance ARN: arn:aws:ec2:us-east-1:317767385440:instance/i-08bec1f4385524788	Activate Windows: Go to Settings to activate Windows.

Instance launched by my peer - Fidelis Nwudo using my AMI

## Homework 5:

- Launch a windows server 2019 EC2 instance and connect to it using a RDP client of your choice.  
SUBMISSION: Post a screenshot of the Windows Server landing page after successful authentication.  
Attach your screenshot to a Word or pdf document.



## Homework 6: USER DATA

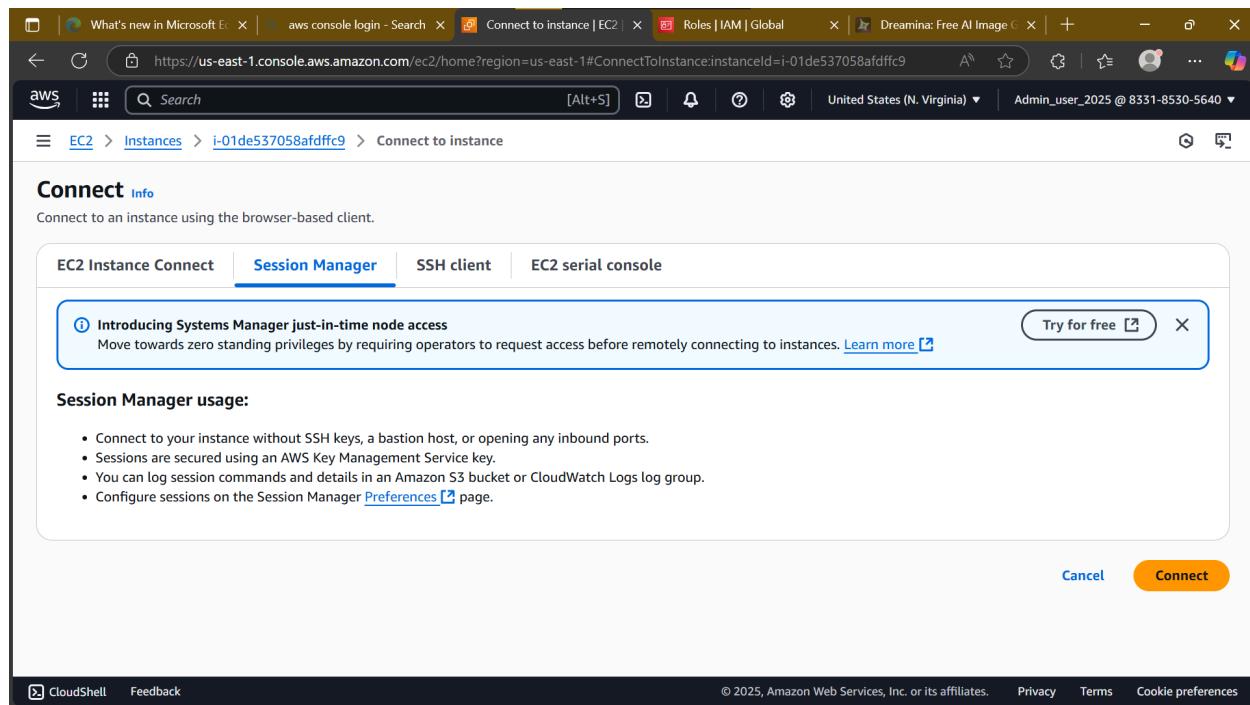
- Automate your HTTP server creation using EC2 user-data.

