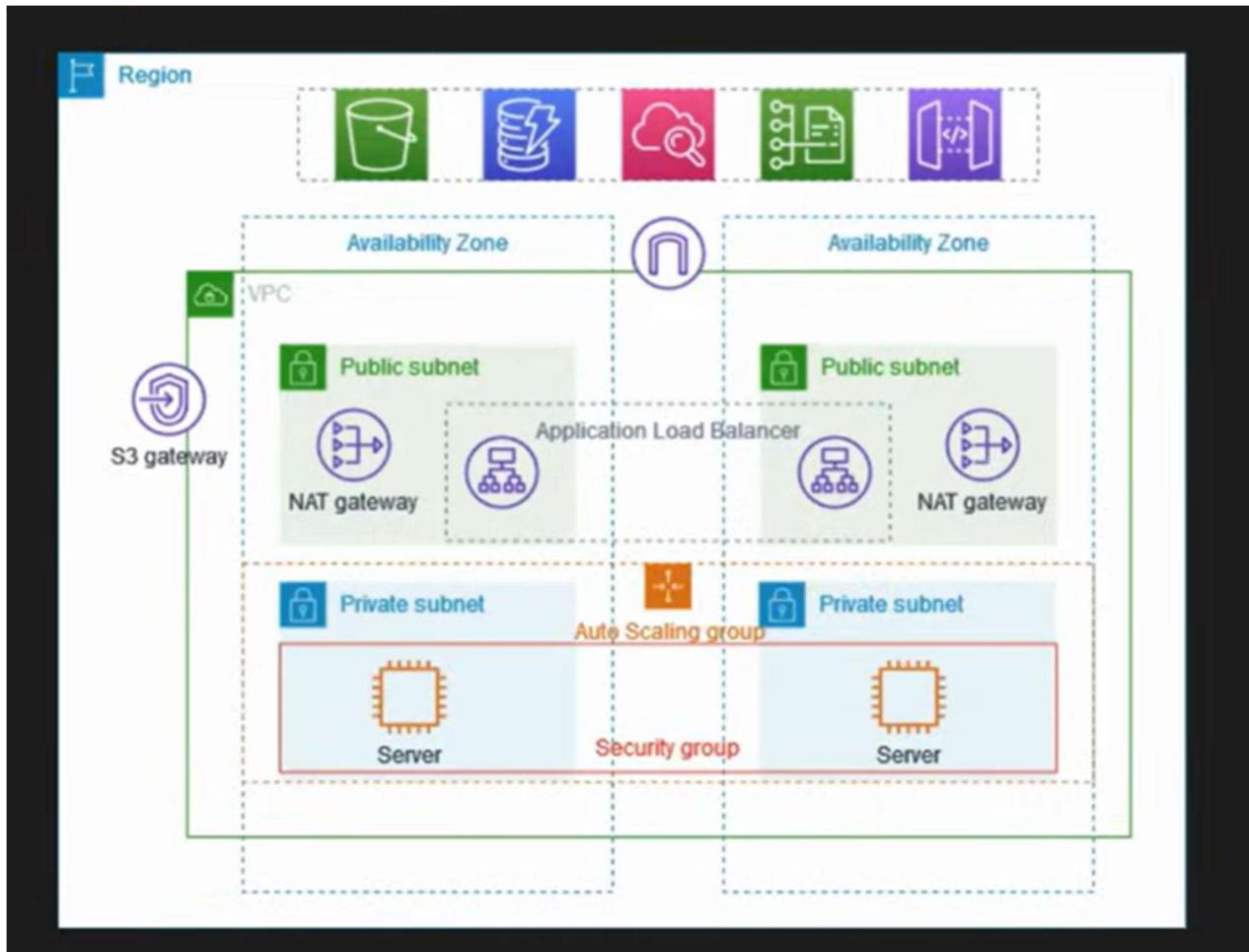


Name:Roshan Saral Kumar

AWS-PROJECT-1 VPC WITH PUBLIC-PRIVATE SUBNET IN PRODUCTION

- **ARCHITECTURE:-**



- **DESCRIPTION:-**

- Using Amazon Web Services (AWS), this project showcases a scalable and reliable cloud infrastructure design that is suited for production-grade deployments. To guarantee high availability and fault tolerance, the architecture makes use of a Virtual Private Cloud (VPC) that is configured with both public and private subnets deployed over two availability zones.
- Key Components and Features:
 - VPC Configuration: A logically separated network environment that safely hosts all Amazon Web Services resources.
 - Public Subnets: These subnets are deployed in each Availability Zone and host NAT Gateways, allowing private instances to have secure outbound internet access.

- Private Subnets: Dedicated to application servers and shielded from direct internet access for added security.
- The Application Load Balancer (ALB): distributes incoming traffic across many instances across private subnets to ensure load balancing and fault tolerance.
- Auto Scaling Group: Optimises cost and performance by automatically adjusting the number of EC2 instances based on traffic demand.
- Security Groups: Serve as instance-level virtual firewalls to regulate incoming and outgoing traffic.
- Accessing Amazon S3 securely and effectively without using the public internet is made possible by the S3 Gateway Endpoint

- **ADVANTAGES:-**

- High Availability: Resilience against zone-level failures is ensured by multi-AZ deployment.
- Scalability: Adapting to traffic patterns, Auto Scaling and ALB offer dynamic resource management.
- Security: A secure working environment is ensured by NAT, security groups, and the separation of public and private resources.
- Cost optimisation: Effective usage of scaling tools and NAT gateways aids in controlling operating expenses.

- **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 resiliency, you deploy the NAT gateway in both Availability Zones.

- PROJECT IMPLEMENTATION:-

STEP 1:- Logging in as root user and entering into my aws management console.

The screenshot shows the AWS Management Console Home page with the following sections:

- Recently visited:** S3, EC2, IAM, CloudFormation, AWS Health Dashboard, Billing and Cost Management, AWS Global View.
- Applications:** 0 applications. Region: US East (N. Virginia). Create application button.
- Welcome to AWS:** Getting started with AWS, Learn the fundamentals and find valuable information to get the most out of AWS. Training and development resources.
- AWS Health:** Info, Open issues (0) Past 7 days, Scheduled changes (0) Upcoming and past 7 days, Other notifications (0) Past 7 days.
- Cost and usage:** Upgrade plan, Credits cover your free plan costs. Your access to AWS services will end when credits are depleted or free period ends. Credits remaining: \$119.96 USD, Days remaining: 149 days (Mar 14, 2026). Current month: \$0.00 ▼ 100%, Forecasted month end: Data unavailable.

