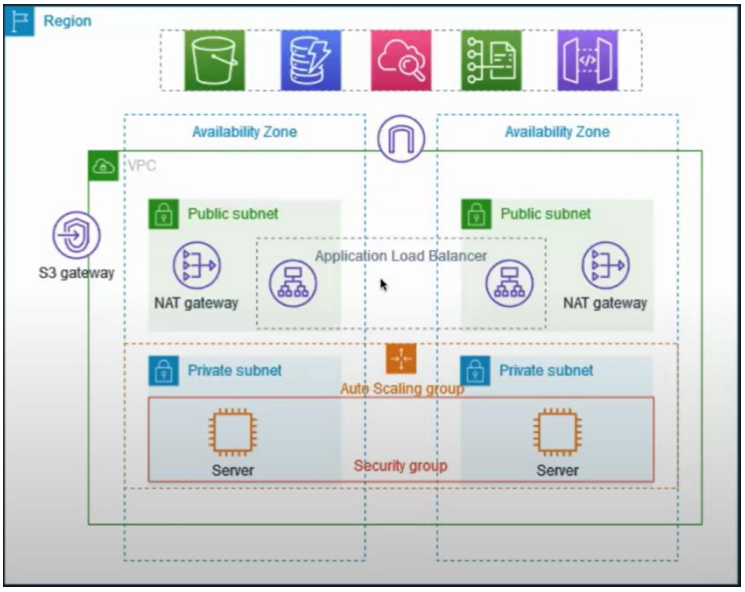
Project Report: Implementing a VPC for a Production Environment

Overview

This project involves creating a Virtual Private Cloud (VPC) to host servers in a production environment. The VPC will provide a secure and scalable network infrastructure within a public cloud, utilizing multiple Availability Zones (AZs) for high availability and redundancy. Key components include an Auto Scaling group, an Application Load Balancer, private and public subnets, and a NAT gateway for internet access.



Terminologies

Virtual Private Cloud (VPC)

A Virtual Private Cloud (VPC) is a private network within a public cloud, such as AWS or Google Cloud, allowing complete control over network settings like IP address ranges, subnets, and security configurations. It provides an isolated section of the cloud to run resources securely with the flexibility and scalability of the cloud.

Network Address Translation (NAT)

Network Address Translation (NAT) maps private IP addresses to a public IP address or a pool of public IP addresses. This enables multiple devices on a private network to access the internet using a single public IP address, enhancing security and conserving IP address space.

Infrastructure

1. **Resiliency with Multiple Availability Zones (AZs)**
    - Servers are deployed across two AZs to ensure redundancy and high availability. Each AZ is a separate data center within the same region.
2. **Auto Scaling Group**
    - Automatically scales the number of servers based on demand to ensure sufficient resources are available.
3. **Application Load Balancer**
    - Distributes incoming traffic across servers to improve availability and performance.
4. **Private Subnets**
    - Servers are deployed in private subnets for security, meaning they are not directly accessible from the internet.
5. **NAT Gateway**
    - Allows servers in private subnets to access the internet for software updates or external services.
6. **Resilient NAT Gateway**
    - Deployed across both AZs to ensure continuous internet connectivity even if one AZ fails.

Project Architecture

The VPC is designed with both public and private subnets across two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. Servers run in private subnets, launched and terminated by an Auto Scaling group, and receive traffic from the load balancer. The servers can access the internet via the NAT gateway.

Components and Configuration

VPC

- **CIDR Block**: 10.0.0.0/16

Subnets

- **Public Subnets**:
    - Subnet 1 (AZ1): 10.0.1.0/24
    - Subnet 2 (AZ2): 10.0.2.0/24

- **Private Subnets**:
    - Subnet 3 (AZ1): 10.0.3.0/24
    - Subnet 4 (AZ2): 10.0.4.0/24

Internet Gateway (IGW)

- Attach an Internet Gateway to the VPC to allow public subnets to access the internet.

NAT Gateway

- Deploy a NAT Gateway in each public subnet to provide internet access to private subnets.

Route Tables

- **Public Route Table**: Route 0.0.0.0/0 to IGW.
- **Private Route Table**: Route 0.0.0.0/0 to NAT Gateway.

Auto Scaling Group

- Launches and terminates servers in private subnets based on demand.

