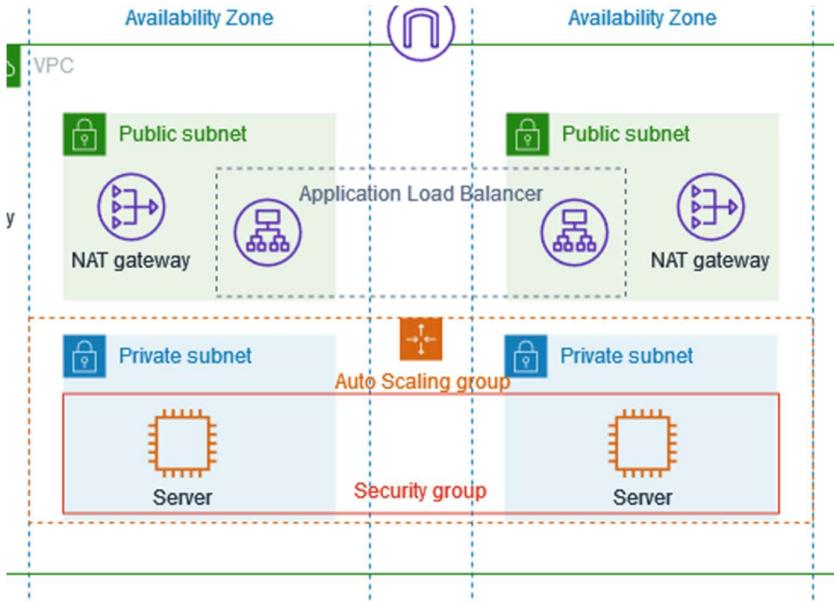


AWS PROJECT- VPC with Public-Private Subnet.



Main Goal:-

The VPC creates a private network for your instances. Although the Application Load Balancer (ALB) is deployed in a public subnet, the instances are placed in private subnets, making them inaccessible directly from the internet. This setup ensures that only the ALB can communicate with the instances.

For example, think of the VPC as a gated community—while the ALB acts as the front gate accessible to the public, the instances (your application or website) reside securely inside. When an external user accesses the application through the ALB's public DNS, the ALB forwards the request to the instances in the private subnet and returns the response.

Thus, users do not directly access the instances; they interact with the ALB, which securely handles communication with the backend instances within the same VPC.

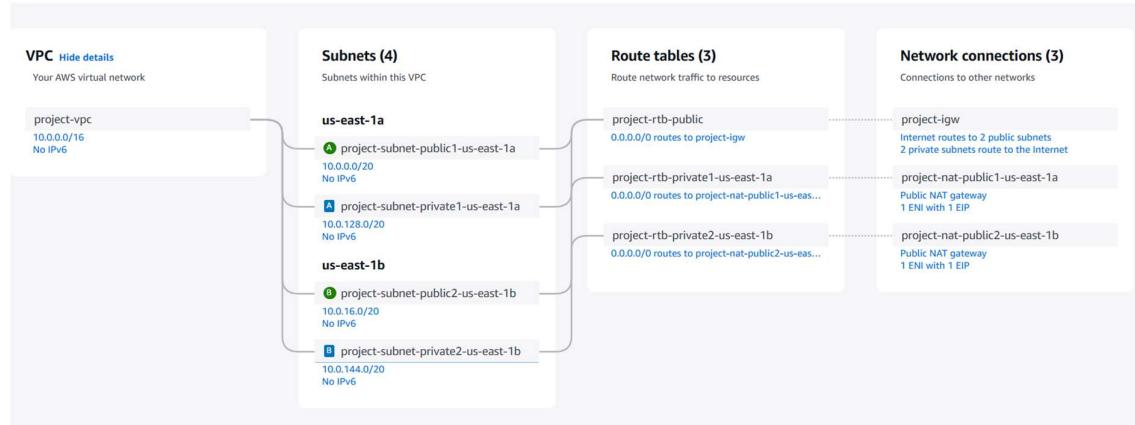
WORKING STEPS-

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)

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

Give the Specifications for the Ec2 Instances ..