Step 2:- SEARCHING FOR VPC ISOLATED CLOUD SERVICES IN AWS MANAGEMENT CONSOLE.

The screenshot shows the AWS Management Console search results for the term "VPC".

Services

- VPC** Isolated Cloud Resources
- AWS Global View** AWS Global View provides a global dashboard and search functionality that lets yo...
- AWS Firewall Manager** Central management of firewall rules

Features

- Dashboard** ■ VPC feature
- Route 53 VPCs** ■ Route 53 feature
- VPC links** ■ API Gateway feature

Documentation

- VPC** Implementation Guide
- VPC to VPC connectivity** AWS Whitepaper

Were these results helpful?

Training area Data unavailable

STEP 3:- Once we click on VPC we have to create the VPC AT THE VPC DASHBOARD BY CLICKING ON “ Create VPC ”.

The screenshot shows the AWS VPC Dashboard. At the top, there are two buttons: "Create VPC" (highlighted in orange) and "Launch EC2 Instances". Below them is a note: "Note: Your Instances will launch in the United States region." On the left, a sidebar lists categories like "Virtual private cloud", "Security", "PrivateLink and Lattice", etc. The main area is titled "Resources by Region" and shows the following data:

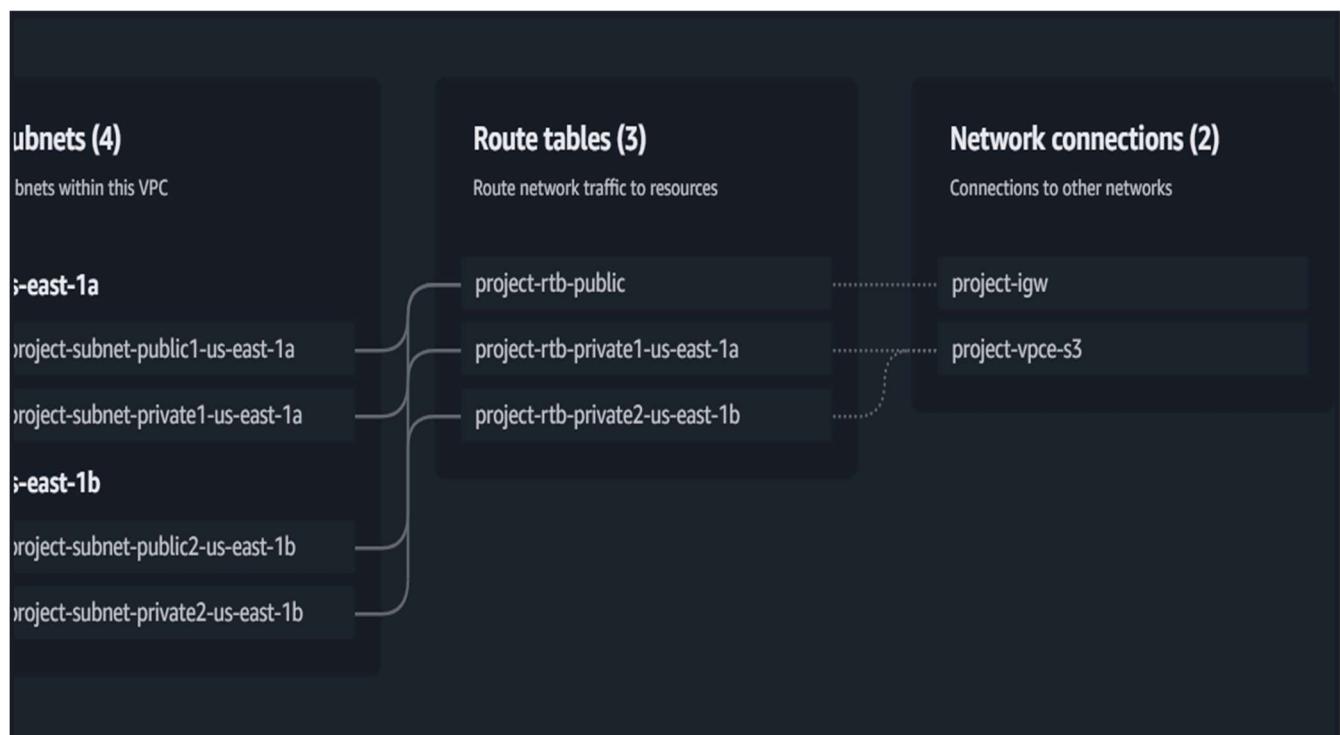
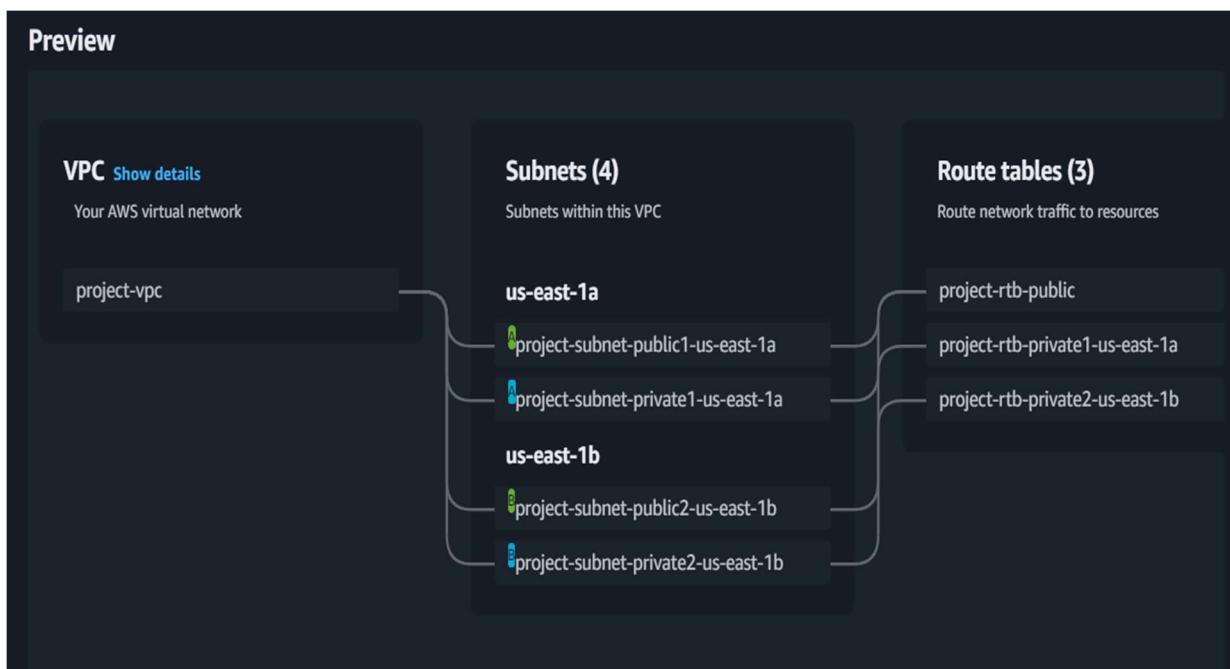
Resource Type	Region	Count
VPCs	N. Virginia	1
Subnets	N. Virginia	6
Route Tables	N. Virginia	1
Internet Gateways	N. Virginia	1
Egress-only internet gateways	N. Virginia	0
Carrier gateways	N. Virginia	0
DHCP option sets	N. Virginia	1
Elastic IPs	N. Virginia	0
Managed prefix lists	N. Virginia	0
NAT gateways	N. Virginia	0
Peering connections	N. Virginia	0
Route servers	New	0
Network ACLs	N. Virginia	1
Security Groups	N. Virginia	1
Egress-only Internet Gateways	N. Virginia	0
Customer Gateways	N. Virginia	0
DHCP option sets	N. Virginia	1
Virtual Private Gateways	N. Virginia	0
Endpoints	N. Virginia	0
Site-to-Site VPN Connections	N. Virginia	0
Instance Connect	N. Virginia	0