Application Load Balancer

- Distributes incoming traffic across servers in private subnets.

Implementation Procedure

Step 1: Create a VPC

1. Create a VPC with a CIDR block of 10.0.0.0/16.

Step 2: Create Subnets

1. Create two public subnets in different AZs with CIDR blocks 10.0.1.0/24 and 10.0.2.0/24.
2. Create two private subnets in different AZs with CIDR blocks 10.0.3.0/24 and 10.0.4.0/24.

Step 3: Configure Internet Gateway

1. Create an Internet Gateway and attach it to the VPC.
2. Create a public route table and add a route for 0.0.0.0/0 pointing to the Internet Gateway.
3. Associate the public route table with the public subnets.

Step 4: Create NAT Gateway

1. Create a NAT Gateway in each public subnet.

2. Create a private route table and add a route for 0.0.0.0/0 pointing to the NAT Gateway.
3. Associate the private route table with the private subnets.

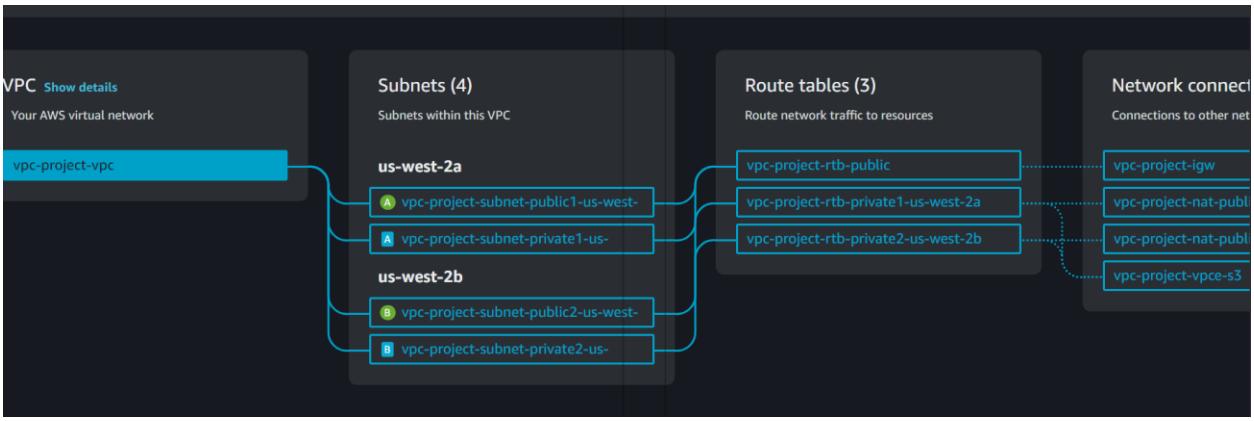
Step 5: Configure Auto Scaling Group

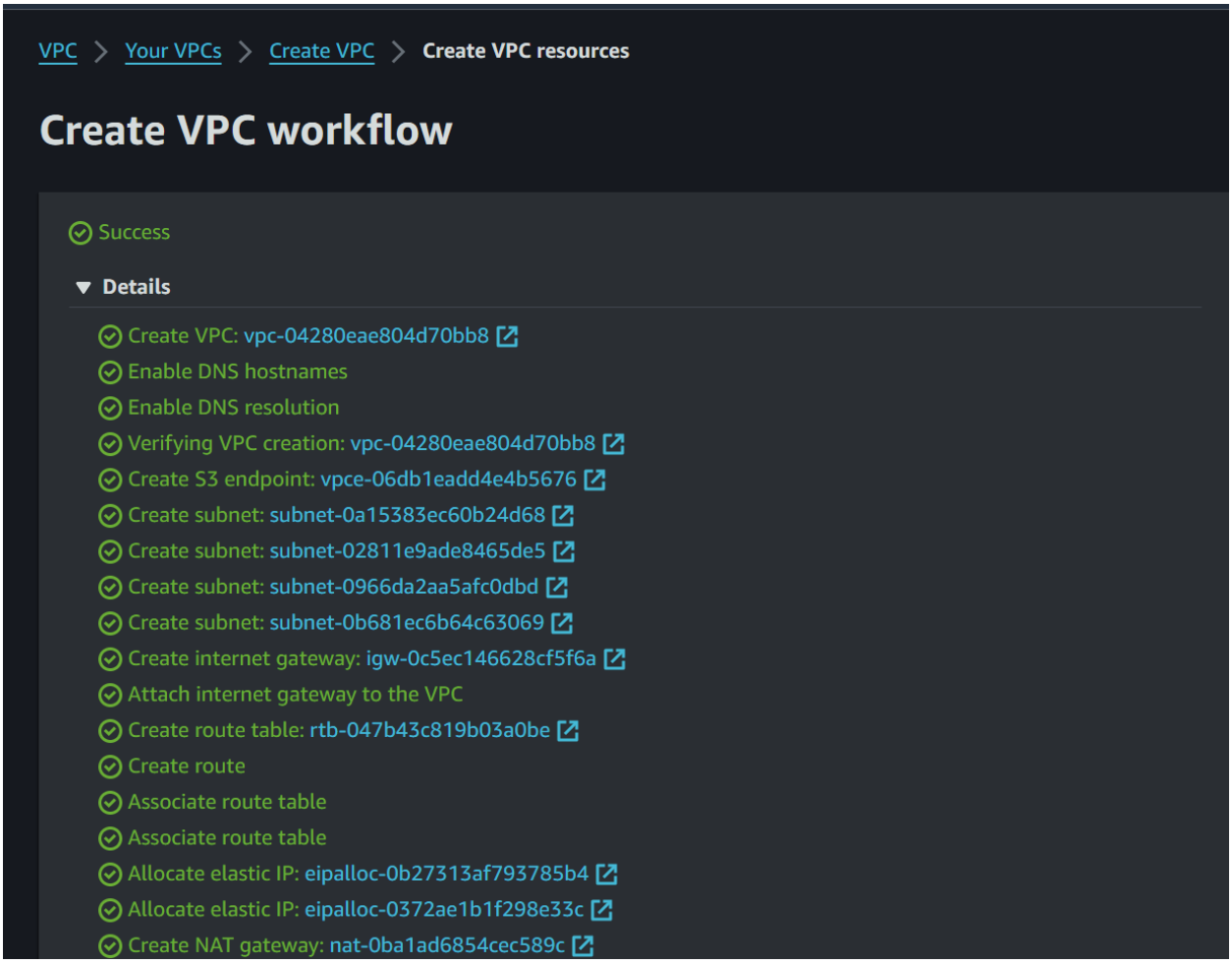
1. Create a launch configuration or launch template specifying the AMI, instance type, and security group.
2. Create an Auto Scaling group with the desired capacity, minimum and maximum instances, and target the private subnets.

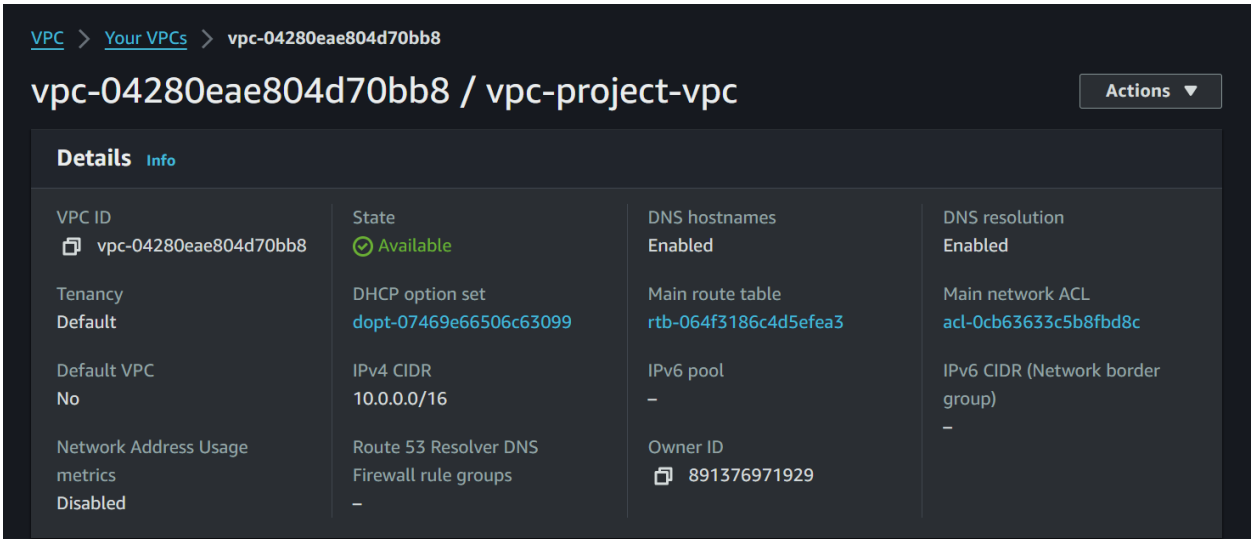
Step 6: Configure Application Load Balancer

