# Scalable Web Hosting Architecture on AWS using EC2, ALB, and S3

**Prepared by :**
Mitrajsinh Solanki

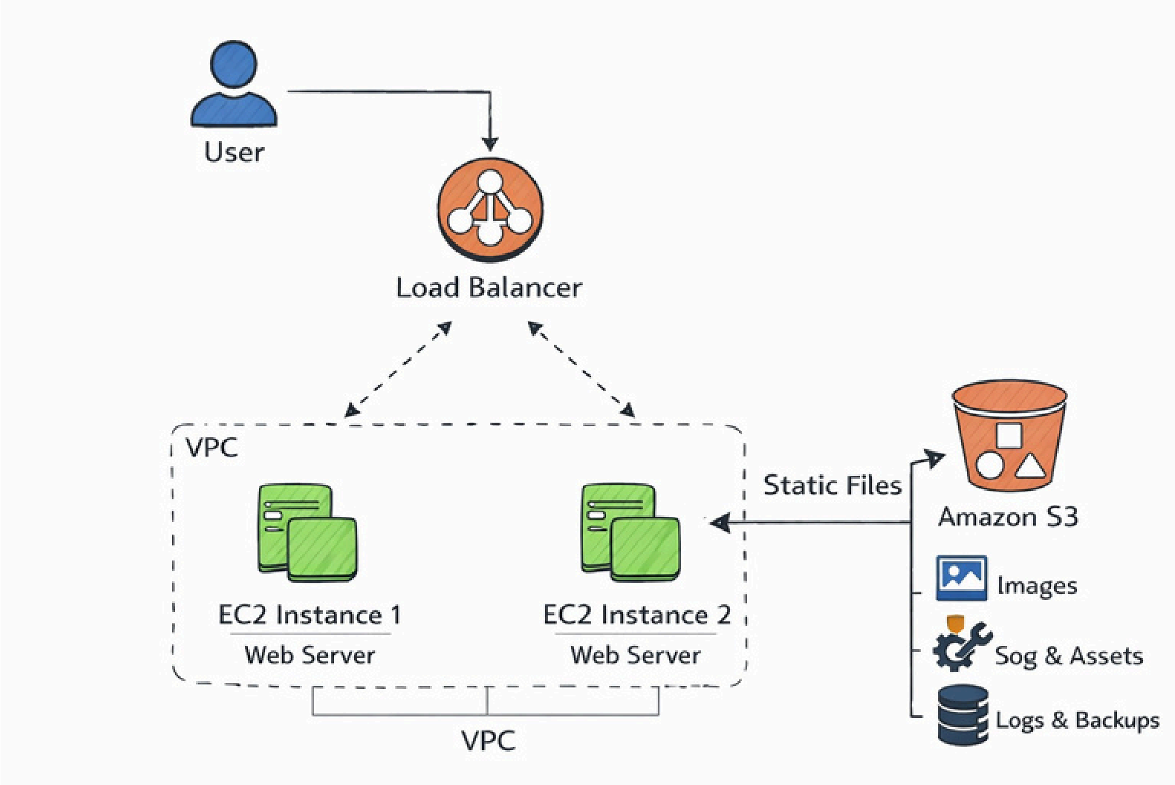
# Objective

The objective of this project is to deploy a highly available and scalable web application on AWS by using EC2 instances inside a Virtual Private Cloud (VPC), distributing traffic using an Application Load Balancer, and storing static files and logs in Amazon S3.

# AWS Services Used

VPC
EC2
Application load balancer
Security group
S3

# System architecture



- User sends request to the website
- Request goes to Application Load Balancer
- Load Balancer distributes traffic to multiple EC2 instances
- EC2 servers fetch static files from Amazon S3
- Logs and backups are stored in S3

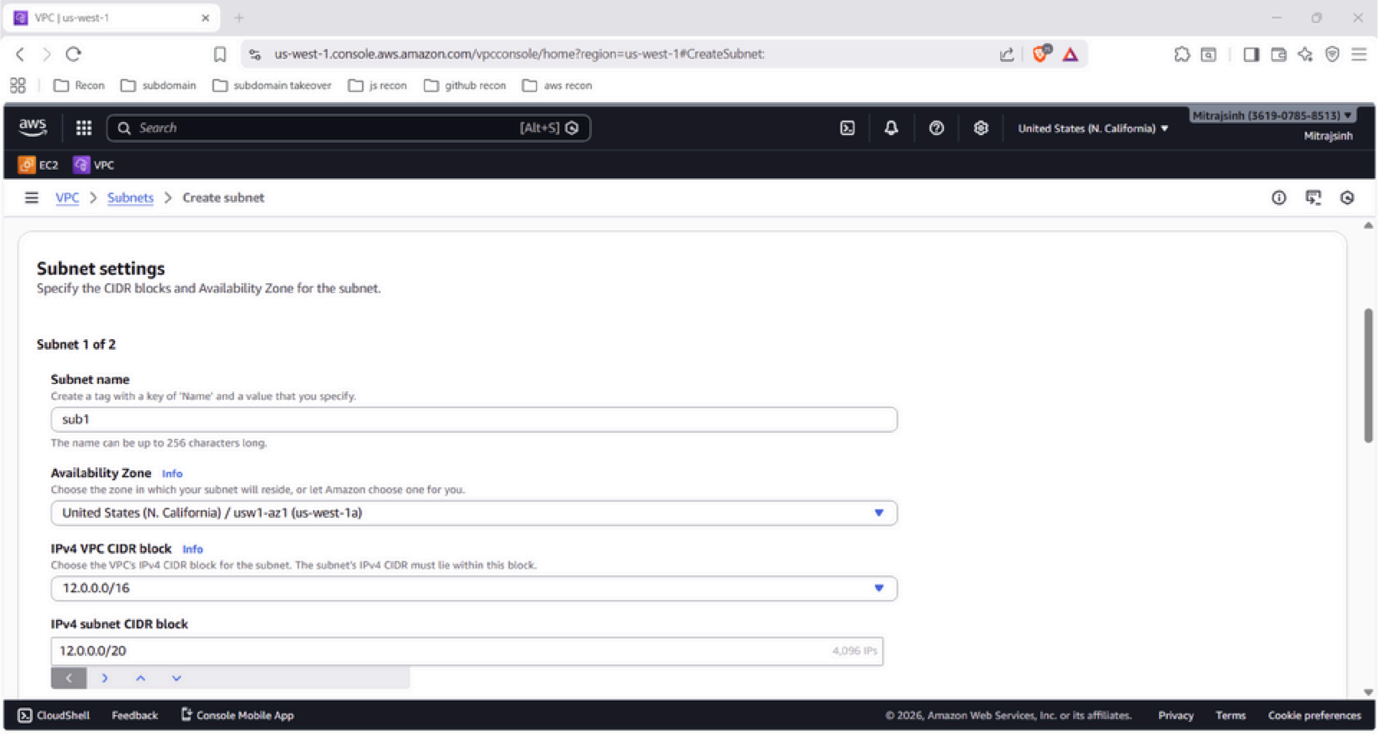
# Step-by-Step Implementation

## ◆ Step 1: Create a VPC

**A custom Virtual Private Cloud (VPC) was created to isolate the network environment for the project. The VPC was configured with a custom IPv4 CIDR block to define the private network range.**

Configurations details:

- VPC Name: myvpc
- IPv4 CIDR Block: 12.0.0.0/16
- Region: us-west-1


VPC | us-west-1
us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#CreateSubnet:
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
aws
Search
[Alt+S]
United States (N. California) ▾
Mitrajsinh (3619-0785-8513) ▾
Mitrajsinh
EC2
VPC
VPC > Subnets > Create subnet
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

**Subnet settings**
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 2**

**Subnet name**
Create a tag with a key of 'Name' and a value that you specify.
sub1
The name can be up to 256 characters long.

