AWS CLOUD SOLUTIONS PROJECT 3

STEP-BY-STEP PROCESS OF SETTING UP A 3-TIER ARCHITECTURE OF A WEB APP

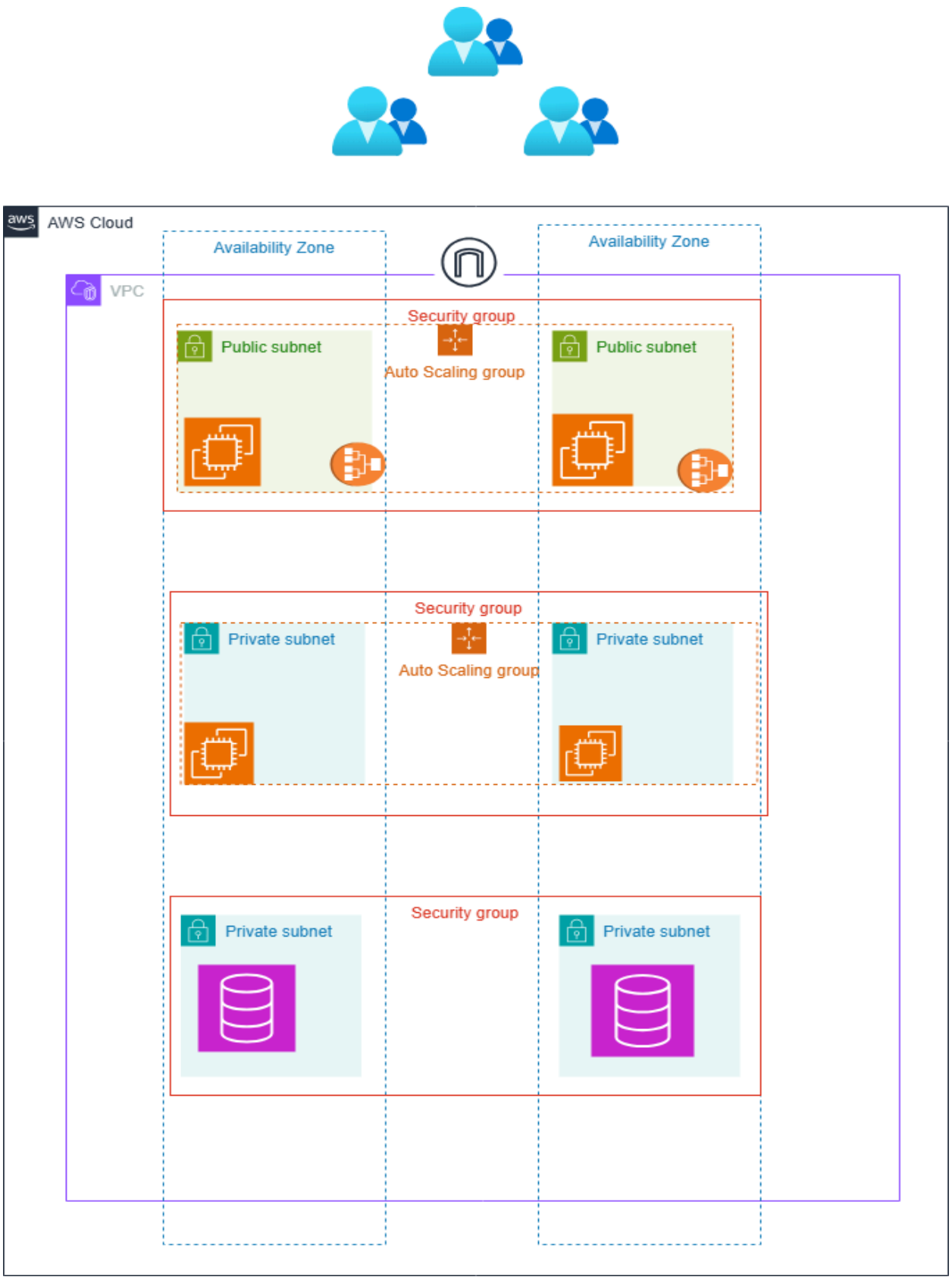
A 3-tier architecture is a software architecture pattern where the application is broken down into three logical tiers:
1. the presentation layer,
2. the application (business logic) layer and
3. the data storage layer.

This architecture is used in a client-server application such as a web application that has the frontend, the backend and the database. Each of these layers or tiers does a specific task and can be managed independently of each other.

In this project, I will be using the following AWS services to design and build a three-tier cloud infrastructure:
Elastic Compute Cloud (EC2), Auto Scaling Group, Virtual Private Cloud (VPC), Elastic Load Balancer (ELB), Security Groups and the Internet Gateway, NAT Gateway, Health check.

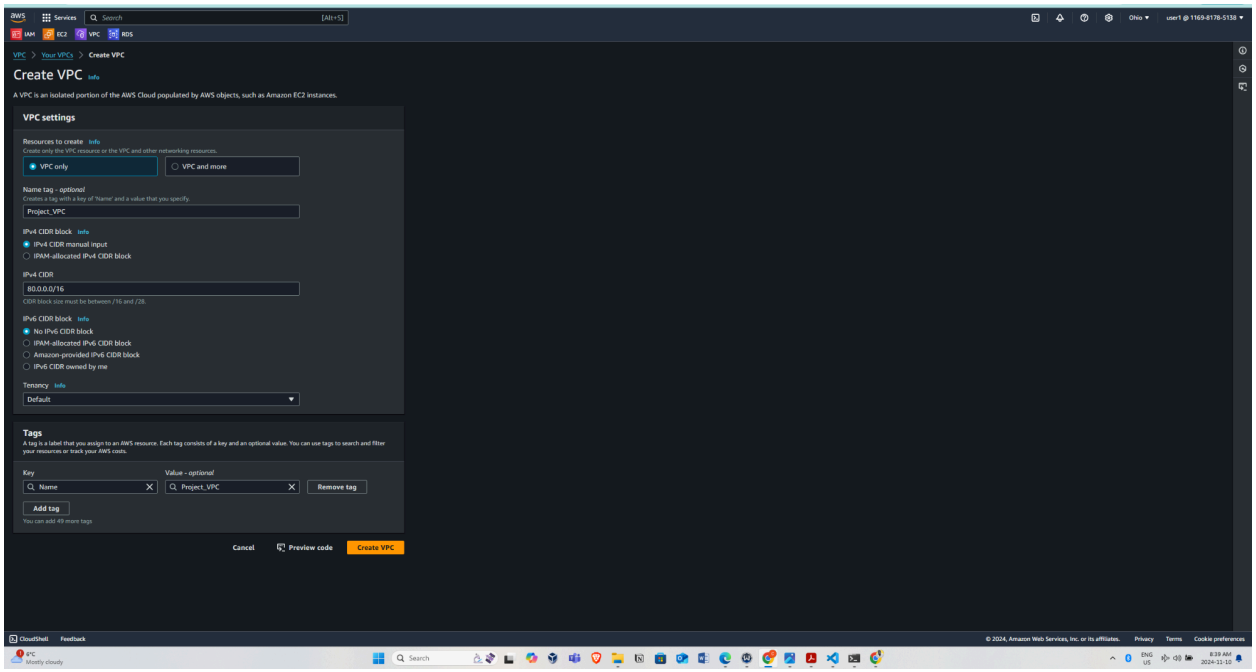
This infrastructure will be designed to be secured, highly available and fault tolerant.

Below is the architectural diagram of this project:


AWS Cloud
Availability Zone
Availability Zone
VPC
Security group
Public subnet
Auto Scaling group
Public subnet
Security group
Private subnet
Auto Scaling group
Private subnet
Security group
Private subnet
Private subnet

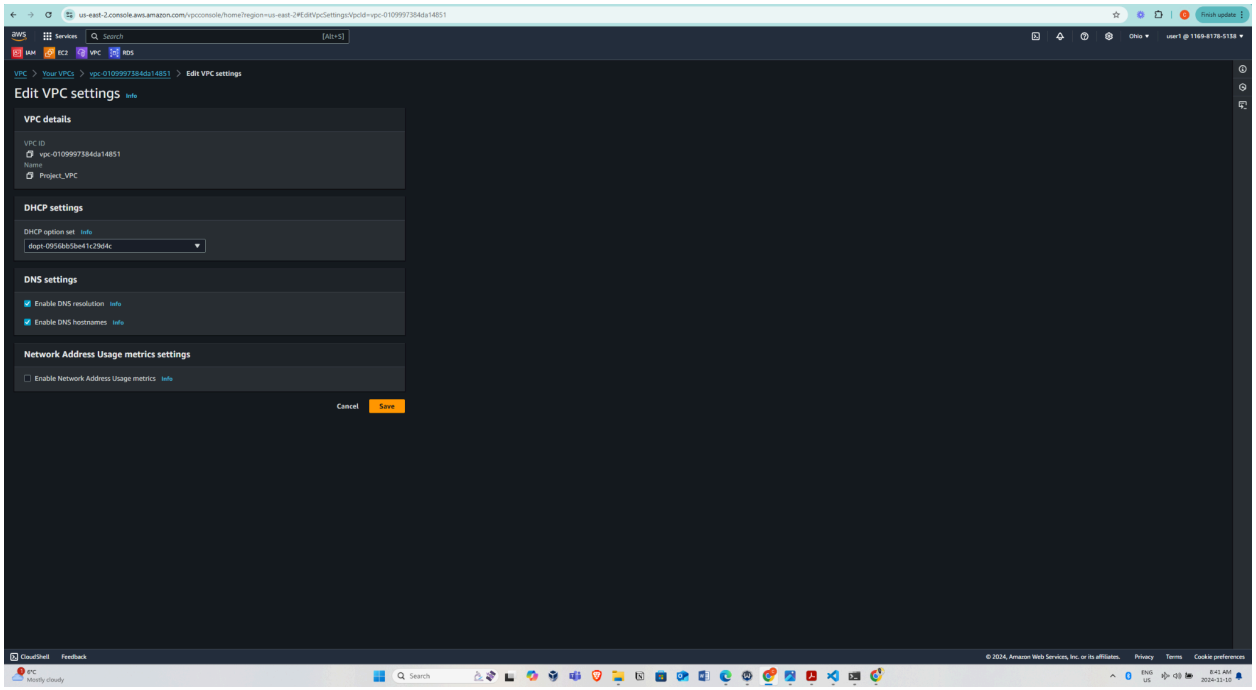
Step 1

Creating a VPC



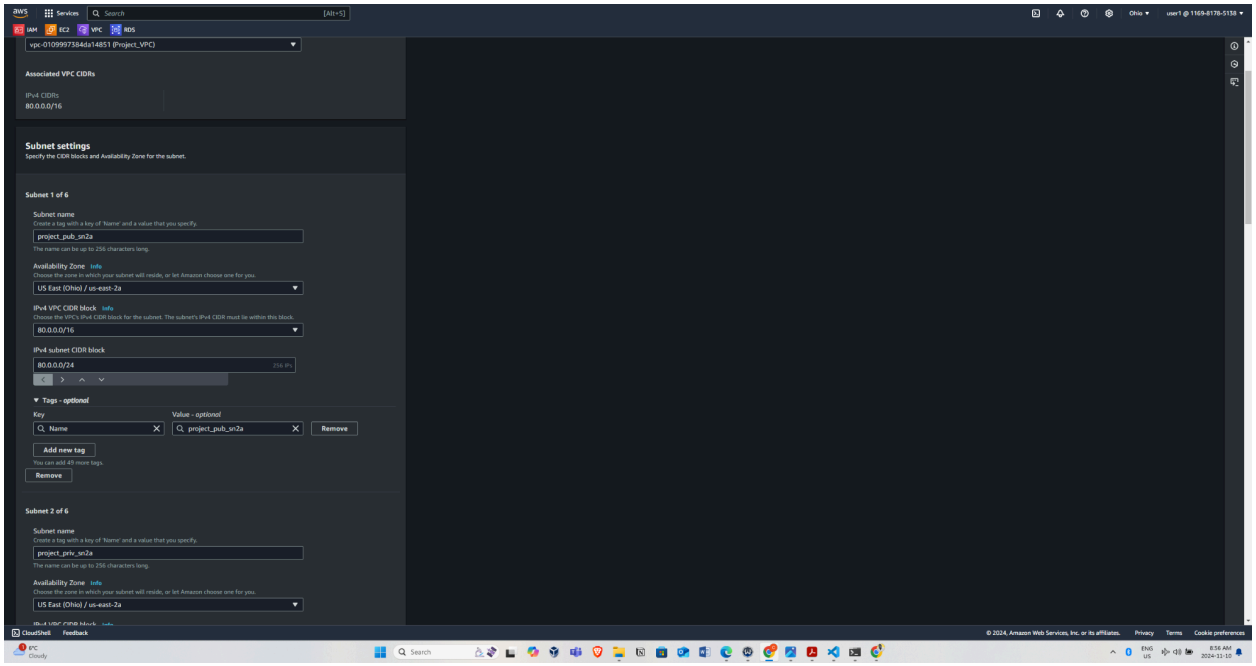
Step 1b

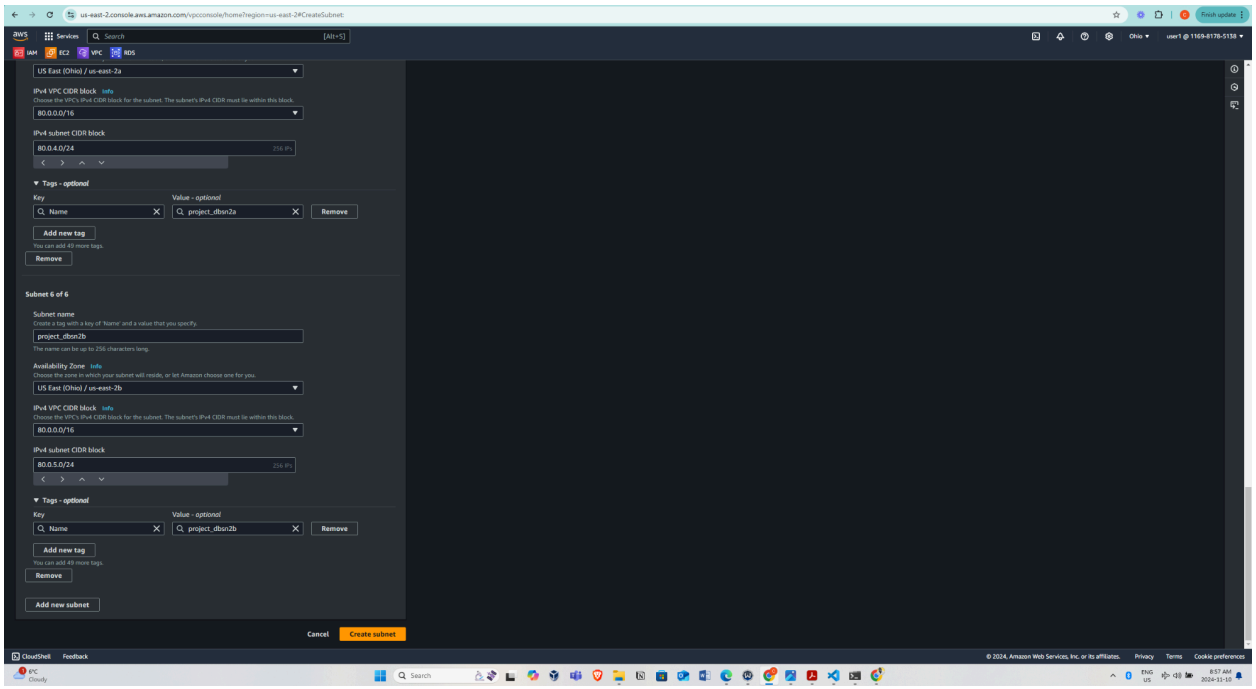
Edit VPC settings and enable DNS host names. This would allow resources provisioned within this VPC that require a DNS host name to be allocated a DNS host name.

