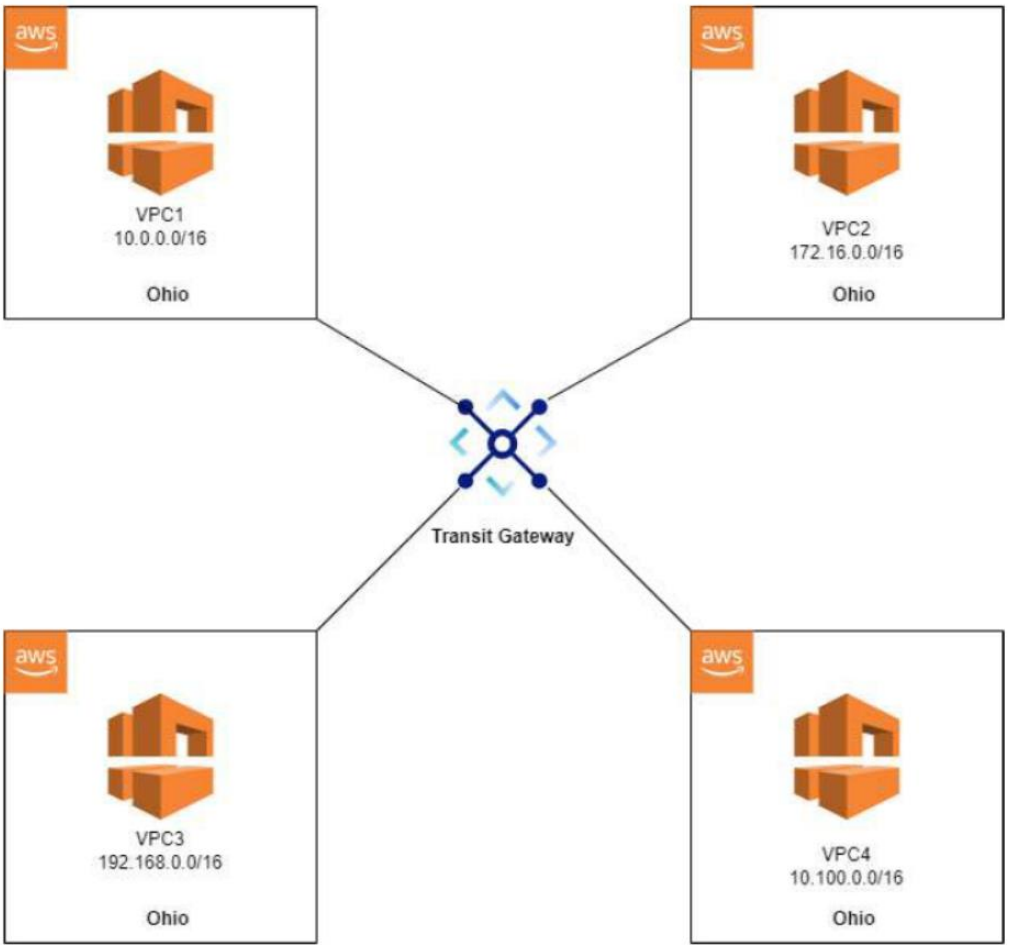
AWS Secure Network

Table of Contents

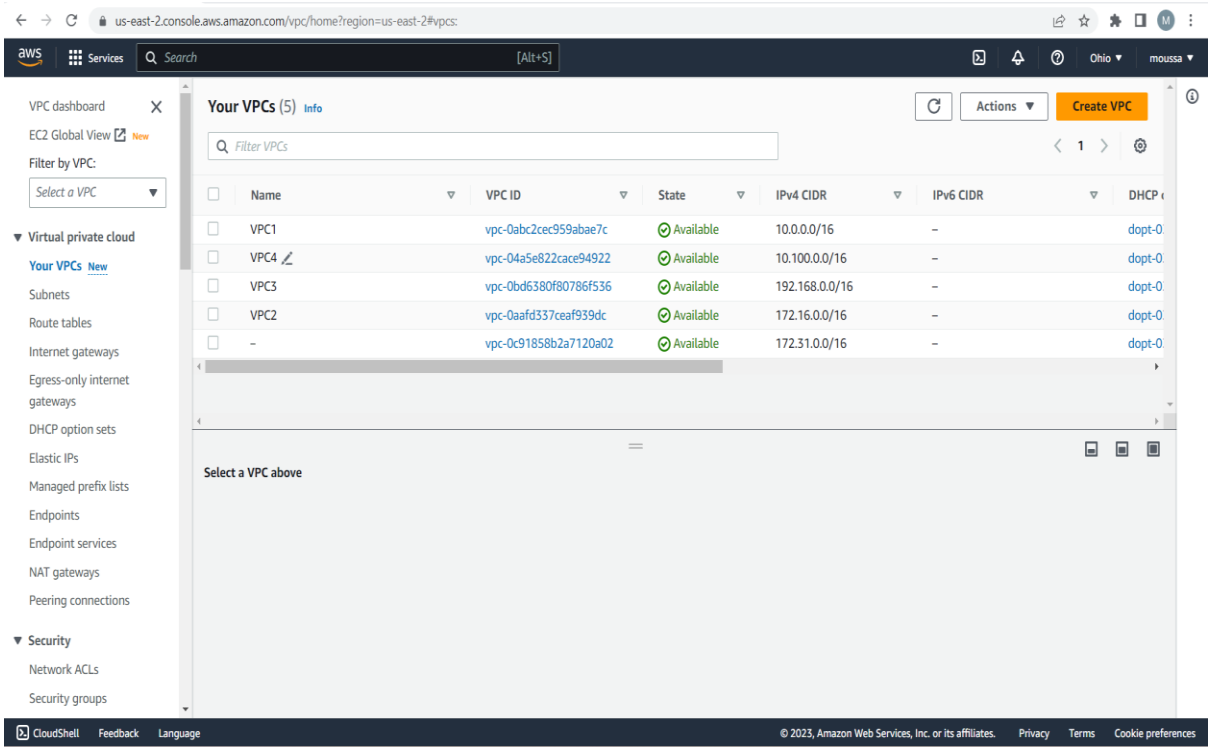
The primary objective of this project is to develop a secure cloud network design that effectively mitigates the risks associated with unauthorized access, data breaches, and network vulnerabilities. By implementing industry best practices and leveraging state-of-the-art security technologies, this project aims to establish a robust security framework that ensures the confidentiality, integrity, and availability of data stored and transmitted within the cloud infrastructure of a small business.

I. Proposed Architecture



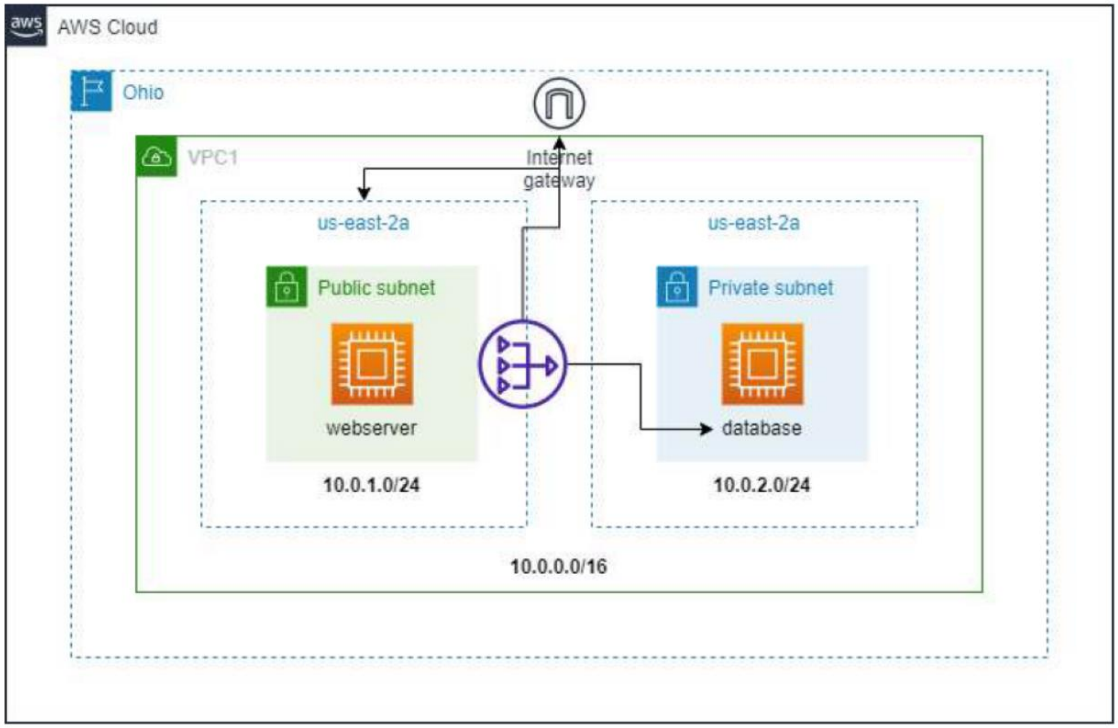
Here is our Network topology with 4 VPCs located in Ohio, each VPC is a different Network. A transit gateway interconnects all the VPCs.