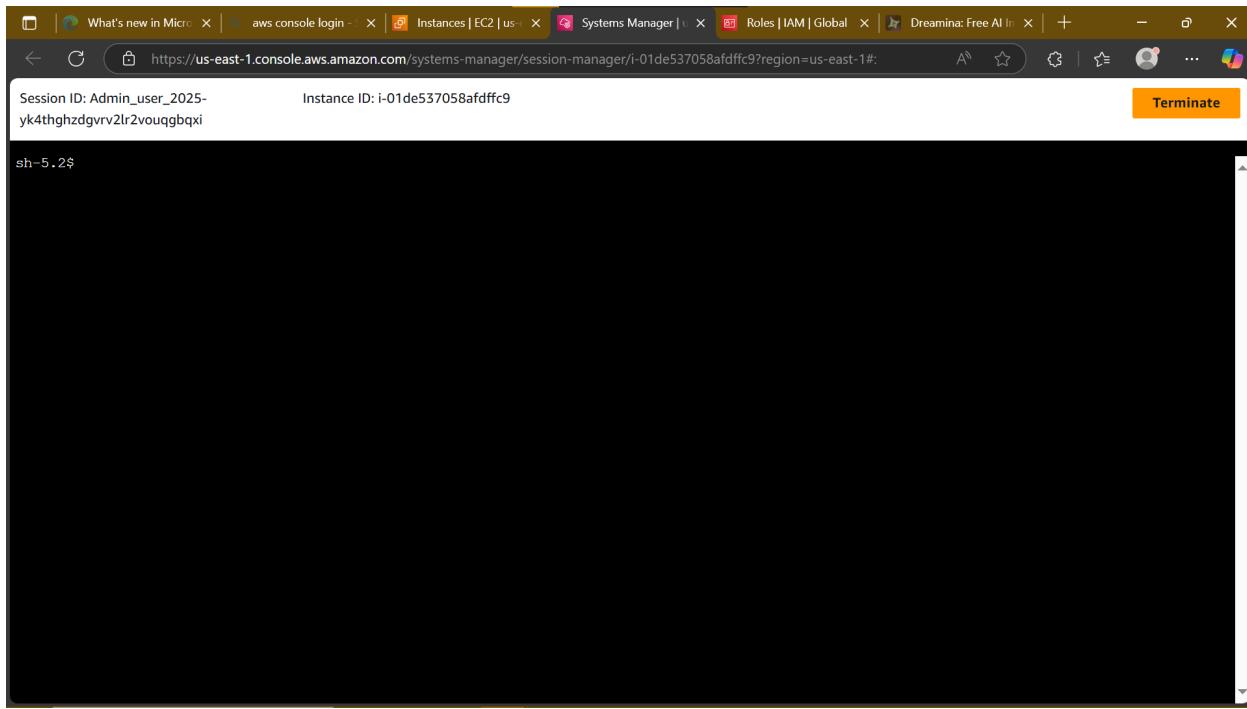
User-data script is provided below.

user-data Script:

```
#!/bin/bash
#updating software package
yum update -y
#installing apache
yum install -y httpd
#enabling httpd service
systemctl start httpd
#enabling httpd service at boot time
systemctl enable httpd
#display web page
echo "Automated Apache web server deployment using EC2 UserData – Student name" >
/var/www/html/index.html
```

- Make sure to attach an SSM role at launch that will allow you to login into the instance within your browser.
- Take screenshots, replace Student name by your name





Connected to my instance using the aws portal



Automated Apache web server deployment using EC2 UserData – Ebubechukwu Azozie

## Homework 7: AWS CLI

NB:refer to the video recording to see how to install and configure AWS CLI or google it