On the right side, there are three sections: "Service Health" (with a link to "View complete service health details"), "Settings" (with links to "Block Public Access", "Zones", and "Console Experiments"), and "Additional Information" (with links to "VPC Documentation", "All VPC Resources", "Forums", and "Report an Issue"). A large callout box on the right is titled "AWS Network Manager" with the subtext: "AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity management,".

STEP 4: AFTER CLICKING ON CREATE VPC WE HAVE TWO OPTIONS ONE IS VPC ONLY AND THE OTHER IS VPC AND MORE. CHOOSE VPC AND MORE SO THAT WE DON'T HAVE TO MAKE MANUAL SUBNETS IF WE WANT MANUAL SUBNETS OR VPCS WE CAN GO FOR VPC ONLY.

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. The top navigation bar includes 'VPC', 'Your VPCs', and 'Create VPC'. The main title is 'Create VPC' with an 'Info' link. A descriptive text states: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.' On the left, under 'VPC settings', there are two radio button options: 'VPC only' (unchecked) and 'VPC and more' (checked). Below this, the 'Name tag auto-generation' section is shown, with 'Auto-generate' checked and 'project' entered. The 'IPv4 CIDR block' section shows '10.0.0.0/16' and '65,536 IPs'. The 'IPv6 CIDR block' section has 'No IPv6 CIDR block' selected. Under 'Tenancy', 'Default' is chosen. At the bottom, the 'Number of Availability Zones (AZs)' section shows '2' selected from a range of 1 to 3, with a link to 'Customize AZs'. On the right, a 'Preview' panel shows a summary of the VPC configuration, including the name 'VPC Show details Your AWS virtual network project-vpc'. A vertical bracket on the right side of the preview panel indicates it spans from the 'VPC settings' section to the 'Preview' section.

→ VPC AND MORE GIVES US A PREVIEW DIAGRAMS:-



Step 5: AFTER THE PREVIEW IS SEEN WE ARE TRYING TO ELIMINATE THE S3 BUCKET AS IT IS OFF NOT MUCH USE IN THE FOLLOWING OPTIONS AND WE ARE TRYING TO RENAME THE PROJECT AS “aws-prod-example” and everything that is private and public subnets are made by default . so in the ARCHITECTURE GIVEN WE NEED 2 PUBLIC SUBNETS AND 2 PRIVATE SUBNETS ALONG WITH A NAT GATEWAY PER AVAILABILITY ZONE.SO WE ARE MODIFYING THE OPTIONS AND REMOVING S3 BY CLICKING ON THE NONE OPTION FOR VPC ENDPOINTS.