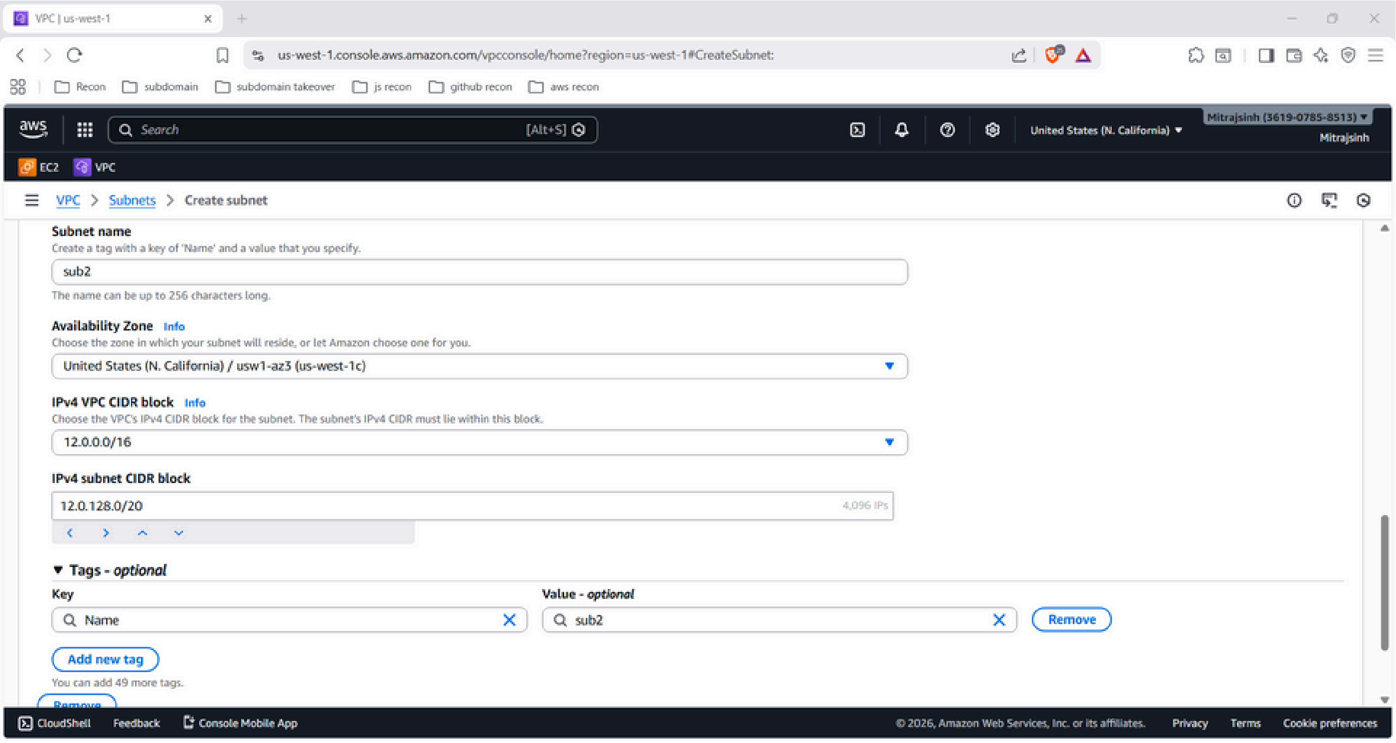
**Availability Zone** Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. California) / usw1-az1 (us-west-1a) ▾

**IPv4 VPC CIDR block** Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
12.0.0.0/16 ▾

**IPv4 subnet CIDR block**
12.0.0.0/20 4,096 IPs

## ◆ Step 2: Create a subnet

Two subnets were created in different Availability Zones to ensure high availability and fault tolerance. These subnets were configured with unique CIDR blocks inside the VPC.



VPC | us-west-1
us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#CreateSubnet:
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
Search
[Alt+S]
United States (N. California) ▾
Mitrajsinh (3619-0785-8513) ▾
Mitrajsinh
EC2
VPC
VPC > Subnets > Create subnet

**Subnet name**
Create a tag with a key of 'Name' and a value that you specify.
sub2
The name can be up to 256 characters long.

**Availability Zone** Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. California) / usw1-az3 (us-west-1c)

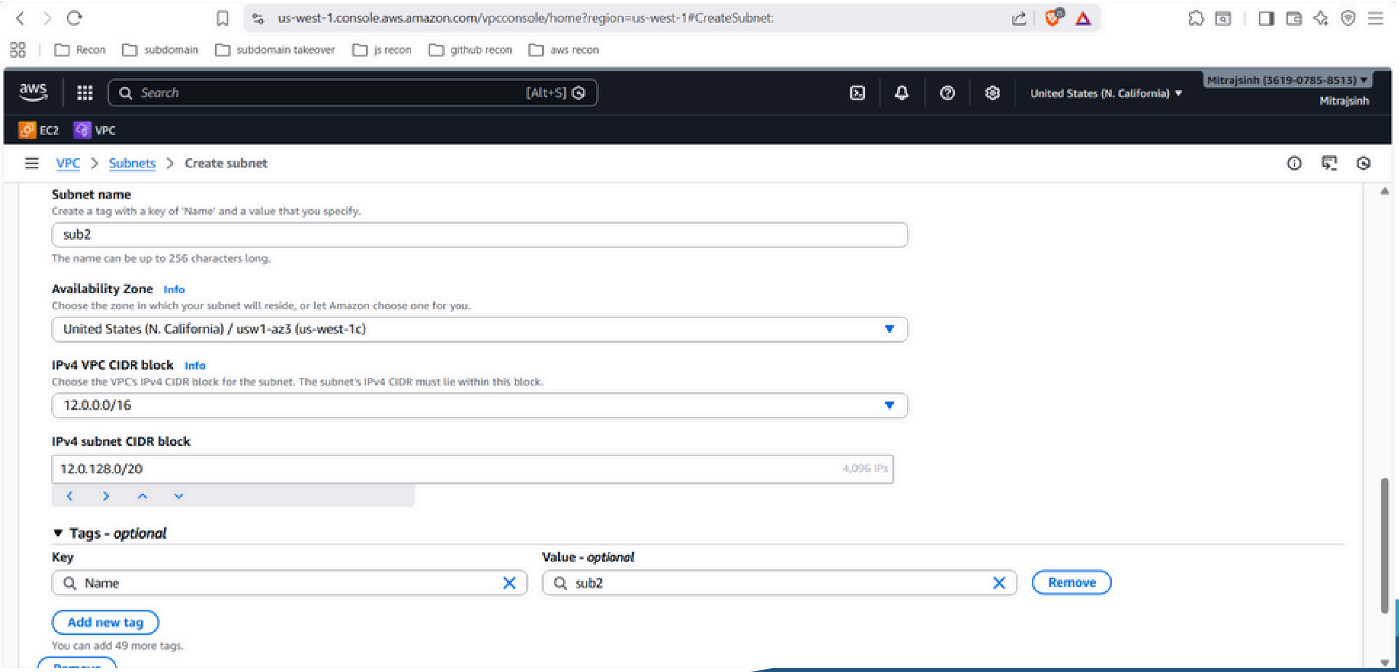
**IPv4 VPC CIDR block** Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
12.0.0.0/16

**IPv4 subnet CIDR block**
12.0.128.0/20 4,096 IPs

▾ **Tags - *optional***

| Key | Value - *optional* | |
|---|---|---|
| Name | sub2 | Remove |

You can add 49 more tags.



us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#CreateSubnet:

**Subnet name**
Create a tag with a key of 'Name' and a value that you specify.
sub2
The name can be up to 256 characters long.