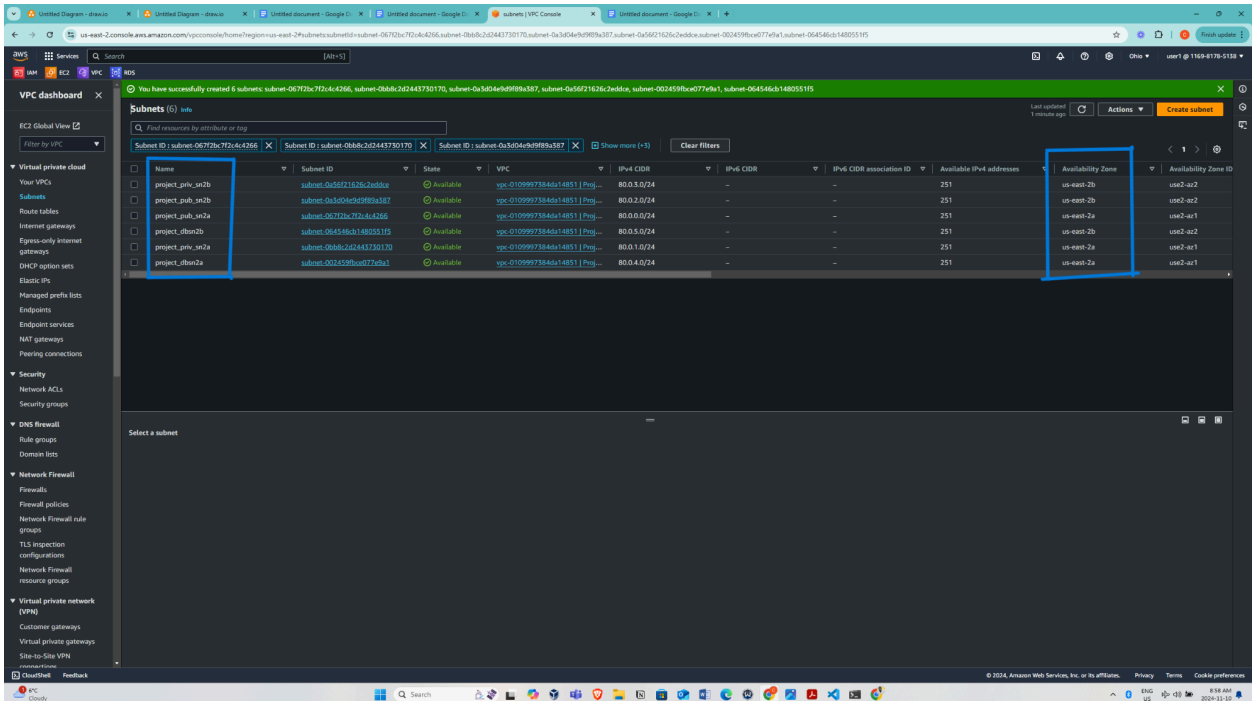


Step 3

Create 6 subnets in 2 availability zones. I will be creating 2 public subnets in each AZs, 2 private subnets in each AZs and 2 database subnets which would still be private in each Azs. This makes a total of 6 subnets.

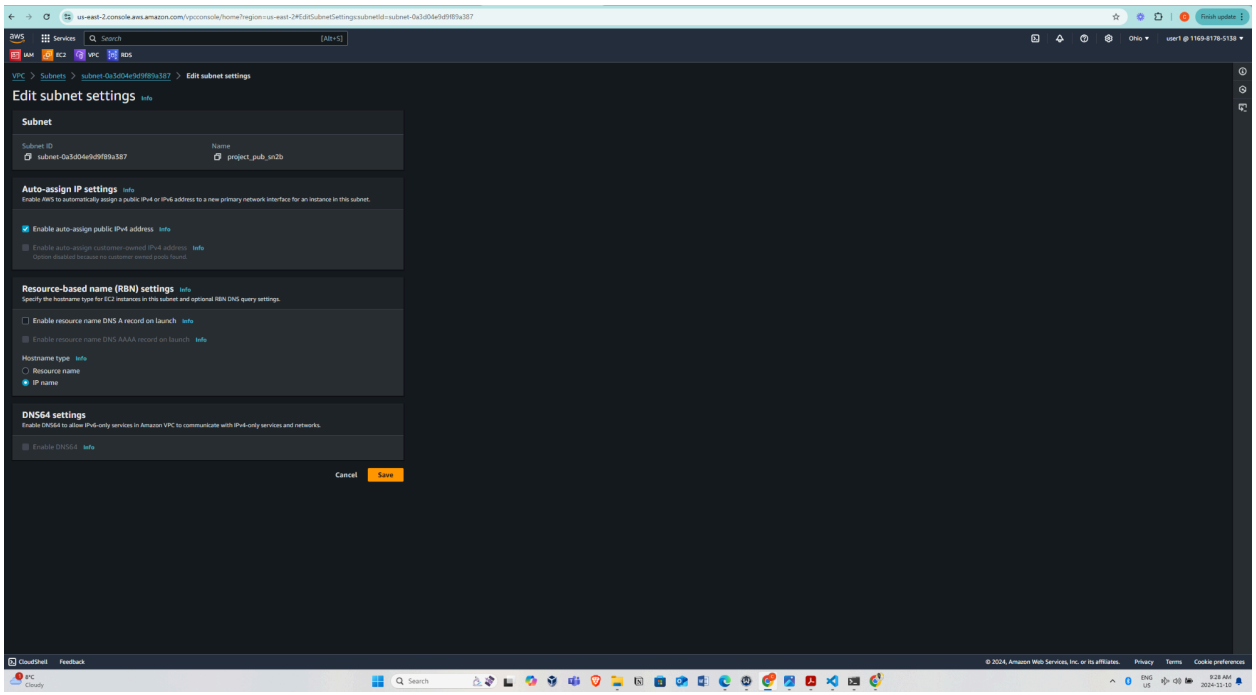






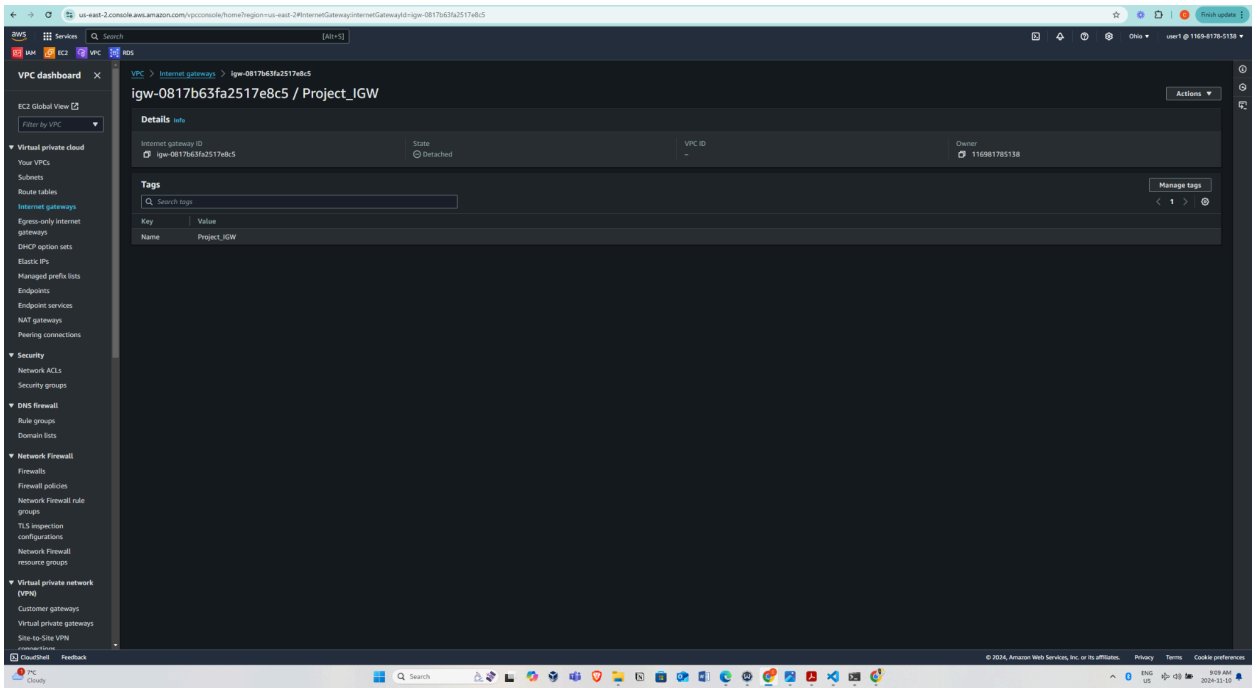
Step 3b

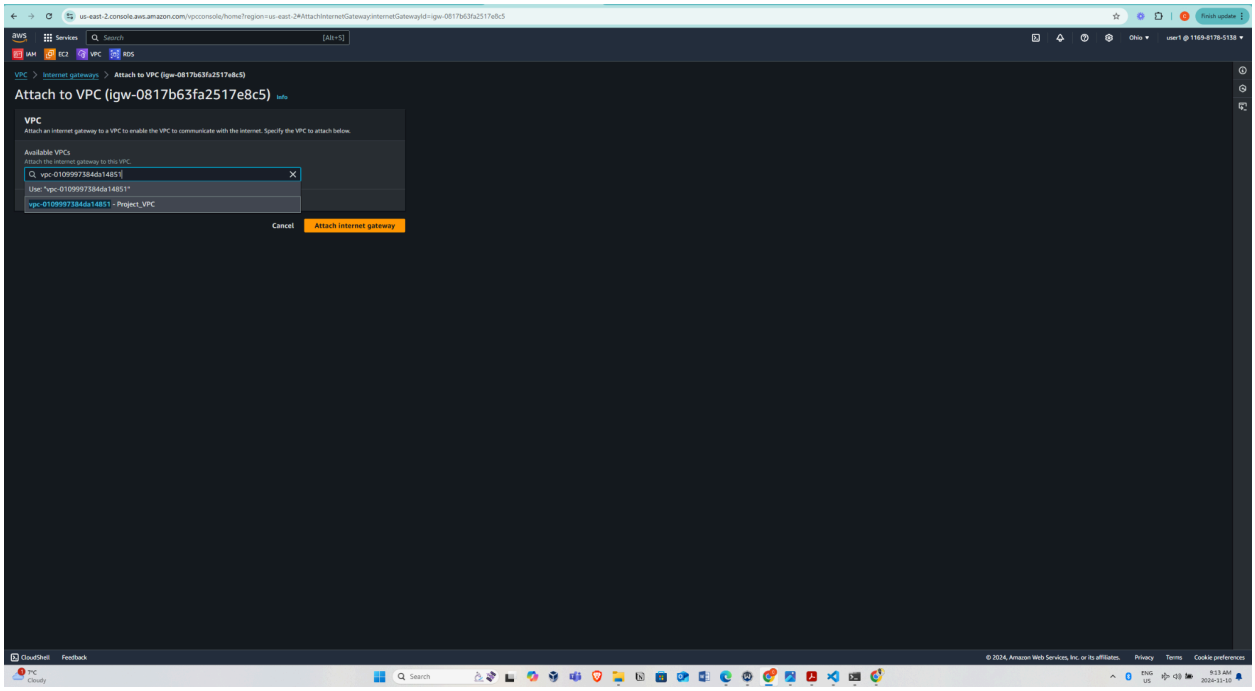
Edit subnet settings of the public subnets and enable auto-assign IP addresses. This would dynamically allocate public IP addresses to resources provisioned within the public subnets.



Step 4

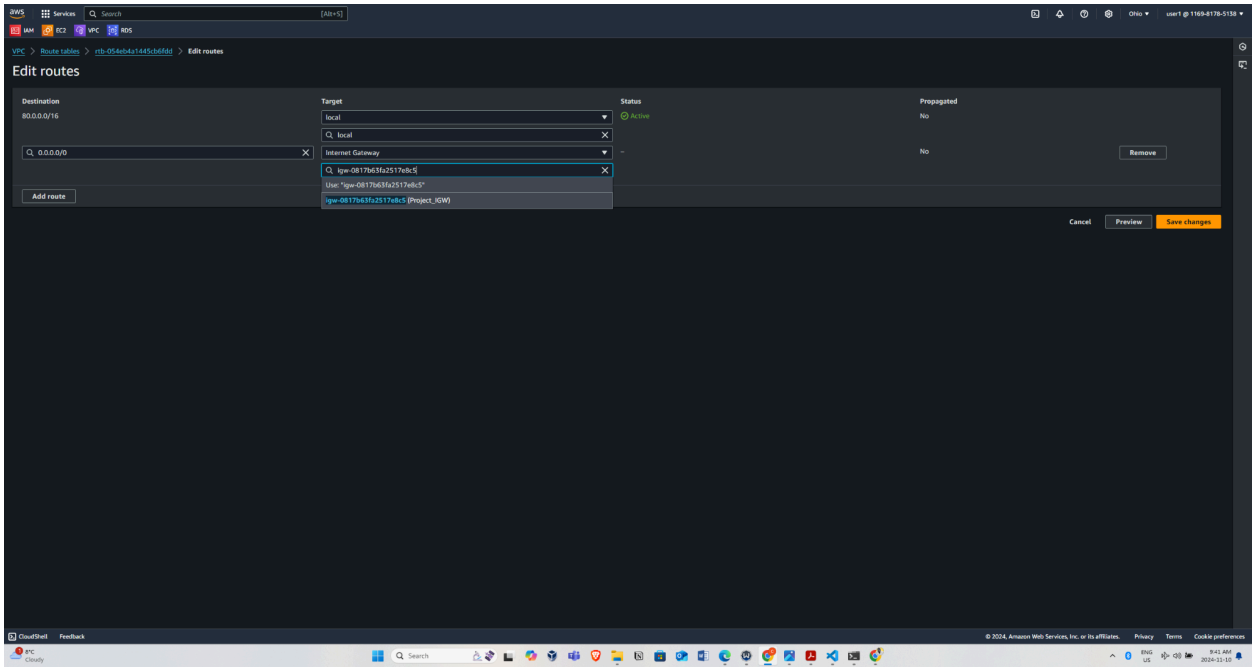
Create an internet gateway and attach it to the VPC