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 | 2 | 3

► **Customize AZs**

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

► **Customize subnets CIDR blocks**

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None | In 1 AZ | 1 per AZ

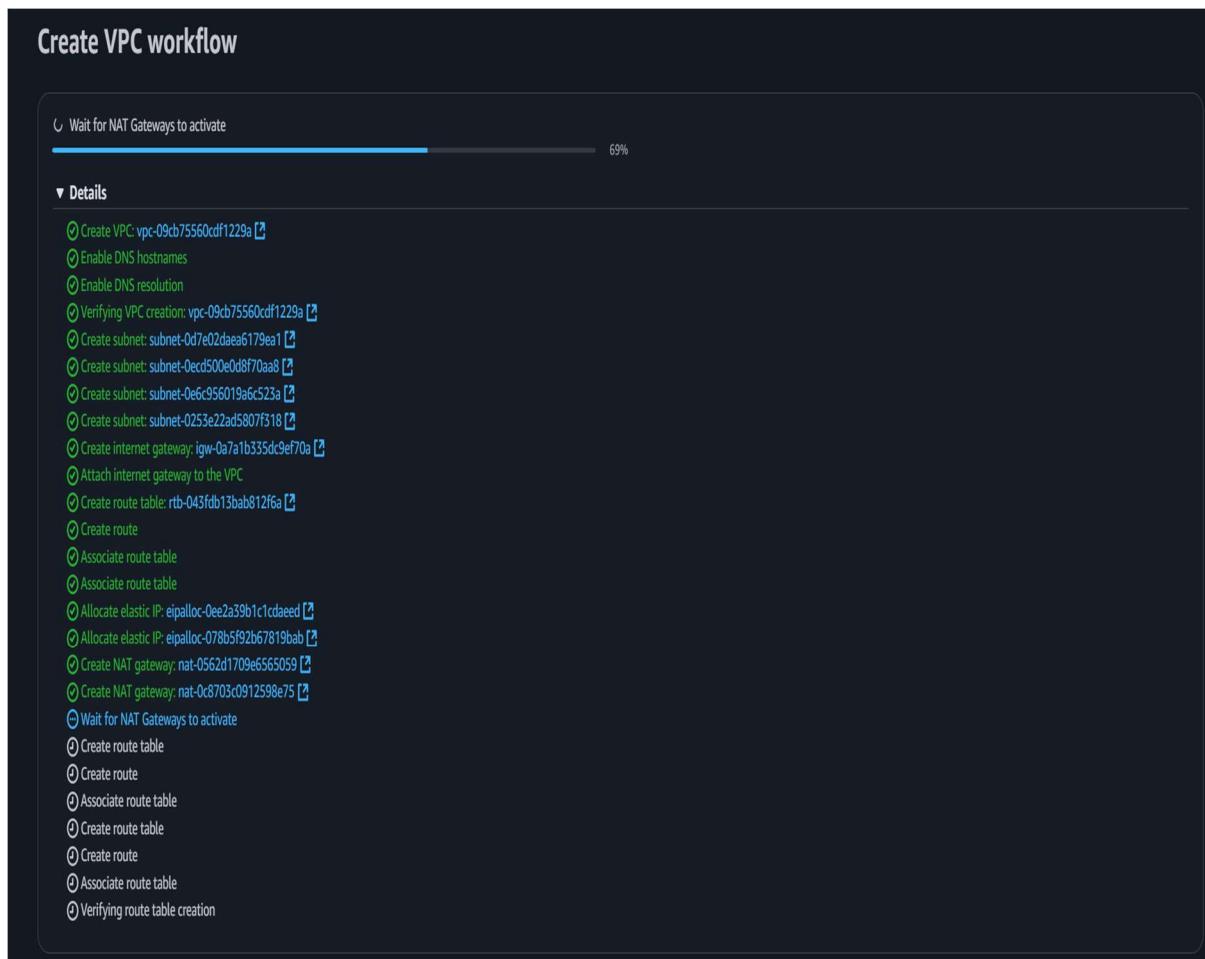
VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

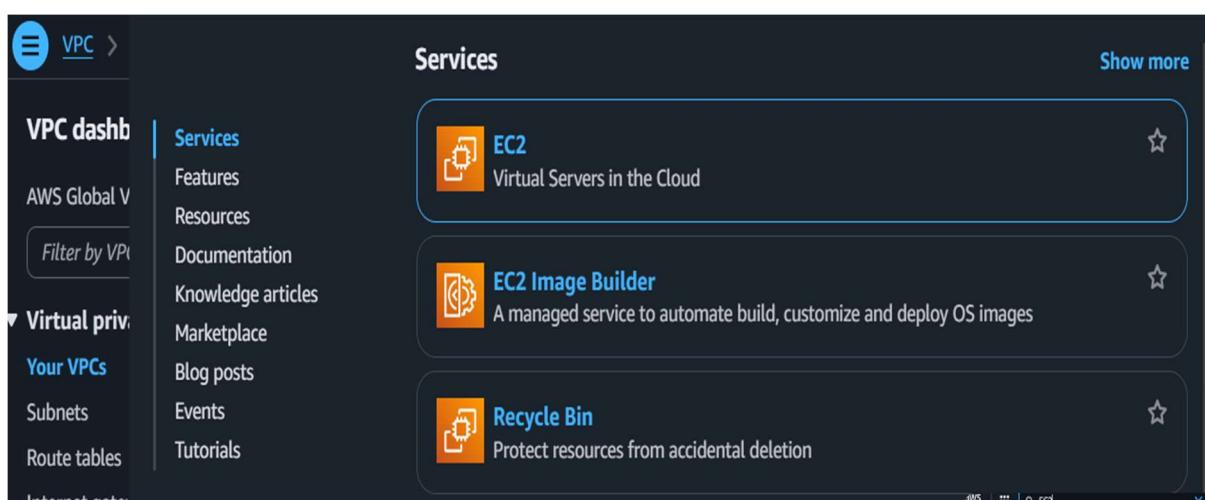
None | S3 Gateway

AFTER SELECTING THE RIGHT OPTIONS CLICK ON CREATE VPC.

VPC WORKFLOW AND WE HAVE TO WAIT FOR THE NAT GATEWAYS TO ACTIVATE



STEP 6: Search for AUTOSCALING GROUP IN EC2 IN AWS MANAGEMENT CONSOLE.



STEP 7:- CLICK ON AUTOSCALING GROUPS IN THE EC2 DASHBOARD AND CLICK ON CREATE AUTOSCALING GROUPS. AND THEN CLICK ON LAUNCH TEMPLATES TO

PROVIDE AUTO SCALING GROUPS EASILY FOR THE TRAFFIC FLOWING IN.
AFTER THIS WE ARE CREATING A LAUNCH TEMPLATE

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The top navigation bar includes the AWS logo, search bar, and account information for 'United States (N. Virginia)'. The breadcrumb trail shows the user is at 'EC2 > Launch templates > Create launch template'. The main title is 'Create launch template'. Below it, a note states: 'Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time with multiple versions.' The first step, 'Launch template name and description', is active. It contains a 'Launch template name - required' field with the value 'MyTemplate', a note that it must be unique and up to 128 characters, and a 'Template version description' field with the value 'A prod webserver for MyApp'. There is also a note about the maximum length of 255 characters. Under 'Auto Scaling guidance', there is an info link and a checkbox for 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling', which is checked. Below this, two expandable sections are shown: 'Template tags' and 'Source template'. The second step, 'Launch template contents', is partially visible below. At the bottom, a section for 'Application and OS Images (Amazon Machine Image) - required' is expanded, showing a note about selecting an AMI.

aws | Search [Alt+S] | United States (N. Virginia)

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time with multiple versions.

Launch template name and description

Launch template name - *required*

MyTemplate

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

A prod webserver for MyApp

Max 255 chars

Auto Scaling guidance | Info

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags

► Source template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) - *required* | Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, you can upload one.

