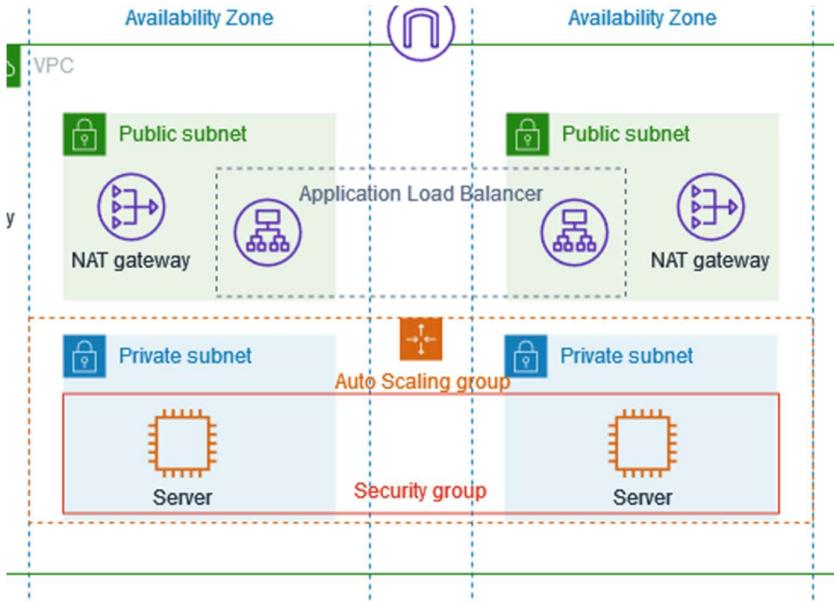


AWS PROJECT- VPC with Public-Private Subnet.



MAIN GOAL –

- The VPC creates a private network for your instances. Even though the ALB is public (in a public subnet), your instances are in a private subnet. This means no one can directly access your instances from the internet—they can only reach them through the ALB.
- Example: Think of the VPC as a gated community. The ALB is the front gate that's open to visitors, but your house (the instances) is inside, and no one can just walk in without going through the gate.
- The Application or Website is Stored inside the Private Subnet , Only the Load Balancer can Access the Private Subnet, AS The Load-Balancer present at PUBLIC SUBNET of SAME VPC. So when an Outside user trying to access to the Website with Load-Balancer URL, The Load-Balancer Actually send request to Instance that is in Private Subnet.
- So Here The user is Not Accessing the website Directly, first he is Accessing the LoadBalancer and then Instance.