Image ID
ami-084568db4383264d4

Username [ubuntu](#)

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

Boot mode uefi-preferred

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

Instance type [t2.micro](#) [Info](#) [Get advice](#)

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: 0.025 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible [All generations](#) [Compare instance types](#)

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](#) [Create new subnet](#)

Don't include in launch template

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

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](#)

Here Create a Security Group that allows only from PORT 22, PORT 8000(i.e Our Load Balancer only Can Access these Private Ec2 Instances)

[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-0a7dcb65ca48ec058 \(project-vpc\)](#)

Inbound Security Group Rules

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

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

Security group rule 2 (TCP, 8000, 0.0.0.0/0)

Type	Protocol	Port range	Action
Custom TCP	TCP	8000	Remove
Source type	Source	Description - optional	
Anywhere	Add CIDR, prefix list or security group	e.g. SSH for admin desktop	
0.0.0.0/0 X			

[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
Step 2
Step 3 - optional
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Step 6 - optional
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 1	Max desired capacity 4
----------------------------------	----------------------------------

Equal or less than desired capacity

Create Auto Scaling group

F)

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

Recentos 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 64-bit (x86)	AMI ID ami-084568db4383264d4	Publish Date 2025-03-05	Username ubuntu	Verified provider
------------------------------	---------------------------------	----------------------------	--------------------	-------------------

Be Careful, as now we are putting the Bastion-Host in **OUR VPC** that is Created , and in **the PUBLIC Subnet of that VPC**.

And PUT Auto-Assign public IP - **ENABLE**

▼ 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 IPv6 ...	Elastic IP	IPv6 IPs	Monitoring	Security gr...		
		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 3rd -2025 BY  
THARUN SOMARAJU</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>

- 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:

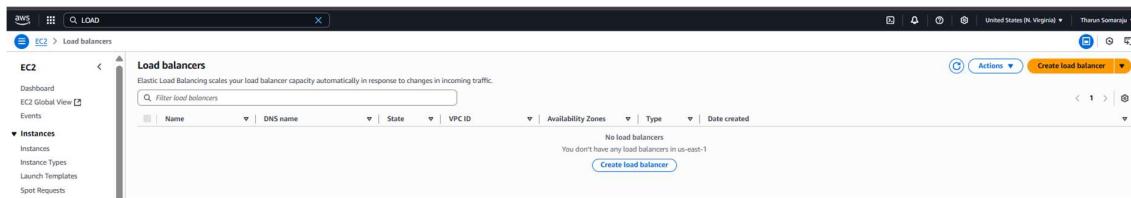
How does EC2 recognize the key-pair?

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-

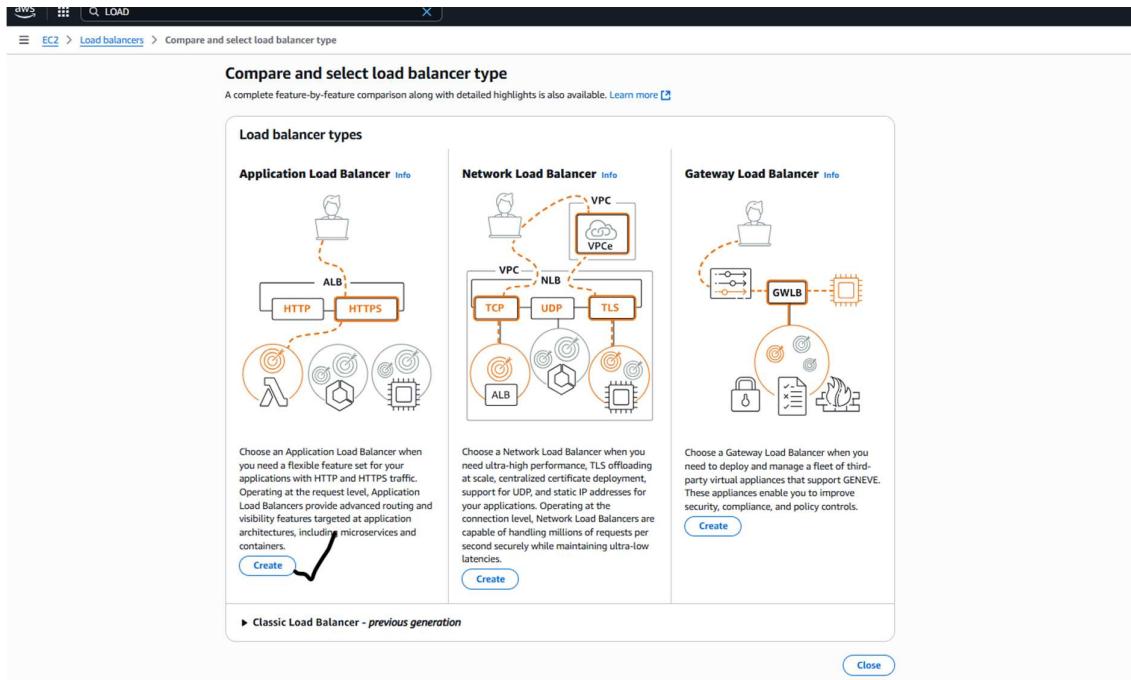
A)

Create LOAD BALANCER



B)

Create APPLICATION LOAD BALANCER



C)