- Launch a linux EC2 instance using AWS CLI and Tag it with “my-ec2-created-through-CLI”. Be carefull with Name tag

```
C:\ Command Prompt - aws ec2 run-instances --image-id ami-02457590d33d576c3 --instance-type t2.micro --key-name awesome-key-east1 --security-group-ids sg-03bc38393a2ca5d1c --subnet-id subnet... --count 1
{
    "ReservationId": "r-082ef481266e39b5e",
    "OwnerId": "833185305640",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "ab7eb2f3-5c5b-4edf-be6e-eae4f2fc9a93",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-06-04T20:06:28+00:00",
                        "AttachmentId": "eni-attach-07bfd4db7194843f4",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-03bc38393a2ca5d1c",
                            "GroupName": "web-sg"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "12:e9:ca:c4:44:2f",
                    "NetworkInterfaceId": "eni-04fad979377c376fd",
                    "OwnerId": "833185305640",
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-0e103c4229e490e1b",
                    "PrivateIpAddress": "10.0.1.228",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateIpAddress": "10.0.1.228"
                        }
                    ]
                }
            ],
            "Ipv6Addresses": [],
            "MacAddress": "12:e9:ca:c4:44:2f",
            "NetworkInterfaceId": "eni-04fad979377c376fd",
            "OwnerId": "833185305640",
            "PrivateIpAddress": "10.0.1.228",
            "PrivateIpAddresses": [
                {
                    "Primary": true,
                    "PrivateIpAddress": "10.0.1.228"
                }
            ],
            "SourceDestCheck": true,
            "Status": "in-use",
            "SubnetId": "subnet-0e103c4229e490e1b",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-ec2-created-through-CLI"
                }
            ]
        }
    ]
}
```

```
C:\ Command Prompt - aws ec2 run-instances --image-id ami-02457590d33d576c3 --instance-type t2.micro --key-name awesome-key-east1 --security-group-ids sg-03bc38393a2ca5d1c --subnet-id subnet... --count 1
{
    "ReservationId": "r-082ef481266e39b5e",
    "OwnerId": "833185305640",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "ab7eb2f3-5c5b-4edf-be6e-eae4f2fc9a93",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-06-04T20:06:28+00:00",
                        "AttachmentId": "eni-attach-07bfd4db7194843f4",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-03bc38393a2ca5d1c",
                            "GroupName": "web-sg"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "12:e9:ca:c4:44:2f",
                    "NetworkInterfaceId": "eni-04fad979377c376fd",
                    "OwnerId": "833185305640",
                    "PrivateIpAddress": "10.0.1.228",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateIpAddress": "10.0.1.228"
                        }
                    ],
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-0e103c4229e490e1b",
                    "Tags": [
                        {
                            "Key": "Name",
                            "Value": "my-ec2-created-through-CLI"
                        }
                    ]
                }
            ],
            "Ipv6Addresses": [],
            "MacAddress": "12:e9:ca:c4:44:2f",
            "NetworkInterfaceId": "eni-04fad979377c376fd",
            "OwnerId": "833185305640",
            "PrivateIpAddress": "10.0.1.228",
            "PrivateIpAddresses": [
                {
                    "Primary": true,
                    "PrivateIpAddress": "10.0.1.228"
                }
            ],
            "SourceDestCheck": true,
            "Status": "in-use",
            "SubnetId": "subnet-0e103c4229e490e1b",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "my-ec2-created-through-CLI"
                }
            ]
        }
    ]
}
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store. The main area displays a table of instances. One instance is selected, showing its details. The instance ID is i-0fb04be8c31cb9886, it's named "my-ec2-create...", it's in the "Running" state, and its type is t2.micro. It has a public IP address of 18.234.188.184 and a private IP address of 10.0.1.228. The "Details" tab is selected, showing the instance summary.

Terminate the instance through CLI.

```
Command Prompt
PrivateIpAddress": "10.0.1.228",
"PrivateIpAddresses": [
{
    "Primary": true,
    "PrivateIpAddress": "10.0.1.228"
}
],
"SourceDestCheck": true,
"Status": "In-use",
"SubnetId": "subnet-0e103c4229e490e1b",
^C
C:\Users\azozi>aws ec2 stop-instances --instance-ids i-0fb04be8c31cb9886
```

I used the command line aws ec2 stop-instances --instance-ids i-0fb04be8c31cb9886

```

C:\ Command Prompt
    "PrivateIpAddress": "10.0.1.228",
    "PrivateIpAddresses": [
        {
            "Primary": true,
            "PrivateIpAddress": "10.0.1.228"
        }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-0e103c4229e490e1b",
^C

C:\Users\azoziz>aws ec2 stop-instances --instance-ids i-0fb04be8c31cb9886
{
    "StoppingInstances": [
        {
            "InstanceId": "i-0fb04be8c31cb9886",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}

C:\Users\azoziz>

```

Instance stopped

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (selected), AMIs, AMI Catalog, and Elastic Block Store. The main content area has a header with tabs for Instances (1/6), Info, Connect, Instance state (dropdown), Actions (dropdown), and Launch instances (dropdown). Below this is a search bar and a filter section with 'All states' dropdown. A table lists two instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	i-05d2c7306d22c10c	i-05d2c7306d22c10c	Terminated	t2.micro	-	<a href="#">View alarms +</a>
<input checked="" type="checkbox"/>	my-ec2-create...	i-0fb04be8c31cb9886	Stopping	t2.micro	-	<a href="#">View alarms +</a>

Below the table, a detailed view is shown for the instance with Instance ID i-0fb04be8c31cb9886. It includes tabs for Details (selected), Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under Details, there's an 'Instance summary' section with fields for Instance ID (i-0fb04be8c31cb9886), Public IPv4 address (18.234.188.184), Private IPv4 addresses (10.0.1.228), IPv6 address (-), Instance state (Stopping), and Public DNS (-).

- Using AWS CLI, Automate your HTTP server creation using EC2 user-data.

User-data Script:

```
#!/bin/bash
yum update -y
yum install httpd -y
```

```
echo '<html><body><h1 style="color:Blue;">Welcome to CloudSpace website launched using user data through AWS CLI by STUDENT_NAME</h1></body></html>' > /var/www/html/index.html
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

NB: use AWS CLI to pass your user data. Refer to the documentation as much as you can.