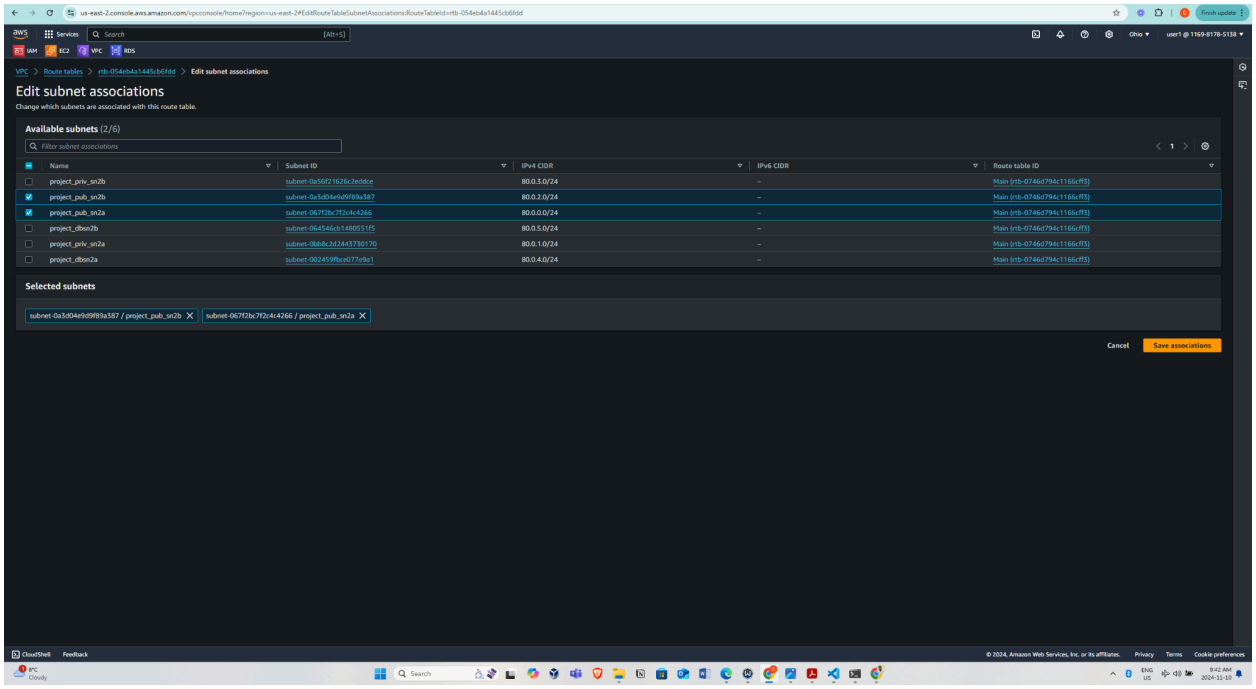




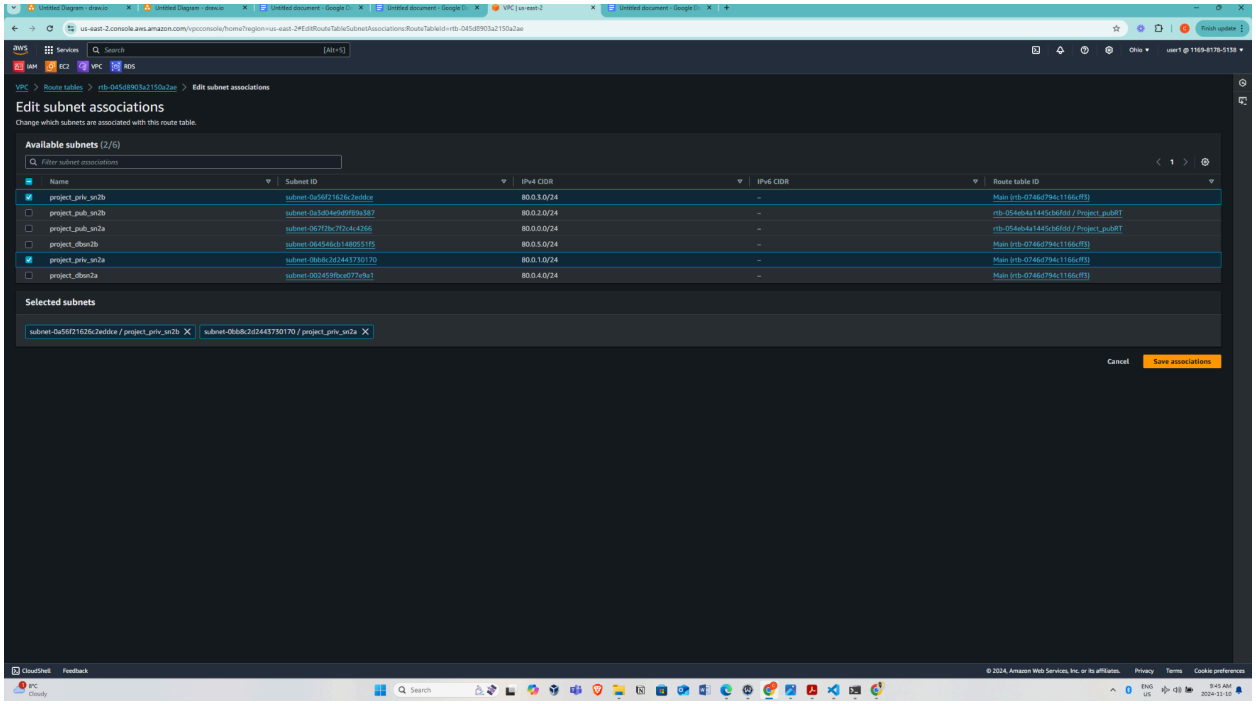
Step 5

Create 2 route tables and associate them with their respective subnets. A public route table which will have a route pointing to the IGW and a private route table with a route pointing to Nat_GW.



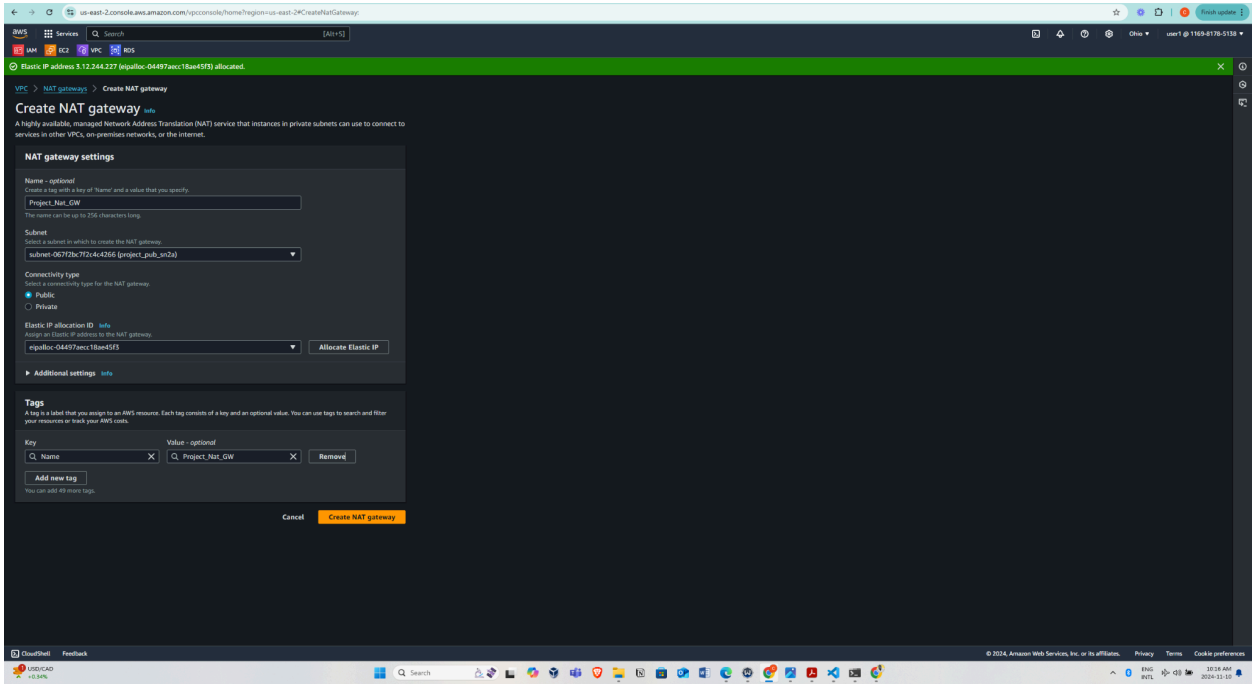


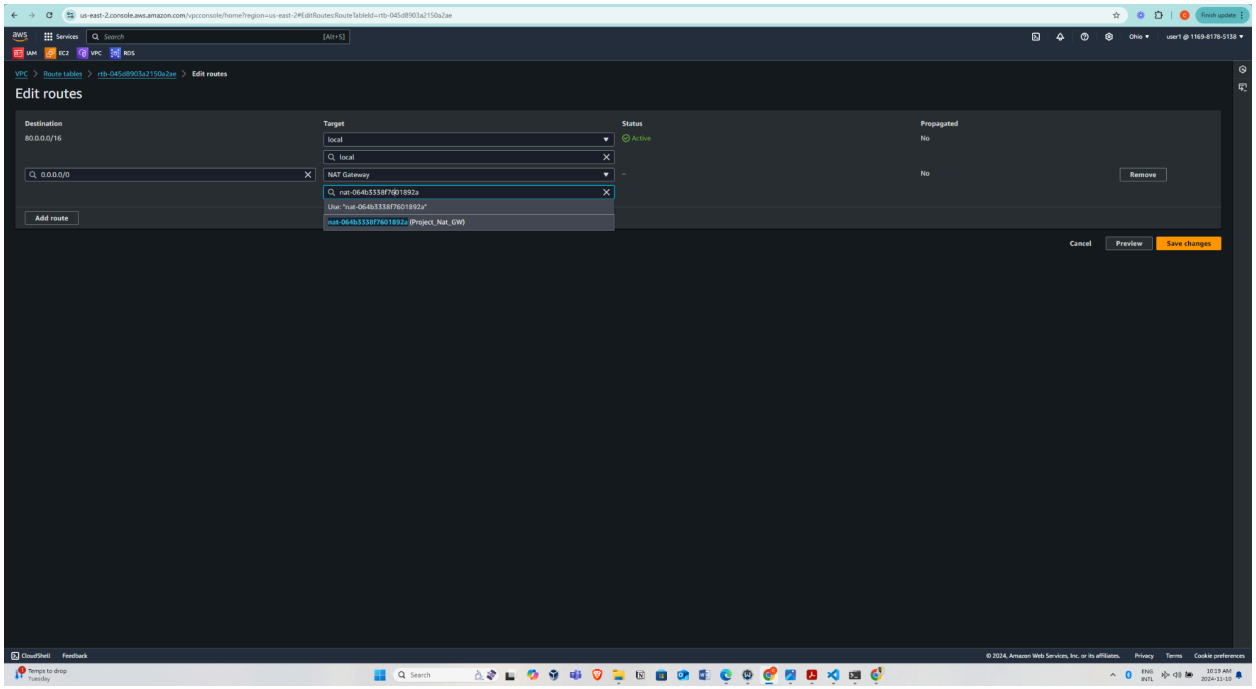
Private subnet association



Step 6

Create a Nat_GW in one of the public Subnets. Then add a new route to the private route table pointing to the Nat_GW. This will enable resources in the private subnet which needs to go to the internet for legitimate reasons to have a route to access the internet.