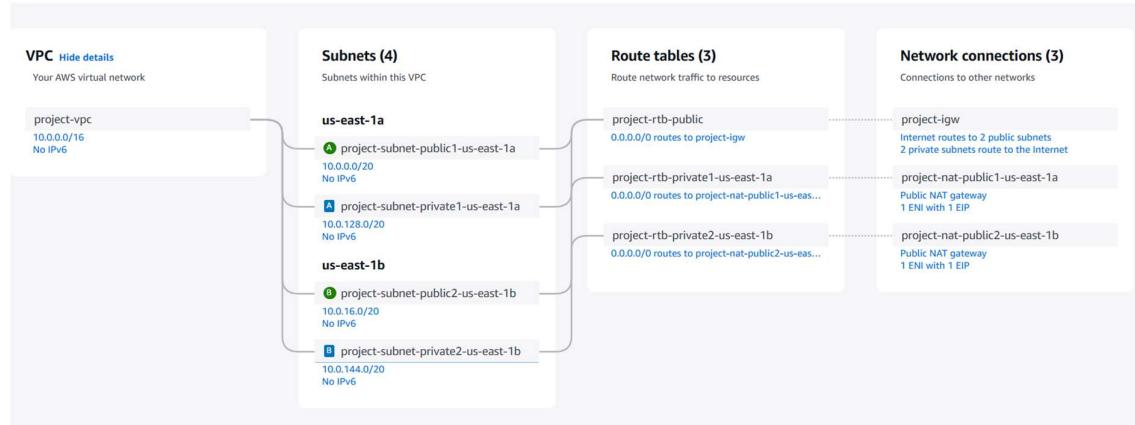
WORKING STEPS-

STEP1-

VPC and Subnet Setup

- **VPC Name:** project-vpc
- **CIDR Block:** 10.0.0.0/16
- Created **4 subnets** (2 public and 2 private) across **2 Availability Zones** (us-east-1a & us-east-1b).
 - **Public Subnets:**
 - project-subnet-public1-us-east-1a
 - project-subnet-public2-us-east-1b
 - **Private Subnets:**
 - project-subnet-private1-us-east-1a
 - project-subnet-private2-us-east-1b
- **Internet Gateway (IGW)** attached for public access.
- **NAT Gateways** created in each public subnet to allow outbound internet access from private subnets.
- **Route Tables** properly configured:
 - Public route table routes to the IGW.
 - Private route tables route to respective NAT gateways.

A)



B)

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. The left panel, titled 'VPC settings', contains fields for 'Resources to create' (set to 'VPC and more'), 'Name tag auto-generation' (set to 'Auto-generate' with 'project' as the name), 'IPv4 CIDR block' (set to '10.0.0.0/16'), and 'IPv6 CIDR block' (set to 'No IPv6 CIDR block'). The right panel, titled 'Preview', shows the created VPC named 'project-vpc' with CIDR '10.0.0.0/16' and 65,536 IPs. It also displays four subnets under 'us-east-1a' and one subnet under 'us-east-1b', all named 'project-sub' with specific IP ranges.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation [Info](#)
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
project

IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Preview

VPC [Hide details](#)
Your AWS virtual network

project-vpc
10.0.0.0/16
No IPv6

Subnets (4)
Subnets within the VPC

us-east-1a

- [A project-sub](#)
10.0.0.0/20
No IPv6
- [B project-sub](#)
10.0.128.0/20
No IPv6

us-east-1b

- [B project-sub](#)
10.0.16.0/20
No IPv6

[project-sub](#)

C)

aws [Alt+S]

VPC > Your VPCs > Create VPC

Tenancy Info
Default

Number of Availability Zones (AZs) Info
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

DNS options Info
 Enable DNS hostnames
 Enable DNS resolution

▶ Additional tags

Create VPC

Create VPC

Create VPC workflow

Success

Details

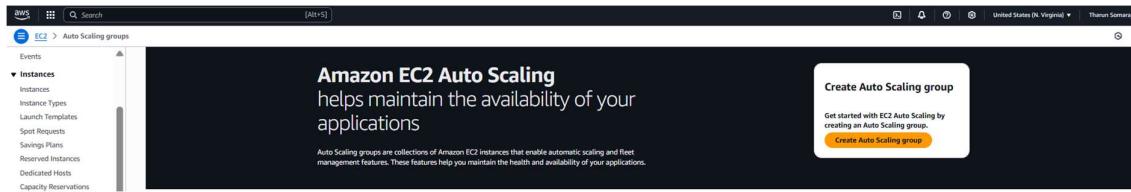
- ✓ Create VPC: vpc-0a7dc65ca48ec058 []
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-0a7dc65ca48ec058 []
- ✓ Create subnet: subnet-001a3eaaef3bb1c1c4 []
- ✓ Create subnet: subnet-04c9462925e161eaee []
- ✓ Create subnet: subnet-0bbf042fa35f9faad3 []
- ✓ Create subnet: subnet-07651453e13724b7 []
- ✓ Create internet gateway: igw-0ff13cb4b2218e802 []
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: rtb-057ca394cc10711fa []
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Allocate elastic IP: eipalloc-0b65fd212fd6d47:96 []
- ✓ Allocate elastic IP: eipalloc-0211aa20ab1d85a5 []
- ✓ Create NAT gateway: nat-0019e71ab261ef1fe []
- ✓ Create NAT gateway: nat-059a91db1b24df55d []
- ✓ Wait for NAT Gateways to activate
- ✓ Create route table: rtb-01fe1d5d1ca7b8886 []
- ✓ Create route
- ✓ Associate route table
- ✓ Create route table: rtb-0f360fd9b510037b4 []
- ✓ Create route
- ✓ Associate route table
- ✓ Verifying route table creation