Replace STUDENT\_NAME by your Name.

SUBMISSION: Post a screenshot of the CLI commands for launching and terminating the EC2 instance. Attach your screenshot(s) to a Word or pdf document.

Code used for the CLI to run the instance while passing the userdata in a text file named userdata.txt

```
aws ec2 run-instances ^
--image-id ami-02457590d33d576c3 ^
--instance-type t2.micro ^
--key-name awesome-key-east1 ^
--security-group-ids sg-03bc38393a2ca5d1c ^
--subnet-id subnet-0e103c4229e490e1b ^
--associate-public-ip-address ^
--user-data file://userdata.txt ^
--tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=my-ec2-created-through-CLI}]"
^
--count 1
```

### Userdata Script used

```
#!/bin/bash
```

```
yum update -y
```

```
yum install httpd -y
```

```
echo '<html><body><h1 style="color:Blue;">Welcome to CloudSpace website launched using user data through AWS CLI by Ebubechukwu Azozie</h1></body></html>' > /var/www/html/index.html
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

```

Command Prompt - aws ec2 run-instances --image-id ami-02457590d33d576c3 --instance-type t2.micro --key-name awesome-key-east1 --security-group-ids sg-03bc38393a2ca5d1c --subnet-id subnet-0e103c4229e490e1b --associate-public-ip-address --user-data file://userdata.txt --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=my-ec2-created-through-CLI}]"
More? --count 1
{
  "ReservationId": "r-0a7876bef17ba7e93",
  "OwnerId": "833185305640",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "3e66b568-de5a-4af6-95cb-abb2c2bae9ab",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2025-06-13T14:58:20+00:00",
            "AttachmentId": "eni-attach-0babab300388163c",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching",
            "NetworkCardIndex": 0
          },
          "Description": "",
          "Groups": [
            {
              "GroupId": "sg-03bc38393a2ca5d1c",
              "GroupName": "web-sg"
            }
          ],
          "Ipv6Addresses": [],
          "MacAddress": "52:bd:ba:91:12:35",
          "NetworkInterfaceId": "eni-03c667ca5156f6d41",
        }
      ]
    }
  ]
}

```

From CLI

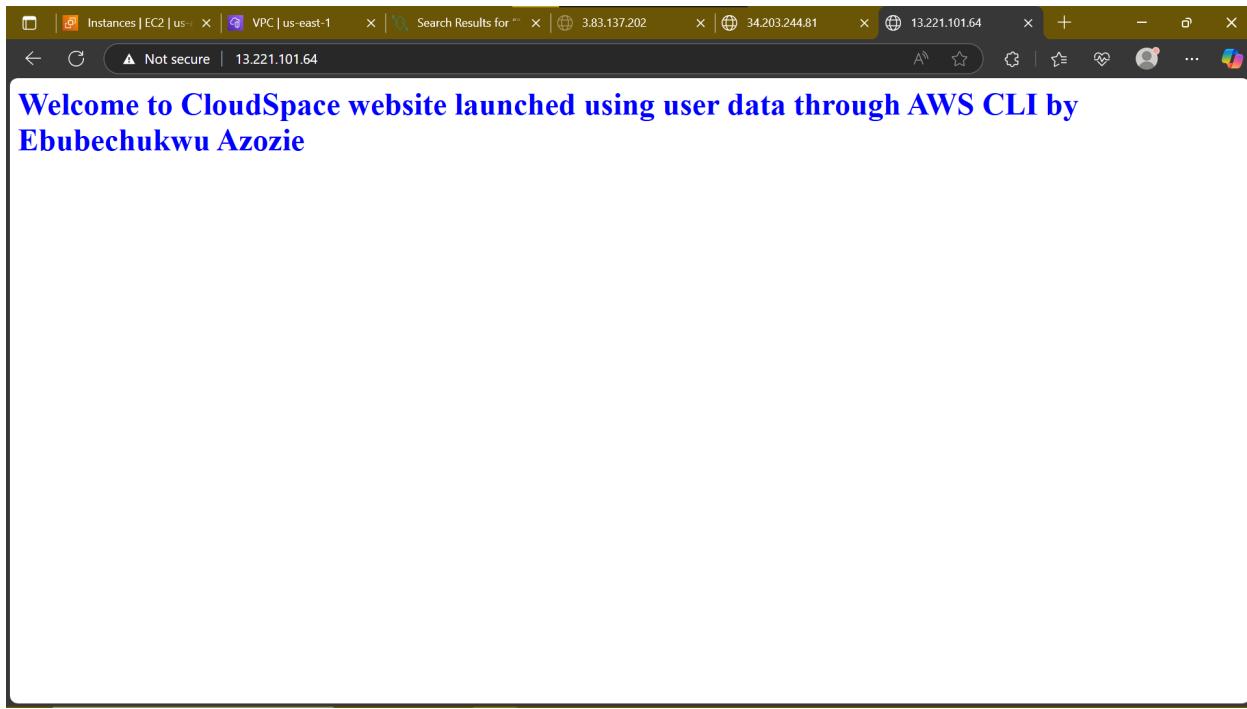
The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (selected), AMIs, AMI Catalog, and Elastic Block Store. Below the sidebar, there are links for CloudShell and Feedback.

The main content area has a header with tabs for Instances (1/3) and Info. It shows a table of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Two instances are listed:

- my-ec2-create...** (Instance ID: i-0359b3fb9091e3bcb) is in the **Running** state, t2.micro type, with 2/2 checks passed. It has a Public IPv4 address of 13.221.101.64 and a Private IPv4 address of 10.0.1.46.
- my-ec2-create...** (Instance ID: i-0ef193f8287c2cb38) is in the **Terminated** state, t2.micro type.

Below the table, there's a detailed view for the first instance (i-0359b3fb9091e3bcb). It includes tabs for Details (selected), Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under Details, there's an Instance summary section with fields for Instance ID (i-0359b3fb9091e3bcb), Public IPv4 address (13.221.101.64), Private IPv4 addresses (10.0.1.46), IP6 address (-), Instance state (Running), and Public DNS (-).

Instance created



Userdata successfully passed through CLI

To stop the instance using CLI

Code used