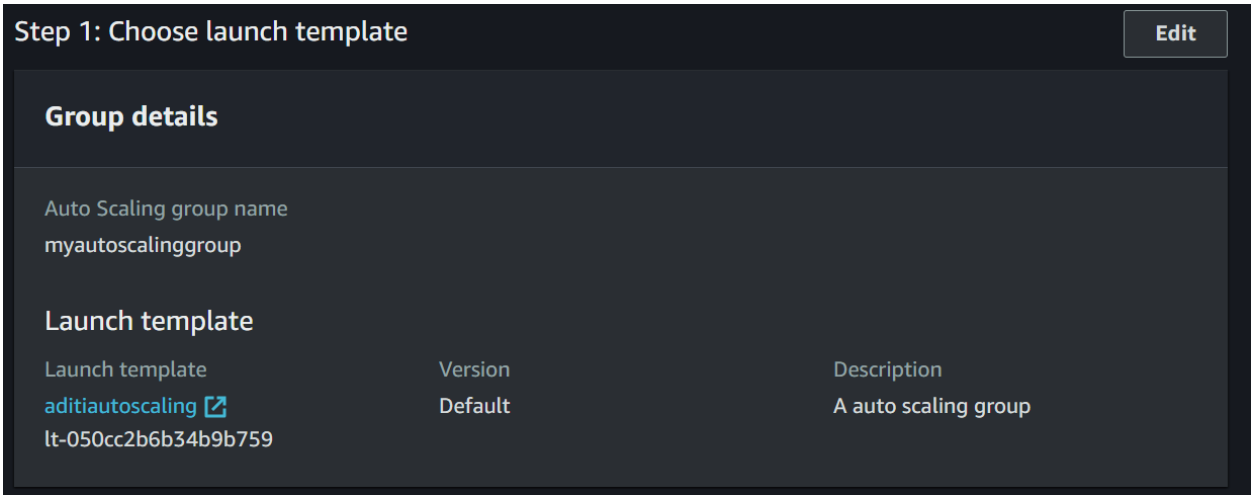
1. Create an Application Load Balancer and specify the public subnets.
2. Create target groups and register the Auto Scaling group instances.
3. Create listeners to route traffic from the load balancer to the target groups.

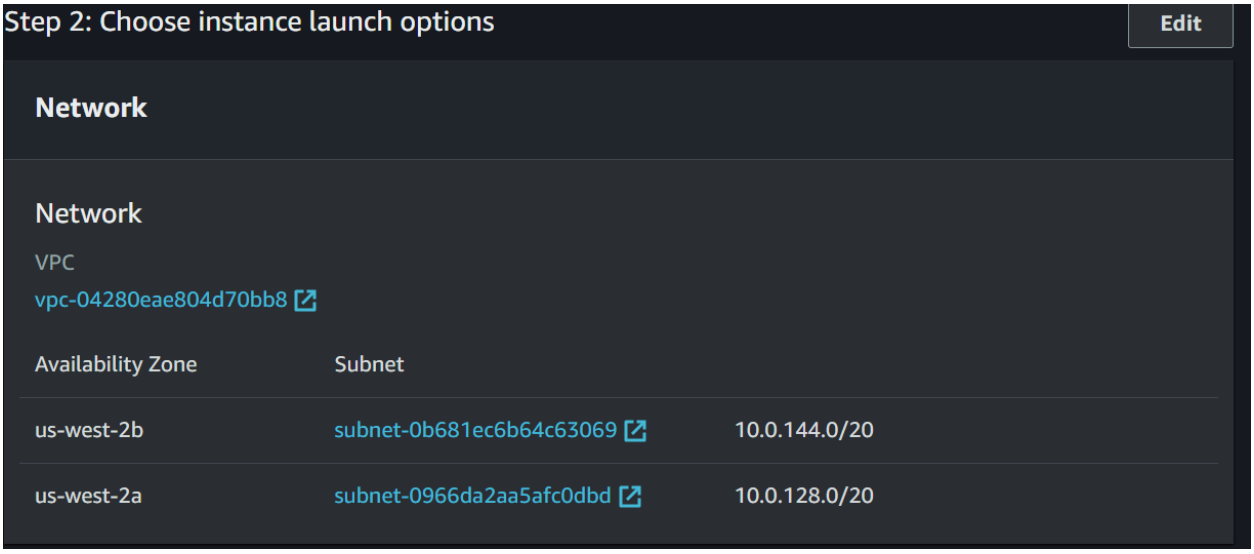
Related Images :