Select the required options like os,instance type etc don't chose default vpc choose the one which we made that is "aws-prod-example-vpc" allow ssh and http traffic that is inbound traffic

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time across multiple versions.

Launch template name and description

Launch template name - *required*

aws-prod-example

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

proof of concept for app deployment in aws private subnet

Max 255 chars

Auto Scaling guidance | [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Quick Start



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-0360c520857e3138f (64-bit (x86)) / ami-026fccd88446aa0bf (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-0360c520857e3138f

Publish Date

2025-08-21

Username

ubuntu

Verified provider

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0392 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

aws_login  [Create new key pair](#)

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template  [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Availability Zone [Info](#)

Don't include in launch template  [Enable additional zones](#)

Not applicable for EC2 Auto Scaling

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Select existing security group](#) [Create security group](#)

Security group name - required

aws-prod-example  [Create security group](#)

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-/.@#=;&;!\$*

Description - required

allow ssh access

VPC [Info](#)

vpc-09cb75560cdf1229a (aws-prod-example-vpc)  [Create security group](#)

Inbound Security Group Rules

- ▼ Security group rule 1 (TCP, 22)

[Remove](#)

Type Info	Protocol Info	Port range Info
ssh	TCP	22
Source type Info	Source Info	Description - optional Info
Custom	<input type="text" value="Add CIDR, prefix list or security group"/>	e.g. SSH for admin desktop
- ▼ Security group rule 2 (TCP, 8000, 0.0.0.0/0)

[Remove](#)

Type Info	Protocol Info	Port range Info
Custom TCP	TCP	8000
Source type Info	Source Info	Description - optional Info
Anywhere	<input type="text" value="Add CIDR, prefix list or security group"/> 0.0.0.0/0 X	e.g. SSH for admin desktop

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

AFTER WE HAVE LAUNCHED THE TEMPLATE WIHOUT ADDING ANY EBS VOLUMES IT WILL BE CREATED THEN GO TO THE AUTOSCALING GROUP IN THE EC2 INSTANCE DASHBOARD AND REFRESH THE PAGE SO THAT WE CAN SEE THE TEMPLATE THEAT WAS LAUNCHED RECENTLY WHCICH IS “aws-prod-example”.

Choose launch template [Info](#)
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#)

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

(C)

Create a launch template [\[?\]](#)

Version

(C)

Create a launch template version [\[?\]](#)

Description proof of concept for app deployment in aws private subnet	Launch template aws-prod-example lt-0c41f940ade161f27	Instance type t3.micro
AMI ID ami-0360c520857e5138f	Security groups -	Request Spot Instances No
Key pair name aws_login	Security group IDs sg-0b532226ea8f784f3	

Additional details

INSTANCE TYPE REQUIREMENTS:-

Instance type requirements [Info](#)

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
aws-prod-example [edit] lt-0c41f940ade161f27	Default	proof of concept for app deployment in aws private subnet

Instance type
t3.micro

Network [Info](#)

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
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-09cb75560cdf1229a (aws-prod-example-vpc)
10.0.0.0/16 [\[edit\]](#) [\[refresh\]](#)

[Create a VPC](#) [\[edit\]](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets [\[refresh\]](#)

use1-az2 (us-east-1a) | subnet-0e6c956019a6c523a (aws-prod-example-subnet-private1-us-east-1a)
10.0.128.0/20 [\[edit\]](#) [\[refresh\]](#)

use1-az4 (us-east-1b) | subnet-0253e22ad5807f318 (aws-prod-example-subnet-private2-us-east-1b)
10.0.144.0/20 [\[edit\]](#) [\[refresh\]](#)

[Create a subnet](#) [\[edit\]](#)

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

BOTH INSTANCES SHOULD BE RUNNING IN MY PRIVATE SUBNETS AS IN THE ARCHTECTURE.

IN THE BELOW PICTURE U MIGHT SEE ME NOT CLICKING ANY LOAD BALNCER BECAUSE I DON'T WANT ANY LOAD BALANCER IN MY ARCHITECTURE FOR PRIVATE SUBNET INSTEAD I WILL CREATE AN APPLICATION LOAD BALNCER FOR MY PUBLIC SUBNET.

Integrate with other services - optional [Info](#)

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 [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

Select Load balancing options

No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

VPC Lattice integration options [Info](#)

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
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#) [\[edit\]](#)

Application Recovery Controller (ARC) zonal shift - new [Info](#)

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

Enable zonal shift
New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.

VPC Lattice integration options [Info](#)

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
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#)

Application Recovery Controller (ARC) zonal shift - *new* [Info](#)

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

Enable zonal shift
New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional [Info](#)

Turn on Elastic Load Balancing health checks
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

Turn on VPC Lattice health checks
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Turn on Amazon EBS health checks
EBS monitors whether an instance's root volume or attached volume stalls. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance on its next periodic health check.

Health check grace period [Info](#)

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

Taking default options and moving on health checks is 300 seconds.

MAXIMUM DESIRED CAPACITY TAKING AS 4 INSTANCES AND MINMUM DESIRED CAPACITY IS 1 INSTANCE AND THE DESIRED CAPACITY IS 2 INSTANCES

Configure group size and scaling - optional [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

2

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
1	4

Equal or less than desired capacity Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

IAM NOT ADDING ANY NOTIFICATIONS LIKE SNS FOR AUTO SCALING GROUPS AND CLICKING ON NEXT FOR LAUNCHING THE AUTOSCALING GROUPS.

The screenshot shows the AWS Auto Scaling Groups page. At the top, a green banner displays the message "aws-prod-example created successfully". Below the banner, the heading "Auto Scaling groups (1) Info" is visible. The main table lists one item: "aws-prod-example" with a status of "aws-prod-example | Version Default" and "0" instances. The "Desired capacity" is set to "2". The "Creation time" is listed as "Fri Oct 17 2025 14:11:58 GMT+0530 (India Standard Time)". The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, Availability Zones, and Creation time.

MY AUTOSCALING GROUP IS LAUNCHED SUCCESSFULLY.

MY UPDATING CAPACITY STATE in autoscaling group has disappeared and 2 instances are launched in EC2.GO TO EC2 AND VERIFY.

