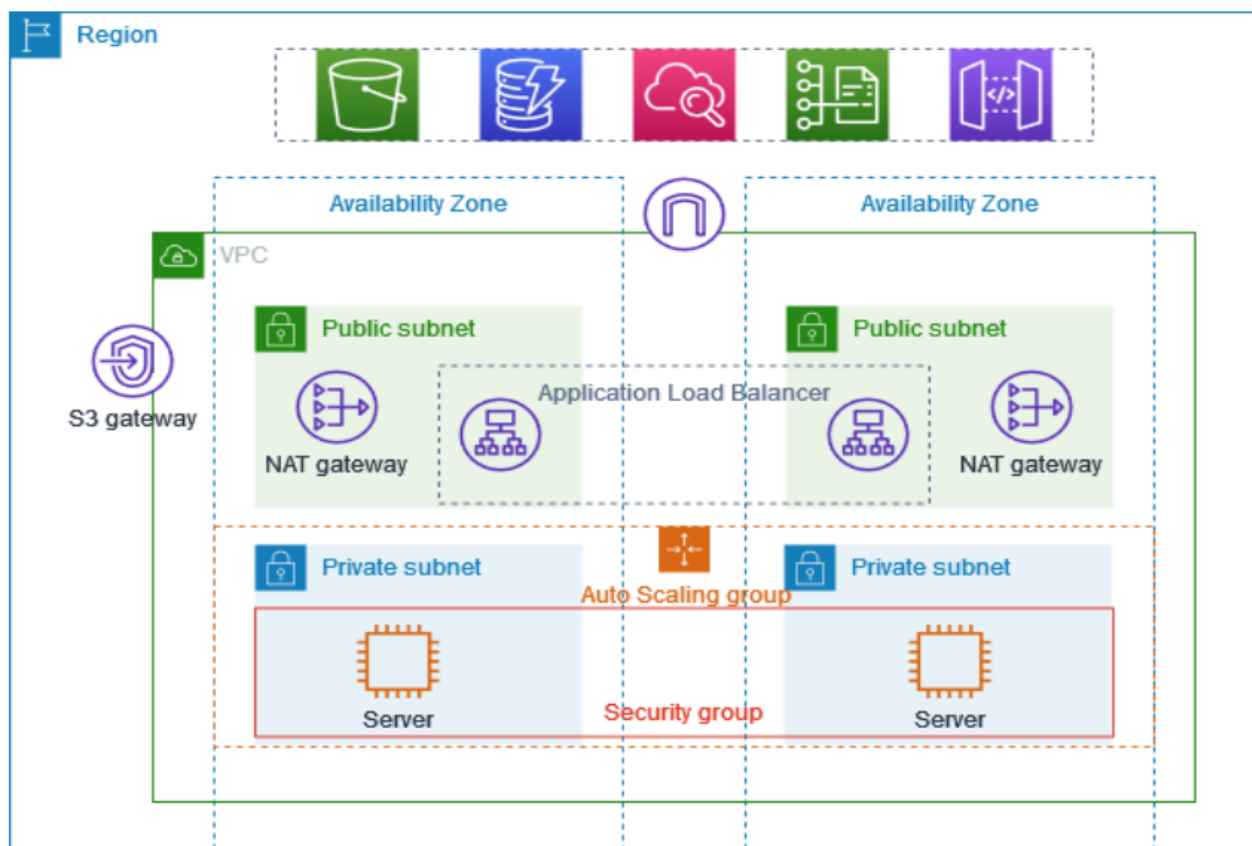


## Project Name: VPC with public-private subnet in Production

### About the Project

This example demonstrates how to create a VPC that you can use for servers in a production environment.

To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve the resiliency, you deploy the NAT gateway in both Availability Zones.



### Overview

The VPC has public and private subnets in two AZs.

Each public subnet contains a NAT gateway and a load balancer node.

The servers run in the private subnets which are launched and terminated by using an Auto Scaling group and receive traffic from the load balancer.