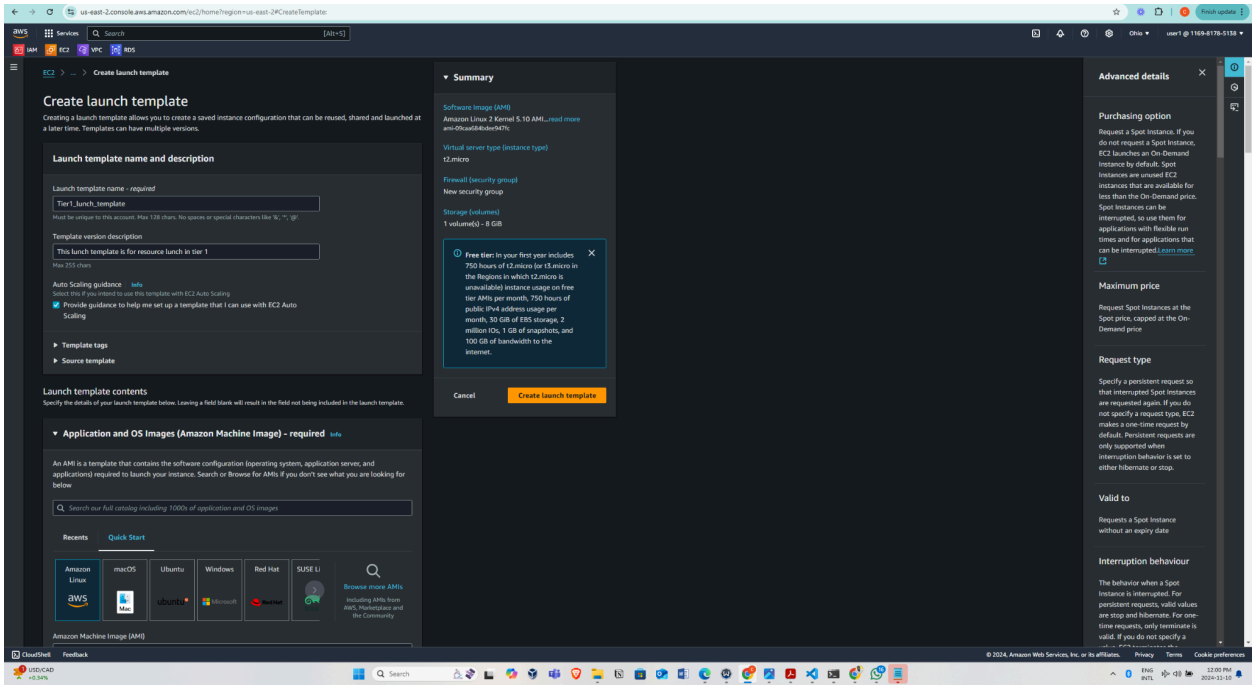


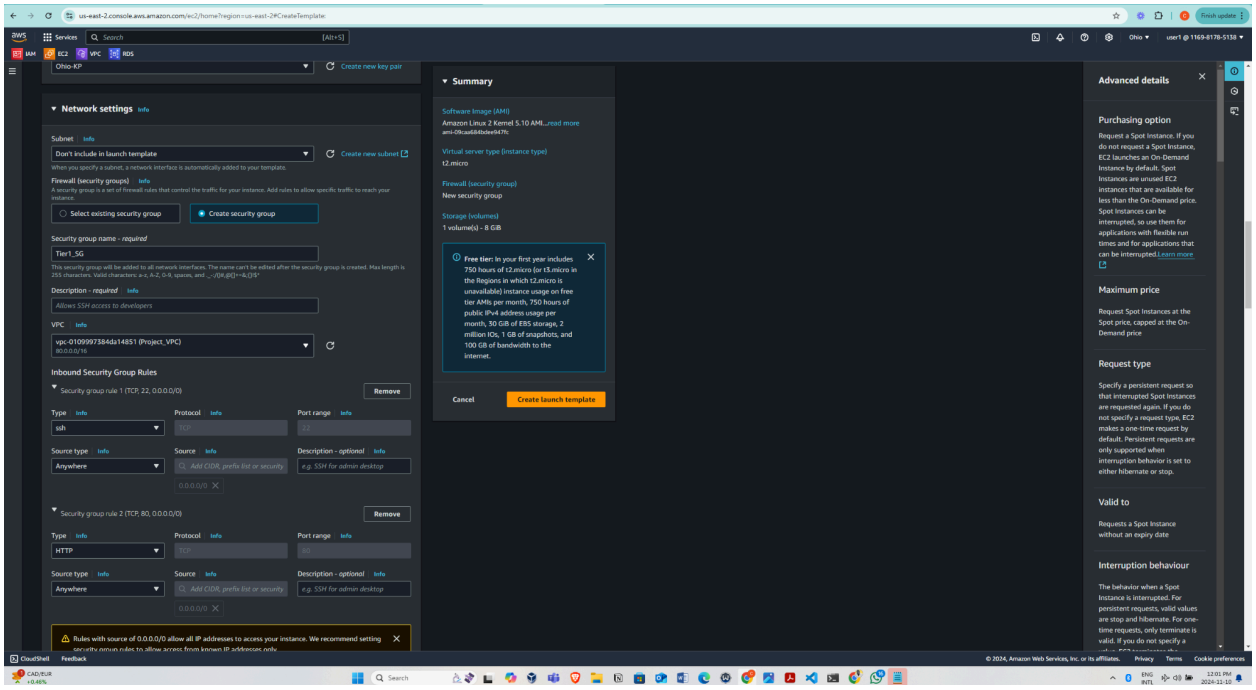


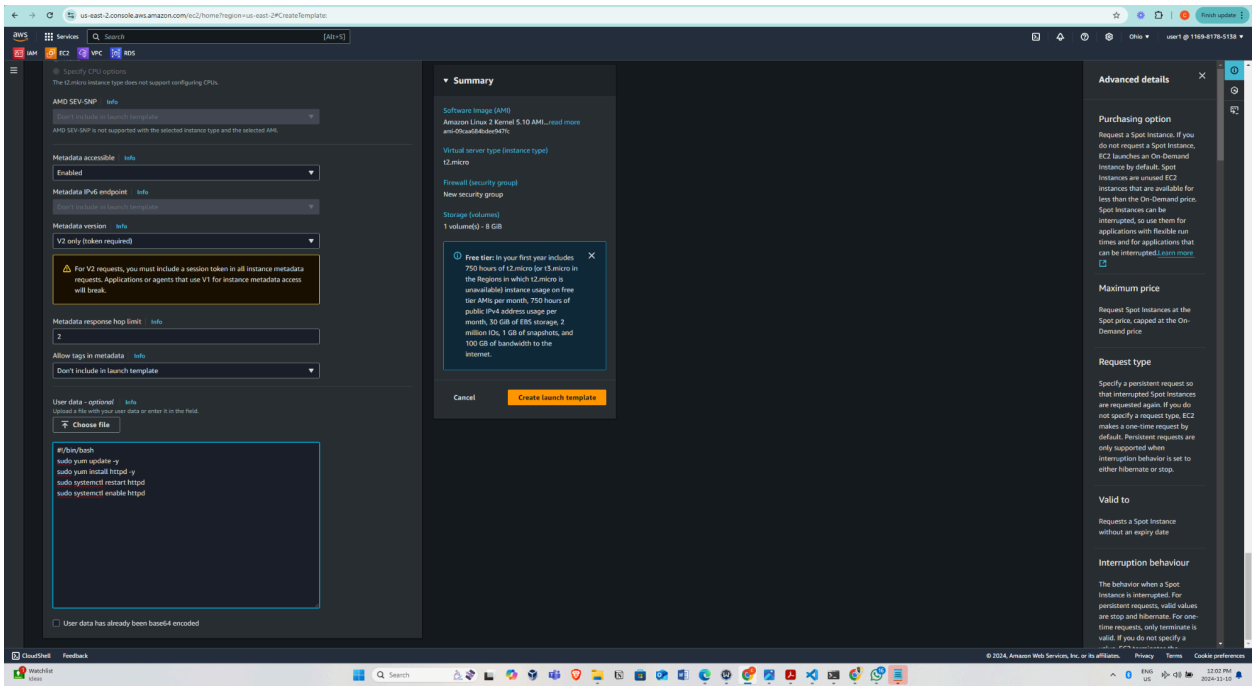
Tier 1 Set up: This is the presentation Tier

Step 1

Create a lunch template for Tier 1

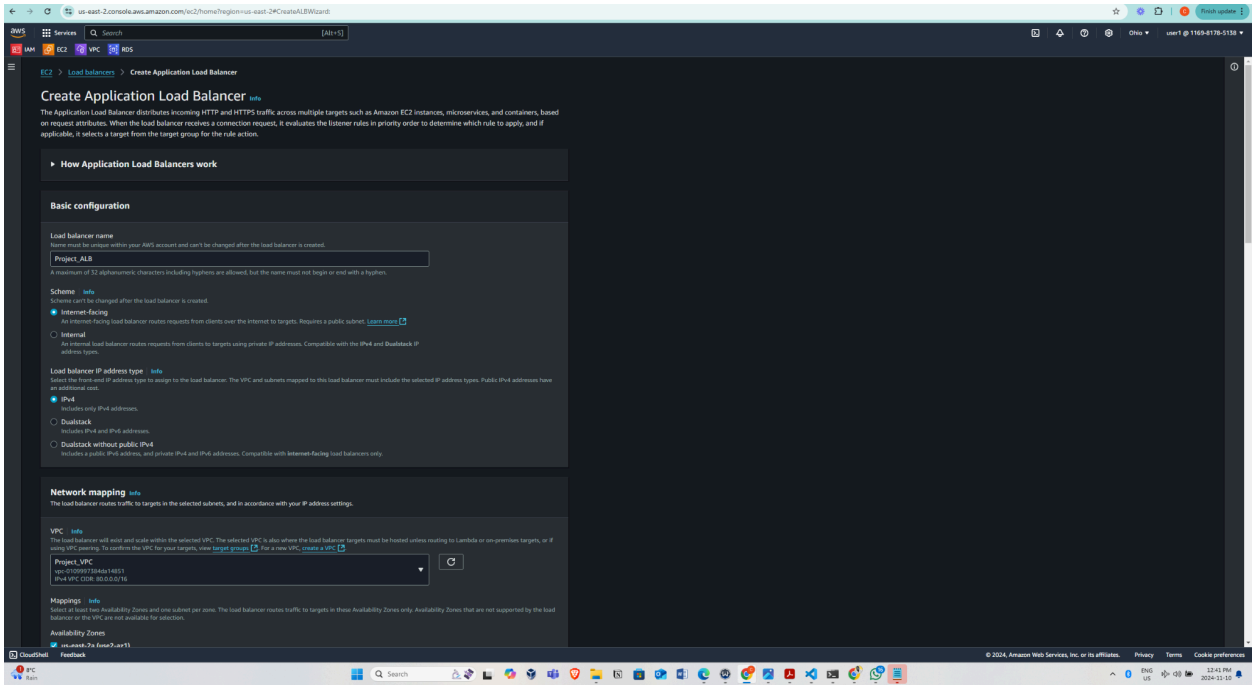


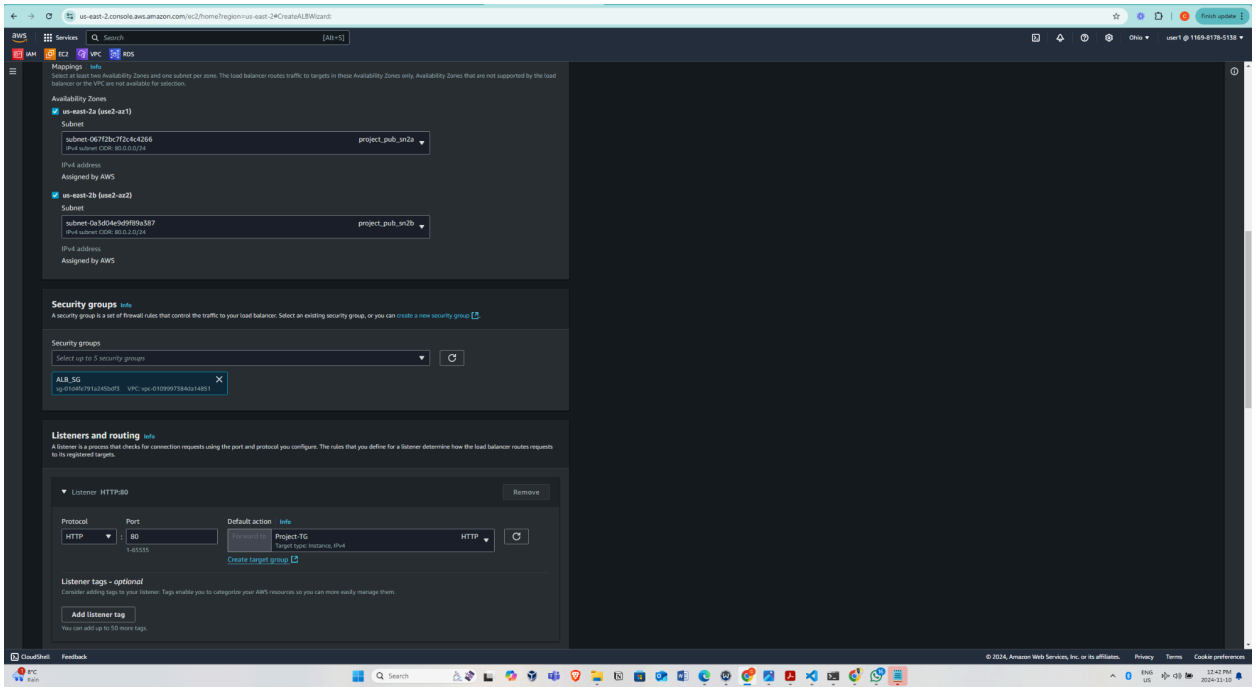

Ohio-KP
Create new key pair
Network settings
Subnet
Don't include in launch template
Create new subnet
Firewall (security groups)
Select existing security group
Create security group
Security group name - required
Tier1_SG
Description - required
Allows SSH access to developers
VPC
vpc-0109997384da14851 (Project_VPC)
Inbound Security Group Rules
Security group rule 1 (TCP, 22, 0.0.0.0/0)
Remove
Type
ssh
Protocol
TCP
Port range
22
Source type
Anywhere
Source
Add CIDR, prefix list or security
Description - optional
e.g. SSH for admin desktop
0.0.0.0/0
Security group rule 2 (TCP, 80, 0.0.0.0/0)
Remove
Type
HTTP
Protocol
TCP
Port range
80
Source type
Anywhere
Source
Add CIDR, prefix list or security
Description - optional
e.g. SSH for admin desktop
0.0.0.0/0
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting
Summary
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-09caa6843bdee947c
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Create launch template
Advanced details
Purchasing option
Request a Spot Instance. If you do not request a Spot Instance, EC2 launches an On-Demand Instance by default. Spot Instances are unused EC2 instances that are available for less than the On-Demand price. Spot Instances can be interrupted, so use them for applications with flexible run times and for applications that can be interrupted.Learn more
Maximum price
Request Spot Instances at the Spot price, capped at the On-Demand price
Request type
Specify a persistent request so that interrupted Spot Instances are requested again. If you do not specify a request type, EC2 makes a one-time request by default. Persistent requests are only supported when interruption behavior is set to either hibernate or stop.
Valid to
Requests a Spot Instance without an expiry date
Interruption behaviour
The behavior when a Spot Instance is interrupted. For persistent requests, valid values are stop and hibernate. For one-time requests, only terminate is valid. If you do not specify a
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Specify CPU options
The t2.micro instance type does not support configuring CPUs.
AMD SEV-SNP
Don't include in launch template
AMD SEV-SNP is not supported with the selected instance type and the selected AMI.
Metadata accessible
Enabled
Metadata IPv6 endpoint
Don't include in launch template
Metadata version
V2 only (token required)
For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.
Metadata response hop limit
2
Allow tags in metadata
Don't include in launch template
User data - optional
Upload a file with your user data or enter it in the field.
Choose file
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo systemctl restart httpd
sudo systemctl enable httpd
User data has already been base64 encoded
Summary
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-09caa6843bdee947c
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Create launch template
Advanced details
Purchasing option
Request a Spot Instance. If you do not request a Spot Instance, EC2 launches an On-Demand Instance by default. Spot Instances are unused EC2 instances that are available for less than the On-Demand price. Spot Instances can be interrupted, so use them for applications with flexible run times and for applications that can be interrupted.Learn more
Maximum price
Request Spot Instances at the Spot price, capped at the On-Demand price
Request type
Specify a persistent request so that interrupted Spot Instances are requested again. If you do not specify a request type, EC2 makes a one-time request by default. Persistent requests are only supported when interruption behavior is set to either hibernate or stop.
Valid to
Requests a Spot Instance without an expiry date
Interruption behaviour
The behavior when a Spot Instance is interrupted. For persistent requests, valid values are stop and hibernate. For one-time requests, only terminate is valid. If you do not specify a
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step2

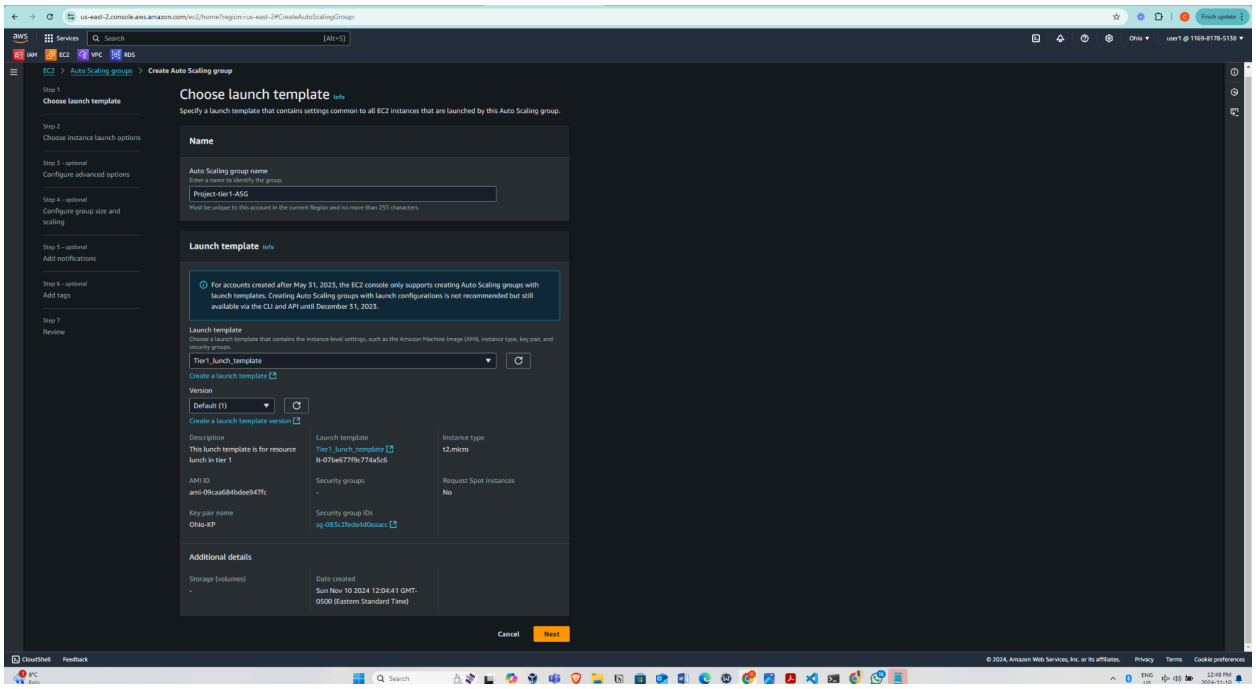
Create an internet facing ALB and create a SG for the ALB allowing only HTTP traffic.