Detailed description: This screenshot shows the AWS CloudFormation console. A new stack named 'HelloWorld' is being created. The 'HelloWorld' function is defined in a file named 'lambda_function.py'. It contains a single export named 'HelloWorld'.

```

version: '1'
functions:
  HelloWorld:
    type: AWS::Lambda::Function
    properties:
      code:
        S3Bucket: !Ref S3Bucket
        S3Key: lambda_function.zip
      handler: index.handler
      runtime: python3.7
      environment:
        variables:
          HELLO_NAME: World
      exports:
        - name: HelloWord
          value: !GetAtt HelloWorld.Arn

```

D)

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

Detailed description: This screenshot shows the AWS CloudFormation console. A new stack named 'HelloWorld' is being created. The 'HelloWorld' function is defined in a file named 'lambda_function.py'. It contains a single export named 'HelloWorld'.

```

version: '1'
functions:
  HelloWorld:
    type: AWS::Lambda::Function
    properties:
      code:
        S3Bucket: !Ref S3Bucket
        S3Key: lambda_function.zip
      handler: index.handler
      runtime: python3.7
      environment:
        variables:
          HELLO_NAME: World
      exports:
        - name: HelloWord
          value: !GetAtt HelloWorld.Arn

```

E)

Create A new Security Group for Load Balancer allowing only **PORT 80** because

Why only allow port 80 on LB's Security Group?

- **Port 80 (HTTP)** is used by users to access the website via the Load Balancer.
- The LB **never needs SSH (port 22) or app ports (like 8000)** from the public.

- Limiting to port 80 reduces your **exposure to unnecessary attacks**.

Security groups [Info](#)
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#)

Select up to 5 security groups

⚠ Application Load Balancers require at least one security group. If none are selected, the VPC's default security group will be applied.

Create security group [Info](#)
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)
 Name cannot be edited after creation.

Description [Info](#)

VPC Info

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional	Delete
HTTP	TCP	80	Anywhere	<input type="text" value="0.0.0.0/0"/>	

Add rule

Outbound rules [Info](#)

Type	Protocol	Port range	Destination	Description - optional	Delete
All traffic	All	All	Custom	<input type="text" value="0.0.0.0/0"/>	

Add rule

Create security group

F) Add the Security Group that is Created i.e PORT 80

Listeners and routing [Info](#)
A listener is a process that checks for connection requests using the port and protocol you configure. The roles you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Protocol **Port**
 :

Default action [Info](#)
Forward to

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) Specify the Target Groups(Our Private Subnet Ec2 Instances)

And We are Specifying the PORT 8000 For the Target Group Because , The Inbound Security Group for Instances are **PORT 8000 AND PORT 22..**

So Therefore we put the PORT 8000 here for LB to Contact Ec2

The screenshot shows the 'Specify group details' step of the target group creation wizard. It includes sections for 'Basic configuration', 'Choose a target type' (selected: Instances), 'Target group name' (TG-June2nd-Ec2), 'Protocol : Port' (HTTP, port 8000), and a summary of the target group.

H)

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

The screenshot shows the 'Register targets' step. It lists available instances (Bastion-host, i-0264390a772d461c3, i-0e0708c1decce2b7, i-07fb4ff0bed5206d7) and allows selecting ports for routing traffic. A red arrow points to the 'Include as pending below' button.

The screenshot shows the 'Review targets' step with two pending targets (i-0264390a772d461c3, i-0e0708c1decce2b7). A red arrow points to the 'Create target group' button.

PRESS CREATE TARGET GROUP

I)

Now add the Created Target Group to the Load-Balancer

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: **June3TG** Target type: Instance, IPv4 **HTTP** [Edit](#) [Remove](#)

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.

[Create target group](#)

PRESS CREATE THE LOAD-BALANCER

J)

After Successful creation of Load Balancer

Now Go Copy the URL of Load Balancer for testing..and check it in the Chrome or any browser ..

aws LOAD

EC2 > Load balancers > MYLoadBalJun3

EC2

- Dashboard
- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store

Successfully created load balancer: **MYLoadBalJun3**
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

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.