The screenshot shows the AWS Auto Scaling Groups page. The "aws-prod-example" group now has "2" instances. The "Desired capacity" is still "2". The "Creation time" is listed as "Fri Oct 17 2025 14:11:58 GMT+0530 (India Standard Time)". The table structure remains the same as the previous screenshot.

IN EC2:- 2 INSTANCES ARE LAUNCHED.

The screenshot shows the AWS Instances page. It displays two EC2 instances: "i-05a0f16e5efb15985" and "i-04458b4341a6b7056", both in a "Running" state. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, Elastic IP, and IPv6 IPs.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs
i-05a0f16e5efb15985	i-05a0f16e5efb15985	Running	t3.micro	0/3 checks passed	View alarms +	us-east-1a	-	-	-	-
i-04458b4341a6b7056	i-04458b4341a6b7056	Running	t3.micro	0/3 checks passed	View alarms +	us-east-1b	-	-	-	-

AFTER WE HAVE LAUNCHED THE INSTANCES IN THE PRIVATE SUBNET WE HAVE TO INSTALL THE APPLICATIONS FIRST IN THE PRIVATE SUBNET BY ENTERING INTO THE EC2 INSTANCES WHICH ARE PRIVATE WHICH MEANS THEY DON'T HAVE AN PUBLIC IPV4 ADDRESS INSTEAD THEY HAVE PRIVATE IPV4 ADDRESS SO WE HAVE TO USE "BASTION HOST" TO ENTER INTO THESE INSTANCES WHICH ACTS AS A MEDIATOR BETWEEN THE PUBLIC AND PRIVATE SUBNETS.

CREATING BASTION HOST BY LAUNCHING AN INSTANCE

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

bastion-host

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

AWS Mac

Ubuntu Microsoft Red Hat SUSE Linux Debian

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-0360c520857e5138f (64-bit (x86)) / ami-026fccc88446aa0bf (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

EDIT THE KEY PAIR AND NETWORK SETTINGS THE NETWORK CONFIGURATION SHOULD CHANGE TO aws-prod-vpc that is the vpc which we created otherwise bastion-host will not be able to access the ec2 instances which are present in the private subnet.

The screenshot shows the AWS Launch Wizard configuration interface. It includes sections for Key pair (login), Network settings, Auto-assign public IP, and Firewall (security groups). The Key pair (login) section shows 'aws_login' selected. The Network settings section shows 'vpc-09cb75560cdf1229a (aws-prod-example-vpc)' and 'subnet-0253e22ad5807f318'. The Auto-assign public IP section has 'Disable' selected. The Firewall (security groups) section shows 'Create security group' selected, with a note about security group naming rules.

Entering into the bastion-host in wsl.

```
roshan@Roshan-S:/mnt/c/Users/rosha$ cp /mnt/c/Users/rosha/Downloads/aws_login.pem ~/
roshan@Roshan-S:/mnt/c/Users/rosha$ chmod 400 ~/aws_login.pem
roshan@Roshan-S:/mnt/c/Users/rosha$ scp -i ~/aws_login.pem ubuntu@34.205.74.56:/home/ubuntu
aws_login.pem
100% 1674      6.7KB/s  00:00
```

I am doing ssh to my bastion-host to enter the bastion instance in my public subnet to access the ec2 instance running in my private subnet.

```
roshan@Roshan-S:/mnt/c/Users/rosha$ ssh -i ~/aws_login.pem ubuntu@34.205.74.56
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

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

System information as of Fri Oct 17 09:32:10 UTC 2025

  System load:  0.0          Temperature:      -273.1 C
  Usage of /:   26.0% of 6.71GB  Processes:       111
  Memory usage: 22%           Users logged in:  0
  Swap usage:   0%            IPv4 address for ens5: 10.0.0.115

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
```

AS WE CAN SEE BY DOING THIS I HAVE SUCCESSFULLY GOT INTO MY PRIVATE SUBNET IN WSL . BY DOING SSH TO MY PRIVATE SUBNET IN USE-EAST-1A.

```
ubuntu@ip-10-0-0-115:~$ ls
aws_login.pem
ubuntu@ip-10-0-0-115:~$ ssh -i aws_login.pem ubuntu@10.0.133.242
The authenticity of host '10.0.133.242 (10.0.133.242)' can't be established.
ED25519 key fingerprint is SHA256:CPYKSvzPAX6oX9CNtkrOYjB26imp0/sipY0l+zoNDAA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.133.242' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

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

System information as of Fri Oct 17 09:36:10 UTC 2025

  System load:  0.0          Temperature:      -273.1 C
  Usage of /:   25.8% of 6.71GB  Processes:       110
  Memory usage: 25%           Users logged in:  0
  Swap usage:   0%            IPv4 address for ens5: 10.0.133.242

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

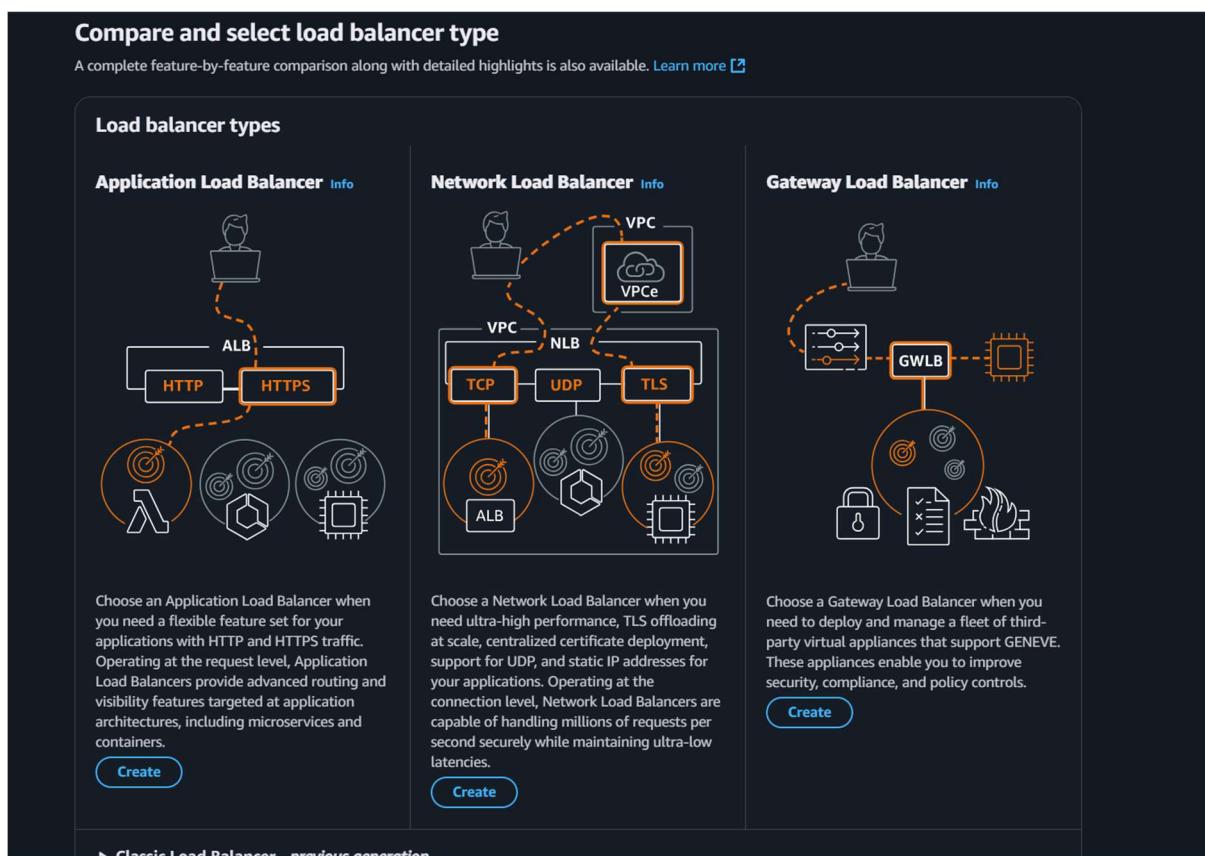
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
```