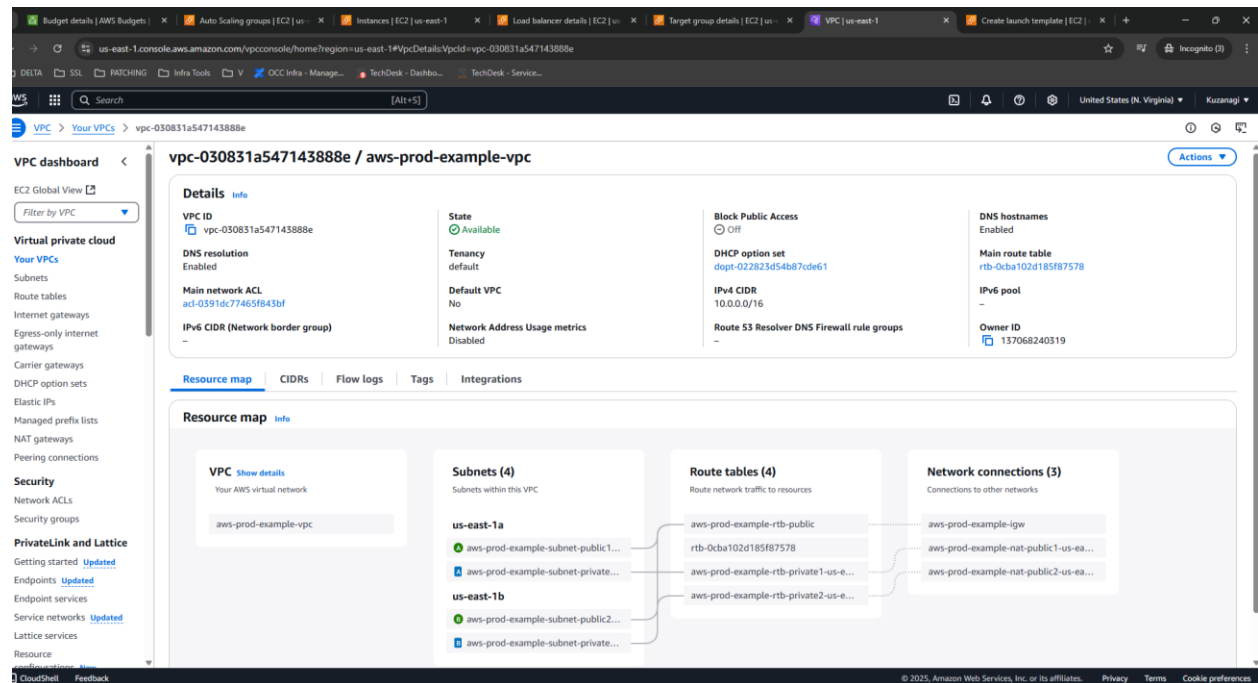
The servers can connect to the internet by using the NAT gateway. We launch a bastion instance for connecting to the servers in the private subnets for their configuration which includes creating the html landing page and running an http server using python.

## Instructions:

Step-by-step instructions can be followed on [Day-7 | AWS Project Used In Production | Complete Implementation](#)

AWS Console setup:

VPC



Auto Scaling Group

us-east-1-console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroupDetails:aws-example-prod-asgview=details

aws

EC2 > Auto Scaling groups > aws-example-prod-asg

### aws-example-prod-asg Capacity overview

arn:aws:autoscaling:us-east-1:137068240319:autoScalingGroup:8ac9111c-4b48-44f9-a6a2-ad49f69a4cce:autoScalingGroupName/aws-example-prod-asg

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	1 - 4	Units (number of instances)	-

Date created  
Wed Mar 26 2025 11:12:55 GMT+0800 (Philippine Standard Time)

Details Integrations - new Automatic scaling Instance management Instance refresh Activity Monitoring