**Availability Zone** Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. California) / usw1-az3 (us-west-1c)

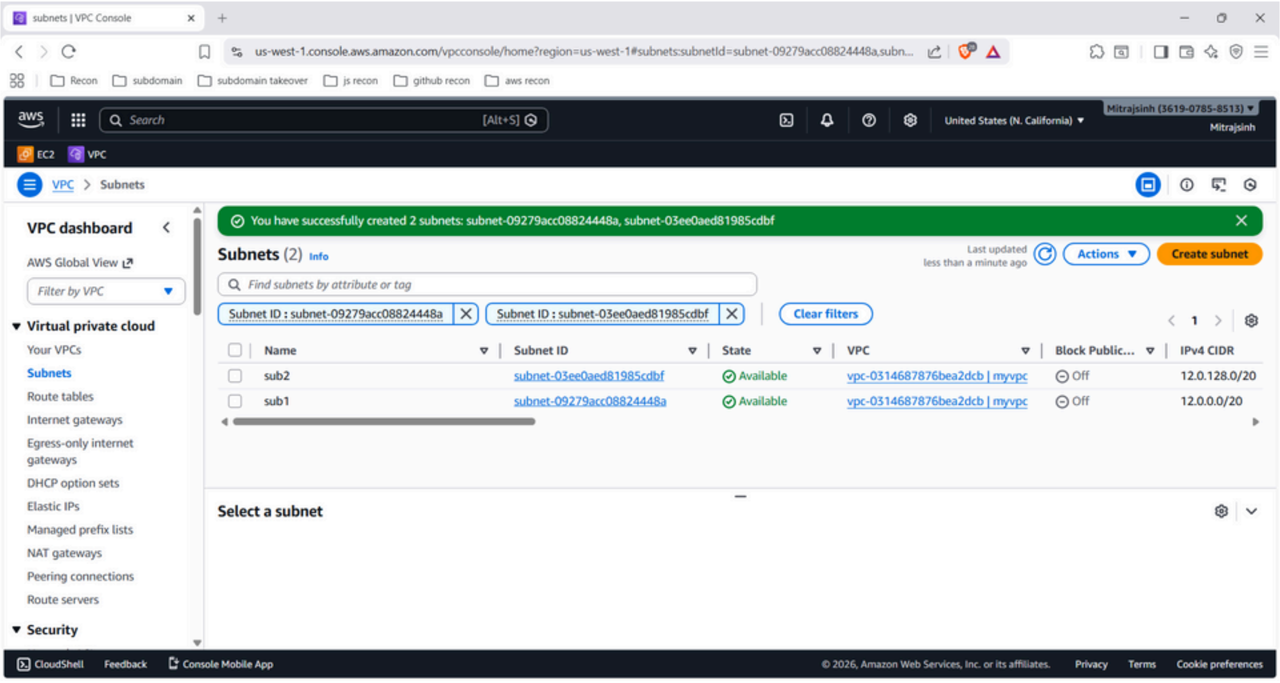
**IPv4 VPC CIDR block** Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
12.0.0.0/16

**IPv4 subnet CIDR block**
12.0.128.0/20 4,096 IPs

▾ **Tags - *optional***

Key: Name
Value - *optional*: sub2

You can add 49 more tags.



You have successfully created 2 subnets: subnet-09279acc08824448a, subnet-03ee0aed81985cdbf

Subnets (2) Info

Last updated less than a minute ago

Actions ▾ | Create subnet

Find subnets by attribute or tag

Subnet ID : subnet-09279acc08824448a ✕ | Subnet ID : subnet-03ee0aed81985cdbf ✕ | Clear filters

| □ | Name | Subnet ID | State | VPC | Block Public... | IPv4 CIDR |
|---|---|---|---|---|---|---|
| □ | sub2 | subnet-03ee0aed81985cdbf | ✓ Available | vpc-0314687876bea2dcb \| myvpc | ⊖ Off | 12.0.128.0/20 |
| □ | sub1 | subnet-09279acc08824448a | ✓ Available | vpc-0314687876bea2dcb \| myvpc | ⊖ Off | 12.0.0.0/20 |

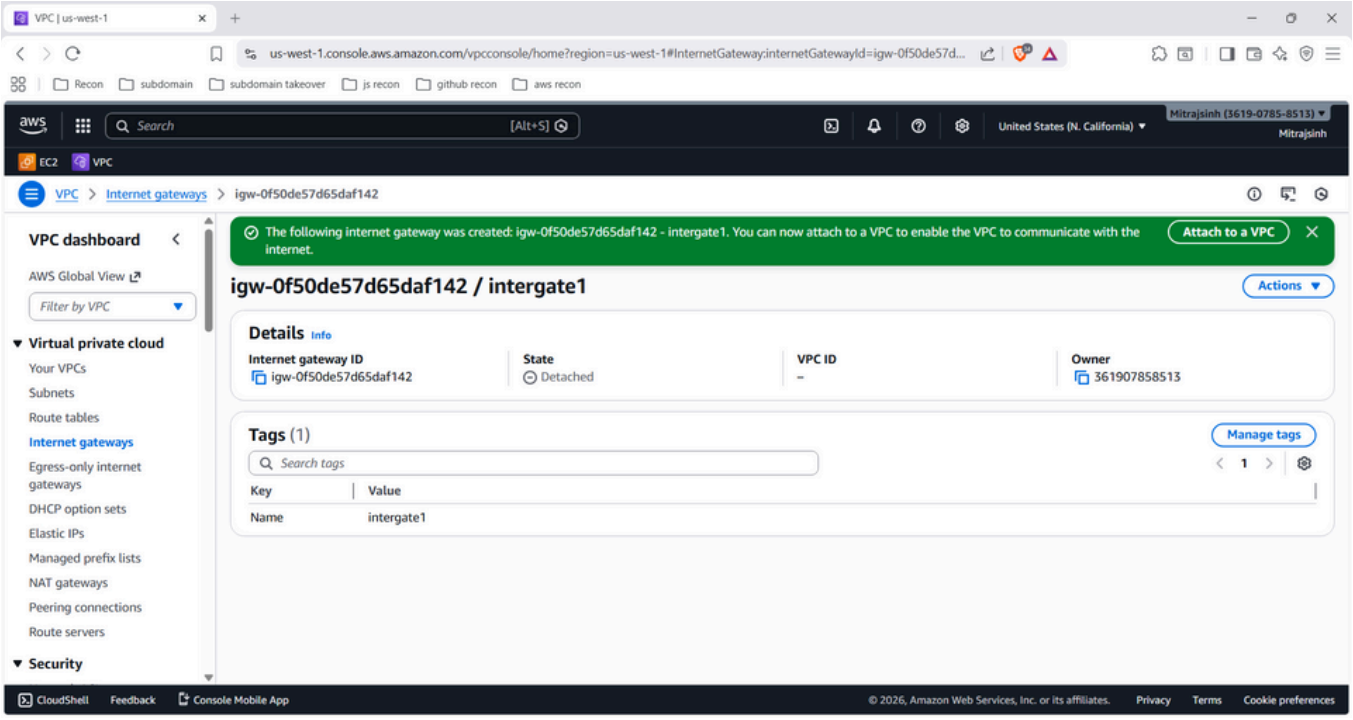
Select a subnet

## ◆ Step 3: Create and Attach Internet Gateway

An Internet Gateway was created and attached to the VPC to enable internet access for public subnets and EC2 instances.

Purpose:

To allow communication between VPC resources and the internet.



The following internet gateway was created: igw-0f50de57d65daf142 - intergate1. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-0f50de57d65daf142 / intergate1