- **VPC1: 10.0.0.0/16**
- **VPC2: 172.16.0.0/16**
- **VPC3: 192.168.0.0/16**
- **VPC4: 10.100.0.0/16**

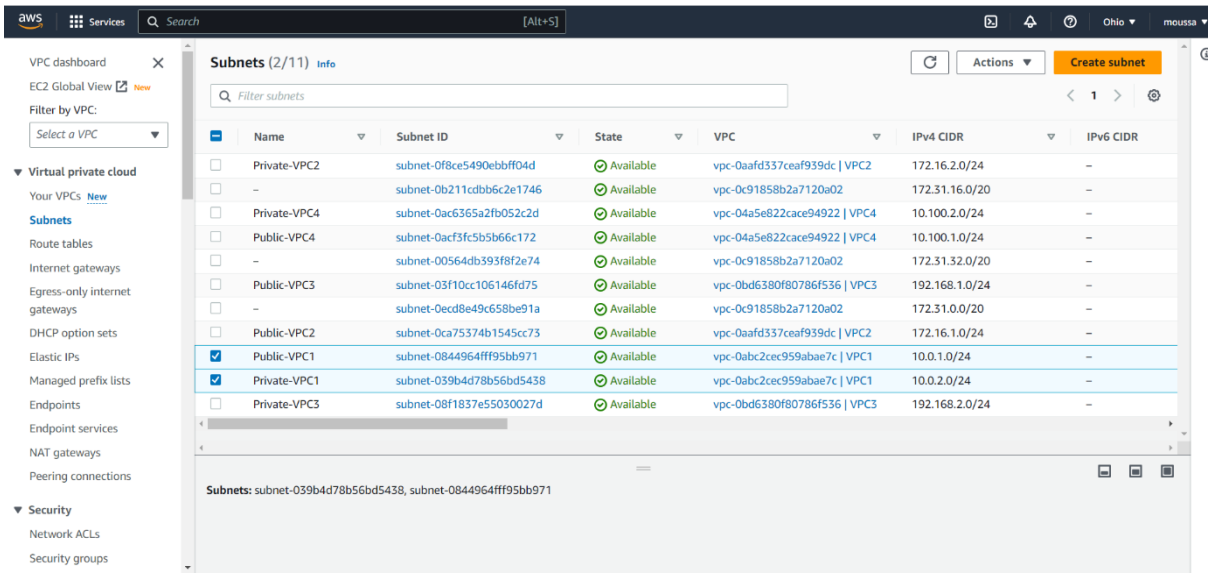


Each VPC has purposes and objectives based on the business's needs and requirements.

VPC1 Diagram



The VPC1 have 2 subnets in the same availability zone which is **us-east-2a.**



- **Public subnet: 10.0.1.0/24**

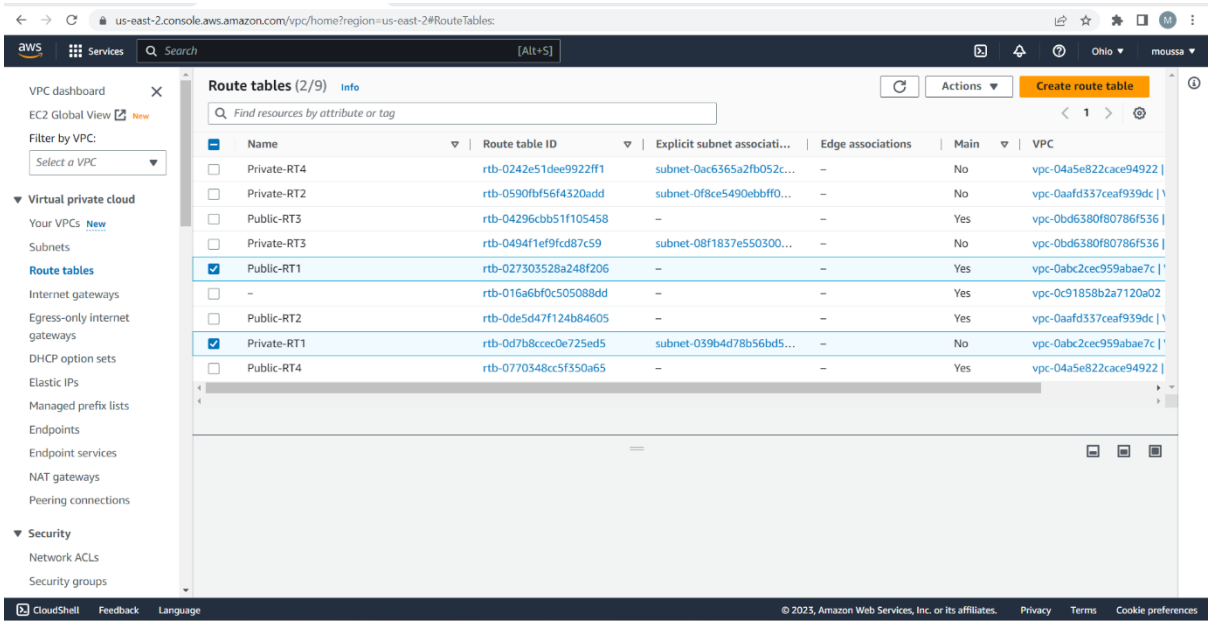
  The public subnet contains an EC2 instance that we consider as a web server, the internet gateway enables this instance to have access to the internet and people from the internet access the instance. This makes instances in the public subnet expose to cyber threat.

- **Private subnet: 10.0.2.0/24**

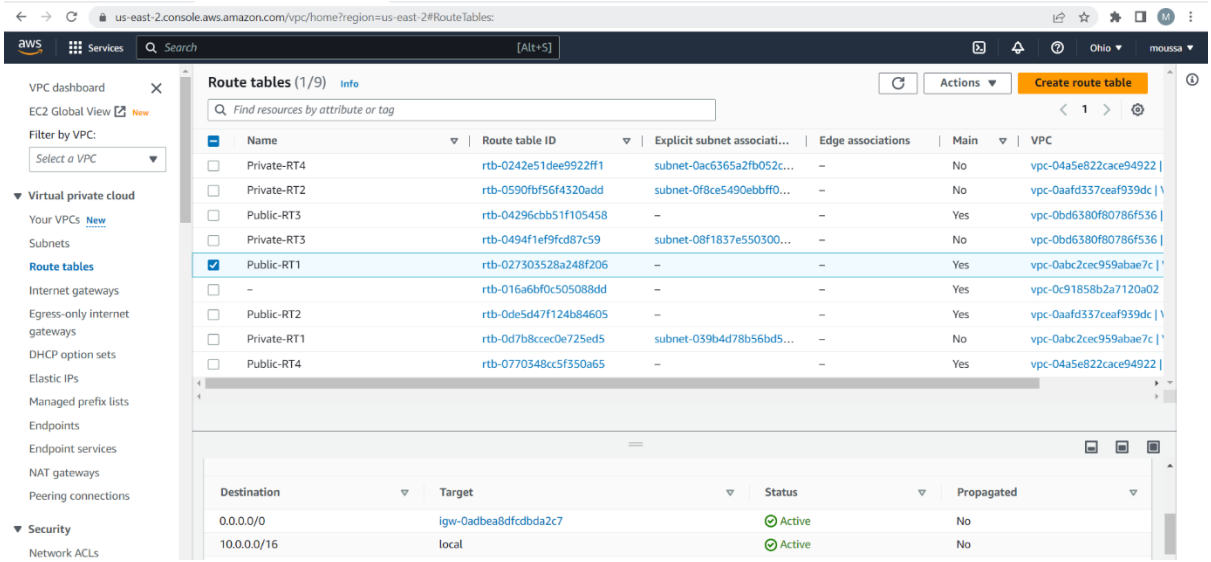
The private subnet contains an EC2 instance that we consider as a database. The instances in the private subnet are not accessible from the internet that is why it is more secure than the public subnet. However, the instance in the private subnet might need to access to the internet. This will be possible by using the NAT gateway that is in the public subnet, so the private instance will be routed to the internet gateway and access the internet.

We will isolate the others VPC for future purposes.

- **VPC1 route tables:** VPC1 have two route table **Public-RT1** for the public subnet and **Private-RT1** the private subnet.