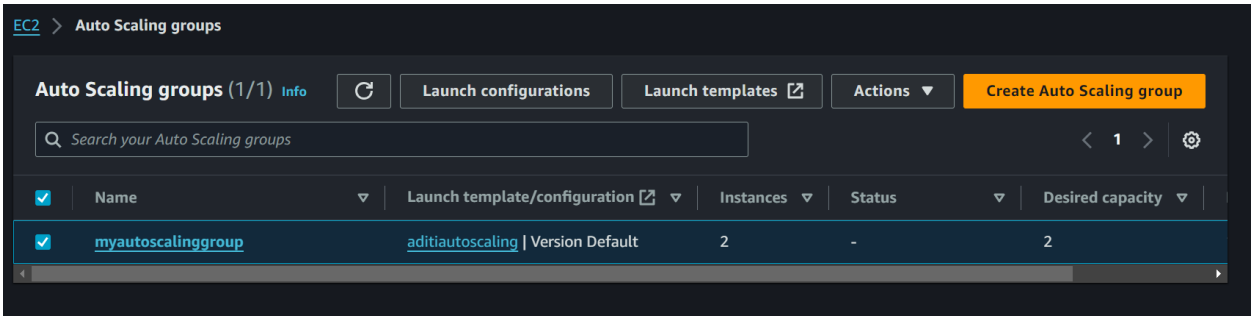


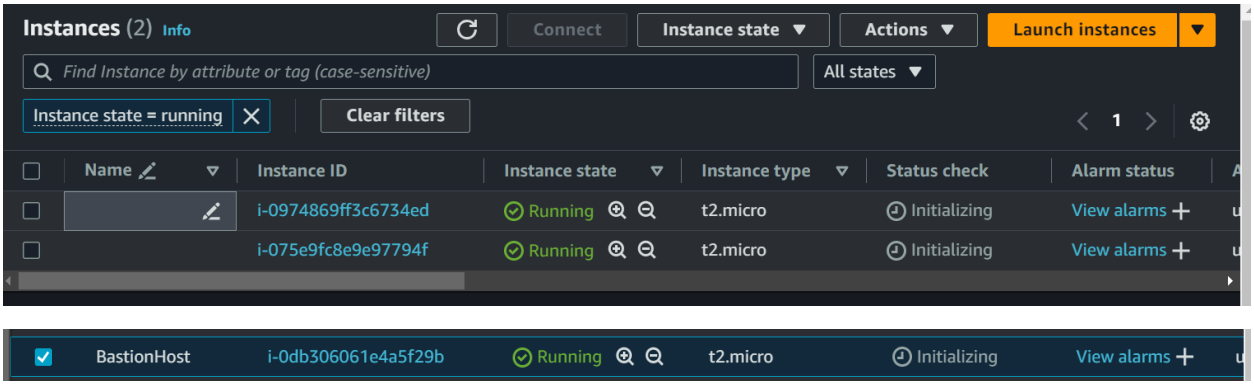

VPC > Your VPCs > Create VPC > Create VPC resources
Create VPC workflow
Success
Details
Create VPC: vpc-04280eae804d70bb8
Enable DNS hostnames
Enable DNS resolution
Verifying VPC creation: vpc-04280eae804d70bb8
Create S3 endpoint: vpce-06db1eadd4e4b5676
Create subnet: subnet-0a15383ec60b24d68
Create subnet: subnet-02811e9ade8465de5
Create subnet: subnet-0966da2aa5afc0dbd
Create subnet: subnet-0b681ec6b64c63069
Create internet gateway: igw-0c5ec146628cf5f6a
Attach internet gateway to the VPC
Create route table: rtb-047b43c819b03a0be
Create route
Associate route table
Associate route table
Allocate elastic IP: eipalloc-0b27313af793785b4
Allocate elastic IP: eipalloc-0372ae1b1f298e33c
Create NAT gateway: nat-0ba1ad6854cec589c