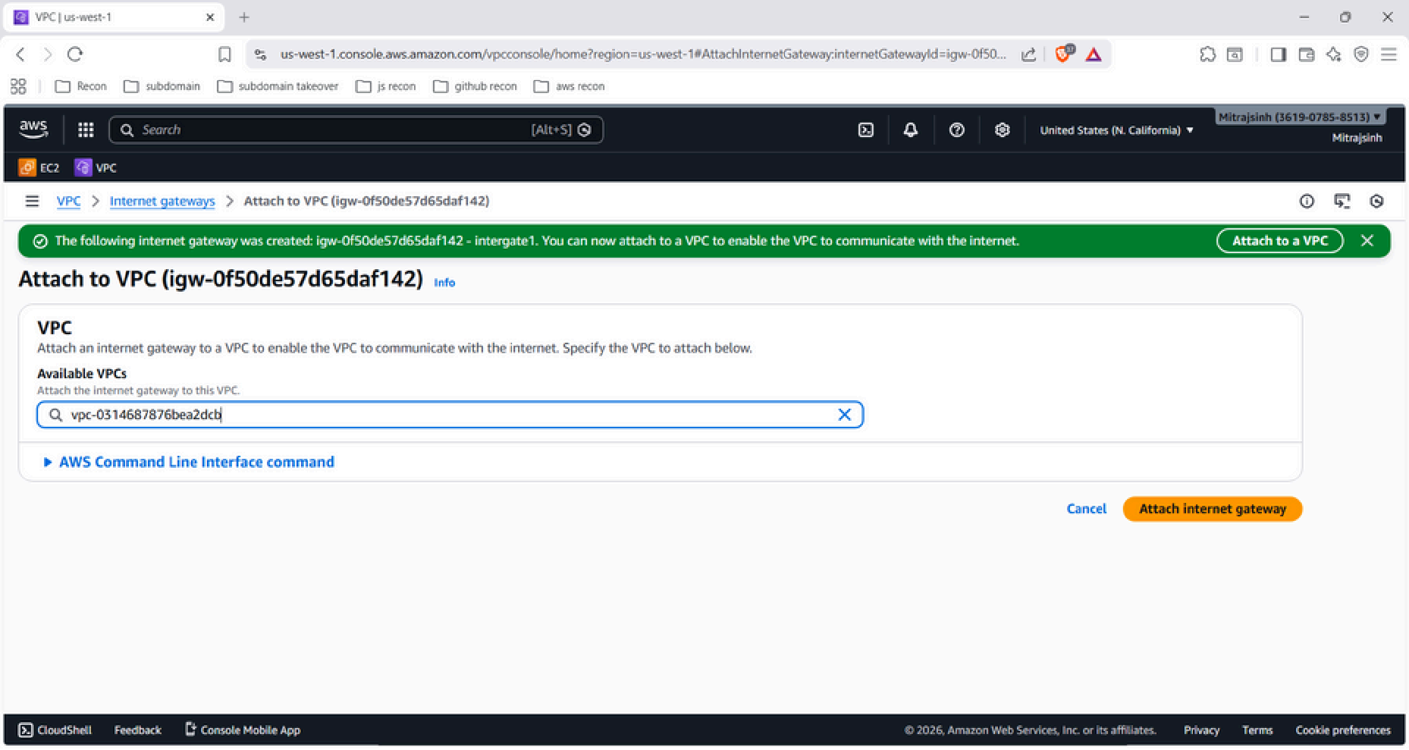
Details Info

| Internet gateway ID | State | VPC ID | Owner |
|---|---|---|---|
| igw-0f50de57d65daf142 | Detached | – | 361907858513 |

Tags (1)

| Key | Value |
|---|---|
| Name | intergate1 |

- Now attach this internet gateway to a VPC


VPC | us-west-1
us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#AttachInternetGateway:internetGatewayId=igw-0f50...
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
Search
[Alt+S]
United States (N. California) ▾
Mitrajsinh (3619-0785-8513) ▾
Mitrajsinh
EC2
VPC
VPC > Internet gateways > Attach to VPC (igw-0f50de57d65daf142)
Attach to a VPC
▸ AWS Command Line Interface command
Cancel
Attach internet gateway
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

The following internet gateway was created: igw-0f50de57d65daf142 - intergate1. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to VPC (igw-0f50de57d65daf142) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

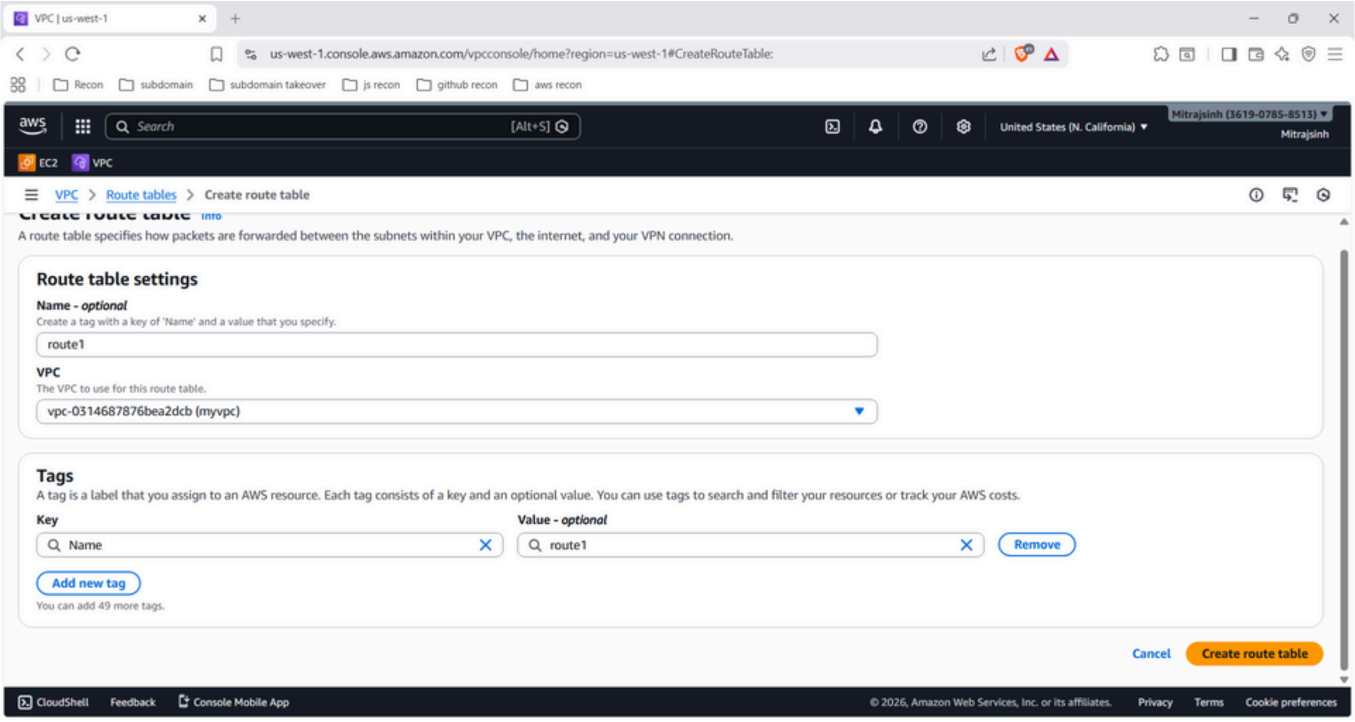
Available VPCs

Attach the internet gateway to this VPC.

vpc-0314687876bea2dcb

## Step 4: Create Route table

- A custom route table was created to control traffic routing inside the VPC.



VPC | us-west-1
us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#CreateRouteTable:
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
Search
[Alt+S]
United States (N. California) ▾
Mitrajsinh (3619-0785-8513) ▾
Mitrajsinh
EC2
VPC
VPC > Route tables > Create route table
Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
route1

VPC
The VPC to use for this route table.
vpc-0314687876bea2dcb (myvpc)