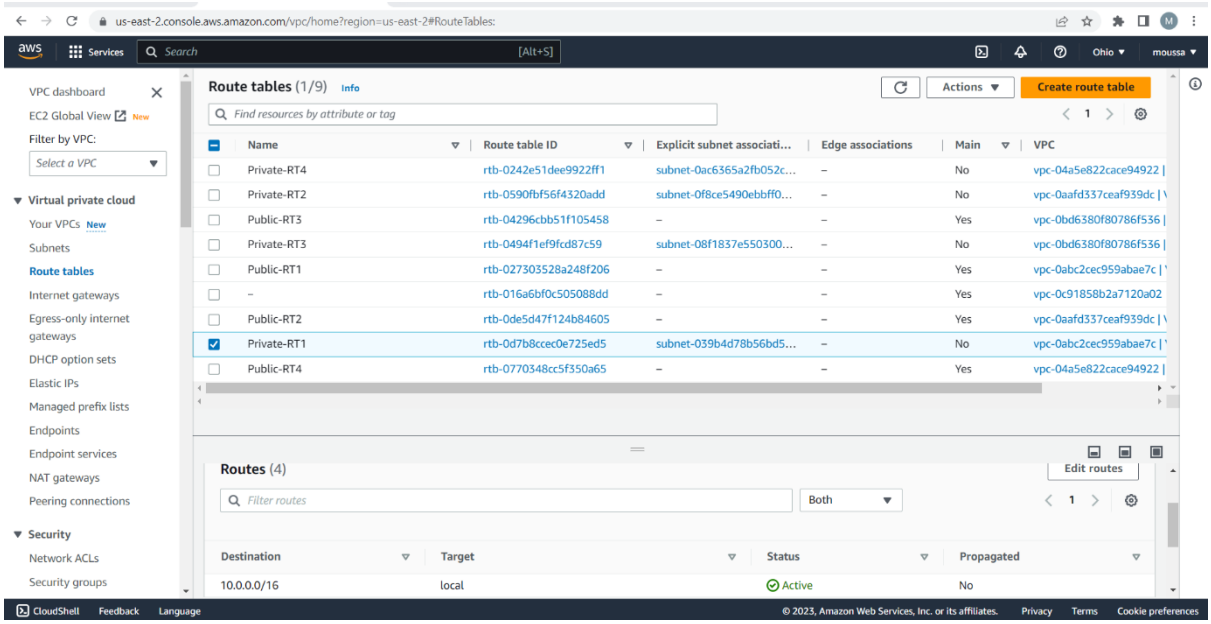


Public-RT1



Public-RT1 have a route for communication within the VPC and a route to the internet thanks to the internet gateway. The public subnet can access to the internet and from the internet we can access the public subnet.

Private-RT1

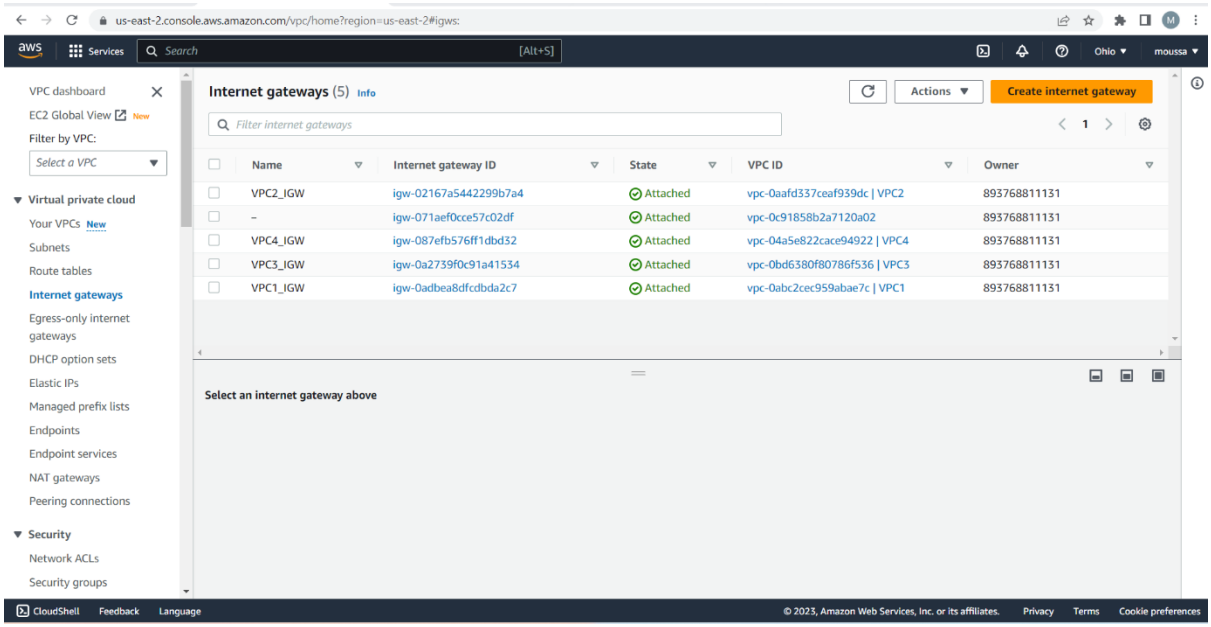


Private-RT1 has a route for communication within the VPC, can access the internet thanks to a NAT gateway in the public subnet but we cannot access it from the internet that make the private subnet more secure.

- **Internet access**

An internet gateway enables resources in your public subnets (such as EC2 instances) to connect to the internet if the resource has a public IPv4 address or an IPv6 address. Similarly, resources on the internet can initiate a connection to resources in your subnet using the public IPv4 address or IPv6 address. For example, an internet gateway enables you to connect to an EC2 instance in AWS using your local computer.

The internet gateway is only for the public subnets.



II. Security Groups

AWS Security Groups are virtual firewalls that control inbound and outbound traffic for Amazon EC2 instances, allowing you to define rules to permit or deny traffic based on protocol, port, and source/destination IP addresses

Web server security group rules:

Inbound Rule for HTTP (Port 80):

Type: HTTP

Protocol: TCP

Port Range: 80

Source: 0.0.0.0/0 (allowing access from any IP address)

Description: Allow inbound HTTP traffic to the web server.

Inbound Rule for HTTPS (Port 443):

Type: HTTPS

Protocol: TCP

Port Range: 443

Source: 0.0.0.0/0 (allowing access from any IP address)

Description: Allow inbound HTTPS traffic to the web server.

Inbound Rule for SSH (Secure Shell) access (Port 22):

Type: SSH

Protocol: TCP

Port Range: 22

Source: 0.0.0.0/0

Description: Allow inbound SSH traffic only from our all IP address. The EC2 instance private key is required so This rule ensures secure remote access to the server.

Inbound Rule for ICMP (Ping) requests:

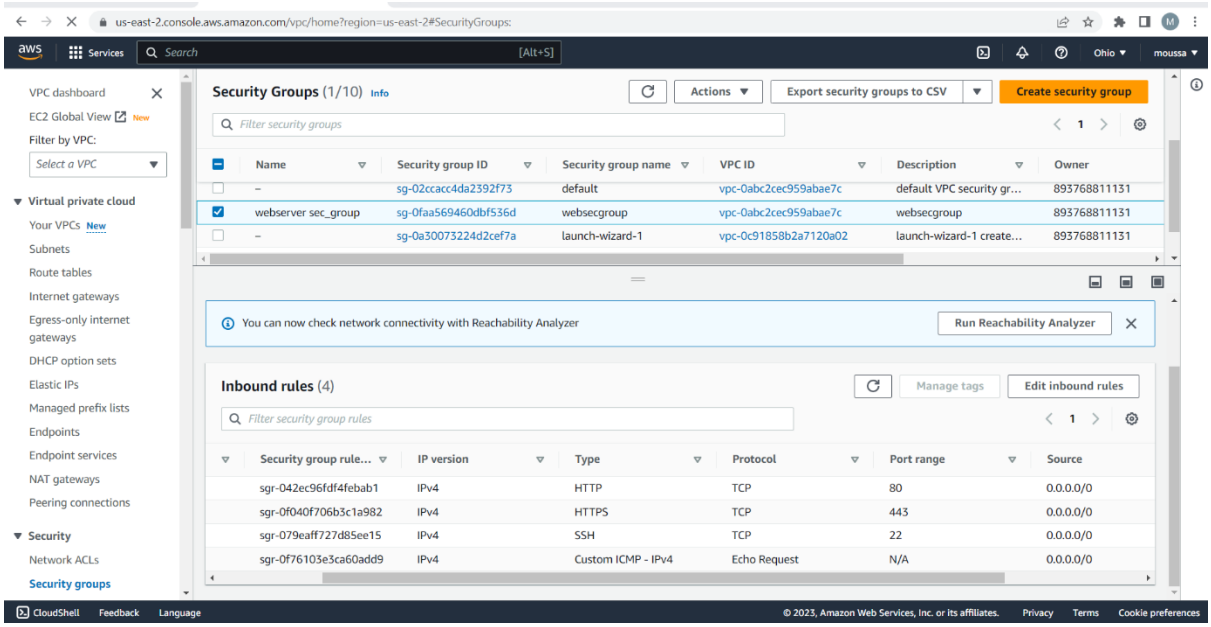
Type: ICMP (IPv4)

Protocol: ICMP

Port Range: N/A

Source: 0.0.0.0/0 (allowing access from any IP address)

Description: Allow inbound ICMP requests, such as ping, for troubleshooting purposes.