View VPC

STEP 2-

A)

Launch Ec2 Ubuntu Instance in each Private Subnets, by Auto Scaling



B)

Create Launch Template for Ec2 Specifications (which we use in the Future for any Creations)

Step 1
 Choose launch template
 Step 2
 Choose instance launch options
 Step 3 - optional
 Integrate with other services
 Step 4 - optional
 Configure group size and scaling
 Step 5 - optional
 Add notifications
 Step 6 - optional
 Add tags
 Step 7 - Review

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

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

Select a launch template

Create a launch template

[Cancel](#) [Next](#)

Create the Launch Template as it automatically loads the page in other Tab

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. Templates can have multiple versions.

Launch template name and description

Launch template name - required

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

Template version description

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 is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

[AMI from catalog](#) [Recents](#) [Quick Start](#)

Name
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

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

Verified provider **Free tier eligible**

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Image ID
ami-084568db4383264d4

Username ubuntu

Catalog	Published	Architecture	Virtualization	Root device type	ENAv Enabled
Quick Start AMIs	2025-03-05T09:18:37.000Z	x86_64	hvm	ebs	Yes

Boot mode
uefi-preferred

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

Instance type

t2.micro	Family: t2	1 vCPU	1 GiB Memory	Current generation: true	On-Demand Windows base pricing: 0.0162 USD per Hour	Free tier eligible
					On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour	<input type="radio"/> All generations
					On-Demand SUSE base pricing: 0.0116 USD per Hour	Compare instance types
					On-Demand RHEL base pricing: 0.025 USD per Hour	
					On-Demand Linux base pricing: 0.0116 USD per Hour	

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 key30April [Create new key pair](#)

▼ Network settings [Info](#)

Subnet [Info](#)

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

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](#)

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

Security group name - required MySGJune2

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 [Info](#) allows ssh and http

VPC [Info](#)

vpc-0a7dc65ca48ec058 (project-vpc) [Remove](#)

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

Type	Protocol	Port range
ssh	TCP	22
Source type	Source	Description - optional
Anywhere	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	Protocol	Port range
Custom TCP	TCP	8000
Source type	Source	Description - optional
Anywhere	Add CIDR, prefix list or security group	e.g. SSH for admin desktop

[Add security group rule](#)

[Advanced network configuration](#)

Create launch template

C)

Load the Created Launch Template in the Auto Scaling Section

D)

Create AutoScaling in the Private Subnet of Created VPC- private1A and private1B

E) Configure group size and Auto Scaling of the Ec2 Instances

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Integrate with other services

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

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

Create Auto Scaling group

G) Create another Ec2 Instance - Bastion-Host(Jump Server) in the Public Subnet of Our VPC, as from our own PC Cmd-line we need to SSH to Bastion

aws | Search [Alt+S]

EC2 > Instances > Launch 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 is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

Recent | Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Debian

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
 ami-084568db4383264d4 (64-bit (x86)) / ami-0c4e709339fa8521a (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	AMI ID	Publish Date	Username
64-bit (x86)	ami-084568db4383264d4	2025-03-05	ubuntu

Verified provider

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

Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour	On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour	On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 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 - required

[Create new key pair](#)

▼ Network settings [Info](#)

VPC - required | [Info](#)

10.0.0.0/16

Subnet | [Info](#)

subnet-001a3eaad33b4c1c4	project-subnet-public1-us-east-1a
VPC: vpc-0b7dcb65ca48ec058	Owner: 948098480976 Availability Zone: us-east-1a
Zone type: Availability Zone	IP addresses available: 4090 CIDR: 10.0.0.0/24

[Create new subnet](#)

Auto-assign public IP | [Info](#)

Enable

Additional charges apply when outside of free tier allowance

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.