In the private subnet we are creating a simple index.html file. And running the server in 8000 port which was developed in the aws management console.

```
ubuntu@ip-10-0-133-242:~$ vim index.html
ubuntu@ip-10-0-133-242:~$ python3 -m http.server 8000
```

CREATE LOAD BALANCERS FOR SENDING THE TRAFFIC TO THESE EC2 INSTANCES CREATED IN THE PRIVATE SUBNETS THE VERIFICATION IS THAT IN USE-EAST-1A THE PYTHON3 IS INSTALLED SO IT HITS AND GIVES A MESSAGE WITHOUT ANY ERROR BUT WHEN WE ARE TRYING TO ACCESS THE SECOND PRIVATE SUBNET EC2 INSTANCE IT WILL GIVE AN ERROR MESSAGE AS PYTHON IS NOT INSTALLED



DIFFERENT TYPES OF LOAD BALANCERS WE CAN CHOOSE.

LOAD BALNCER SHOULD BE INTERNET FACING BECAUSE WE ARE TRYING TO ACCESS VIA THE PUBLIC SUBNET.

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
 aws-prod-example
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

Internet-facing
• Serves internet-facing traffic.
• Has public IP addresses.
• DNS name resolves to public IPs.
• Requires a public subnet.

Internal
• Serves internal traffic.
• Has private IP addresses.
• DNS name resolves to private IPs.
• Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type [Info](#)
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only.

NETWORK MAPPING IS FOR THE PUBLIC SUBNETS NOT PRIVATE SUBNETS FOR APPLICATION LOAD BALNCER COOHSE ONLY PUBLIC SUBNETS.

Load balancer IP address type [Info](#)
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only.

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

vpc-09cb75560cdf1229a (aws-prod-example-vpc)

IP pools [Info](#)
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view [Pools](#) in the Amazon VPC IP Address Manager console.

Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets [Info](#)
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az2)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-0d7e02dea6179ea1
IPv4 subnet CIDR: 10.0.0.0/16

us-east-1b (use1-az4)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-0ec500e0d8f70aa8
IPv4 subnet CIDR: 10.0.16.0/20

ADD SECURITY GROUP aws-prod-example and make a target group by adding target groups mention target group as ec2 instances.

The screenshot shows the AWS Load Balancer configuration interface. At the top, there's a section for 'Security groups' with a note about firewall rules. Below it, 'Listeners and routing' is shown, with a 'Listener HTTP:80' entry. Under 'Default action', 'Forward to target groups' is selected. In the 'Target group' section, a new target group is being created with the name 'aws-prod-example'. The 'Instances' target type is chosen, and the 'Protocol' is set to 'HTTP' with port '80'. A weight of '1' is assigned with a 'Percent' of '100%'. There's also a note about adding up to 4 more target groups.

CREATING TARGET GROUPS:-

The screenshot shows the 'Create target group' wizard. Step 1, 'Create target group', is selected. It shows basic configuration settings: 'Basic configuration' (load balancing to targets in a specific VPC), 'Choose a target type' (Instances selected, showing benefits like EC2 Auto Scaling support), 'Target group name' (left blank), and 'Protocol' (HTTP, port 80). Step 2, 'Register targets', is shown as the next step.

WHILE REGISTERING THE TARGETS DON'T SELECT BASTION-HOST SELECT THE OTHER TWO INSTANCES.

Register targets

This is an optional step to create a target group. However, to ensure your load balancer routes traffic to this target group you must register your targets.

Available instances (2/3)

Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID
i-0f9c93ba8c43777a7	bastion-host	Running	launch-wizard-1	us-east-1a	10.0.0.115	subnet-0d7e02daea6179e
<input checked="" type="checkbox"/> i-05a0f16e5fb15985		Running	aws-prod-example	us-east-1a	10.0.133.242	subnet-0e6c956019a6c52
<input checked="" type="checkbox"/> i-04458b4341a6b7056		Running	aws-prod-example	us-east-1b	10.0.148.243	subnet-0253e22ad5807f3

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80
1-65535 (separate multiple ports with commas)

Review targets

THE LOAD BALANCER is in the active state.

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	Status	Type	Scheme	IP address type	VPC ID	Availability Zones	Security groups	DNS name
<input checked="" type="checkbox"/> aws-prod-example	Active	application	Internet-facing	IPv4	vpc-09cb75560cdf1229a	2 Availability Zones	sg-0b53226ea8f784f3	aws-prod-example-896504...