```
aws ec2 stop-instances --instance-ids i-0359b3fb9091e3bcb
```

```

Instances | EC2 | us-east-1 | VPC | us-east-1 | Search Results for " | 3.83.137.202 | 34.203.244.81 | 13.221.101.64 | + - ⌂
Command Prompt
Microsoft Windows [Version 10.0.19045.5854]
(c) Microsoft Corporation. All rights reserved.

C:\Users\azozzi>aws ec2 stop-instances --instance-ids i-0359b3fb9091e3bcb
{
    "StoppingInstances": [
        {
            "InstanceId": "i-0359b3fb9091e3bcb",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}

C:\Users\azozzi>
C:\Users\azozzi>

```

The screenshot shows a Windows Command Prompt window with the AWS CLI command `aws ec2 stop-instances --instance-ids i-0359b3fb9091e3bcb` executed. The output indicates that the instance with ID `i-0359b3fb9091e3bcb` is being stopped, transitioning from a previous state of running (code 16) to a current state of stopping (code 64). The AWS Management Console interface is visible in the background, showing the instance summary for the stopped instance.

The screenshot shows the AWS Management Console EC2 Instances page. The left sidebar is expanded to show the 'Instances' section. The main pane displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
practice	i-01a6d5e83f8984ade	Terminated	t2.micro	-	<a href="#">View alarms +</a>
my-ec2-create...	i-0359b3fb9091e3bcb	Stopping	t2.micro	-	<a href="#">View alarms +</a>
my-ec2-create...	i-0ef193f8287c2cb38	Terminated	t2.micro	-	<a href="#">View alarms +</a>

Below the table, the details for the instance `i-0359b3fb9091e3bcb` are shown. The instance is currently in a stopping state. Its public IPv4 address is 13.221.101.64 and its private IPv4 address is 10.0.1.46. It has no public DNS assigned.

Instance stopped

To terminate/delete the initially stopped instance

CLI command:

aws ec2 terminate-instances --instance-ids i-0359b3fb9091e3bcb

```
cmd Command Prompt
        "PreviousState": {
            "Code": 16,
            "Name": "running"
        }
    }
}

C:\Users\azozzi>
C:\Users\azozzi>aws ec2 terminate-instances --instance-ids i-0359b3fb9091e3bcb
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0359b3fb9091e3bcb",
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

C:\Users\azozzi>
C:\Users\azozzi>S_
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (selected), AMIs, and AMI Catalog. Below the sidebar, there's a CloudShell link and a Feedback button.