Create security group

Select existing security group

Security group name - required

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 [Info](#)

Inbound Security Group Rules

▼ Security group rule 1 (TCP: 22, 0.0.0.0/0)

Type	Info	Protocol	Info	Port range	Info
ssh	TCP	22			
Source type	Info	Source	Info	Description	Info
Anywhere		Add CIDR, prefix list or security group		e.g. SSH for admin desktop	
0.0.0.0/0					

Rules from 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.

[Add security group rule](#)

[Advanced network configuration](#)

▼ Configure storage [Info](#)

Advanced

1x	8 GB	gp3	Root volume, 3000 IOPS, Not encrypted
----	------	-----	---------------------------------------

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click here to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0x File systems

[Edit](#)

Summary

Number of instances [Info](#)

1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64 [read more](#)
ami-084568b4383264d4

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 of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or equivalent Amazon Lambda functions) used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

LAUNCH THE Instance.

STEP 3-

A)

We Can see 3 instance – One is Bastion and two are Private Subnet Instances.

Instances (3) info															
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security gr...			
i-0a0d70f61dece2b7	i-0a0d70f61dece2b7	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	-	-	-	-	disabled	MySGuinc...			
Bastion-host	i-0264390a772d461c3	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-44-202-73-88.com.	44.202.73.88	-	-	disabled	Launch-wiz...			
	i-07fb4ff08ed5206d7	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	-	-	-	-	disabled	MySGuinc...			

B)

Go to Bastion-host instance and note the Public IP of it.

EC2 > Instances > i-0264390a772d461c3

Instance summary for i-0264390a772d461c3 (Bastion-host) [Info](#)

Updated less than a minute ago

Instance ID: i-0264390a772d461c3

IPv6 address: -

Hostname type: IP name: ip-10-0-1-186.ec2.internal

Public IPv4 address: 44.202.73.88 | [Open address](#)

Instance state: Running

Private IP DNS name (IPv4 only): ip-10-0-1-186.ec2.internal

C)

FOR copying key **key30April.pem** from the Local Pc to Bastion (as we want to login to private subnet from bastion) in the Command Prompt of the PC

```
scp -i "C:\Users\Somaraju Tharun\Downloads\key30April.pem" "C:\Users\Somaraju Tharun\Downloads\key30April.pem" ubuntu@44.202.73.88:/home/ubuntu/
```

```
C:\Users\Somaraju Tharun>scp -i "C:\Users\Somaraju Tharun\Downloads\key30April.pem" "C:\Users\Somaraju Tharun\Downloads\key30April.pem" ubuntu@44.202.73.88:/home/ubuntu/
The authenticity of host '44.202.73.88 (44.202.73.88)' can't be established.
ED25519 key fingerprint is SHA256:u0wpy+nyW0DzTBFeJ4rFQ2sbm+z3n1H4XAfFMx+P0Q8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '44.202.73.88' (ED25519) to the list of known hosts.
key30April.pem                                         100% 1678      5.3KB/s   00:00

C:\Users\Somaraju Tharun>
```

SUCCESSFULLY COPIED THE Key-Pair to Bastion from the Local PC

D)

To verify if the key got copied or not

```
-ssh -i "C:\Users\Somaraju Tharun\Downloads\key30April.pem" ubuntu@44.202.73.88
```

```
-$ ls
```

You will see the key

```
C:\Users\Somaraju Tharun>ssh -i "C:\Users\Somaraju Tharun\Downloads\key30April.pem" ubuntu@44.202.73.88
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 Mon Jun  2 12:28:17 UTC 2025

System load: 0.0          Processes:           106
Usage of /: 25.3% of 6.71GB   Users logged in: 0
Memory usage: 20%           IPv4 address for enX0: 10.0.1.186
Swap usage:  0%

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

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-186:~$ ls
key30April.pem
ubuntu@ip-10-0-1-186:~$ |
```

E)

chmod 400 is used to secure the .pem file by allowing only the owner to read it, which SSH requires for safe access.