**Tags**

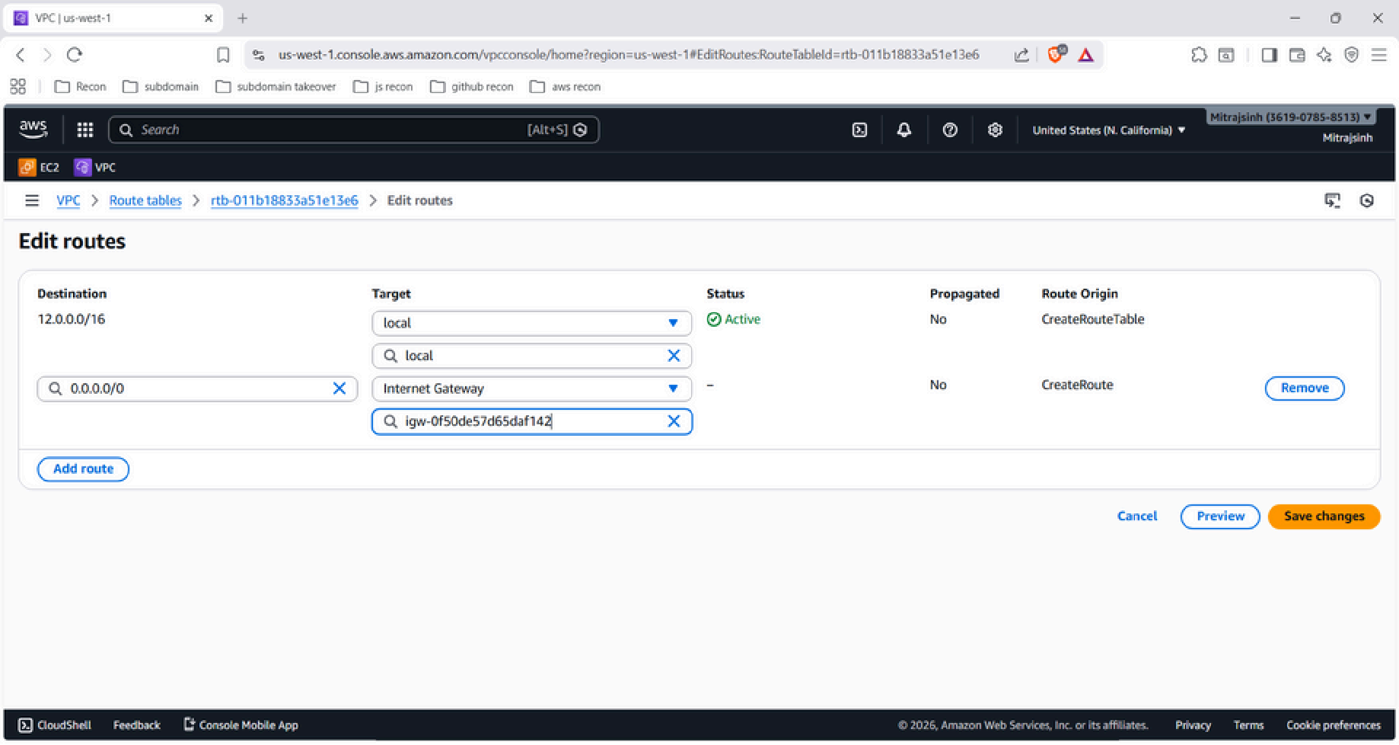
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Key | Value - *optional* |
|---|---|
| Name | route1 |

You can add 49 more tags.

## Step 5: Configure Route Table

- A route was added to the route table to forward all outbound traffic (0.0.0.0/0) to the Internet Gateway. This makes the subnet a public subnet.


VPC | us-west-1
us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#EditRoutes:RouteTableId=rtb-011b18833a51e13e6
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
Search
[Alt+S]
United States (N. California)
Mitrajsinh (3619-0785-8513)
Mitrajsinh
EC2
VPC
VPC > Route tables > rtb-011b18833a51e13e6 > Edit routes
Add route
Remove
Cancel
Preview
Save changes
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Edit routes

| Destination | Target | Status | Propagated | Route Origin |
|---|---|---|---|---|
| 12.0.0.0/16 | local<br>local | Active | No | CreateRouteTable |
| 0.0.0.0/0 | Internet Gateway<br>igw-0f50de57d65daf142 | - | No | CreateRoute |

Route details:

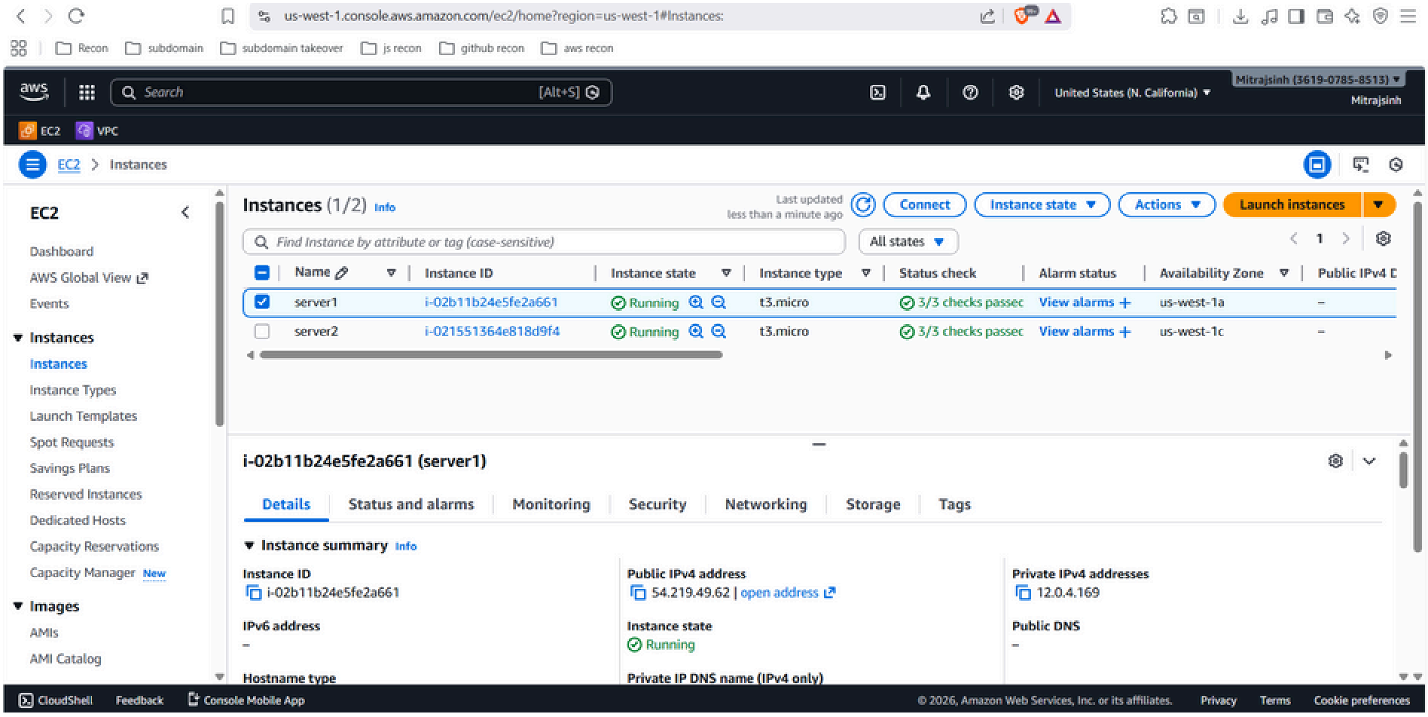
- Destination : 0.0.0.0/0
- Target : Internet gateway

## Step 6: Launch EC2 instance

Two Windows-based EC2 instances were launched to host the web application. The instances were deployed in a custom VPC and public subnets to enable internet access and load balancing.

Instance configuration details :

- Instance Name: server1,server2
- AMI : Microsoft Windows
- Instance Type: t2.micro (Free Tier)
- VPC: myvpc
- Subnet: Public Subnet



Instances (1/2) Info

| | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 D |
|---|---|---|---|---|---|---|---|---|
| ☑ | server1 | i-02b11b24e5fe2a661 | Running | t3.micro | 3/3 checks passec | View alarms + | us-west-1a | – |
| □ | server2 | i-021551364e818d9f4 | Running | t3.micro | 3/3 checks passec | View alarms + | us-west-1c | – |

i-02b11b24e5fe2a661 (server1)