The main content area has a header with tabs for Instances (1/3) and Info. It shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
practice	i-01a6d5e83f8984ade	Terminated	t2.micro	-	<a href="#">View alarms +</a>
my-ec2-create...	i-0359b3fb9091e3bcb	Terminated	t2.micro	-	<a href="#">View alarms +</a>
my-ec2-create...	i-0ef193f8287c2cb38	Terminated	t2.micro	-	<a href="#">View alarms +</a>

Below the table, there's a detailed view for the instance with Instance ID i-0359b3fb9091e3bcb. The Details tab is selected, showing the following information:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0359b3fb9091e3bcb	-	-
IPv6 address	Instance state	Public DNS
-	Terminated	-

Instance terminated

## Homework 8: EBS & SNAPSHOT

- 1- Launch a linux EC2 instance using AWS CLI and Tag it with “my-ec2-created-through-CLI”. Be carefull with Name tag

```
Command Prompt - aws ec2 run-instances --image-id ami-02457590d33d576c3 --instance-type t2.micro --key-name awesome-key-east1 --security-group-ids sg-03bc38393a2ca5d1c --subnet-id subnet... -- More --
```

```
{ "ReservationId": "r-010c1e2472a869647", "OwnerId": "833185305640", "Groups": [], "Instances": [ { "Architecture": "x86_64", "BlockDeviceMappings": [], "ClientToken": "1cd695e2-7ca2-4844-8932-e4139ba1b3f2", "EbsOptimized": false, "EnaSupport": true, "Hypervisor": "xen", "NetworkInterfaces": [ { "Attachment": { "AttachTime": "2025-06-06T14:25:17+00:00", "AttachmentId": "eni-attach-0c2e848ba841f0cd3", "DeleteOnTermination": true, "DeviceIndex": 0, "Status": "attaching", "NetworkCardIndex": 0 }, "Description": "", "Groups": [ { "GroupId": "sg-03bc38393a2ca5d1c", "GroupName": "web-sg" } ], "Ipv6Addresses": [], "MacAddress": "12:3b:63:6e:56:45", "NetworkInterfaceId": "eni-0aab1c9ed8c6fcfa3", "OwnerId": "833185305640", "PrivateIpAddress": "10.0.1.72", "PrivateIpAddresses": [ { "Primary": true, "PrivateIpAddress": "10.0.1.72" } ], "SourceDestCheck": true, "Status": "in-use", }
```

- 2- Take a snapshot of the Root Volume

The screenshot shows the AWS Management Console in a browser window. The URL is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Snapshots>. The left sidebar shows navigation links for Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (Load Balancers). The main content area displays the 'Schemas (1/2) Info' section, which lists two snapshots:

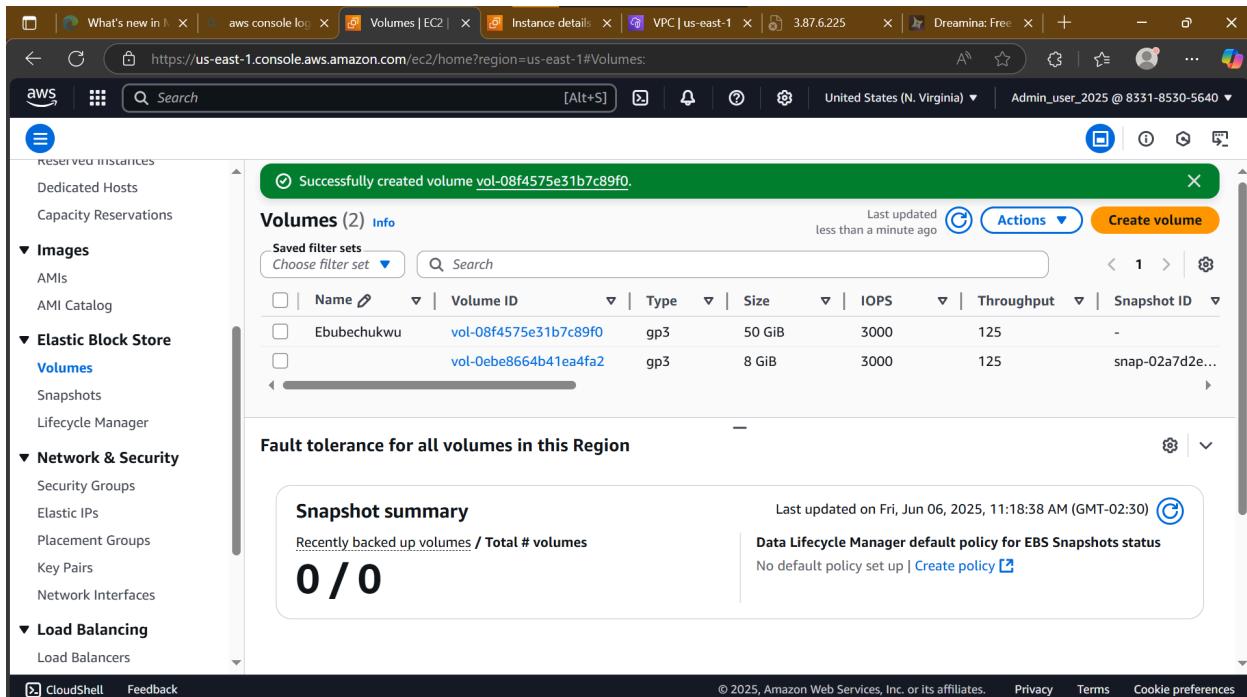
Name	Snapshot ID	Full snapshot size	Volume size	Description	Storage
homework4-a...	snap-004b4ec567ef925a5	1.56 GiB	8 GiB	Created by CreateImage(l...	Standby
<input checked="" type="checkbox"/>	snap-0e0c986f5a4fb15b0	1.81 GiB	8 GiB	vol-0ebe8664b41ea4fa2-J...	Standby