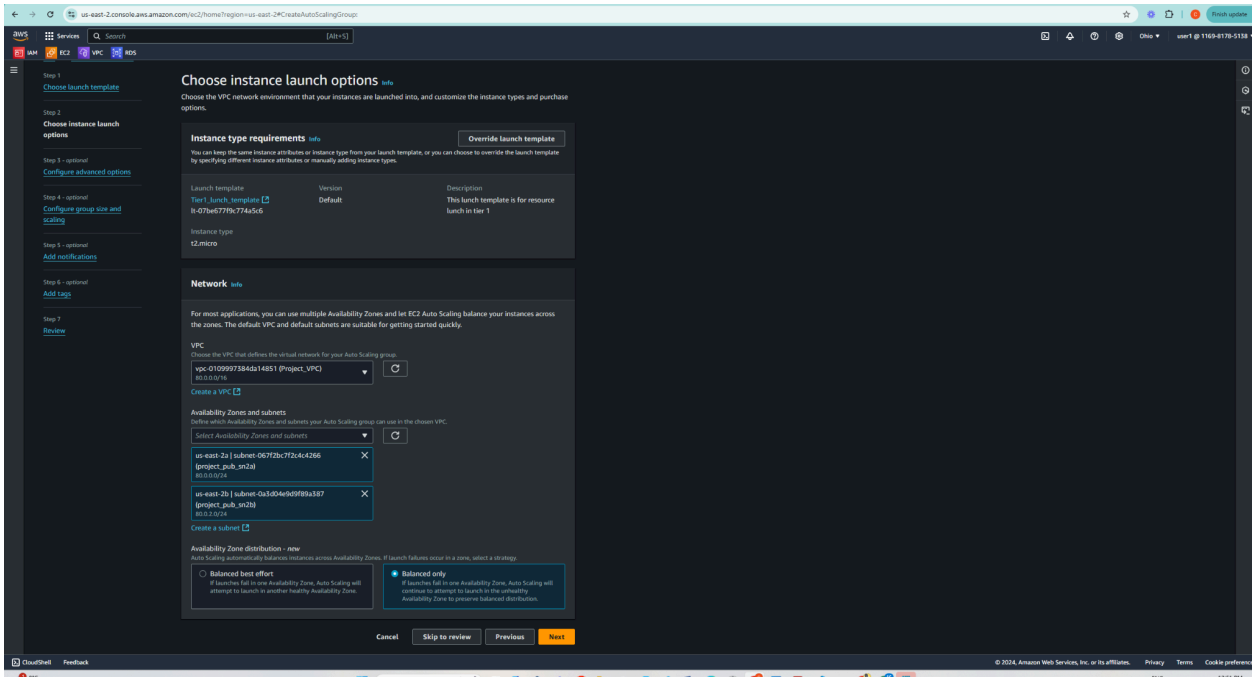


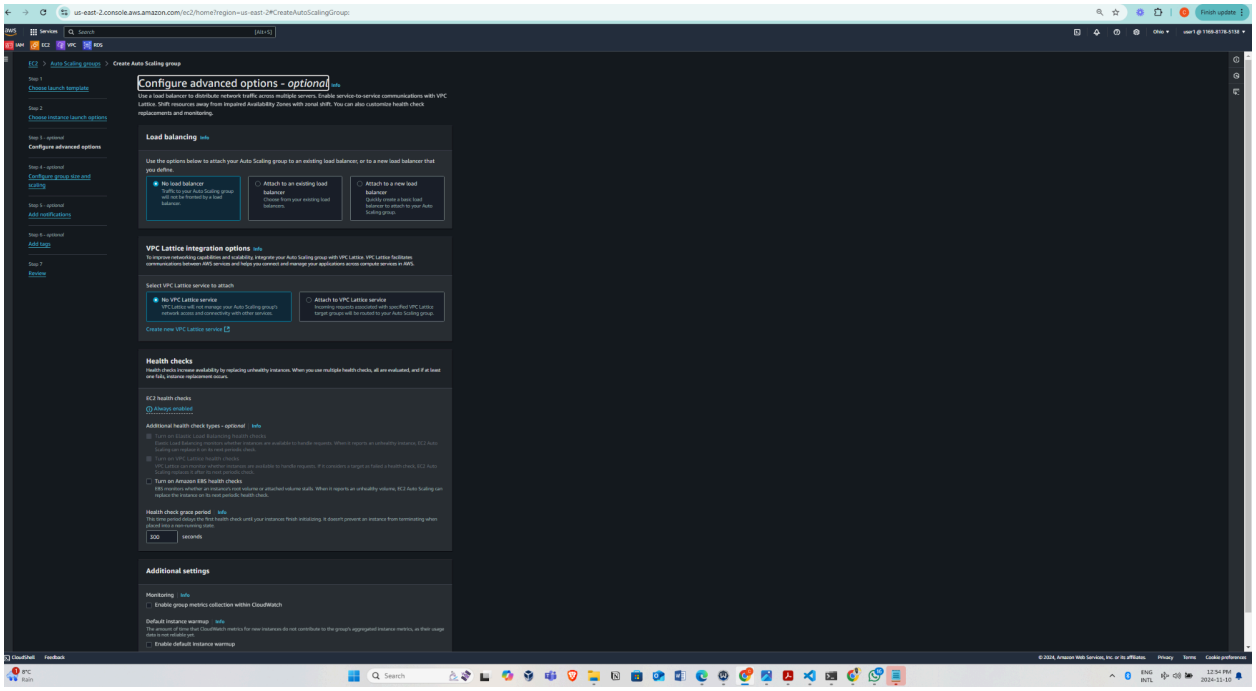


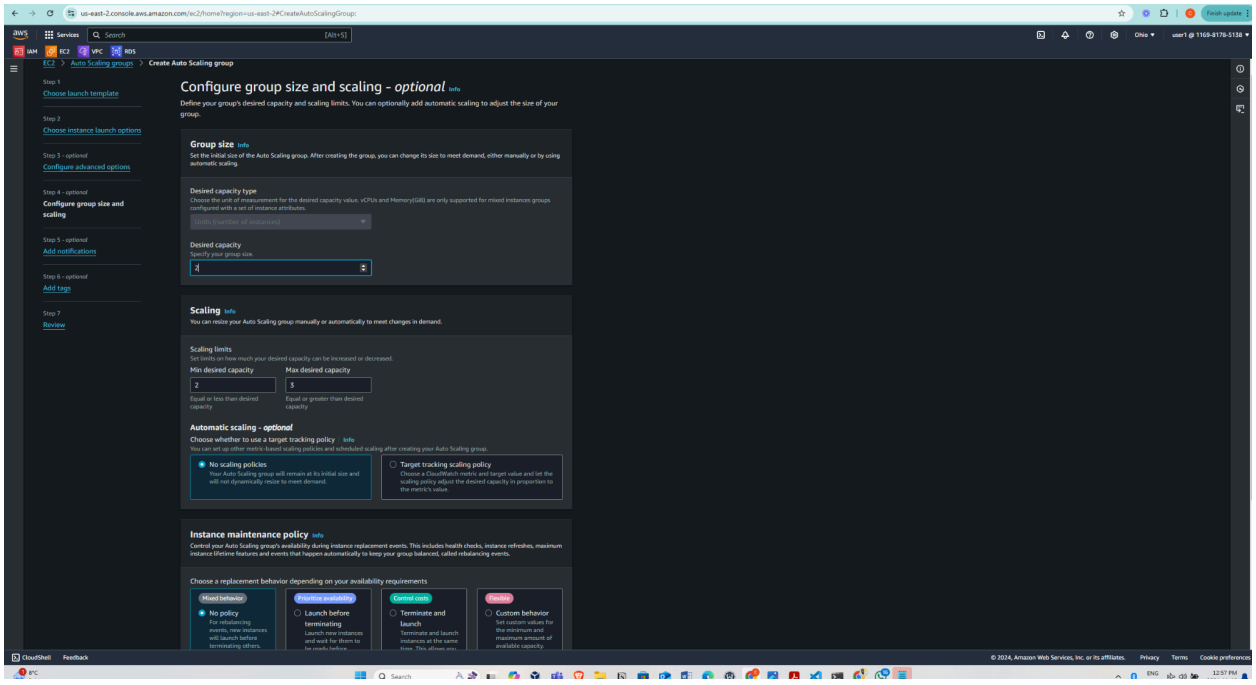
Step 3

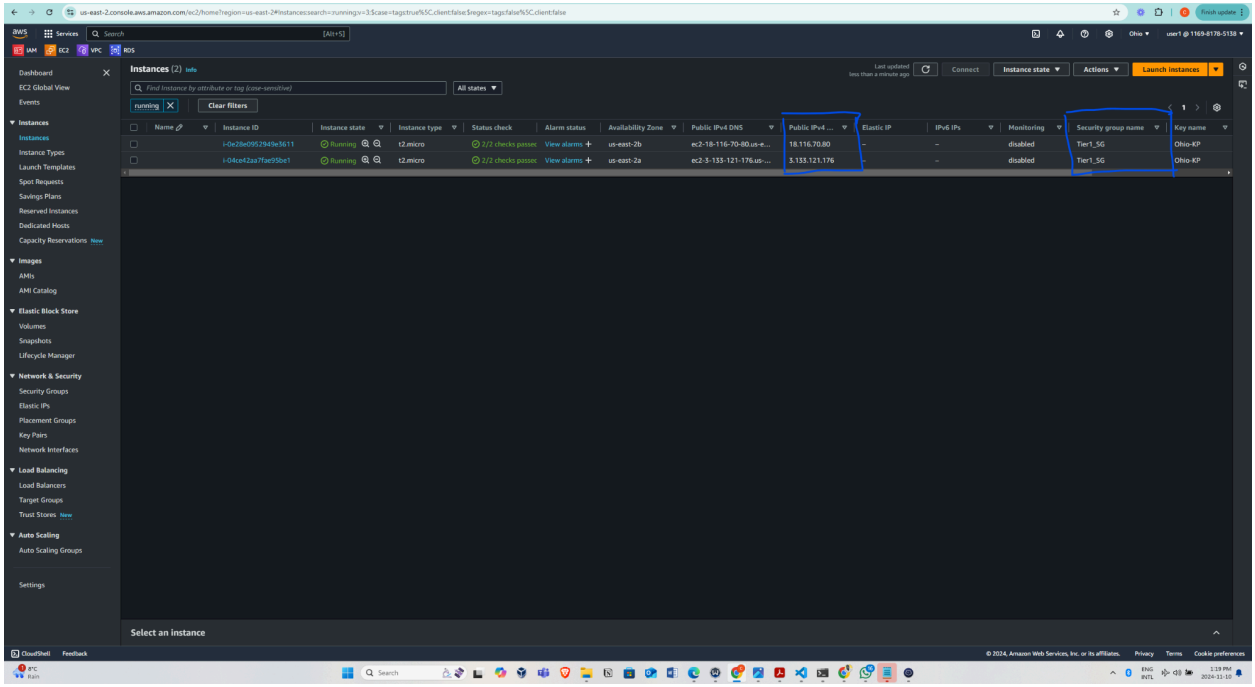
Create an auto scaling group for Tier 1





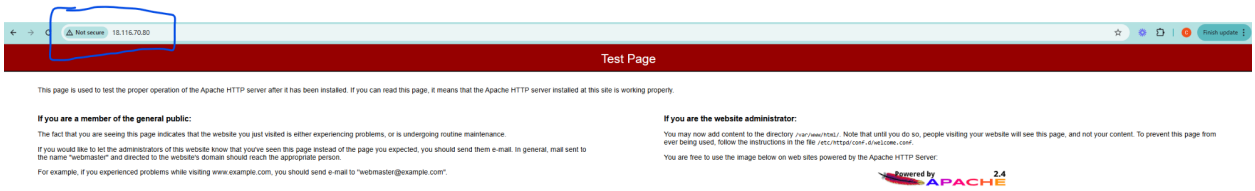

Configure advanced options - optional
Load balancing
VPC Lattice integration options
Health checks
Additional settings

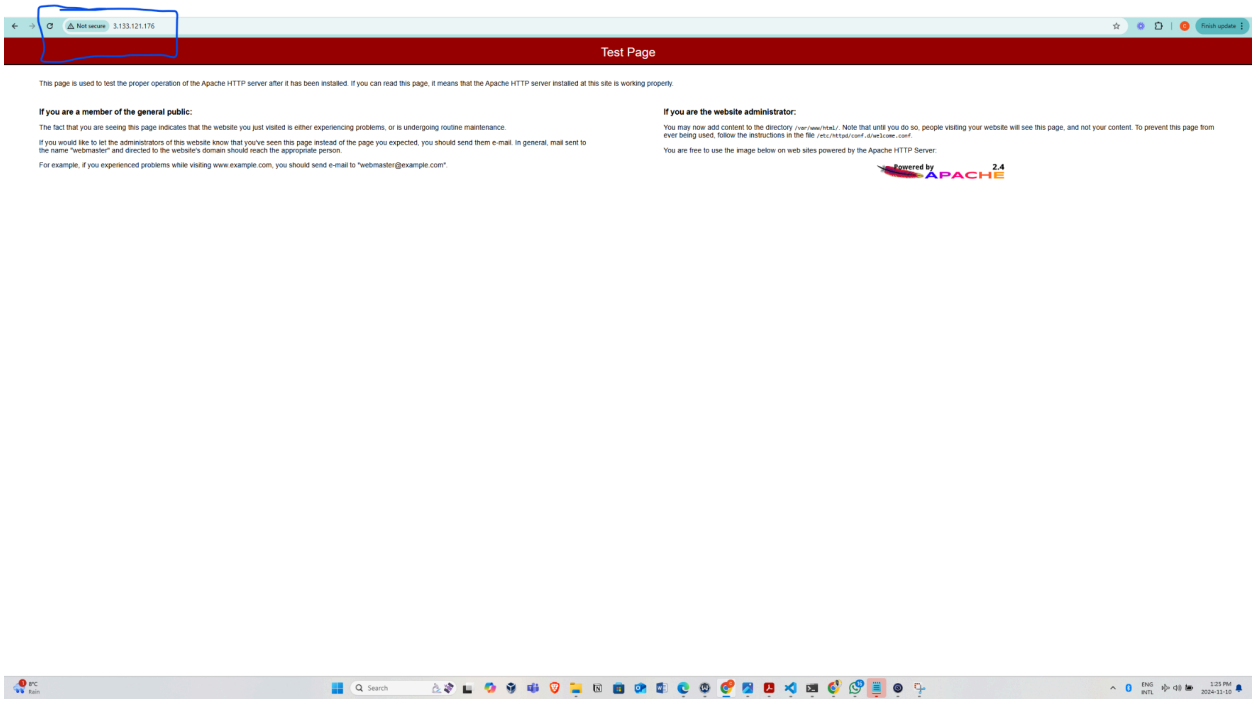

Configure group size and scaling - optional
Group size
Scaling
Instance maintenance policy