Instance summary Info

Instance ID: i-02b11b24e5fe2a661

Public IPv4 address: 54.219.49.62 | open address

Private IPv4 addresses: 12.0.4.169

IPv6 address: –

Instance state: Running

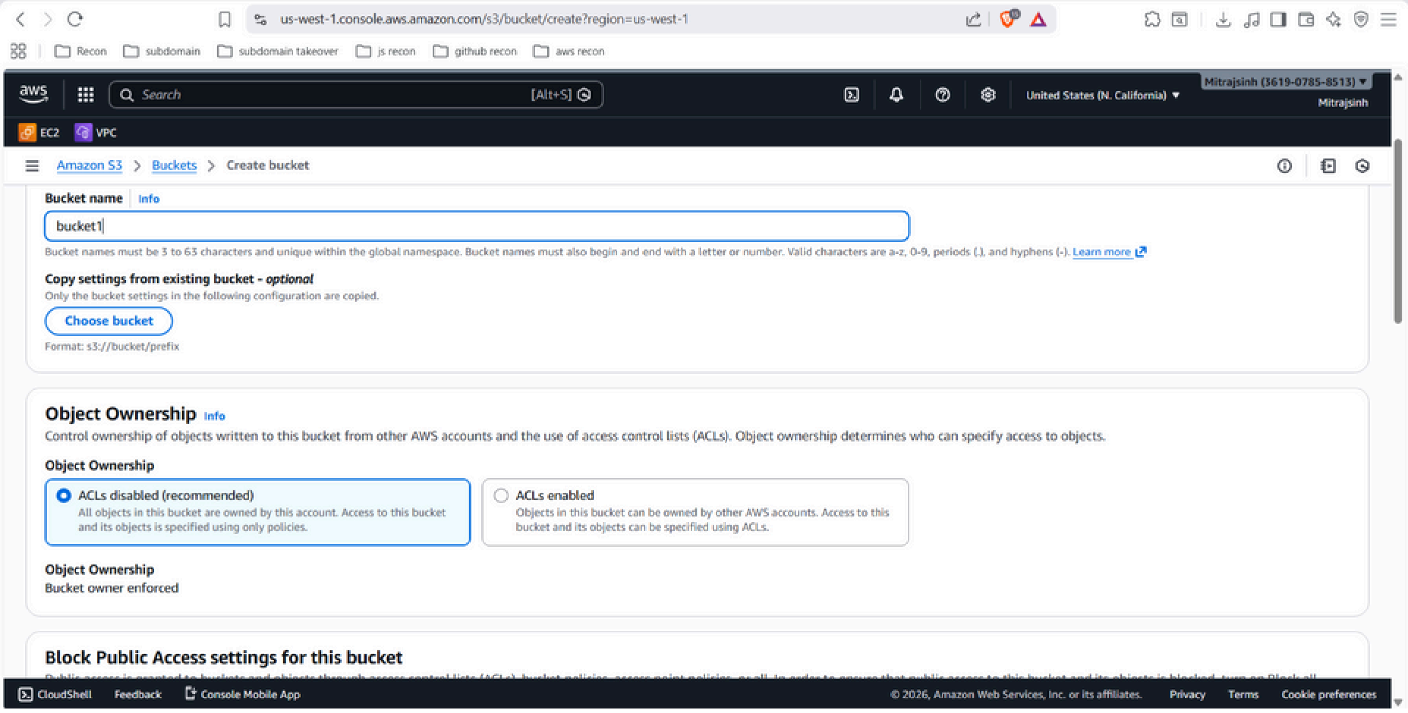
Public DNS: –

Hostname type

Private IP DNS name (IPv4 only)

## Step 7: Create Amazon s3 bucket

- An Amazon S3 bucket was created to store static files, images, and backup data for the web application. S3 provides highly durable and scalable object storage.


us-west-1.console.aws.amazon.com/s3/bucket/create?region=us-west-1
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
aws
Search
[Alt+S]
United States (N. California) ▾
Mitrajsinh (3619-0785-8513) ▾
Mitrajsinh
EC2
VPC
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Amazon S3 > Buckets > Create bucket

Bucket name Info

bucket1

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn more

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Object Ownership

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

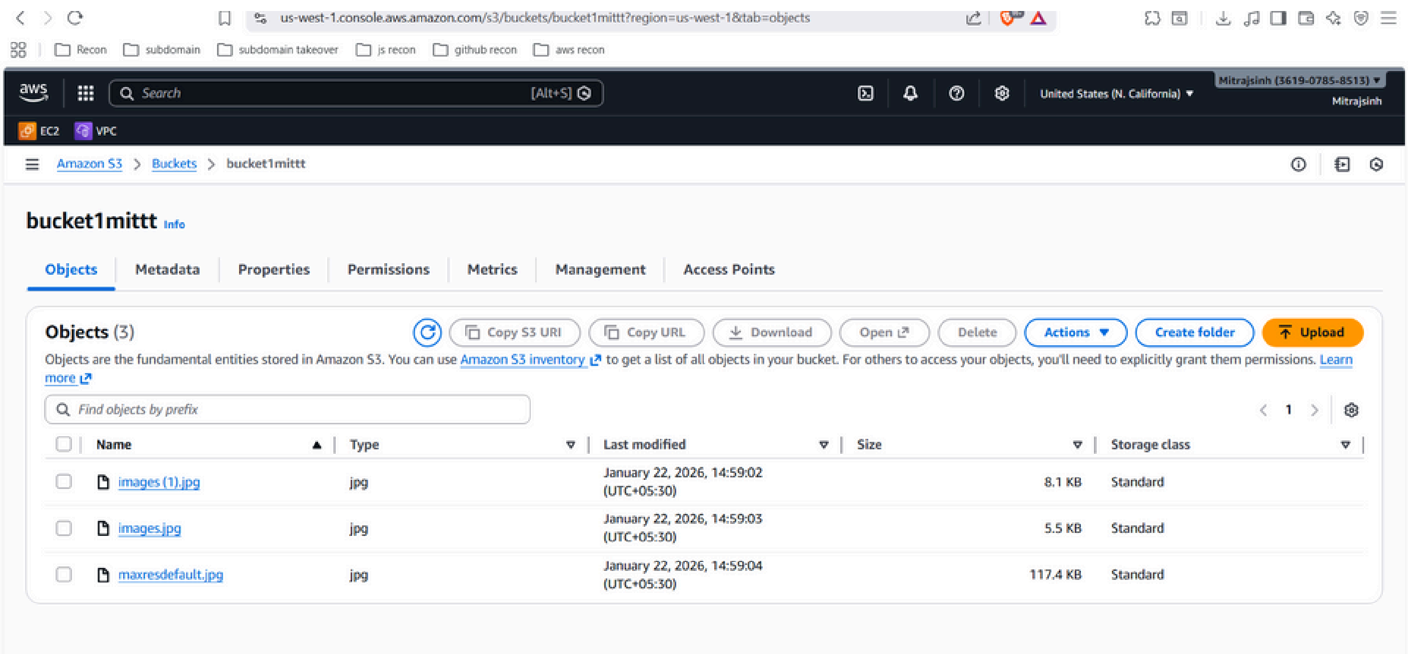
○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

- Images and static files were uploaded to the S3 bucket to simulate real-world website assets and backup storage.



Amazon S3 > Buckets > bucket1mittt

bucket1mittt Info

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

| □ | Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|---|
| □ | images (1).jpg | jpg | January 22, 2026, 14:59:02 (UTC+05:30) | 8.1 KB | Standard |
| □ | images.jpg | jpg | January 22, 2026, 14:59:03 (UTC+05:30) | 5.5 KB | Standard |
| □ | maxresdefault.jpg | jpg | January 22, 2026, 14:59:04 (UTC+05:30) | 117.4 KB | Standard |

## ◆ Step 8: Configure Bucket Policy for Public Access

- A bucket policy was generated using the AWS Policy Generator to allow public read access to the objects stored in the S3 bucket. This policy was added to the bucket to make the uploaded images publicly accessible via object URLs and also enable public access

Bucket Policy Purpose

- Allow public users to view images stored in S3
- Enable static content access for the web application
- Test public object access using S3 URLs

## ◆ Step 8: Install IIS Web Server on Windows EC2

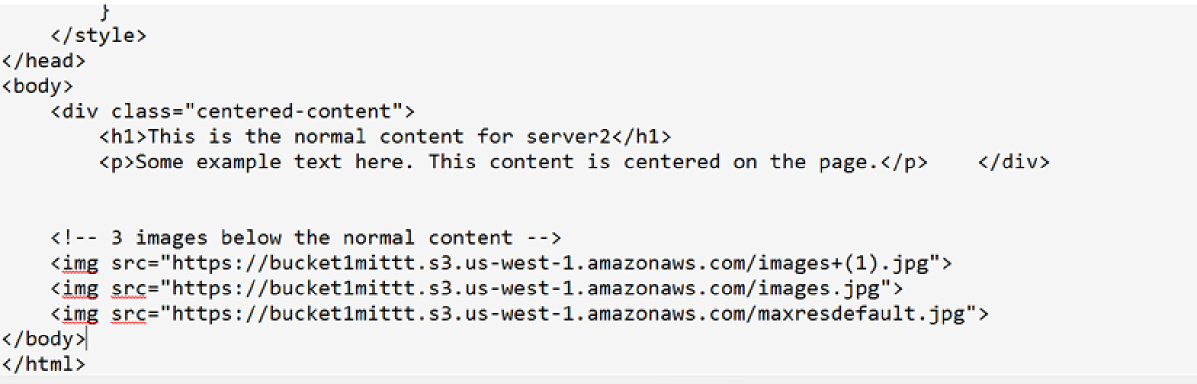
Internet Information Services (IIS) web server was installed on both Windows EC2 instances (server1 and server2) to host the web application.