### Launch template

Launch template lt-038eb8604c9fe864c aws-prod-example-template	AMI ID ami-084568db4383264d4	Instance type t2.micro	Owner arn:aws:iam:137068240319:root
Version Default	Security groups -	Security group IDs sg-0777affe12f5a1f8d	Create time Wed Mar 26 2025 11:08:39 GMT+0800 (Philippine Standard Time)
Description prod example template	Storage (volumes) -	Key pair name devops1	Request Spot Instances No

View details in the launch template console

### Network

Availability Zones us-east-1a, us-east-1b	Subnet ID subnet-05d2ac1d1f8ab3b5a, subnet-	Availability Zone distribution Balanced best effort
--	--	--

## Launch Template

us-east-1-console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplateDetails:launchTemplateId=lt-038eb8604c9fe864c

aws

EC2 > Launch templates > aws-prod-example-template

### aws-prod-example-template (lt-038eb8604c9fe864c)

Actions Delete template

#### Launch template details

Launch template ID lt-038eb8604c9fe864c	Launch template name aws-prod-example-template	Default version 1	Owner arn:aws:iam:137068240319:root
--	---	----------------------	--

Details Versions Template tags

#### Launch template version details

Version  
1 (Default)

Description  
prod example template

Date created  
2025-03-26T03:08:39.000Z

Created by  
arn:aws:iam:137068240319:root

Actions Delete template version

Instance details Storage Resource tags Network interfaces Advanced details

AMI ID ami-084568db4383264d4	Instance type t2.micro	Availability Zone -	Key pair name devops1
Security groups -	Security group IDs sg-0777affe12f5a1f8d		

## Target Group

us-east-1-console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroup/targetGroupArn=arn:aws:elasticloadbalancing:us-east-1:137068240319:targetgroup/aws-prod-example-tg/30a625e681fd4584:tab=targets

EC2 > Target groups > aws-prod-example-tg

**Details**

am:aws:elasticloadbalancing:us-east-1:137068240319:targetgroup/aws-prod-example-tg/30a625e681fd4584

Target type: Instance

Protocol: HTTP

Port: 8000

Protocol version: HTTP1

VPC: vpc-030831a547143888e

IP address type: IPv4

Load balancer: None associated

2 Total targets

2 Healthy

0 Unhealthy

0 Anomalous

0 Unused

0 Initial

0 Draining

**Distribution of targets by Availability Zone (AZ)**

Select values in this table to see corresponding filters applied to the Registered targets table below.

**Registered targets (2)**

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override	Override details
i-092f42b78dbf1d52f		8000	us-east-1a (us-east-1)	Healthy	-	No override	No override is currently active on ...
i-01c44a04f5046d717		8000	us-east-1b (us-east-1)	Healthy	-	No override	No override is currently active on ...

## Load Balancer

us-east-1-console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancer/loadBalancerArn=arn:aws:elasticloadbalancing:us-east-1:137068240319:loadbalancer/app/aws-prod-example-lb/bc1654d5fa9b7414:tab=listeners

EC2 > Load balancers > aws-prod-example-lb

**Application Load Balancers now support public IPv4 IP Address Management (IPAM)**

You can get started with this feature by configuring IP pools in the Network mapping section.

**aws-prod-example-lb**

**Details**

Load balancer type: Application

Status: Active

Scheme: Internet-facing

Hosted zone: Z35SXDOTRQ7X7K

VPC: vpc-030831a547143888e

Availability Zones: subnet-0ae2bf462ee7f7235 (us-east-1a (use1-az4)), subnet-99a86b83cac8 (us-east-1b (use1-az6))

Load balancer IP address type: IPv4

Date created: March 26, 2025, 12:01 (UTC+08:00)

Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:137068240319:loadbalancer/app/aws-prod-example-lb/bc1654d5fa9b7414

**Listeners and rules (1)**

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

Protocol/Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group • aws-prod-example-tg (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applicable