\$chmod 400 key30April.pem

F)

To SSH from the **Bastion Host** into the **private EC2 instance (10.0.151.236)**, use the following command **on the Bastion terminal**:

\$ssh -i key30April.pem ubuntu@10.0.136.212

G)

You are Finally in Ec2 instace Terminal –

ubuntu@ip-10-0-136-212:~\$

```

ubuntu@ip-10-0-1-186:~$ chmod 400 key30April.pem
ubuntu@ip-10-0-1-186:~$ ssh -i key30April.pem ubuntu@10.0.136.212
The authenticity of host '10.0.136.212 (10.0.136.212)' can't be established.
ED25519 key fingerprint is SHA256:824jio+/ctfCC6K4+SJU/FpkZ0FYlVUPSp7hGied9No.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.136.212' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1824-aws x86_64)

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

System information as of Mon Jun  2 12:33:16 UTC 2025

System load:  0.0          Processes:      103
Usage of /:   24.9% of 6.71GB  Users logged in:    0
Memory usage: 20%           IPv4 address for enx0: 10.0.136.212
Swap usage:   0%

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-136-212:~$ |

```

H)

CREATE A HTML PAGE IN THE EC2 TERMINAL

1. Create the file

Run this on your EC2 instance:
 bash
 CopyEdit

\$nano index.html

Paste the following code:
 html
 CopyEdit

```

<!DOCTYPE html>
<html>
<head>
  <title>My AWS Project</title>
</head>
<body>

```

```
<h1>Welcome to My AWS Created on June 2nd -2025</h1>
<p>This page is hosted on a private EC2 instance via a
Bastion Host.</p>
</body>
</html>
```

3. Save and exit:

- Press **Ctrl + O** → **Enter** to save
 - Then **Ctrl + X** to exit Nano
- Your HTML page is ready as index.html.

Create a Python server

```
$ python3 -m http.server 8000
```

 **What happens when user visits Load Balancer?**

1. The request goes to the Load Balancer.
2. Load Balancer sends the request to your private EC2.
3. The Python server running on that EC2 responds with your HTML page.
4. User sees your website!

IMPORTANT POINT TO REMEMBER ABOUT-

How does an EC2 instance recognize the key-pair?

When we

SSH FROM Local PC → Bastion-Host(Public Subnet)

Bastion-Host(Public Subnet) --→ Ec2 (Private Subnet)

Here we are copying KeyPair.pem from local pc to bastion , as we want this KeyPair.pem to be verified with private ec2 when we are doing SSH from bastion to Private ec2.

BELOW IS THE MAIN AND IMPORTANT DESCRIPTION about the ABOVE:-

When you **launch an EC2 instance**, you **attach a key-pair** (like key30April.pem) to it. Here's what happens:

 **Behind the Scenes:**

1. When the EC2 is created with a key-pair:
 - o AWS puts the **public part** of the key into a file called:

/home/ubuntu/.ssh/authorized_keys

2. When you try to connect using:

ssh -i key30April.pem ubuntu@<ec2-ip>

3.  If it matches:
 - o Your **private key** (key30April.pem) is used by your SSH client.
 - o The EC2 instance checks:
 - “Does this private key match the public key I have in authorized_keys?”

3.  If it matches:

- o You are allowed to login.

4.  If it doesn't match:

- o Access is denied.