Outbound Rule for All Traffic:

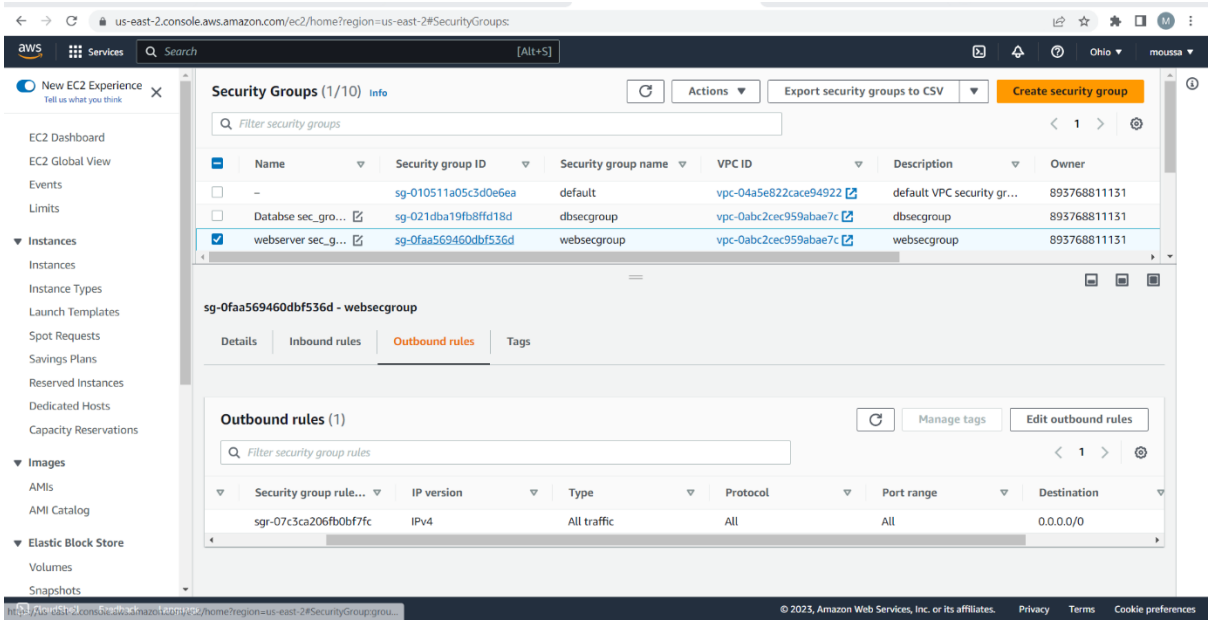
Type: All Traffic

Protocol: All

Port Range: All

Destination: 0.0.0.0/0

Description: Allow all outbound traffic from the web server to any destination. This rule ensures that the server can communicate with other services or resources it requires.


us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#SecurityGroups:
Services
Search
[Alt+S]
Ohio
moussa
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Security Groups (1/10) Info
Actions
Export security groups to CSV
Create security group
Filter security groups
Name
Security group ID
Security group name
VPC ID
Description
Owner
–
sg-010511a05c3d0e6ea
default
vpc-04a5e822cace94922
default VPC security gr...
893768811131
Databse sec_gro...
sg-021dba19fb8ffd18d
dbsecgroup
vpc-0abc2cec959abae7c
dbsecgroup
893768811131
webserver sec_g...
sg-0faa569460dbf536d
websecgroup
vpc-0abc2cec959abae7c
websecgroup
893768811131
sg-0faa569460dbf536d - websecgroup
Details
Inbound rules
Outbound rules
Tags
Outbound rules (1)
Manage tags
Edit outbound rules
Filter security group rules
Security group rule...
IP version
Type
Protocol
Port range
Destination
sgr-07c3ca206fb0bf7fc
IPv4
All traffic
All
All
0.0.0.0/0
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

- **Database** security group rules:

Inbound Rule for MySQL (Port 3306):

Type: MySQL/Aurora

Protocol: TCP

Port Range: 3306

Source: sg-0faa569460dbf536d (web server security group ID)

Description: Allow inbound MySQL traffic from the security group associated with the web server. This rule allows the web server to connect to the database.

Inbound Rule for SSH (Secure Shell) access (Port 22):

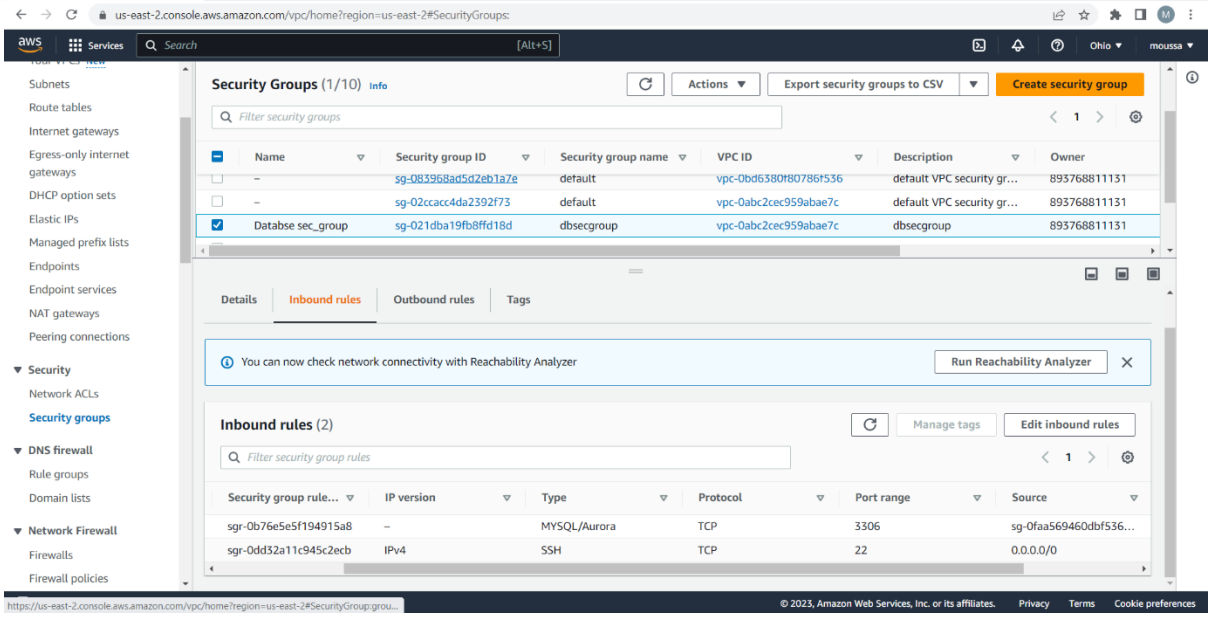
Type: SSH

Protocol: TCP

Port Range: 22

Source: 0.0.0.0/0

Description: Allow inbound SSH traffic only from the public subnet. This rule ensures secure remote access to the database server for administrative purposes.



Outbound Rule for All Traffic:

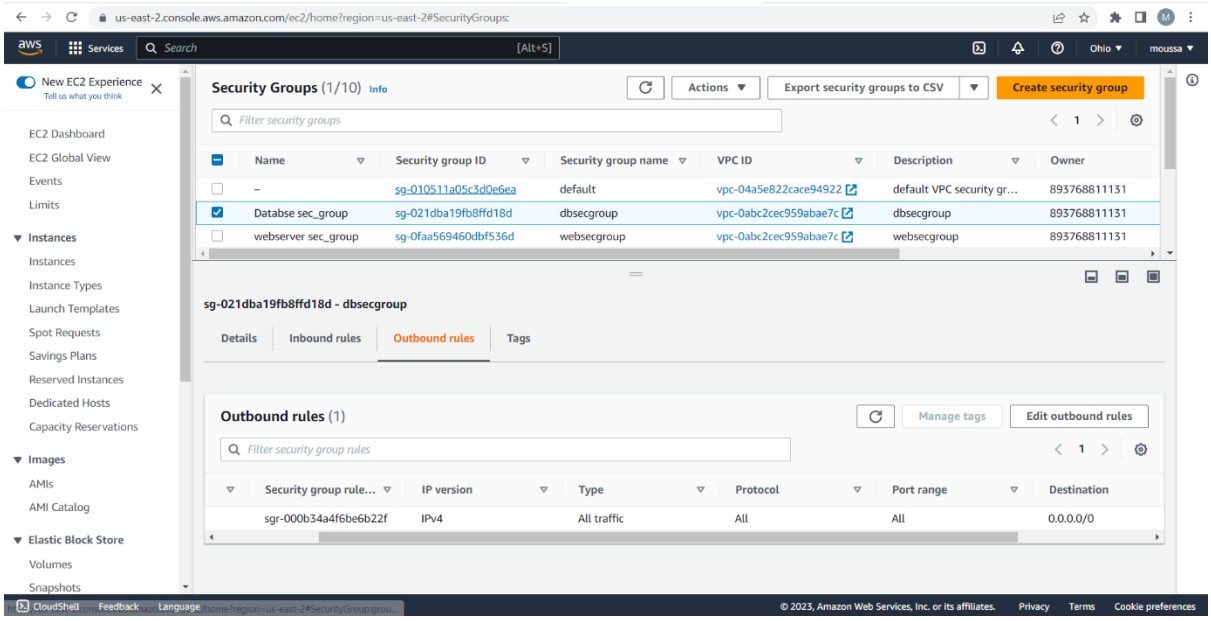
Type: All Traffic

Protocol: All

Port Range: All

Destination: 0.0.0.0/0

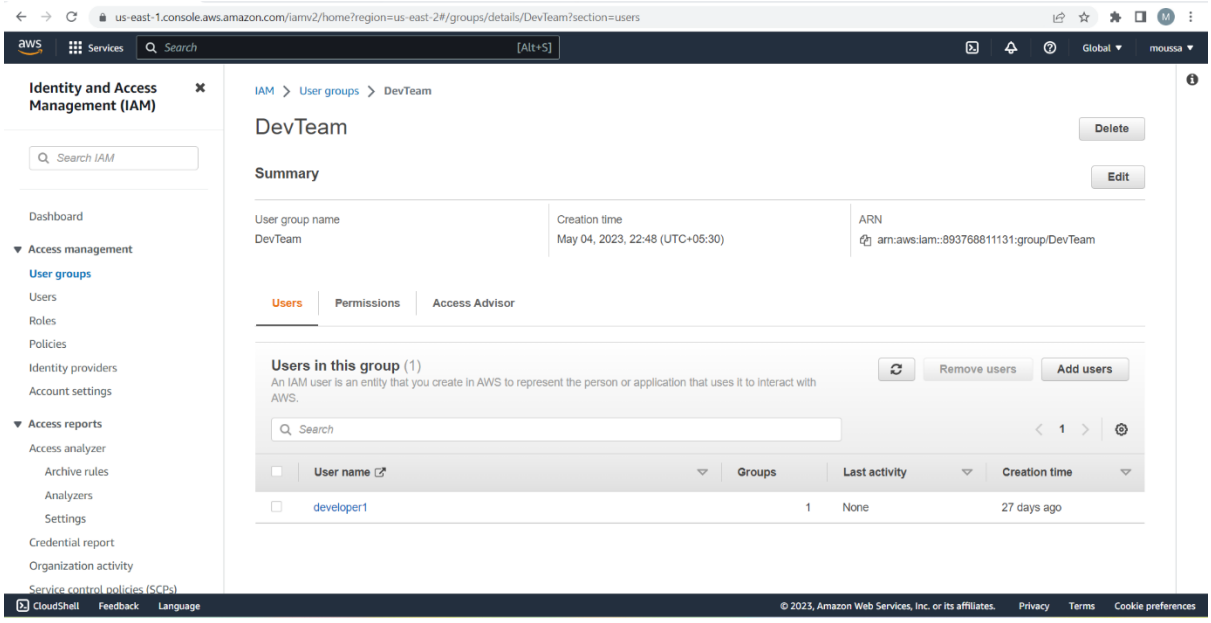
Description: Allow all outbound traffic from the database server to any destination. This rule ensures that the database server can communicate with other services or resources it requires.



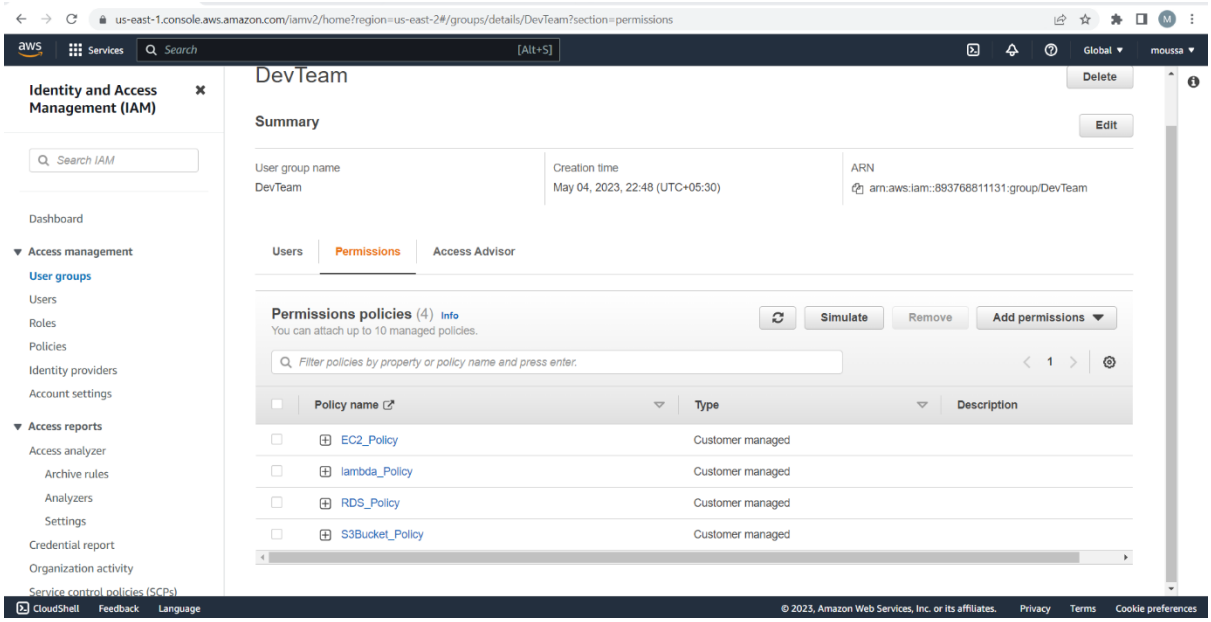
III. Identity and Access Management (IAM)

AWS IAM (Identity and Access Management) is a web service provided by Amazon Web Services (AWS) that enables you to securely control access to AWS services and resources. It helps you manage users, groups, roles, and permissions within your AWS environment.

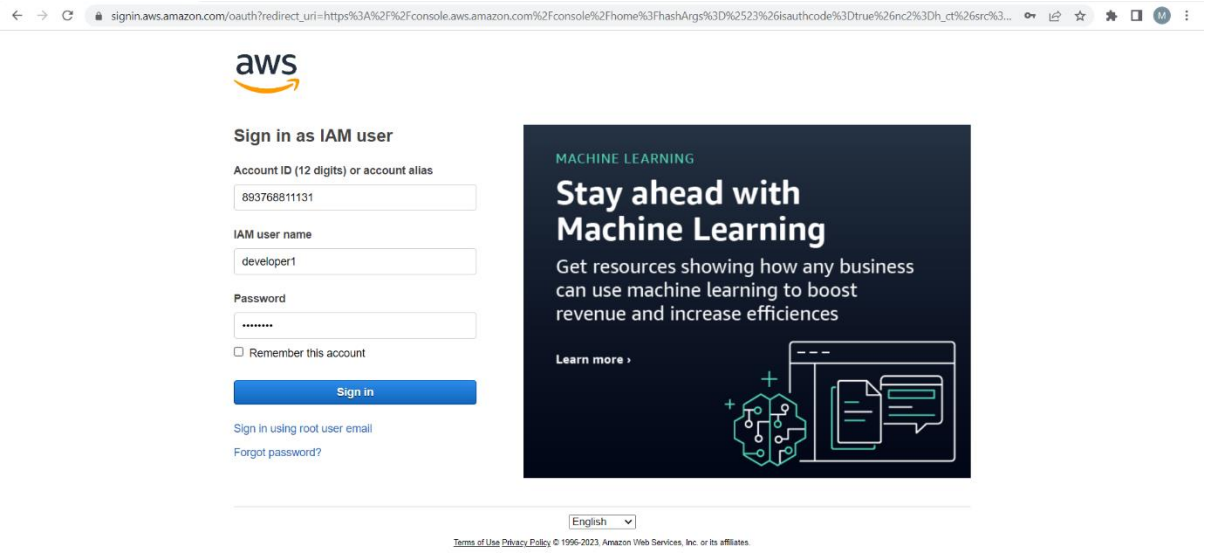
Instead of assigning permissions to each user we have created a group called **DevTeam** so the permissions will be assigned to the group. This will make the management of the users easier.



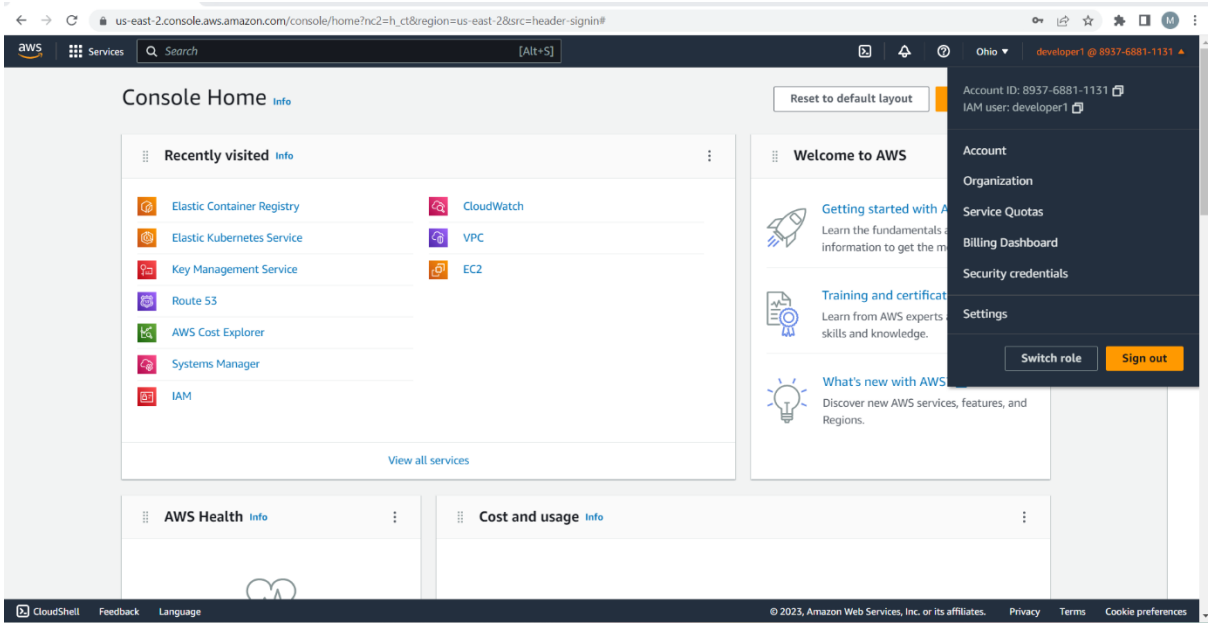
DevTeam is a development team so we suppose that developers may need these few policies including:



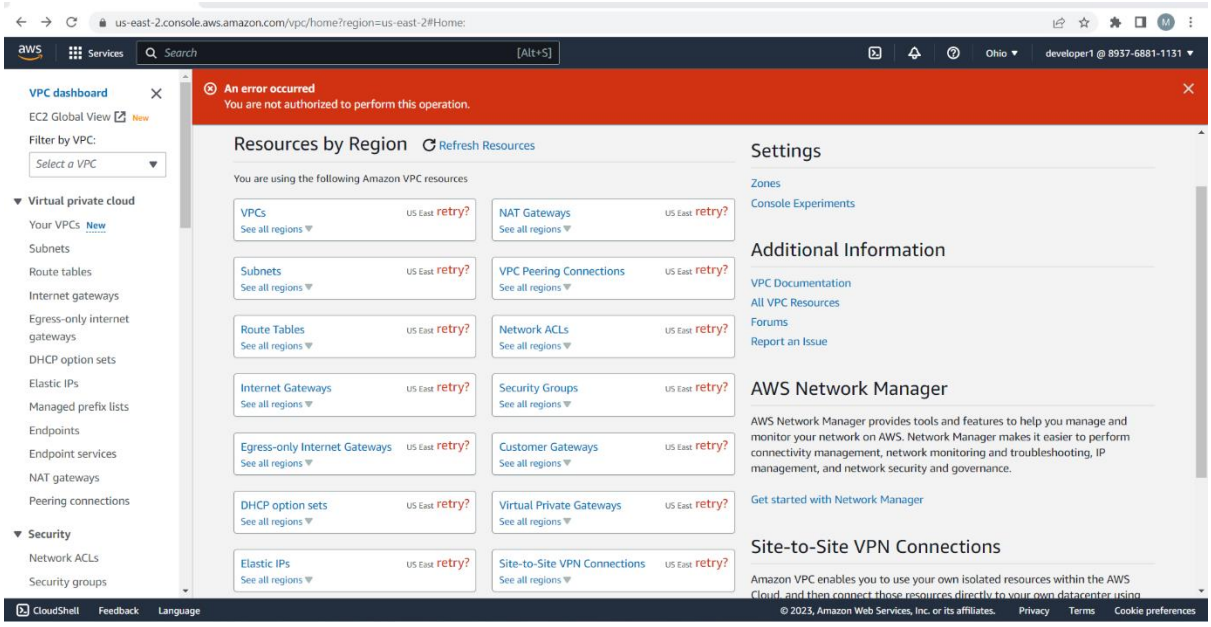
We connect to that user account by using his username and password.



Now we have logged as developer1



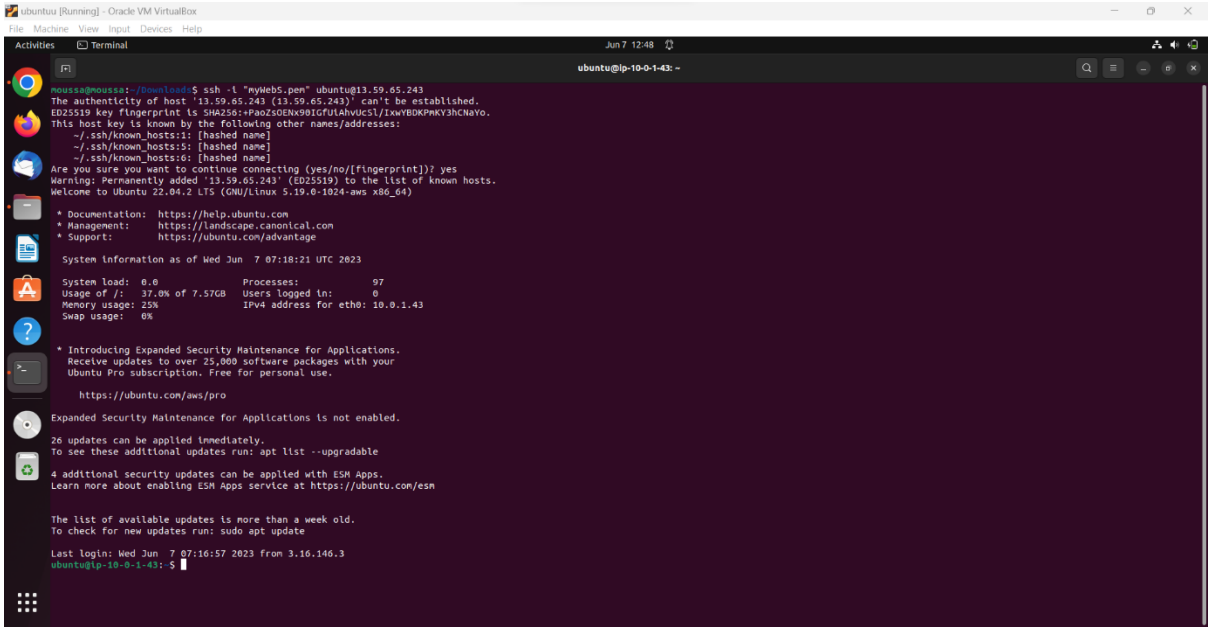
Developer1 is only authorized to perform certain tasks, we can see that his not authorized to access the VPCs.



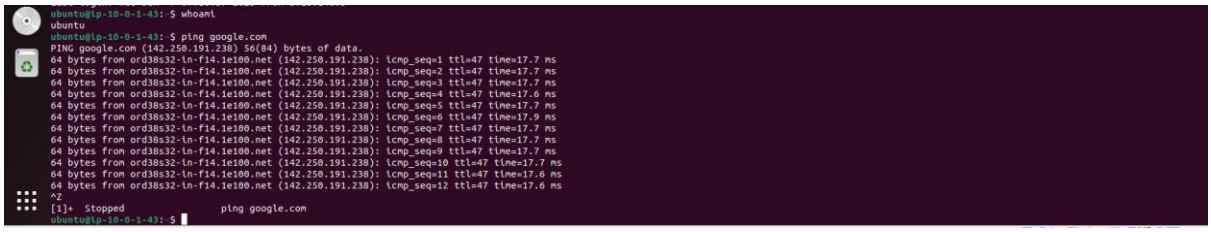
IV. Remote Access to the instances

1. Access to the web server

In order to have a remote access to the web server from our Ubuntu terminal we need the private key of the EC2 instance and the public IP address of the instance.



We are able to connect to the instance and we can see the name of the machine and eventually the machine can go to the internet