Load balancer: aws-prod-example

Details

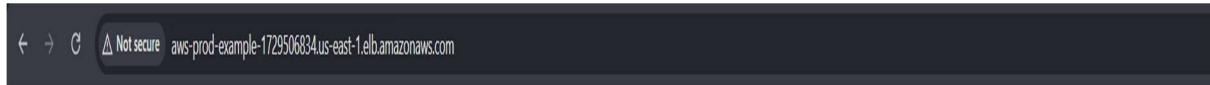
Load balancer type Application	Status Active	VPC vpc-09cb75560cdf1229a	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z355XD0TRQ7X7K	Availability Zones subnet-0d7e02daea6179e1 us-east-1a (use1-az2) subnet-0ecd500e0d8f70aa8 us-east-1b (use1-az2)	Date created October 17, 2025, 16:02 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:518286664533:loadbalancer/app/aws-prod-example/a427c5fd80f6229	DNS name info aws-prod-example-896504700.us-east-1.elb.amazonaws.com (A Record)		

BY using the DNS NAME AT THE APPLICATION LOAD BALANCER AND COPY PASTING IT IN A NEW TAB I WILL BE ABLE TO CREATE MY APPLICATION IN A PRIVATE SUBNET

The screenshot shows the AWS Application Load Balancer (ALB) configuration page for the load balancer named "aws-prod-example". The "Listeners and rules" tab is selected. There is one listener rule defined:

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group aws-prod-example [1] 1 (100%)	1 rule	arn:aws:elasticloadbalancing:us-east-1:518286664533:loadbalancer/app/aws-prod-example/f2826f4886350f0b	Not applicable	Not applicable	Not applicable	Not applicable

OUTPUT FOR First Private subnet (us-east-1a)



MY first AWS PROJECT to demonstrate apps in private subnet

This is a paragraph.

I have also made both the applications healthy and saw how one was healthy and unhealthy using the following DNS NAME of load balancer and I was able to understand how the load balancer functions

The screenshot shows the AWS Elastic Load Balancing console for the 'aws-prod-example' target group. At the top, there's a summary table with the following data:

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 8000	HTTP1	vpc-09cb75560cdf1229a [2]
IP address type	Load balancer		
IPv4	aws-prod-example [2]		

Below the summary table, there's a status breakdown:

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0
	Healthy	Unhealthy	Unused	Initial	Draining
	0 Anomalous				

A note below the status table says: "► Distribution of targets by Availability Zone (AZ) Select values in this table to see corresponding filters applied to the Registered targets table below."

Below the status table, there are tabs for Targets, Monitoring, Health checks, Attributes, and Tags. The Targets tab is selected.

The Registered targets table has the following columns:

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative o...	Override details	Launch...	Anomaly c...
i-05a0f16e5eb15985		8000	us-east-1a (us...)	Healthy	-	No override	No override is curre...	October 1...	Normal
i-04458b4341a6b7056		8000	us-east-1b (us...)	Healthy	-	No override	No override is curre...	October 1...	Normal

The below picture represents the way I was accessing the ec2 instance in my private subnets and checking if there were healthy or not in the load balancer.so what we understand is even if one server is down the other server is able to run in a different availability zone for backups.

```
ubuntu@ip-10-0-148-243:~  X  ubuntu@ip-10-0-133-242:~  X  +  -
10.0.27.135 - - [17/Oct/2025 12:07:47] "GET / HTTP/1.1" 304 -
^C
Keyboard interrupt received, exiting.
ubuntu@ip-10-0-148-243:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.27.135 - - [17/Oct/2025 12:08:12] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:08:12] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:08:27] "GET / HTTP/1.1" 304 -
10.0.0.152 - - [17/Oct/2025 12:08:33] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:08:42] "GET / HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
ubuntu@ip-10-0-148-243:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.0.152 - - [17/Oct/2025 12:09:42] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:09:42] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:09:46] "GET / HTTP/1.1" 304 -
10.0.0.152 - - [17/Oct/2025 12:09:50] "GET / HTTP/1.1" 304 -
10.0.0.152 - - [17/Oct/2025 12:10:03] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:10:12] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:10:33] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:10:42] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:11:03] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:11:12] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:11:33] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:11:42] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:12:03] "GET / HTTP/1.1" 200 -
10.0.27.135 - - [17/Oct/2025 12:12:12] "GET / HTTP/1.1" 200 -
10.0.0.152 - - [17/Oct/2025 12:12:33] "GET / HTTP/1.1" 200 -
```

OUTPUT FOR second private subnet (US-EAST-1B)



While refreshing the pages we will be able to see different outputs and health checks can be maintained by configuring the health checks in the application load balancer.

NEW TERMINOLOGY:- “BASTION-HOST”

A **Bastion Host** is a special-purpose server used to securely access servers in a private network from an external network, usually the internet. It acts as a “jump box” and is heavily secured since it’s exposed to potential attacks. Here are the common **applications and use cases** of a bastion host:

1. Secure Remote Access

- Allows system administrators to access internal servers (like database servers or application servers) in private subnets.
 - Example: SSH or RDP access to EC2 instances in AWS private VPCs.
-

2. Jump Server / Jump Box

- Serves as the only entry point to a network segment.
 - All traffic to sensitive servers goes through the bastion host for monitoring and control.
-

3. Network Segmentation Enforcement

- Helps enforce security policies by isolating public-facing and internal servers.
 - Reduces the attack surface since only the bastion host is exposed to the internet.
-

4. Auditing and Logging

- Tracks all connections and actions for compliance purposes.
 - Example: Logging all SSH sessions from admins for audit purposes.
-

5. VPN Alternative

- Sometimes used as a lightweight alternative to full VPN for secure remote access.
-

6. Temporary Administrative Access

- Can provide temporary access to developers or support staff without opening direct access to internal servers.
-

7. Multi-Cloud or Hybrid Cloud Access

- Acts as a secure entry point for managing servers across multiple cloud providers or on-premises environments.
-

8. Security Layer for Automation

- Scripts or automation tools can route through the bastion host for tasks like updates, backups, or monitoring internal servers securely.
-

Example in AWS:

- An **EC2 instance** in a public subnet acts as a bastion host.
- Admins SSH into this instance, then use it to SSH into EC2 instances in private subnets.
- Combined with security groups, multi-factor authentication, and CloudWatch logging for secure operations.