 **Simple Analogy:**

Think of it like a **lock and key**:

- EC2 has the **lock** (public key).
 - You have the **key** (private key .pem).
 - Only the **right key** can unlock it.
-

📌 So to answer your question:

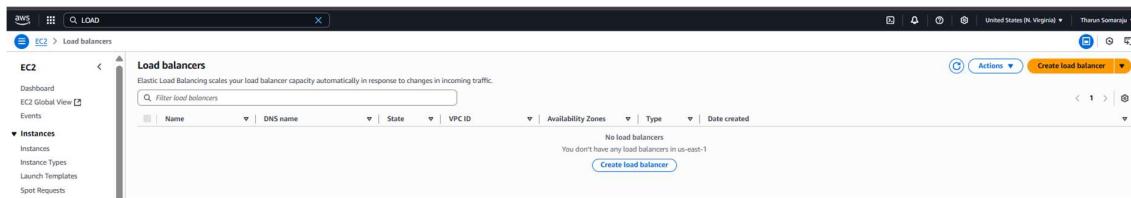
How does EC2 recognize the key-pair?

Because **when the instance is launched, AWS automatically saves the public part of the key-pair** into the EC2's login system. When you try to connect using the private key, it matches it with the stored public key.

STEP 4-

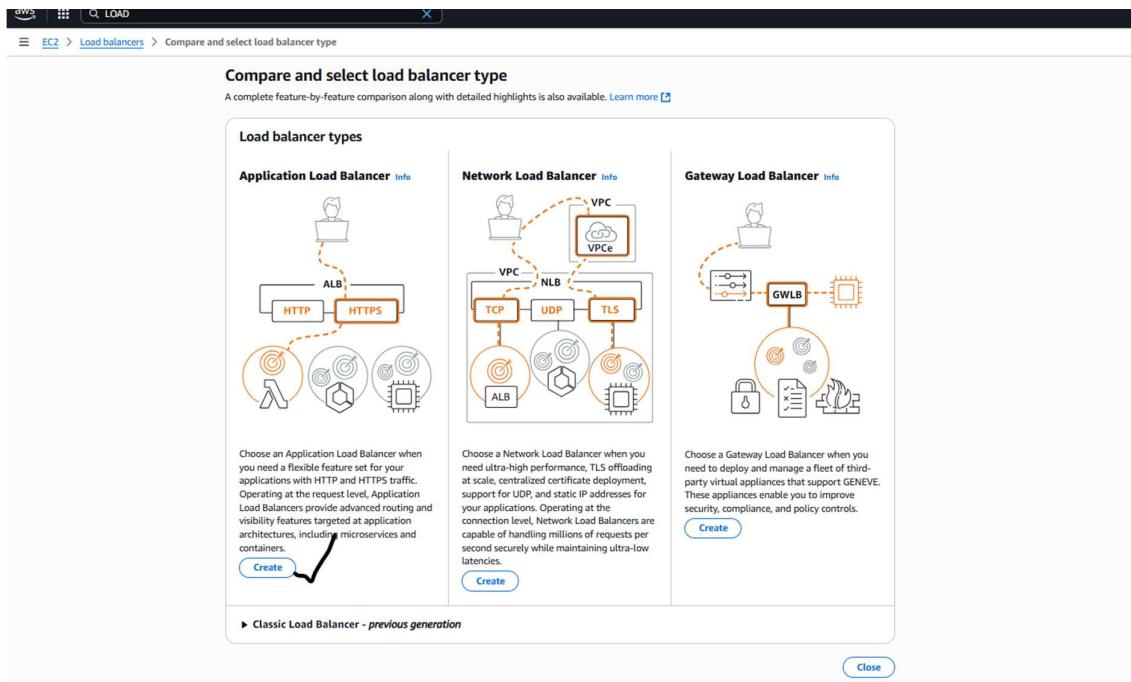
A)

Create LOAD BALANCER



B)

Create APPLICATION LOAD BALANCER



C)

D)

Make sure that Load Balancer in the Public Subnets of OUR VPC

E)

Add the Security Group of the Ec2 instances in Load-Balancer Aswell

F)

Create a Target-group I.e (Our Private Ec2 Instances which are in the Private Subnet)

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP-80

Protocol: **HTTP** Port: **80** 1-65535

Default action [Info](#) Forward to: Select a target group [Create target group](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#) You can add up to 50 more tags.