IIS Installation Steps:

- Open Server Manager
- Click Add Roles and Features
- Select Web Server (IIS)
- Install the instance

## ◆ Step 8: Create HTML Web Page

A custom HTML file (index.html) was created on both server1 and server2 instances. The HTML page contains text content and images fetched from Amazon S3 using object URLs
Deploy this on below path :
C:\inetpub\wwwroot\index.html

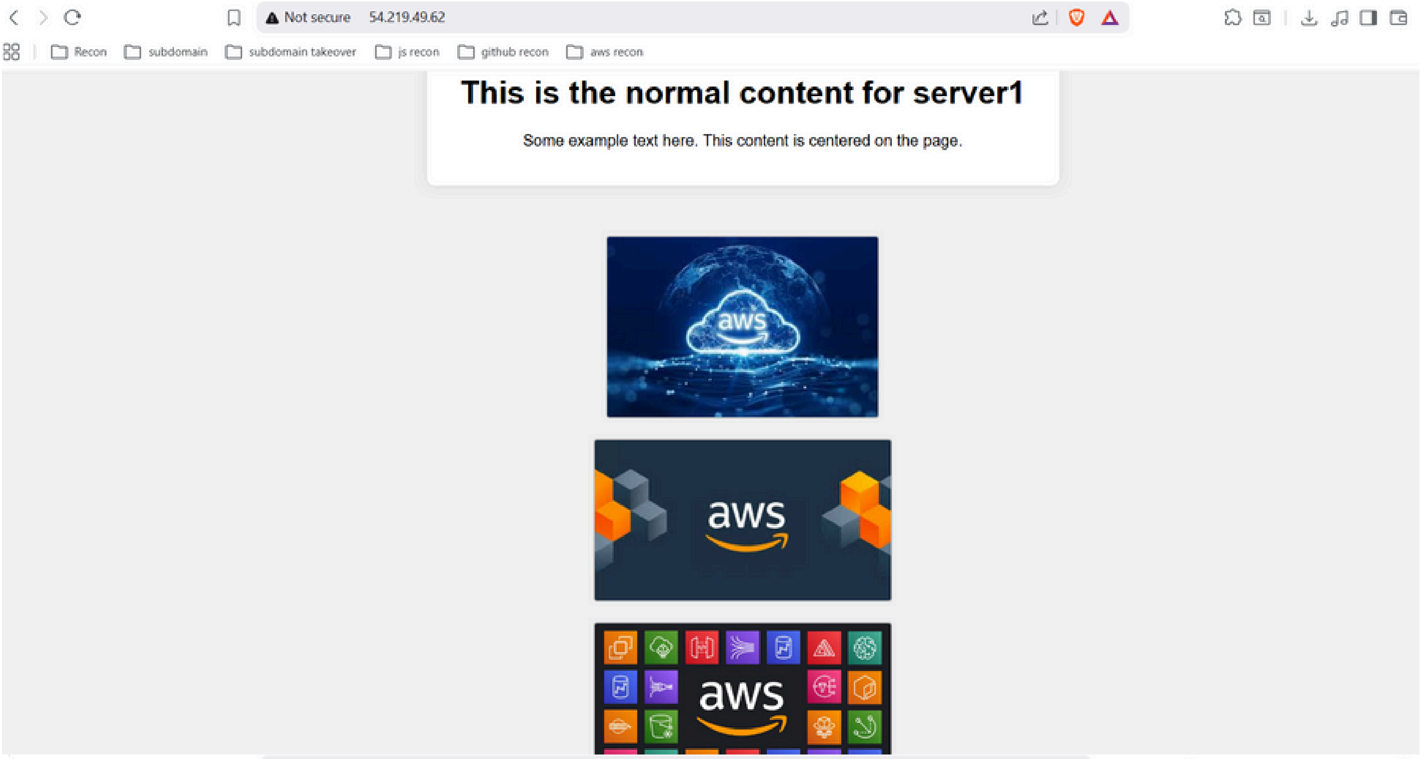


```
        }
    </style>
</head>
<body>
    <div class="centered-content">
        <h1>This is the normal content for server2</h1>
        <p>Some example text here. This content is centered on the page.</p>    </div>


    <!-- 3 images below the normal content -->
    <img src="https://bucket1mittt.s3.us-west-1.amazonaws.com/images+(1).jpg">
    <img src="https://bucket1mittt.s3.us-west-1.amazonaws.com/images.jpg">
    <img src="https://bucket1mittt.s3.us-west-1.amazonaws.com/maxresdefault.jpg">
</body>
</html>
```

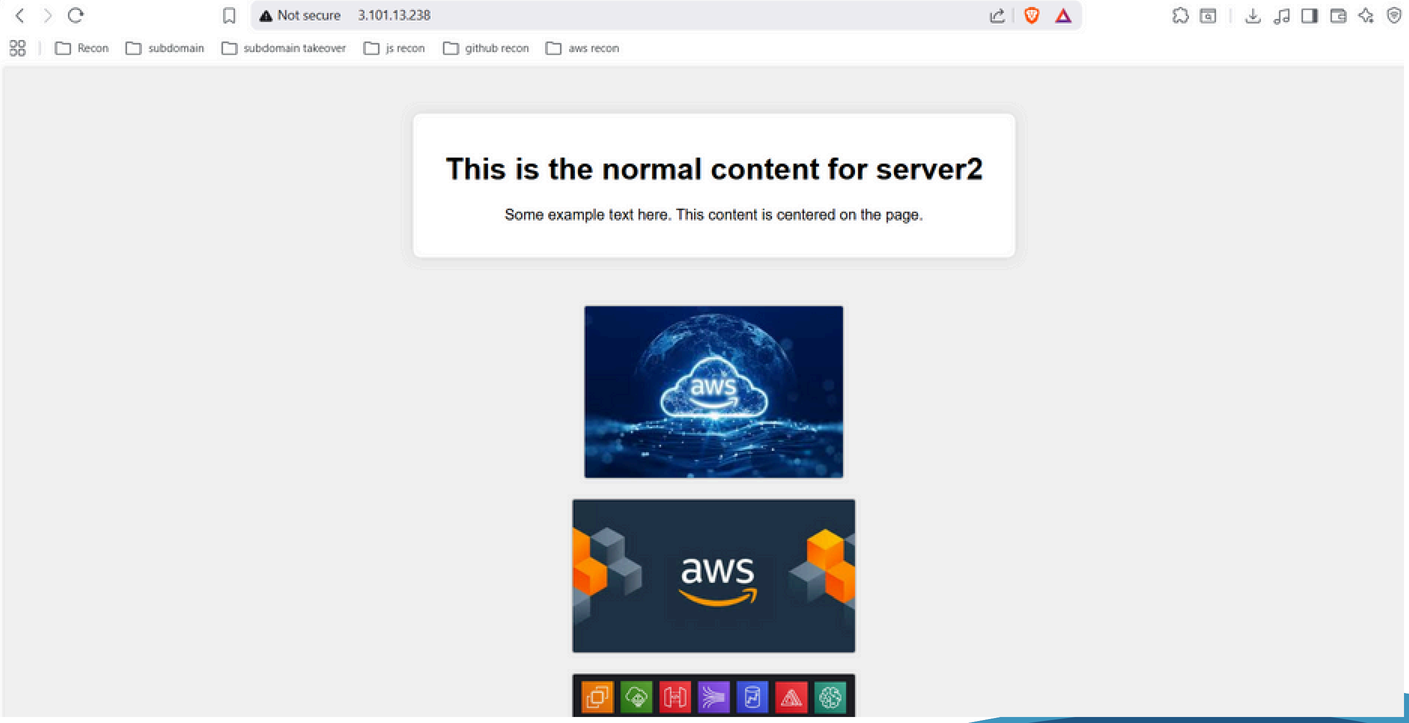
## ◆ Step 9: Website testing

The website was tested using the public IP addresses of both EC2 instances.
 The HTML page successfully displayed text and images stored in Amazon S3, proving integration between EC2 and S3.