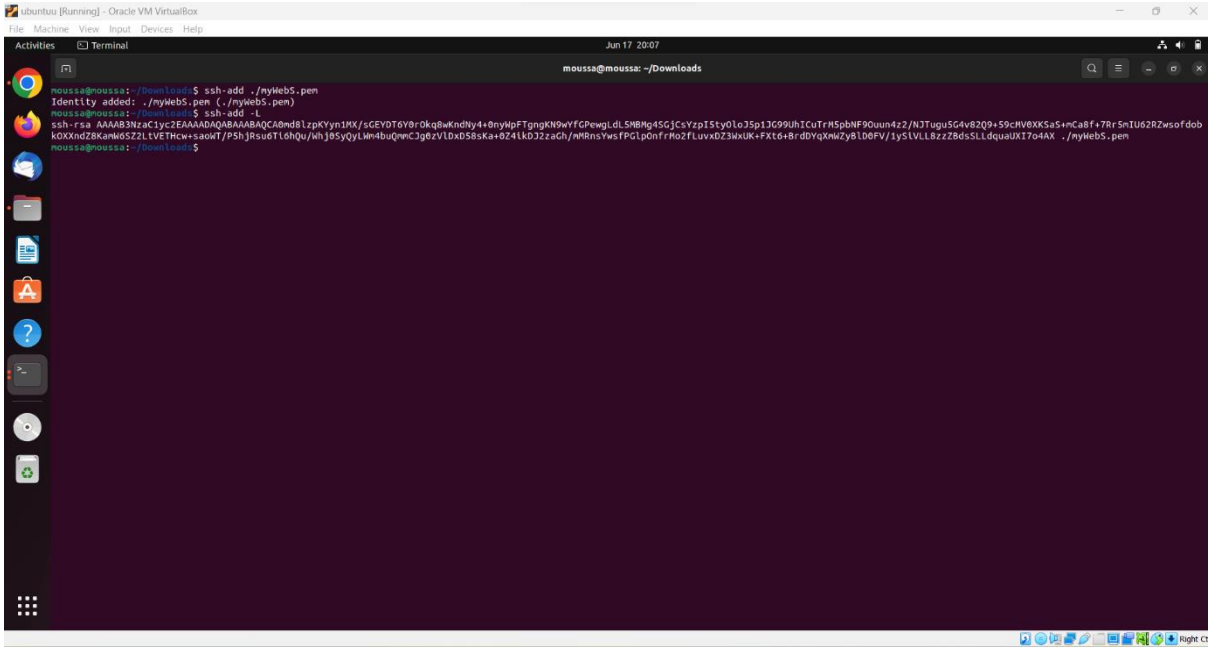


2. Access to the database

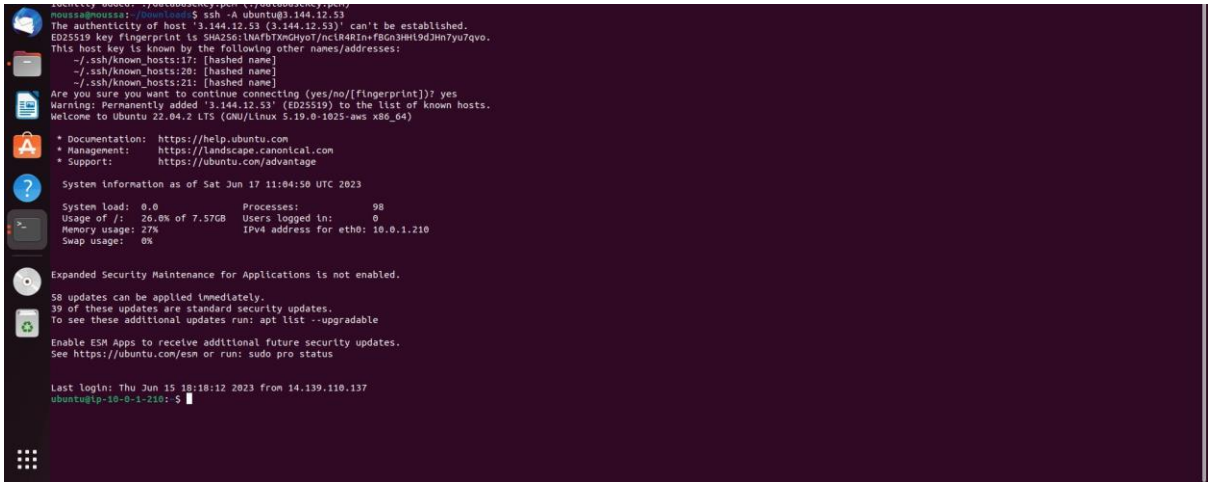
To access the database, one of the most secure ways is by SSH forwarding also known as SSH bastion host.

This method involves using an intermediate server (our web server instance), often called a "bastion host" or "jump host," that is publicly accessible and acts as a bridge to connect to the private EC2 instance. The bastion host resides in a public subnet and has SSH access to both the public and private subnets. You establish an SSH connection to the bastion host first and then use that connection to SSH into the private EC2 instance.

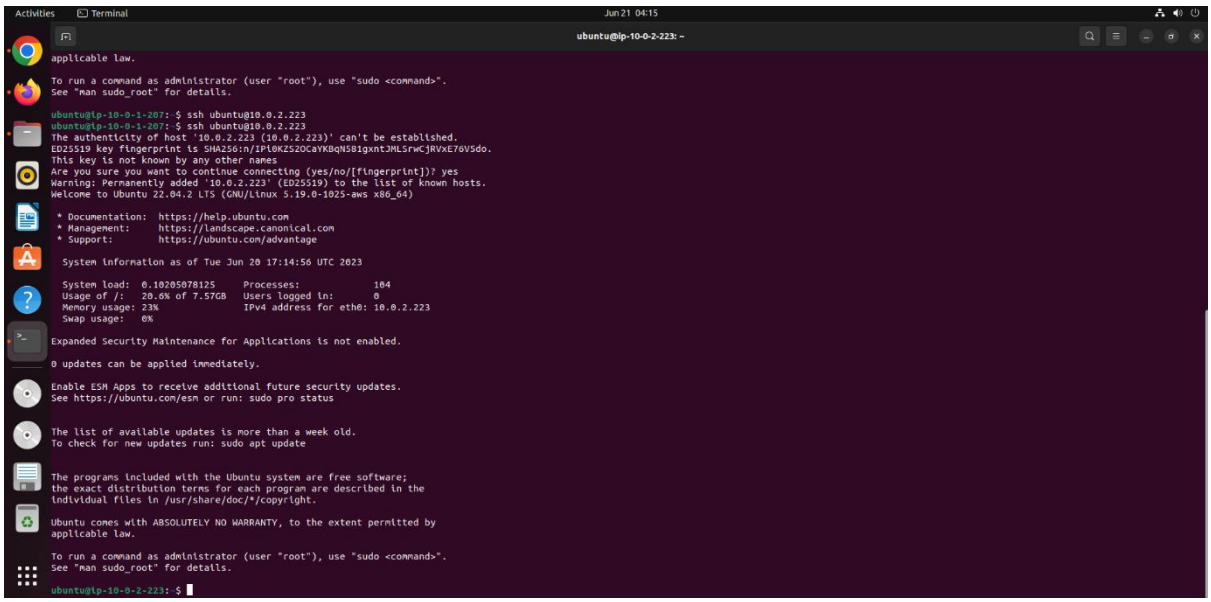
Adding of our private to the ssh agent



Here we are in the bastion host and we did not specify the private key



Thanks to the agent forwarding we are able to access to the private instance without specifying the private key



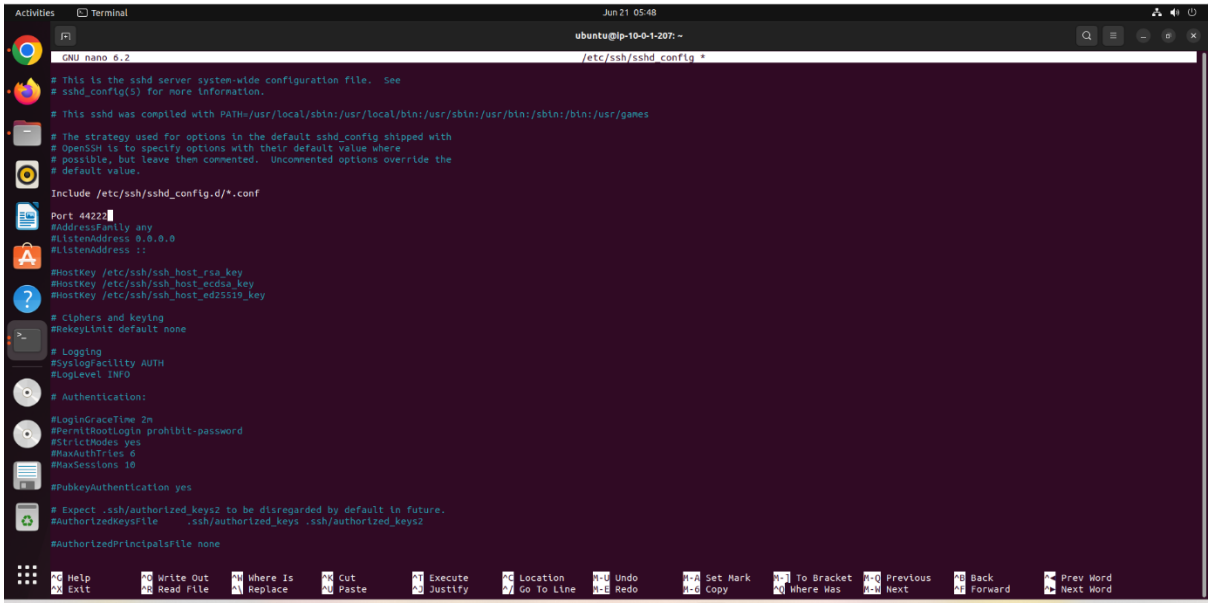
V. Security Hardening

- **SSH connexion**

To enforce the security of the SSH service, Changing the default port number of SSH (Secure Shell) is considered a security best practice. By default, SSH uses port 22 for communication, which is well-known and often targeted by malicious actors for brute-force attacks and automated scanning.

Changing the default SSH port can make it harder for attackers to find and target your SSH service. It adds an extra layer of security through "security through obscurity." While it shouldn't be relied upon as the sole security measure, it can be an effective deterrent against automated attacks.

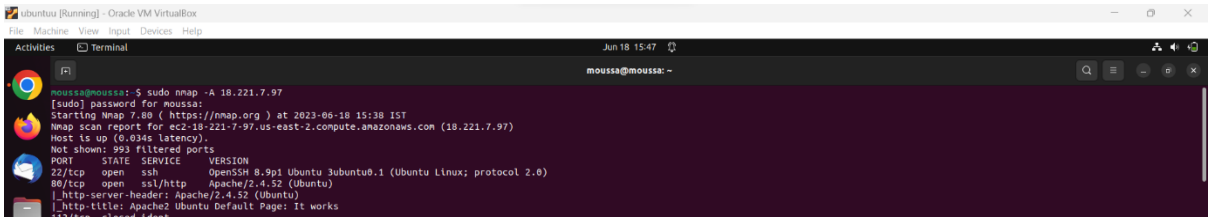
We have changed the SSH port 22 to 4422.



- **Hide apache2 server banner**

The Apache banner refers to the server identification information that is typically included in the HTTP response headers sent by an Apache web server. It provides details about the version of Apache being used and may also include additional information such as the operating system and other server software.