[Add listener](#)

G)

Here the Port as 8000 as we set the Instances Security Group Inbound as Port 8000

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration
Settings in this section can't be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC
- Facilitates the use of Amazon EC2 Auto Scaling [To manage and scale your EC2 capacity](#)

IP addresses

- Supports load balancing to multiple IP addresses and network interfaces on the same instance
- Offers flexibility with microservice based architectures, simplifying inter-application communication
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT

Lambda function

- Facilitates routing to a single Lambda function
- Accessible to Application Load Balancers only

Application Load Balancer

- Offers the Resiliency for a Network Load Balancer to accept and route TCP requests within a specific VPC
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer

Target group name
Tls-June2nd-Ec2

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

Protocol: **HTTP** Port: **8000** 1-65535

H)

Put the Targets as our Private Instances and Press include as Pending Below button

Register targets

This is an optional step to create a target group. However, to ensure that 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	Launch time
i-0264390a772d461c3	Bastion-host	Running	launch-wizard-6	us-east-1a	10.0.1.186	subnet-001a5eaa33b4c1c4	June 2, 2025, 17:48 (UTC)
i-0a0e708c1decce2b7	Running	MySGJune2	us-east-1b	10.0.159.46	subnet-07651453ae15724b7	June 2, 2025, 17:33 (UTC)	
i-07fb4ff08bed5206d7	Running	MySGJune2	us-east-1a	10.0.136.212	subnet-08bf0427a359faad5	June 2, 2025, 17:33 (UTC)	

2 selected

Ports for the selected instances
Port numbers defining traffic to the selected instances:
8000
1-65535 (separate multiple ports with commas)

Include as pending below

2 selections are now pending below. Include more or register targets when ready.

Review targets

Targets (2)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-0a0e708c1deceeb7		8000	Running	MySGJune2	us-east-1b	10.0.159.46	subnet-07651453ac13724b7	June 2, 2025, 17:33 (UTC+05:30)
i-07fbff0f8ed5206d7		8000	Running	MySGJune2	us-east-1a	10.0.156.212	subnet-0bbf042fa35f9aa5d	June 2, 2025, 17:35 (UTC+05:30)

2 pending

Cancel Previous Create Target group

CREATE TARGET GROUP

I)

Now add the Created Target Group to the Load-Balancer

Security groups

Select up to 5 security groups

MySGJune2

Listeners and routing

Protocol: HTTP Port: 80 Listener tags - optional

Default action: Forward to TG-June2nd-E2 Target type: Instance, IPv4

Create target group

CREATE THE LOAD-BALANCER

J)

After Successful creation of LoadBalancer

We See this Error in the LISTERNERS AND RULES Section

Details

Load balancer type: Application Status: Provisioning VPC: vpc-07dc65ca48ec058 Load balancer IP address type: IPv4

Scheme: Internet-facing Hosted zone: Z555X0OTRQ7X7PK Date created: June 2, 2025, 18:34 (UTC+05:30)

Listeners and rules (1)

Protocol: HTTP:80 Default action: Forward to target group TG-June2nd-E2 (100%) Target group stickiness: Off

Here This is Showing the Error Because

The Load Balancer is Listening on Port 80, But its Own Security Group(same as Ec2 instance's SG) does not allowing the Inbound traffic on Port , So that is why we NOW ENABLE THE PORT 80 IN Security Group

Go to Security Section and Select the Security Group

The screenshot shows the AWS Load Balancer configuration page for 'LoadBalancerJune2nd'. The 'Security' tab is highlighted with a blue underline. The 'Listeners and rules' tab is also visible. The 'Security groups' section shows one group named 'MySGJune2' which allows ssh and http.

Now Edit Inbound Rules

The screenshot shows the AWS Security Groups configuration page for 'sg-0bcc24ce2225008b7 - MySGJune2'. The 'Inbound rules' tab is selected. A new rule is being added, with 'Type' set to 'HTTP', 'Protocol' to 'TCP', 'Port range' to '80', and 'Source' to 'Anywhere'. A note at the bottom states: 'Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The 'Edit inbound rules' button is circled.