Below the table, a detailed view for the selected snapshot (snap-0e0c986f5a4fb15b0) is shown:

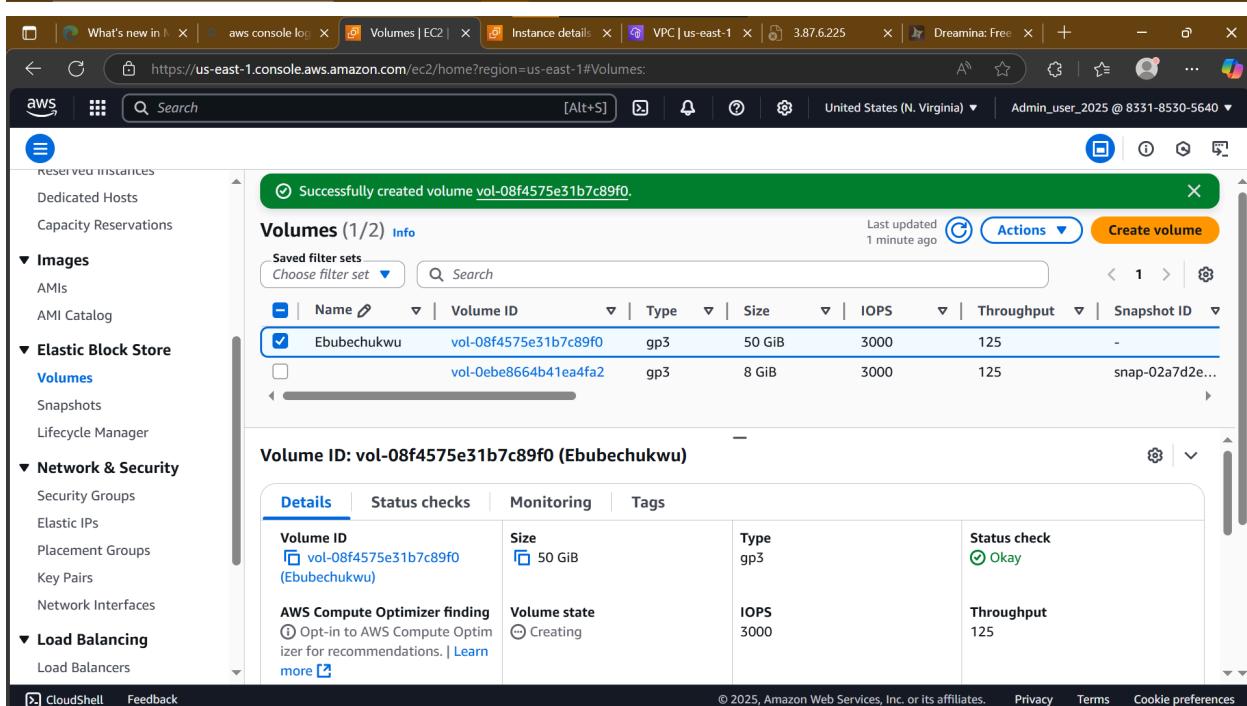
Snapshot ID: snap-0e0c986f5a4fb15b0					
Snapshot ID	Full snapshot size	Progress	Snapshot status		
<a href="#">snap-0e0c986f5a4fb15b0</a>	<a href="#">1.81 GiB</a>	<a href="#">100%</a>	<span>Completed</span>		
Owner	Started	Product codes	Fast snapshot restore		
<a href="#">833185305640</a>	<a href="#">Fri Jun 06 2025 12:10:43 GMT-0230 (Newfoundland Daylight Time)</a>	-	-		
Description	<a href="#">vol-0ebe8664b41ea4fa2-June072025-homework</a>				

At the bottom of the page, there are links for CloudShell, Feedback, and a footer with copyright information: © 2025, Amazon Web Services, Inc. or its affiliates.