Step 4

Browsing for both Instances

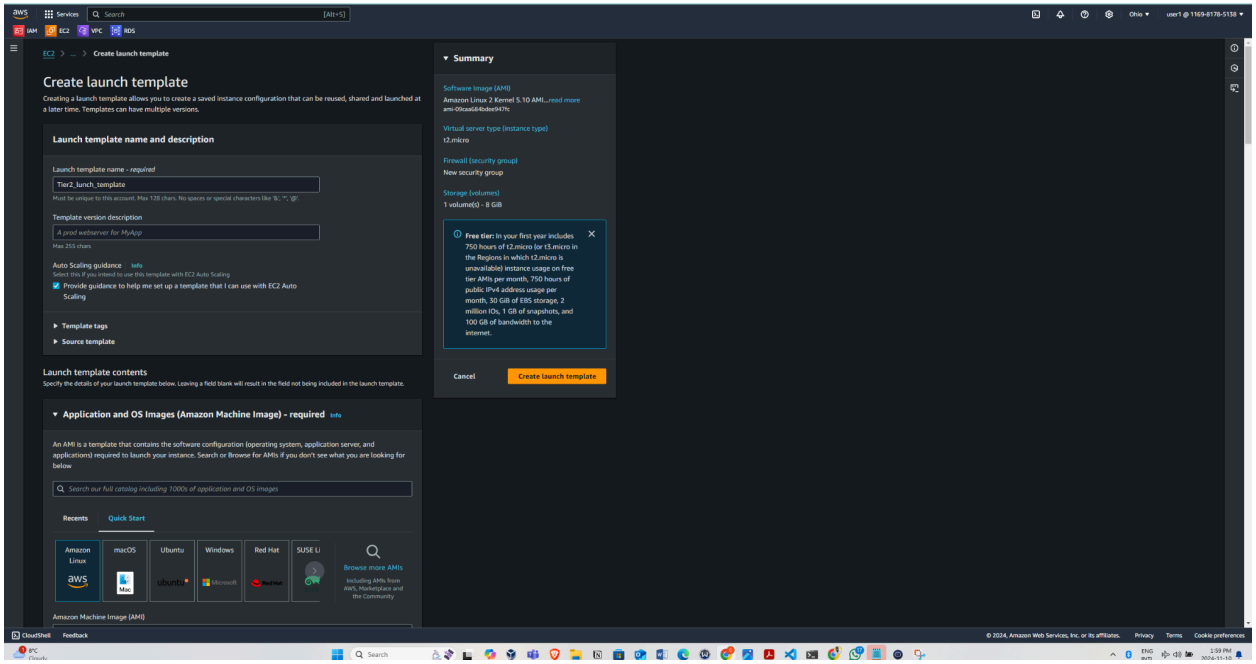


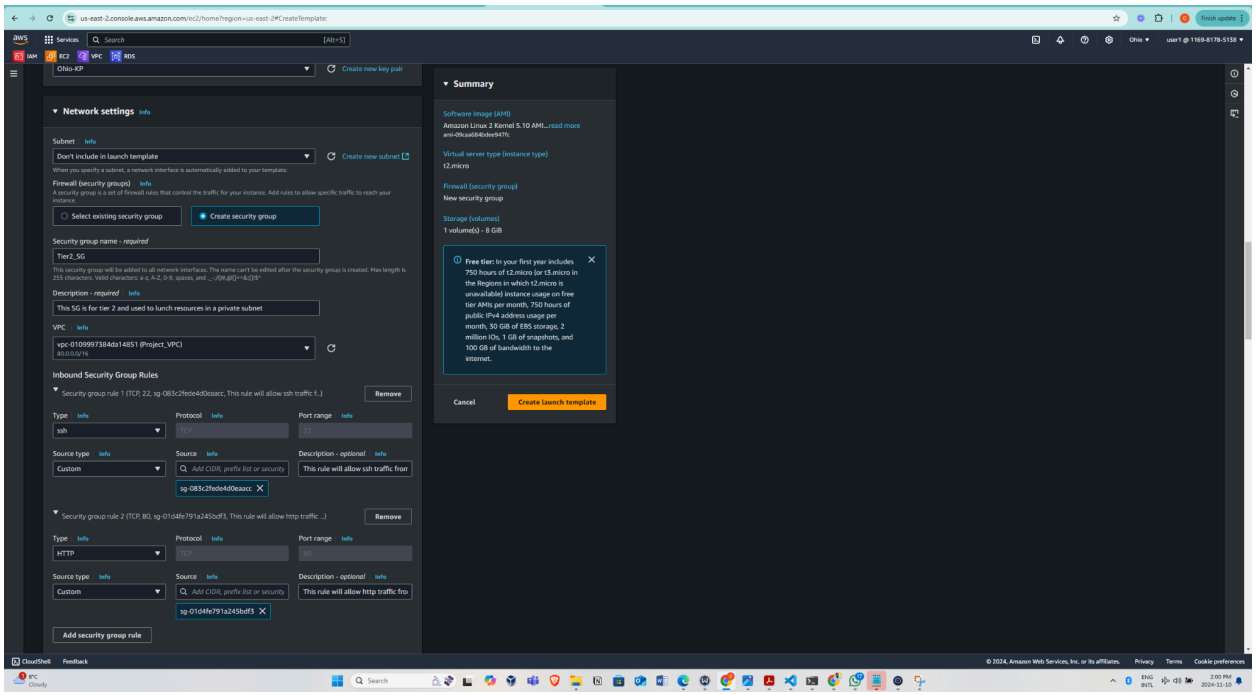


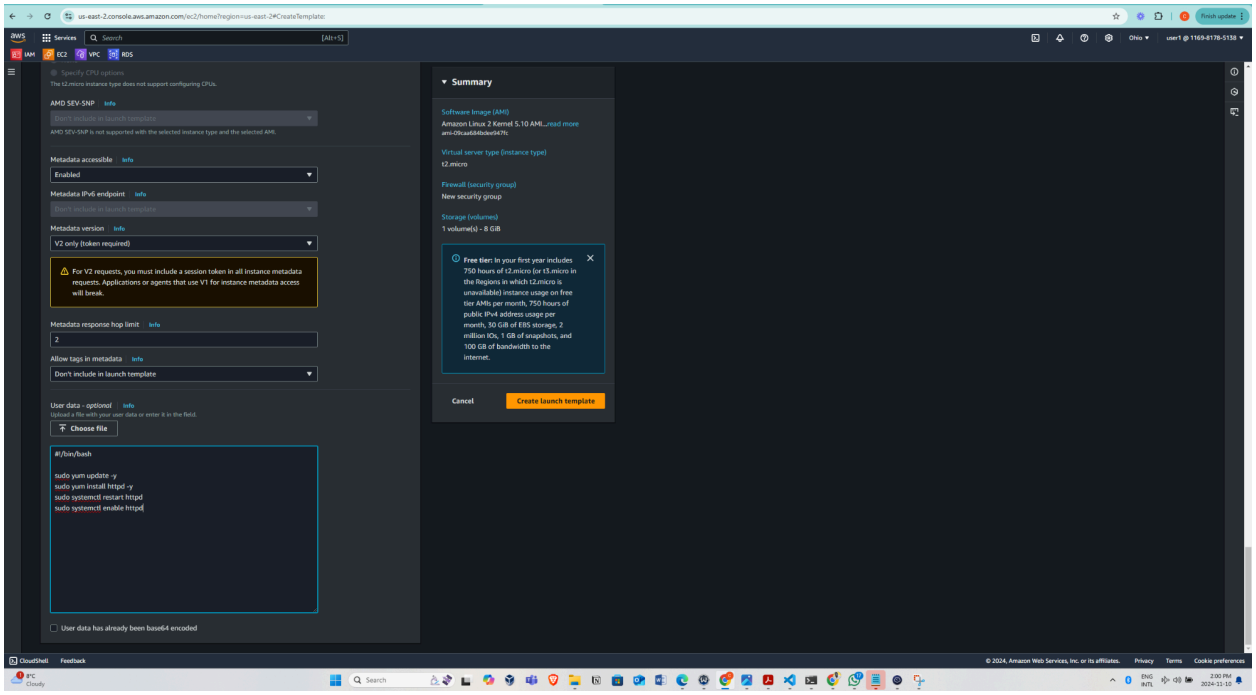
Tier 2 set up: This is the application layer

Step 1

Create a lunch template for tier2. The SG that would be created alongside this lunch template will only allow ssh traffic from Tier1_SG and http traffic from ALB_SG.

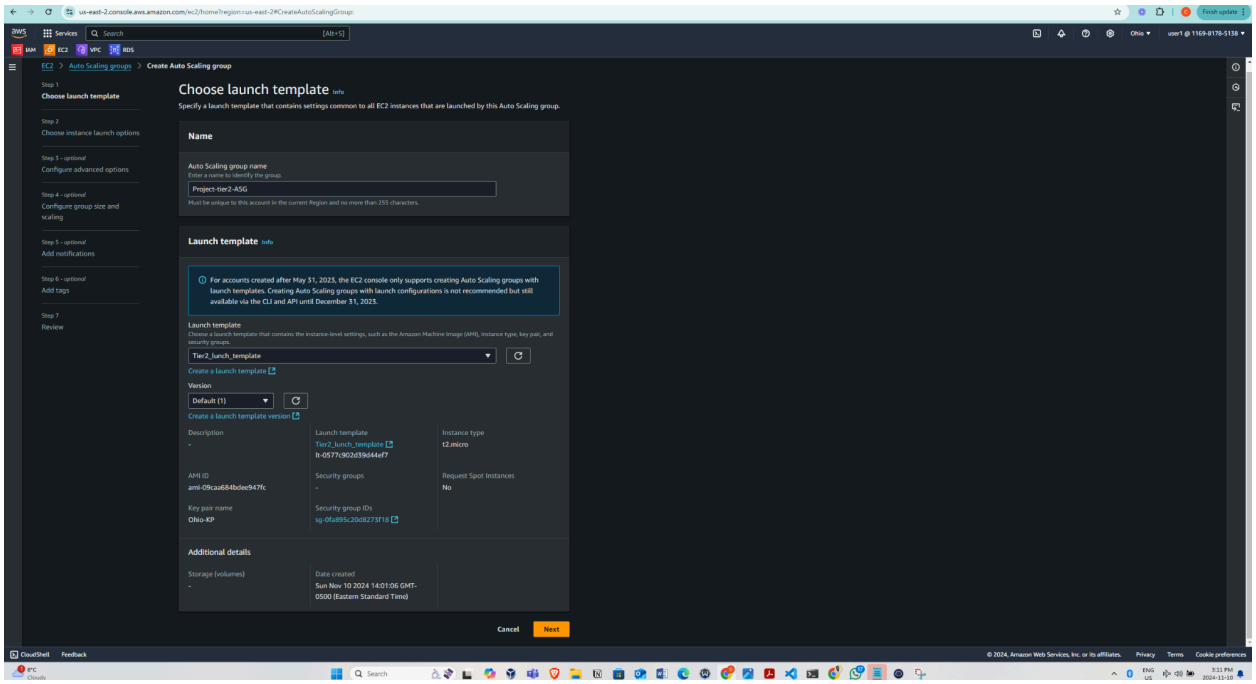


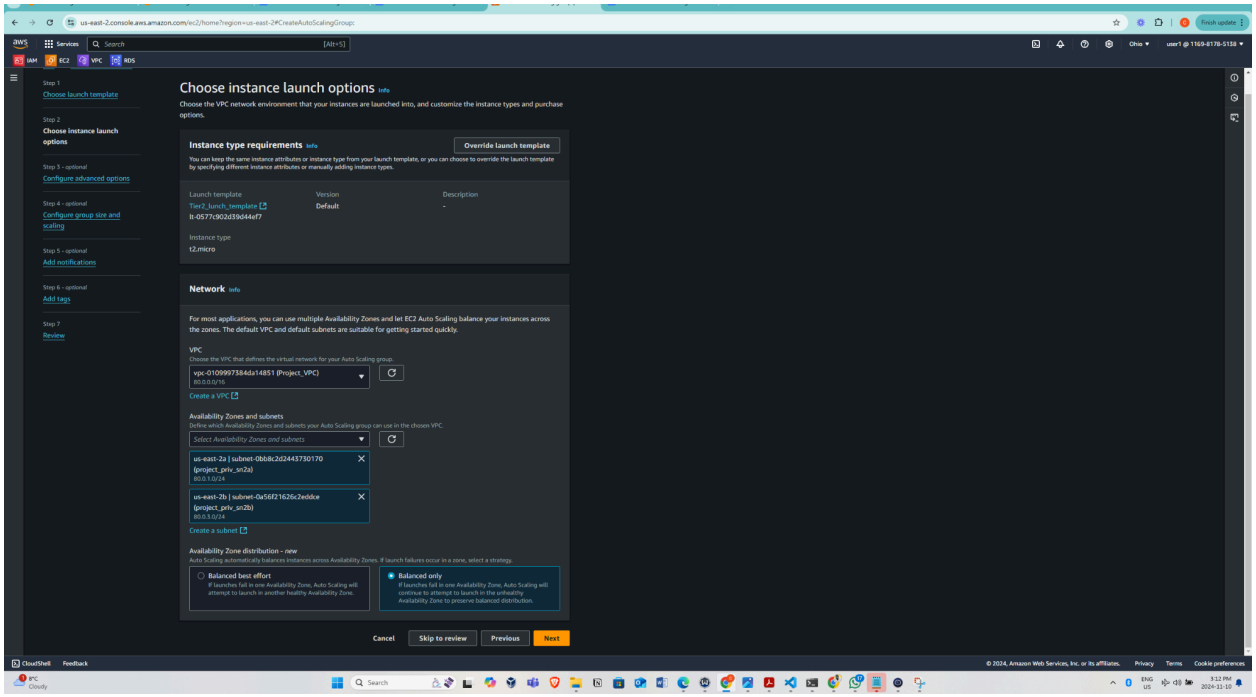

Ohio-KP
Create new key pair
Network settings
Subnet
Don't include in launch template
Create new subnet
Firewall (security groups)
Select existing security group
Create security group
Security group name - required
Tier2_SG
Description - required
This SG is for tier 2 and used to lunch resources in a private subnet
VPC
vpc-0109997384da14851 (Project_VPC)
Inbound Security Group Rules
Security group rule 1 (TCP, 22, sg-083c2fede4d0eaacc, This rule will allow ssh traffic f...)
Remove
Type
ssh
Protocol
TCP
Port range
22
Source type
Custom
Source
sg-083c2fede4d0eaacc
Description - optional
This rule will allow ssh traffic from
Security group rule 2 (TCP, 80, sg-01d4fe791a245bdf3, This rule will allow http traffic ...)
Remove
Type
HTTP
Protocol
TCP
Port range
80
Source type
Custom
Source
sg-01d4fe791a245bdf3
Description - optional
This rule will allow http traffic fro
Add security group rule
Summary
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-05ca66d4bdee9477c
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Create launch template
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

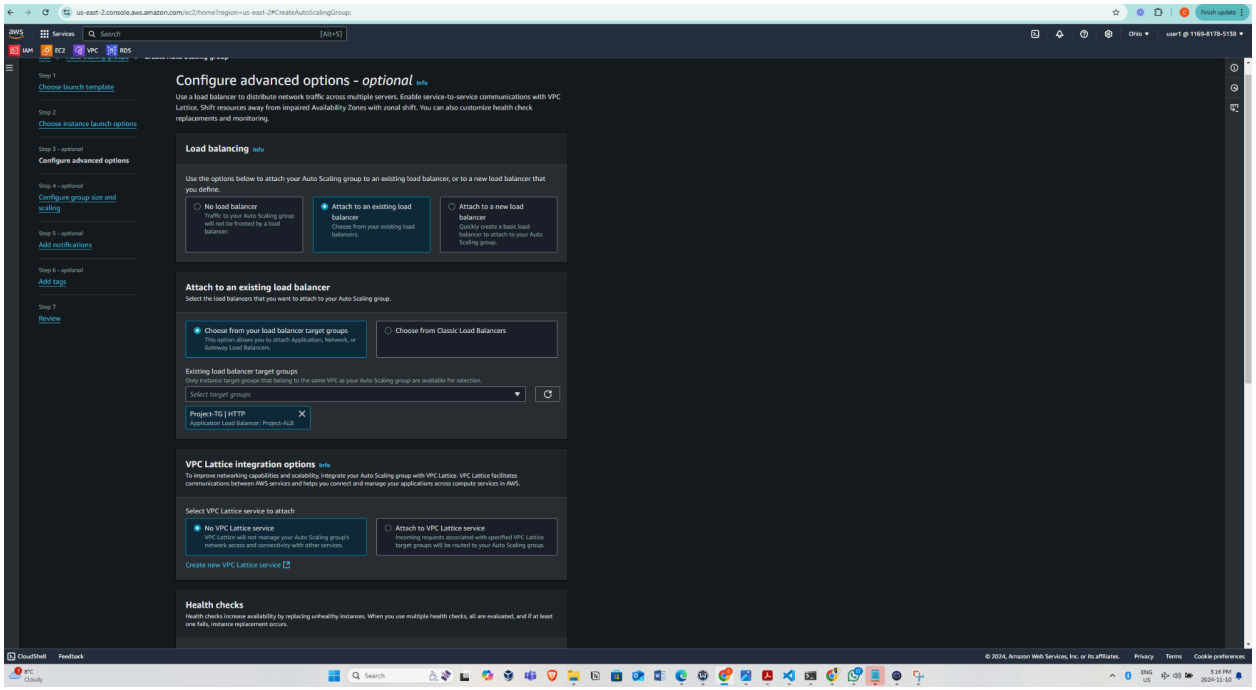

Specify CPU options
The t2.micro instance type does not support configuring CPUs.
AMD SEV-SNP
Don't include in launch template
AMD SEV-SNP is not supported with the selected instance type and the selected AMI.
Metadata accessible
Enabled
Metadata IPv6 endpoint
Don't include in launch template
Metadata version
V2 only (token required)
For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.
Metadata response hop limit
2
Allow tags in metadata
Don't include in launch template
User data - optional
Upload a file with your user data or enter it in the field.
Choose file
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo systemctl restart httpd
sudo systemctl enable httpd
User data has already been base64 encoded
Summary
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-05ca66d4bdee9477c
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Create launch template
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