VPC > Your VPCs > vpc-04280eae804d70bb8
vpc-04280eae804d70bb8 / vpc-project-vpc
Actions
Details Info
VPC ID
vpc-04280eae804d70bb8
State
Available
DNS hostnames
Enabled
DNS resolution
Enabled
Tenancy
Default
DHCP option set
dopt-07469e66506c63099
Main route table
rtb-064f3186c4d5efea3
Main network ACL
acl-0cb63633c5b8fbd8c
Default VPC
No
IPv4 CIDR
10.0.0.0/16
IPv6 pool
–
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
891376971929

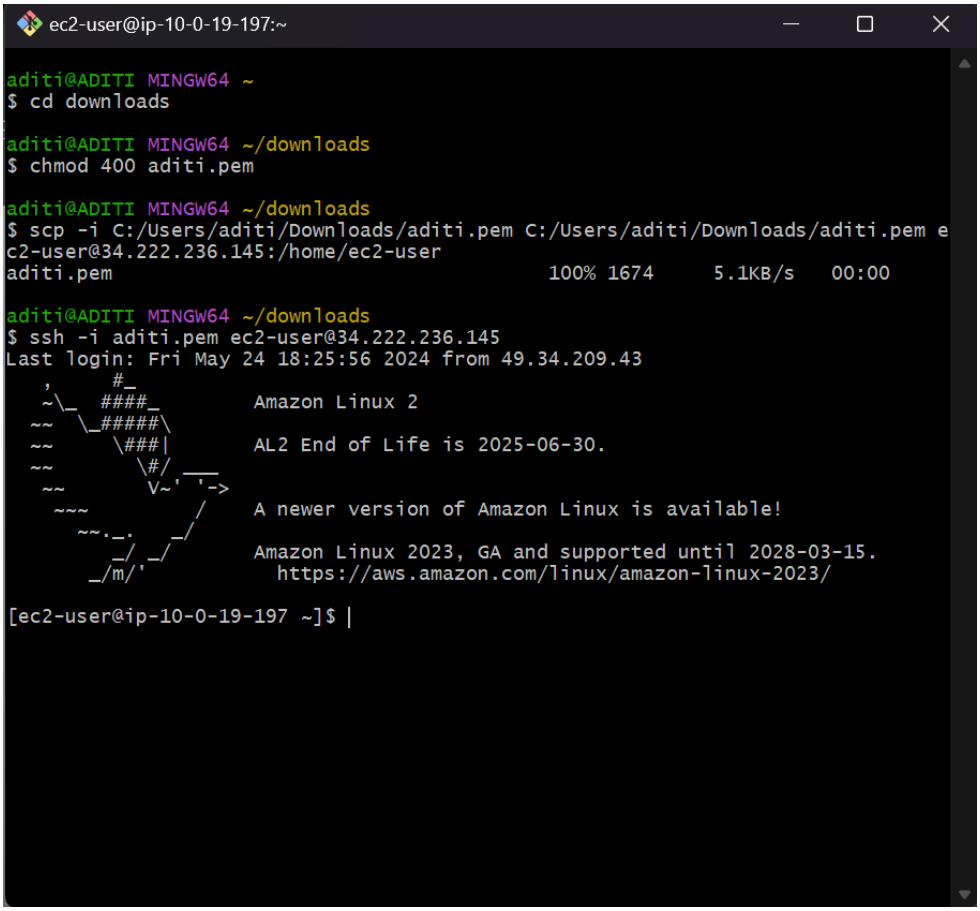

Step 1: Choose launch template
Edit
Group details
Auto Scaling group name
myautoscalinggroup
Launch template
Launch template
aditiautoscaling
lt-050cc2b6b34b9b759
Version
Default
Description
A auto scaling group