SAVE RULES...

NOW Go to the Load Balancer and check The Error is Removed

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar includes sections for Instances, Images, Elastic Block Store, and Network & Security. The main content area displays the details for a load balancer named 'LoadBalancerJune2nd'. Under the 'Listeners and rules' tab, there is one rule defined:

- Protocol:Port:** HTTP:80
- Default action:** Forward to target group: TG-June2nd-Ec2 (100%)
- Target group stickiness:** Off
- Rules:** 1 rule
- ARN:** arn:aws:elasticloadbalancing:us-east-1:948099480976:loadbalancer/app/LoadBalancerJune2nd/2797c0fc456424c5
- Security policy:** Not applicable

Points to Remember –

The Load Balancer Listener Listens on a Specific Port Like (PORT 80 For HTTP)

But if the LB's SG does not allowing that port in its Inbound rules , USER'S Can't reach the LB. So the Listener Port and SG Inbound Port must be same(Both set to 80) to allow traffic from users to reach your website.

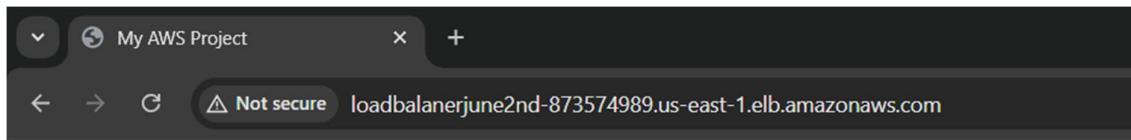
Therefore , The Listener Port and Load Balancer Security Group Inbound Port must match (Port 80) to allow traffic from users to reach your website.

STEP 5

In the Load-Balancer , Copy the DNS URL and Test it in the Chrome browser ,

Now the Traffic only goes to Only one Private Instance, as we Uploaded the HTML in Only one instance.

The screenshot shows the AWS CloudFormation console with the path: AWS > EC2 > Load balancers > LoadBalancerJune2nd. The main view displays the 'LoadBalancerJune2nd' resource. In the 'Listeners and rules' section, there is one listener rule for port 80, which forwards traffic to the target group 'TG_June2nd_ELB'. The 'DNS name copied' link is also highlighted with a red box.



Welcome to My AWS Created on June 2nd -2025

This page is hosted on a private EC2 instance via a Bastion Host.

Finally we can see the Website ,

Here The user is Not Directly accessing the Private Ec2 Instances , instead Via Load Balancer User can Access the Website that is created in the Private Ec2 ..

STEPS TO DISMANTLE THE RESOURCES

Here's a simple checklist to **safely dismantle each AWS tool** from your project and avoid charges:

1. Terminate EC2 Instances

- Go to **EC2 > Instances**
 - Select all running instances (including Bastion and private ones)
 - Click "**Instance State**" > **Terminate**
-

2. Delete Auto Scaling Groups

- Go to **EC2 > Auto Scaling Groups**
 - Delete your Auto Scaling Group (ASG)
-

3. Delete Load Balancer

- Go to **EC2 > Load Balancers**
 - Select and **delete** your Application Load Balancer (ALB)
-

4. Delete Target Groups

- Go to **EC2 > Target Groups**
 - Select and **delete** them (if not auto-removed)
-

5. Delete NAT Gateway

- Go to **VPC > NAT Gateways**
- Select and **delete**

 NAT Gateway can **incur charges even when idle** — make sure to remove it!

6. Release Elastic IP (if any)

- Go to **VPC > Elastic IPs**
 - Disassociate and then **release** unused IPs
-

7. Delete Subnets and Route Tables

- Go to **VPC > Subnets and Route Tables**
 - Delete custom ones **after** EC2s and NAT are gone
-

 **8. Delete VPC**

- Go to **VPC > Your VPCs**
- Select and **delete your custom VPC**

.....**THE END.....**