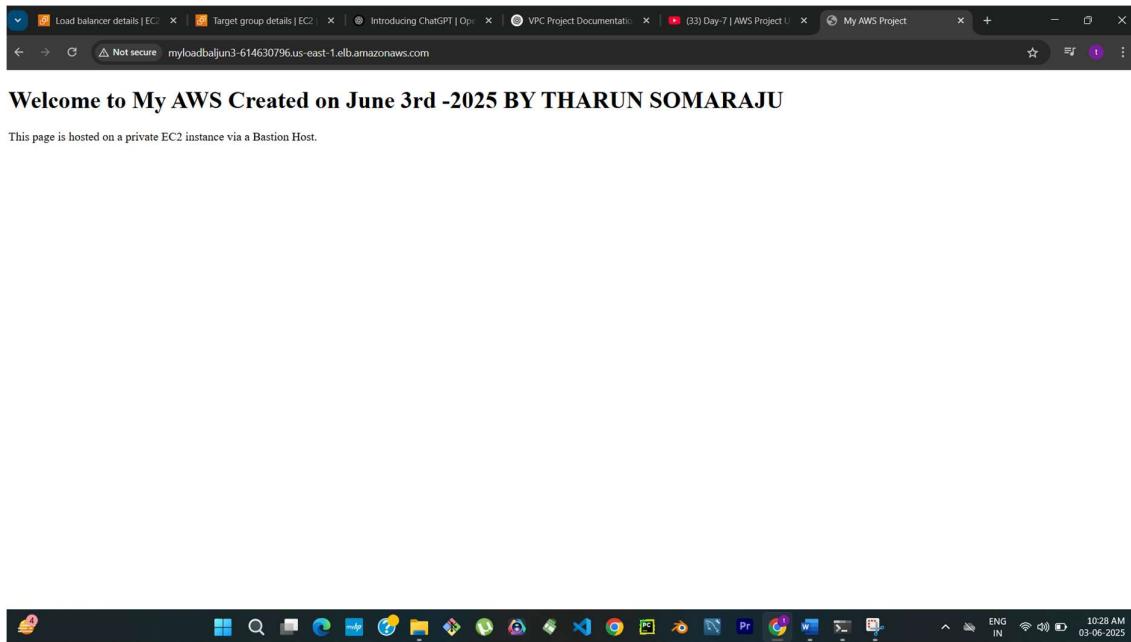
MYLoadBalJun3

Details

Load balancer type Application	Status Provisioning	VPC vpc-0054740456e89fb6b	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z355XDTRQ7X7K	Availability Zones subnet-0658f6853de4ad8 us-east-1a (use1-az2) subnet-091b74bfef42942d us-east-1b (use1-az2)	Date created June 3, 2025, 10:24 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:948099480976:loadbalancer/app/MYLoadBalJun3/c9db8e3601dc95ba	DNS name info MYLoadBalJun3-614630796.us-east-1.elb.amazonaws.com (A Record)		

[Edit IP pools](#) [Actions](#)

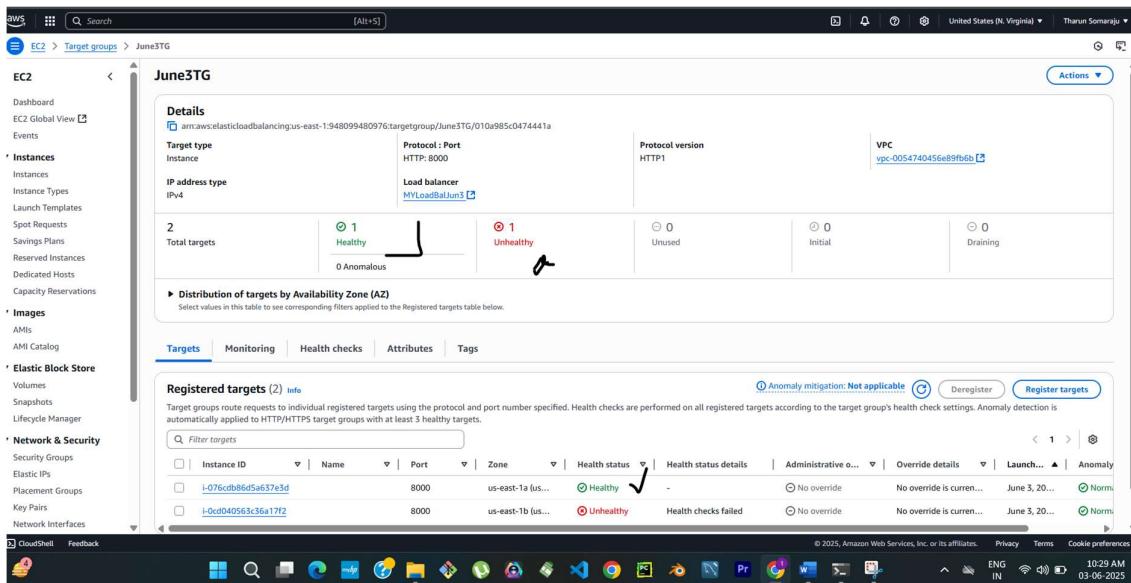
[Listeners and rules](#) [Network mapping](#) [Resource map](#) [Security](#) [Monitoring](#) [Integrations](#) [Attributes](#) [Capacity](#) [Tags](#)



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 ..

Therefore, Now Our Main Goal is Achieved



Therefore , One instance is Healthy and another one not healthy because , We Hosted the Website in only One Ec2 Instance, So the Request Goes for only One instance..

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.....