Step 2: Choose instance launch options
Edit
Network
Network
VPC
vpc-04280eae804d70bb8
Availability Zone
Subnet
us-west-2b
subnet-0b681ec6b64c63069
10.0.144.0/20
us-west-2a
subnet-0966da2aa5afc0dbd
10.0.128.0/20





A public server will be intermediate between private servers and the internet

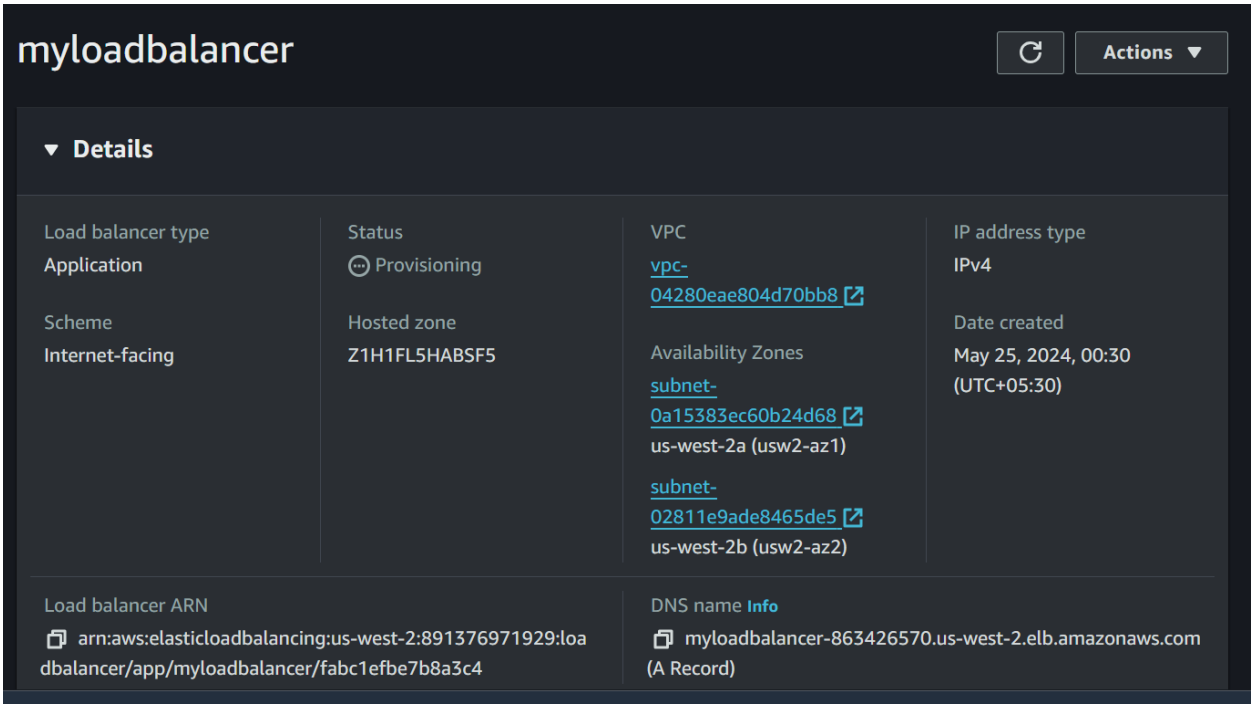

ec2-user@ip-10-0-19-197:~
aditi@ADITI MINGW64 ~
$ cd downloads
aditi@ADITI MINGW64 ~/downloads
$ chmod 400 aditi.pem
aditi@ADITI MINGW64 ~/downloads
$ scp -i C:/Users/aditi/Downloads/aditi.pem C:/Users/aditi/Downloads/aditi.pem e
c2-user@34.222.236.145:/home/ec2-user
aditi.pem 100% 1674 5.1KB/s 00:00
aditi@ADITI MINGW64 ~/downloads
$ ssh -i aditi.pem ec2-user@34.222.236.145
Last login: Fri May 24 18:25:56 2024 from 49.34.209.43
Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-10-0-19-197 ~]$