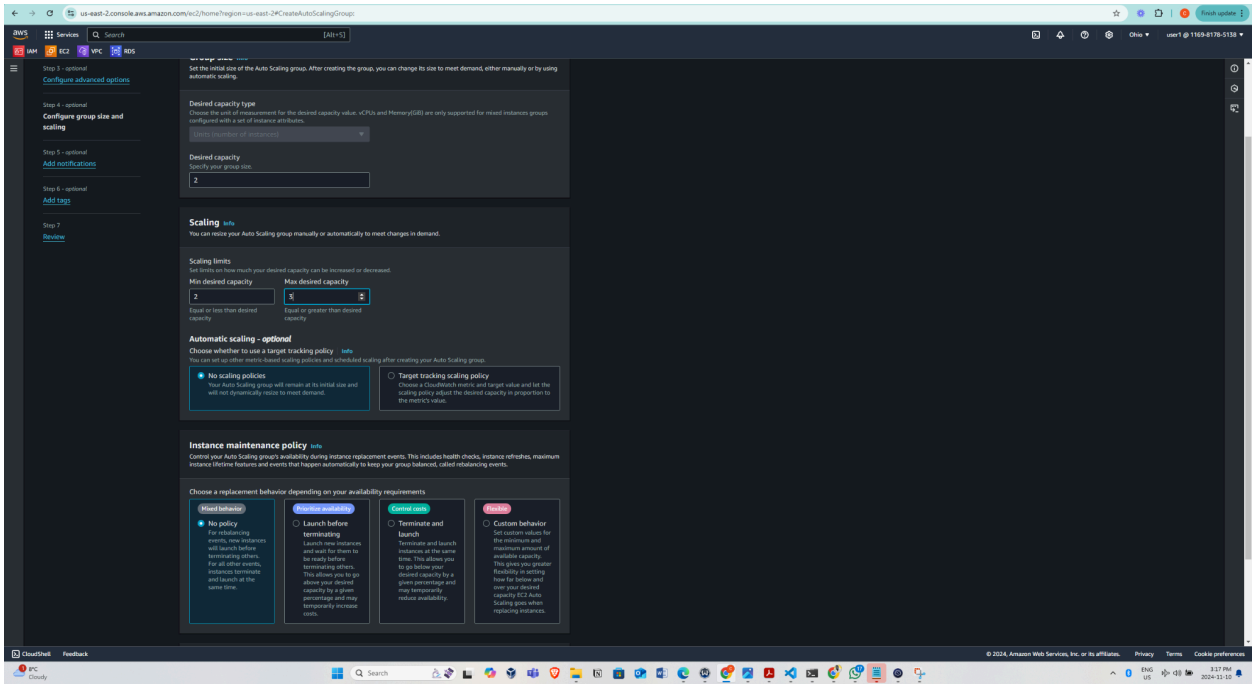
Step2

Create an ASG for tier2.. This ASG is specifically for lunch Instances in the private subnets. The ALB I created earlier will be selected so traffic coming from the internet can access the web servers that would be launched within this ASG




Choose instance launch options
Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.
Instance type requirements
Override launch template
Launch template
Tier2_lunch_template
lt-0577c902d39d44ef7
Version
Default
Description
Instance type
t2.micro
Network
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.
VPC
vpc-0109997384da14851 (Project_VPC)
Create a VPC
Availability Zones and subnets
Select Availability Zones and subnets
us-east-2a | subnet-0bb8c2d2443730170 (project_priv_sn2a)
us-east-2b | subnet-0a56f21626c2eddce (project_priv_sn2b)
Create a subnet
Availability Zone distribution - new
Balanced best effort
Balanced only
Cancel
Skip to review
Previous
Next


Configure advanced options - optional
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.
Load balancing
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
No load balancer
Attach to an existing load balancer
Attach to a new load balancer
Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.
Choose from your load balancer target groups
Choose from Classic Load Balancers
Existing load balancer target groups
Project-TG | HTTP
VPC Lattice integration options
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.
Select VPC Lattice service to attach
No VPC Lattice service
Attach to VPC Lattice service
Create new VPC Lattice service
Health checks



Step 3

Testing Connectivity to Tier 2 Servers:

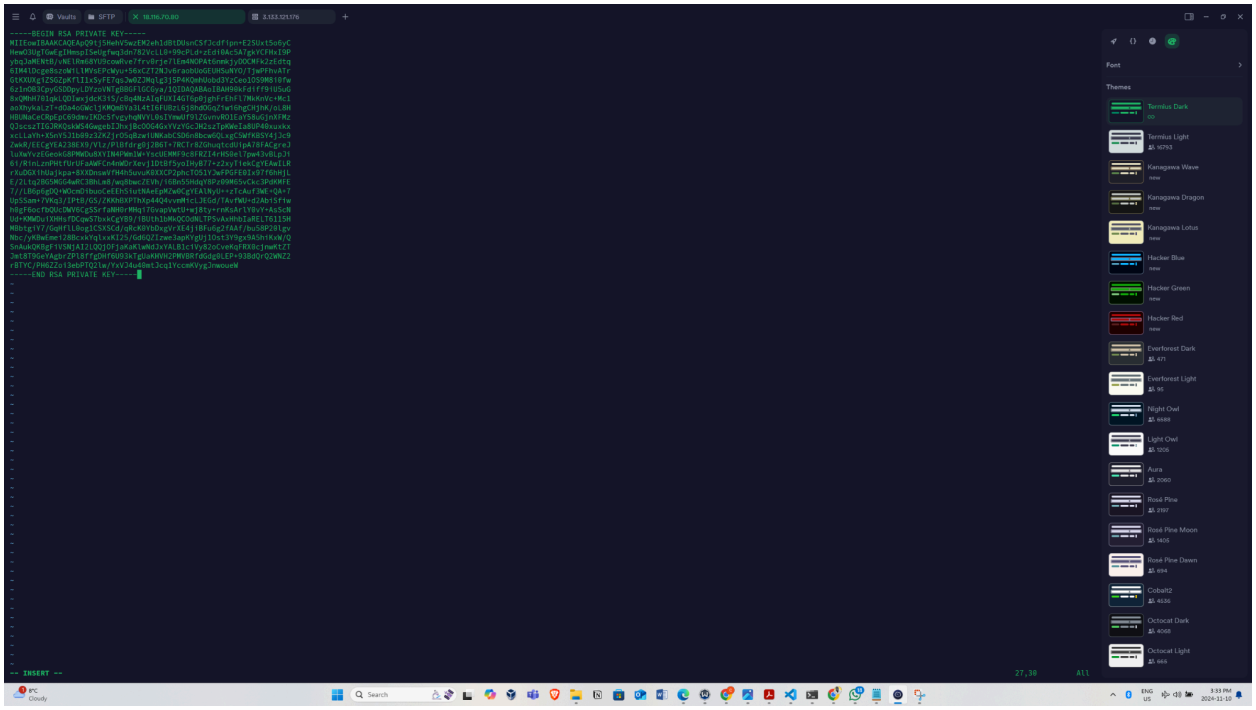
To verify connectivity to servers in the Tier 2 (private subnet), I will use SSH to access them via a bastion host. This involves the following steps:

a. SSH into Tier 1 Host (Public Subnet): First, I connect to the Tier 1 host in the public subnet.
b. SSH from Tier 1 to Tier 2 Host (Private Subnet): Once connected to the Tier 1 host, I use SSH again to access the Tier 2 host in the private subnet.

Since both instances are using the same key pair, I need to transfer the private key to the Tier 1 host before I can successfully SSH into the Tier 2 host.

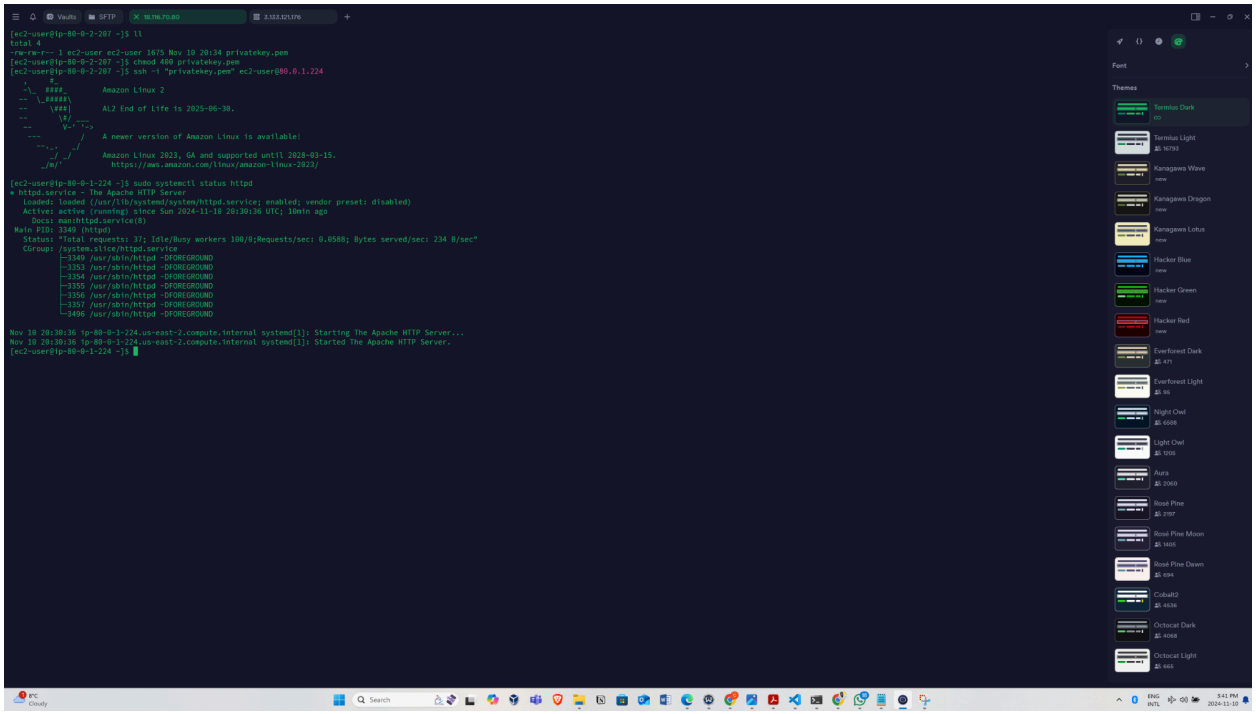
Step 3b

Transfer of the private key into the tier1(public) host



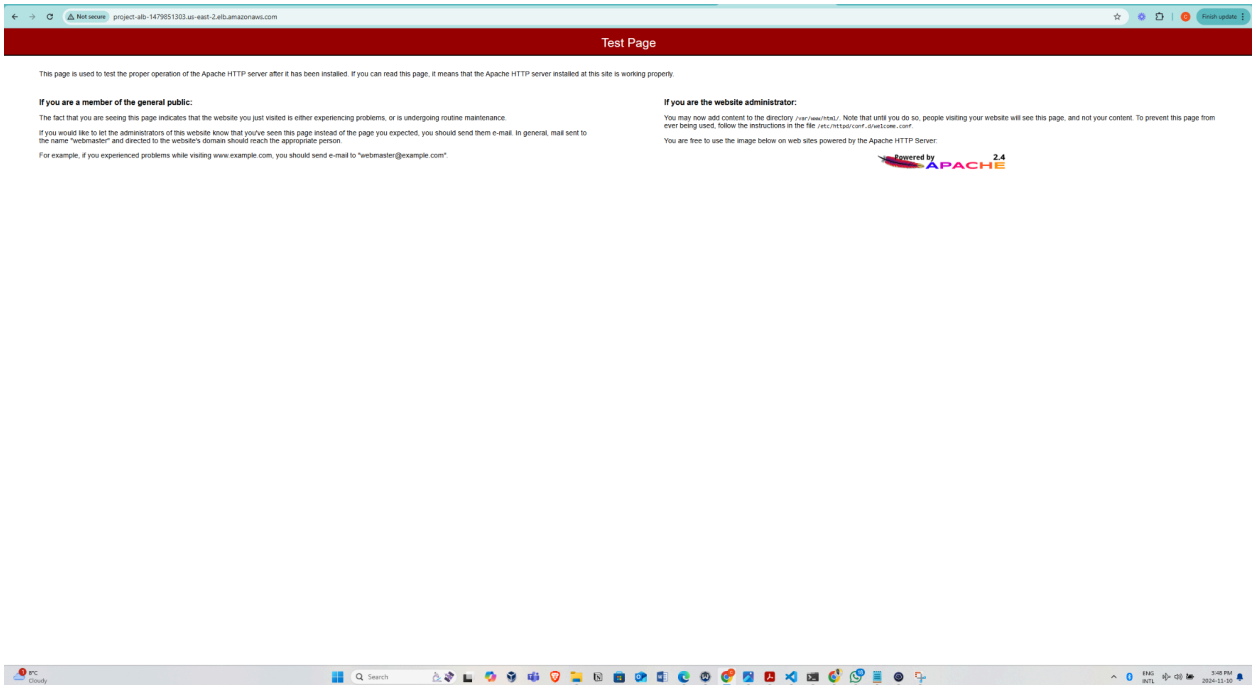
Step 3c

First change the permission of the key to chmod 400. This makes the key private and nobody can access the key. Then ssh into the private server in tier2.