From our first nmap scan we can see the apache banner information

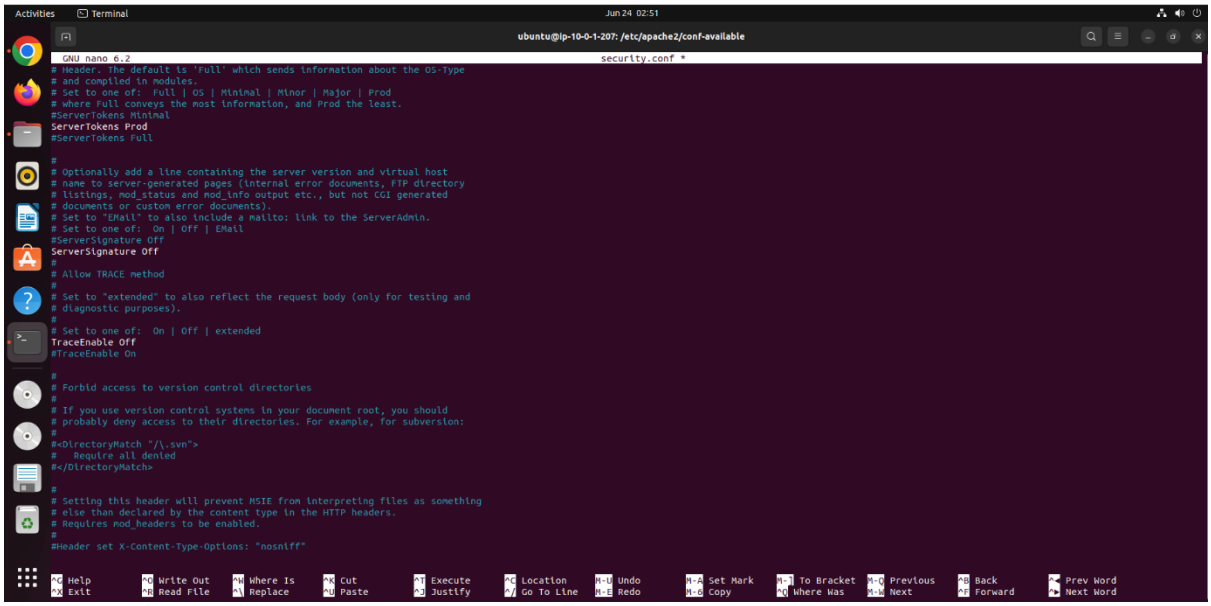


To hide the banner, we need to configure the file **/etc/apache2/apache2.conf** and add the following:

ServerTokens Prod

ServerSignature Off

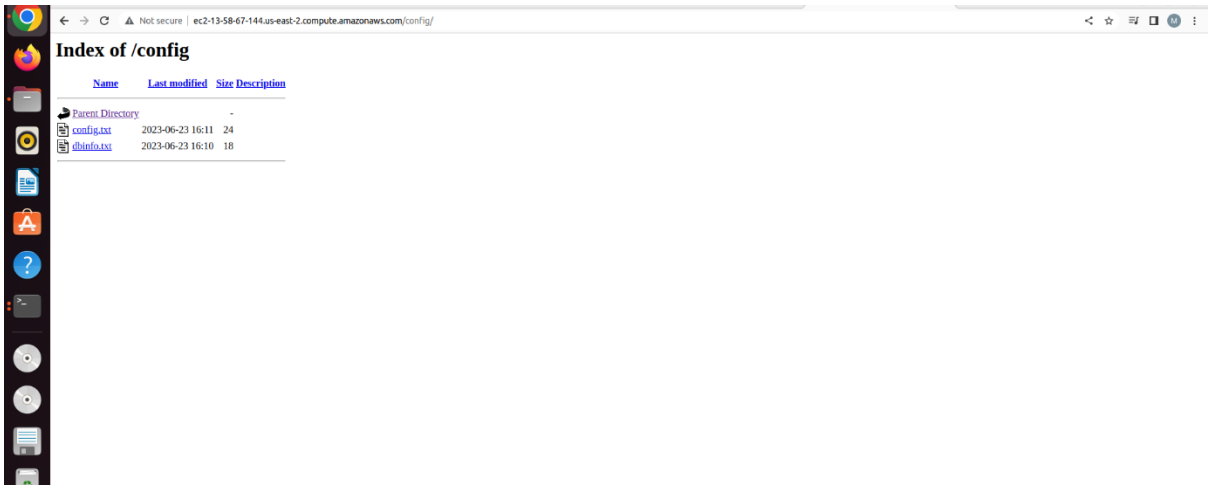
TraceEnable Off



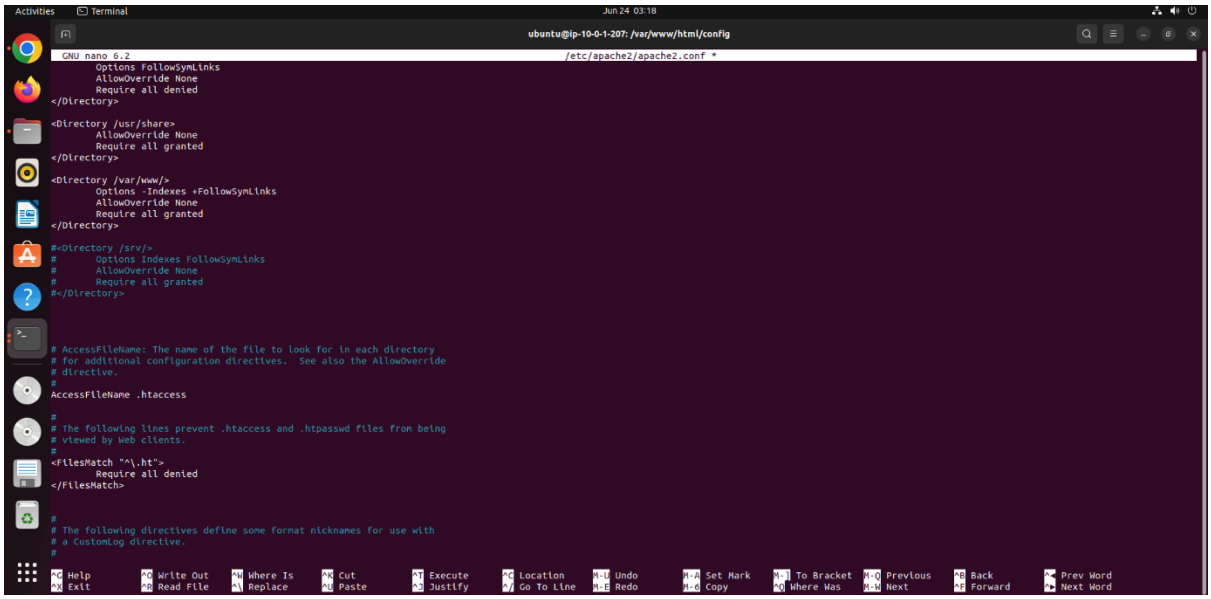
Now we can see that the nmap scan show only apache

- **Disable Apache directory browsing**

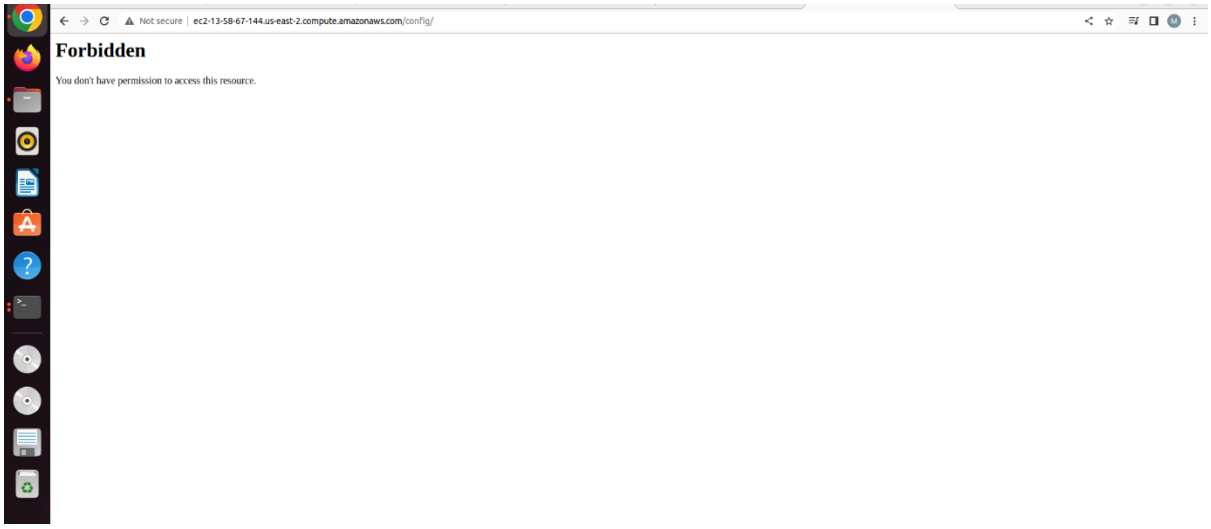
Directory browsing is a feature that can be enabled on a web server, which allows users to view the directory structure of a website without the need to enter any credentials. It is typically enabled on web servers to allow users to easily access files that they need, without having to know the precise path.



To disable the directory browsing we modify the file **/etc/apache2/apache2.conf** by adding "-" to Options -Indexes.



We can see that now we are not able to access the directory



This work is done by simply following AWS configuration best practices. It can be more enhanced by using others tools such as:

- **AWS WAF (Web Application Firewall)**
- **AWS Inspector**
- **AWS CloudWatch**
- **Amazon GuardDuty**
- **AWS Key Management Service (KMS)**
- **Etc…**