3- Create a new Volume (with Name tag “your\_first\_name”, size 50 GB) in the same Availability Zone (AZ) as the EC2 instance. Take screenshot



The screenshot shows the AWS Cloud Console interface for the EC2 service. The left sidebar navigation includes Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Lead Balancing (Load Balancers), and CloudShell/Feedback. The main content area displays a success message: "Successfully created volume vol-08f4575e31b7c89f0." Below this is a "Volumes (2) Info" table with columns: Name, Volume ID, Type, Size, IOPS, Throughput, and Snapshot ID. Two volumes are listed: "Ebubechukwu" (vol-08f4575e31b7c89f0, gp3, 50 GiB, 3000 IOPS, 125 throughput, no snapshot) and "vol-0ebe8664b41ea4fa2" (gp3, 8 GiB, 3000 IOPS, 125 throughput, snap-02a7d2e...). A "Fault tolerance for all volumes in this Region" section indicates "0 / 0" snapshots backed up. A "Snapshot summary" box shows "0 / 0" recently backed up volumes / total # volumes. The bottom right corner of the main content area contains copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates." and links for Privacy, Terms, and Cookie preferences.

This screenshot shows the detailed view of the newly created volume. The left sidebar is identical to the previous screenshot. The main content area shows the "Volume ID: vol-08f4575e31b7c89f0 (Ebubechukwu)" details. The "Details" tab is selected, displaying the following information:

Volume ID	Size	Type	Status check
vol-08f4575e31b7c89f0 (Ebubechukwu)	50 GiB	gp3	Okay

Below this, the "AWS Compute Optimizer finding" section shows a note about opting-in to recommendations. The "Status checks" tab shows the volume is "Creating". The "Monitoring" and "Tags" tabs are also present but not selected.

4- Attach the new volume to the EC2 created in step 1. Take a screenshot showing both Volumes in the console.

The screenshot shows the AWS Cloud console interface. The left sidebar is collapsed. The main content area displays a success message: "Successfully attached volume vol-08f4575e31b7c89f0 to instance i-099402e540aaec251." Below this, the "Volumes (1/2)" section is shown with one item listed:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID
Ebubechukwu	vol-08f4575e31b7c89f0	gp3	50 GiB	3000	125	-

Below the volume list, a detailed view for "Volume ID: vol-08f4575e31b7c89f0 (Ebubechukwu)" is provided. The "Details" tab is selected, showing the following information:

Volume ID	Size	Type	Status check
vol-08f4575e31b7c89f0 (Ebubechukwu)	50 GiB	gp3	Okay

Under the "AWS Compute Optimizer finding" section, there is a note about opting-in to the AWS Compute Optimizer for recommendations.

The screenshot shows the AWS Cloud console interface with the "EC2" menu selected. The left sidebar shows the "Instances" section is expanded, with "Instances" highlighted. The main content area displays the "Instances (1/1)" section with one item listed:

Name	Instance ID	Instance state	Instance type	Status check
my-ec2-create...	i-099402e540aaec251	Running	t2.micro	2/2 checks passed

Below the instance list, a detailed view for "i-099402e540aaec251 (my-ec2-created-through-CLI)" is shown. The "FILTER BLOCK DEVICES" section lists two volumes:

Volume ID	Device name	Volume size (GiB)	Volume State	Attachment status
vol-0ebe8664b41ea4fa2	/dev/xvda	8	In-use	Attached
vol-08f4575e31b7c89f0	/dev/sdb	50	In-use	Attached

At the bottom of the page, the "Volume monitoring (2)" section is visible.

5- login to the EC2 instance and run the command “lsblk”. Take a screenshot showing both Volumes size  
Exemple of lsblk output

```
[ec2-user@ip-10-0-7-44 ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0   0  8G  0 disk
└─xvda1 202:1   0  8G  0 part /
xvdf    202:80  0 50G  0 disk
```