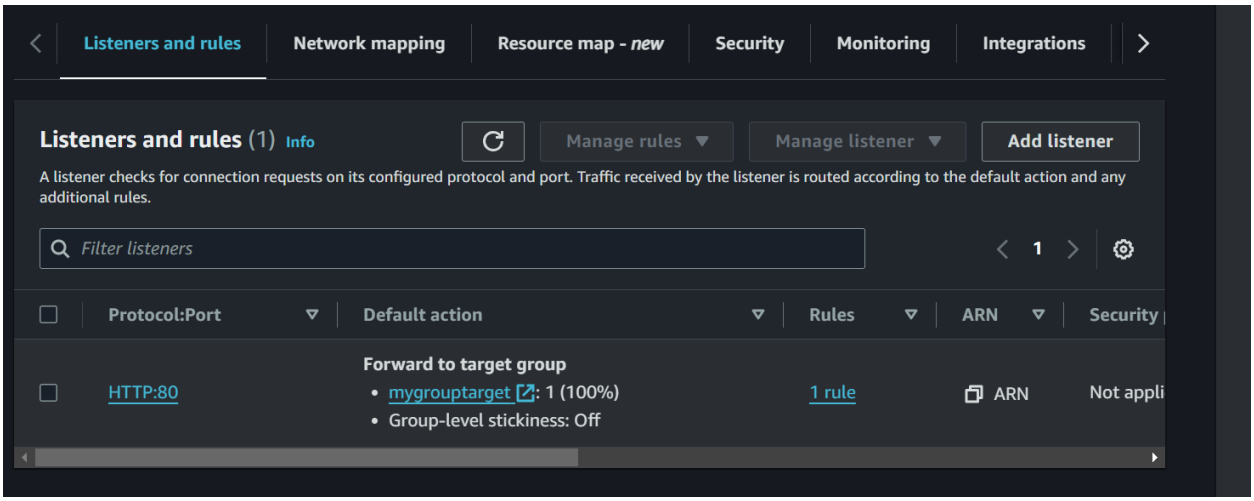
```
aditi@ADITI MINGW64 ~/downloads
$ ssh -i aditi.pem ec2-user@34.222.236.145
Last login: Fri May 24 18:25:56 2024 from 49.34.209.43
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-19-197 ~]$ ls
aditi.pem
[ec2-user@ip-10-0-19-197 ~]$ ssh -i aditi.pem ec2-user@10.0.157.228
The authenticity of host '10.0.157.228 (10.0.157.228)' can't be established.
ECDSA key fingerprint is SHA256:s6SRsQppv3Sig46mnxvcQli3o43gYFtQKUSlP2xDfxQ.
ECDSA key fingerprint is MD5:ba:19:62:2c:aa:7d:00:30:b9:d7:93:85:8a:6e:c4:12.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.157.228' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0444 for 'aditi.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "aditi.pem": bad permissions
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-19-197 ~]$
```

```
[ec2-user@ip-10-0-19-197 ~]$ vim aditi.html
[ec2-user@ip-10-0-19-197 ~]$
```

```
[ec2-user@ip-10-0-19-197 ~]$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```


myloadbalancer
Actions
Details
Load balancer type
Application
Status
Provisioning
VPC
vpc-04280eae804d70bb8
IP address type
IPv4
Scheme
Internet-facing
Hosted zone
Z1H1FL5HABSF5
Availability Zones
subnet-0a15383ec60b24d68
us-west-2a (usw2-az1)
subnet-02811e9ade8465de5
us-west-2b (usw2-az2)
Date created
May 25, 2024, 00:30 (UTC+05:30)
Load balancer ARN
arn:aws:elasticloadbalancing:us-west-2:891376971929:loadbalancer/app/myloadbalancer/fabc1efbe7b8a3c4
DNS name Info
myloadbalancer-863426570.us-west-2.elb.amazonaws.com (A Record)


Listeners and rules
Network mapping
Resource map - new
Security
Monitoring
Integrations
Listeners and rules (1) Info
Manage rules
Manage listener
Add listener
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
Filter listeners
1
Protocol:Port
Default action
Rules
ARN
Security
HTTP:80
Forward to target group
mygrouptarget: 1 (100%)
Group-level stickiness: Off
1 rule
ARN
Not appli

Conclusion

By following this procedure, you can implement a robust and scalable VPC architecture for a production environment. The architecture ensures high availability, security, and efficient resource utilization through the use of multiple Availability Zones, Auto Scaling, and load balancing. The NAT gateway provides secure internet access for servers in private subnets, completing a well-rounded infrastructure setup.