## Bastion setup

```

rellorvm-lnx@ELS-RltUozrUPZ:~/.ssh$ scp -i /home/rellorvm-lnx/.ssh/devops1.pem /home/rellorvm-lnx/.ssh/devops1.pem ubuntu@54.82.110.181:/home/ubuntu/.ssh
devops1.pem                               100% 1678    5.7KB/s   00:00
rellorvm-lnx@ELS-RltUozrUPZ:~/.ssh$ ssh -i devops1.pem ubuntu@54.82.110.181
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Mar 26 03:37:41 UTC 2025

System load:  0.0                Processes:    105
Usage of /:   25.2% of 6.71GB    Users logged in:  0
Memory usage: 20%              IPv4 address for enx0: 10.0.10.114
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Mar 26 03:31:49 2025 from 167.103.65.95
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

```

Running http server on both EC2 instances hosted in the Private subnet

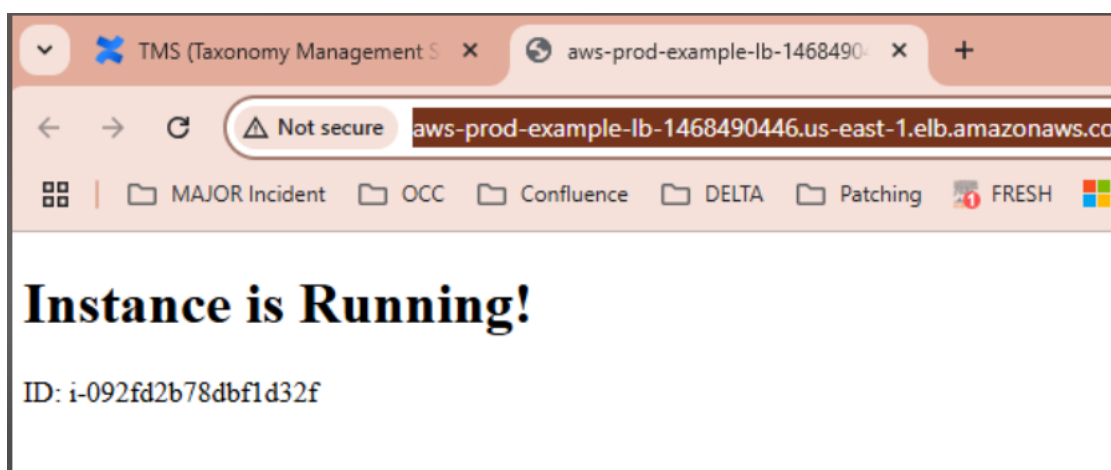
```

ubuntu@ip-10-0-150-73:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.148.44 - - [26/Mar/2025 04:35:19] "GET / HTTP/1.1" 200 -
10.0.4.232 - - [26/Mar/2025 04:35:19] "GET / HTTP/1.1" 200 -
10.0.148.44 - - [26/Mar/2025 04:35:49] "GET / HTTP/1.1" 200 -
10.0.4.232 - - [26/Mar/2025 04:35:49] "GET / HTTP/1.1" 200 -
10.0.4.232 - - [26/Mar/2025 04:35:51] "GET / HTTP/1.1" 200 -

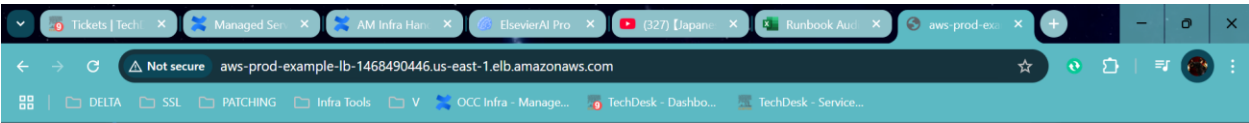
```

**Output:**

Accessing the DNS of the load balancer (shows html landing page set on Instance #1)



Accessing the DNS of the load balancer (shows html landing page set on Instance #2)



## 2nd Instance Running

ID: i-01c44a04f5046d717