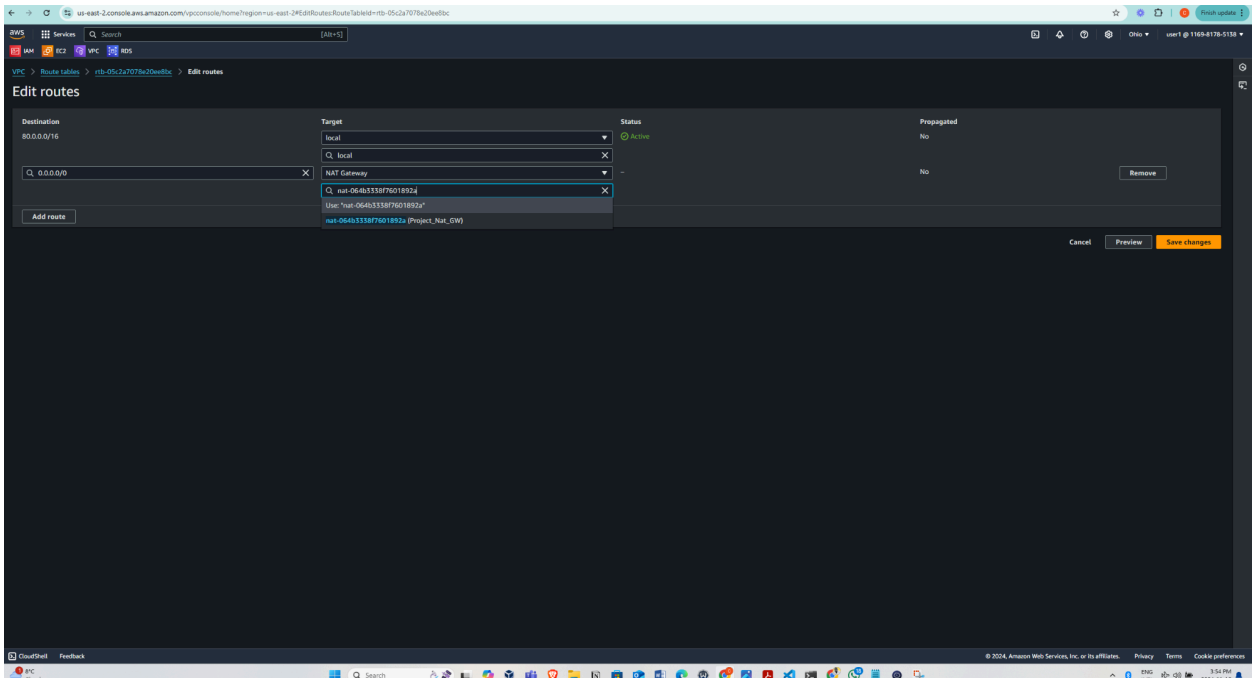
Step4

Browsing for the DNS A record of the loadbalancer to test if the application in the tier2(private) Instances are accessible.



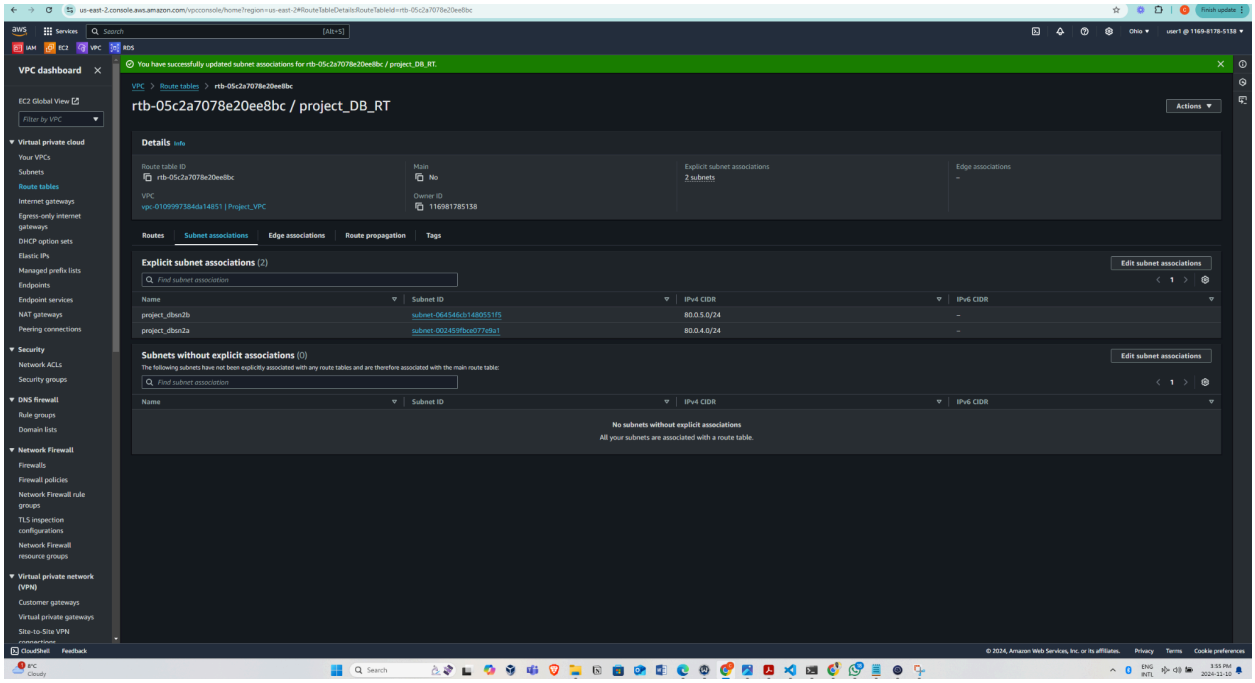
Tier 3 Set up: This is the database layer

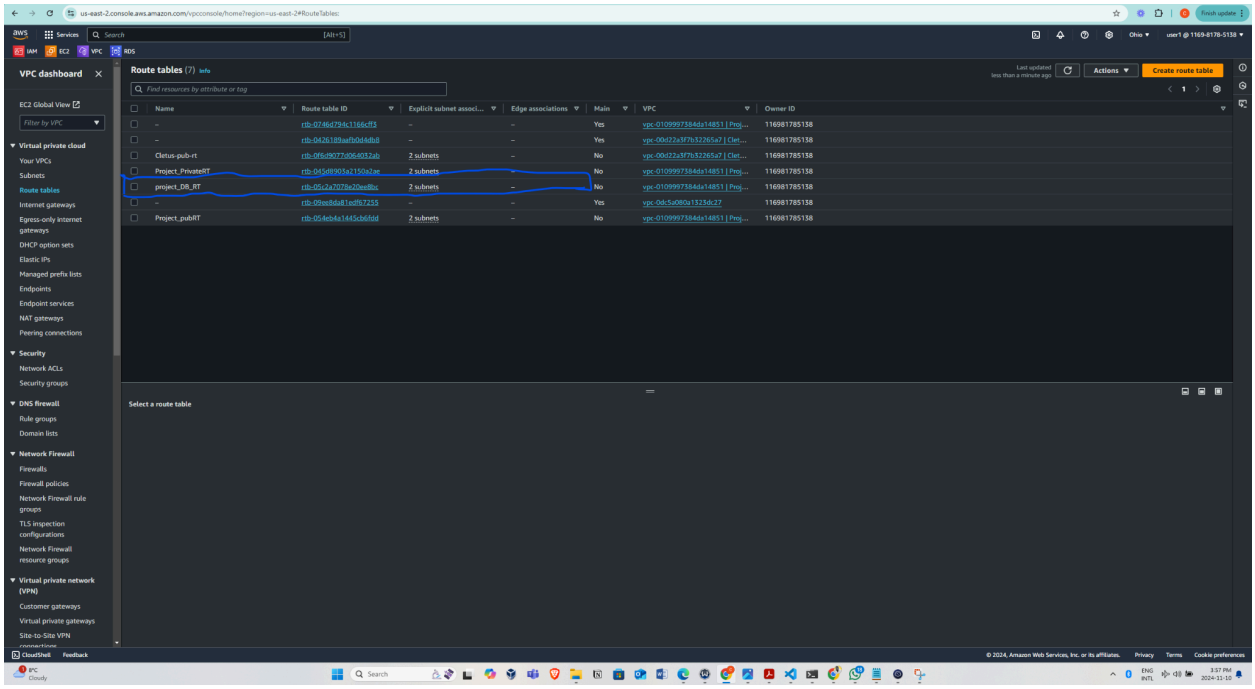
Step 1:Create a new route table for the databases add a route pointing to the Nat-GW



Step 1b

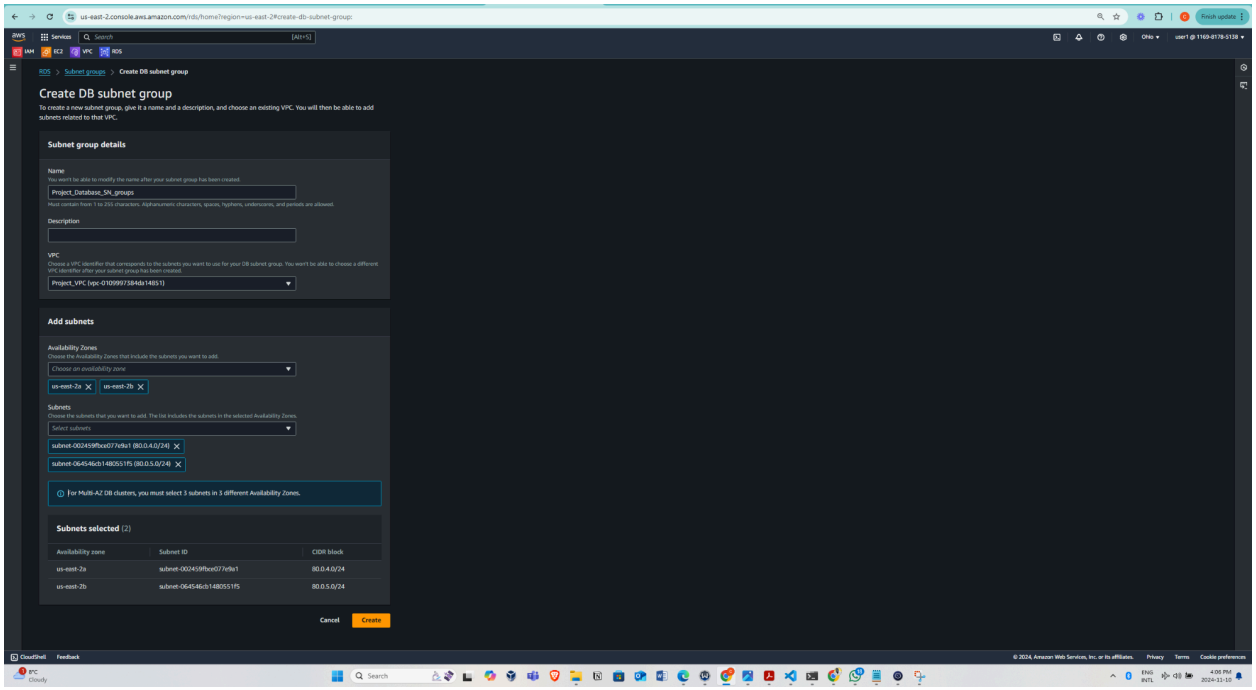
Associate the route table with the database subnets





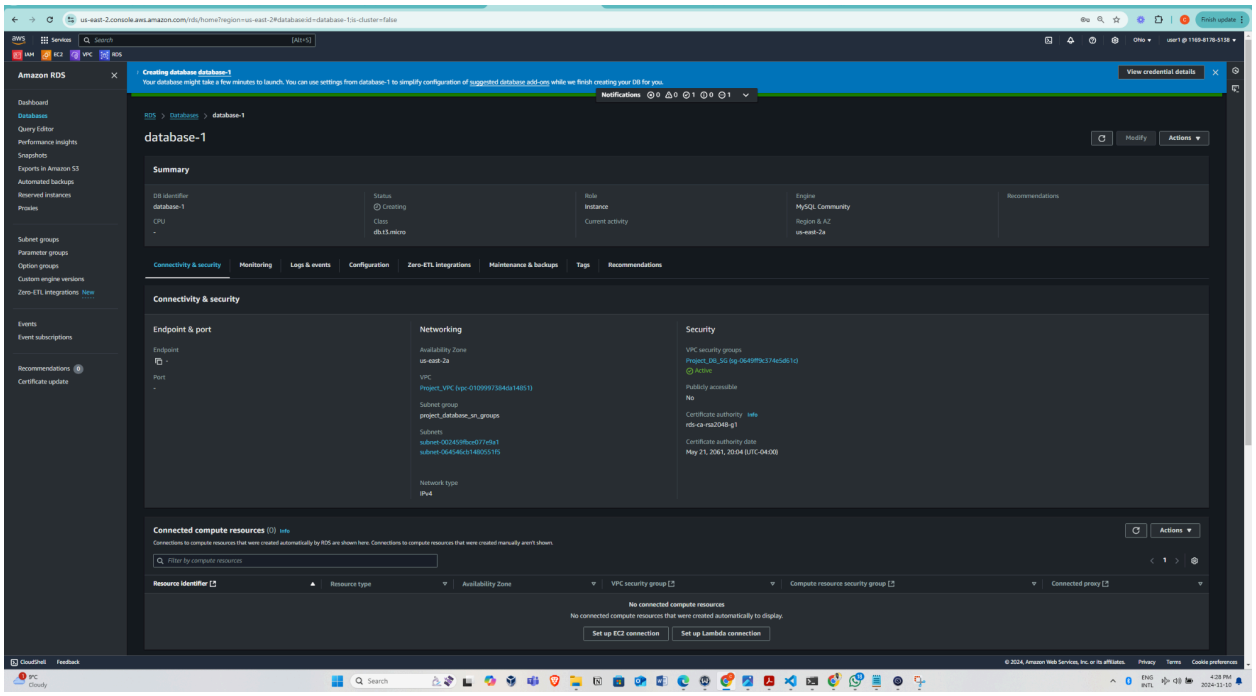
Step 2

Create database subnet groups and then i will add the database subnets i have created in availability zone 2a and 2b.



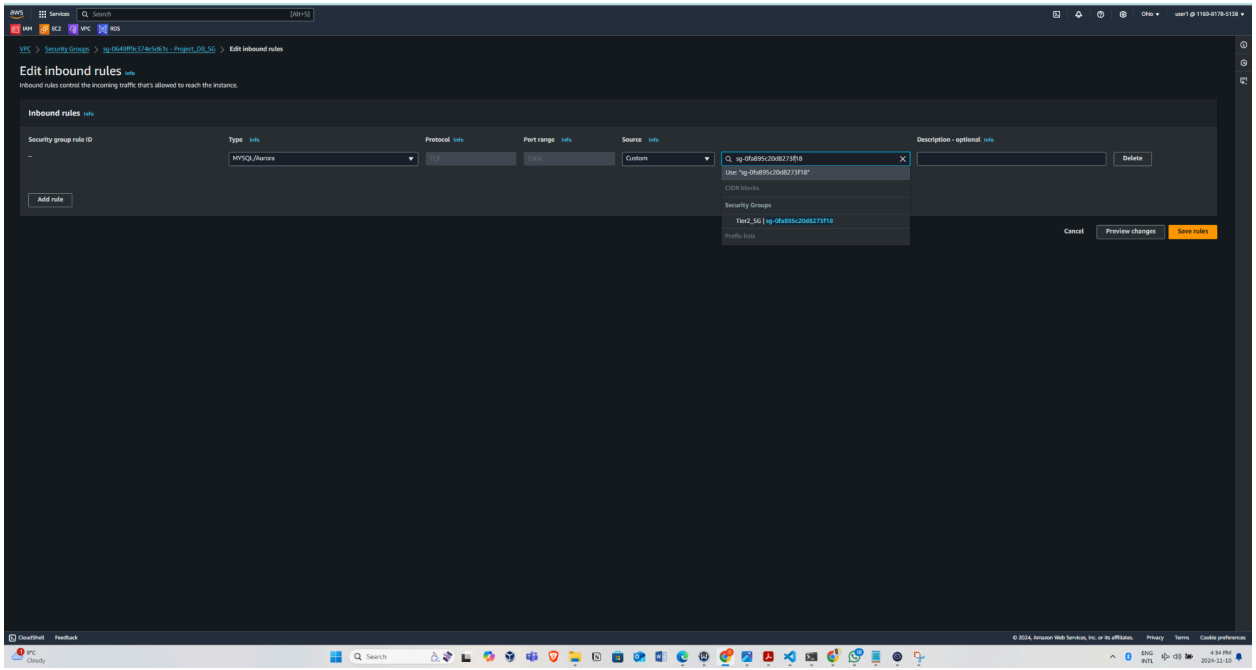
Step 3

Create database instance



Step 4

Add a new rule allowing traffic from tier2-SG on port 3306 and delete the default rule in the SG.



Now a 3-tier web page architecture has been successfully set up.