This is the normal content for server1

Some example text here. This content is centered on the page.

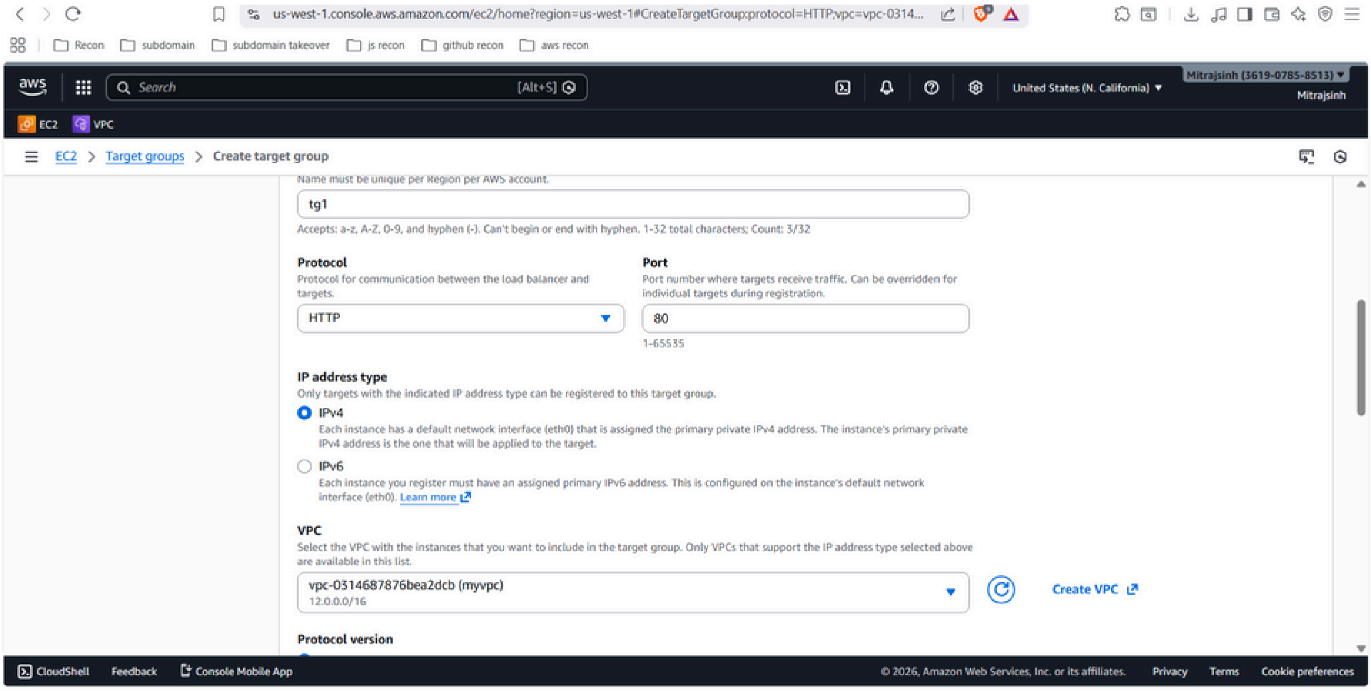


This is the normal content for server2

Some example text here. This content is centered on the page.

# Step 10: Create target group

A target group was created to register EC2 instances that will receive traffic from the Application Load Balancer.



us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#CreateTargetGroup:protocol=HTTP;vpc=vpc-0314...
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
Search
[Alt+S]
United States (N. California) ▾
Mitrajsinh (3619-0785-8513) ▾
Mitrajsinh
EC2
VPC
EC2 > Target groups > Create target group
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Name must be unique per Region per AWS account.

tg1

Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 3/32

**Protocol**
Protocol for communication between the load balancer and targets.
HTTP ▾

**Port**
Port number where targets receive traffic. Can be overridden for individual targets during registration.
80
1-65535

**IP address type**
Only targets with the indicated IP address type can be registered to this target group.

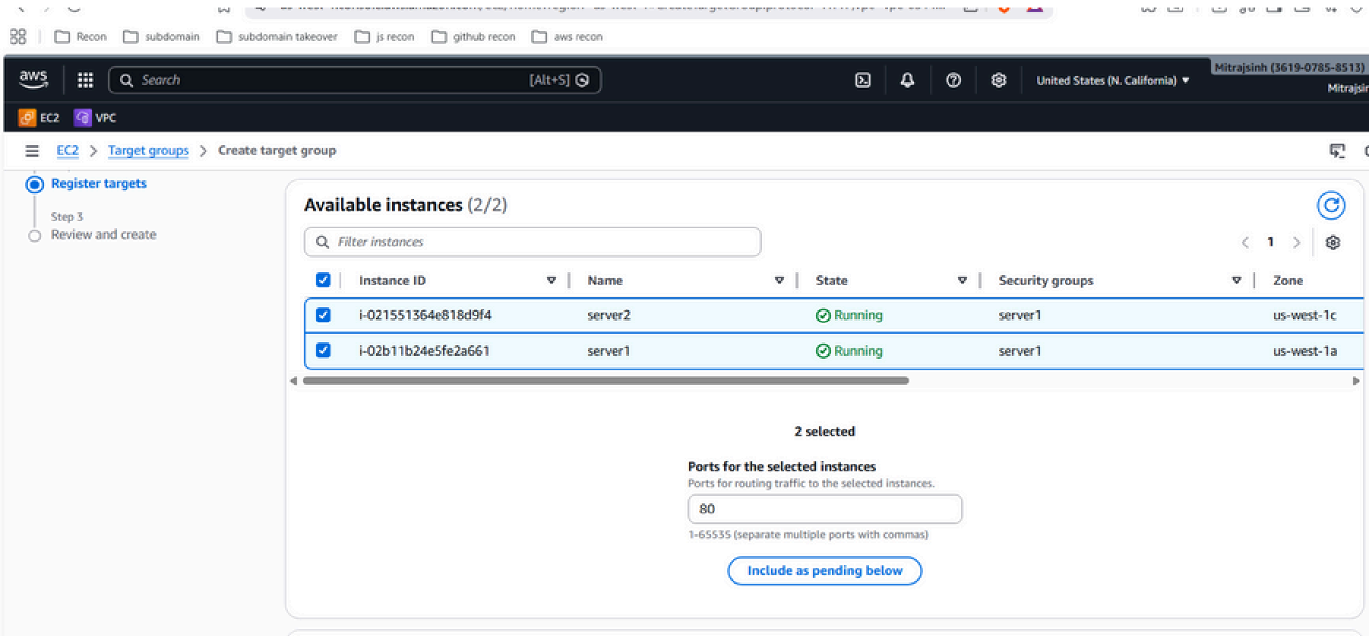
● IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

○ IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

**VPC**
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.
vpc-0314687876bea2dcb (myvpc)
12.0.0.0/16 ▾
Create VPC

**Protocol version**

Both EC2 instances (server1 and server2) were registered to the target group to enable load balancing between multiple servers.



**Available instances** (2/2)

| ☑ | Instance ID | Name | State | Security groups | Zone |
|---|---|---|---|---|---|
| ☑ | i-021551364e818d9f4 | server2 | Running | server1 | us-west-1c |
| ☑ | i-02b11b24e5fe2a661 | server1 | Running | server1 | us-west-1a |

2 selected

**Ports for the selected instances**
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

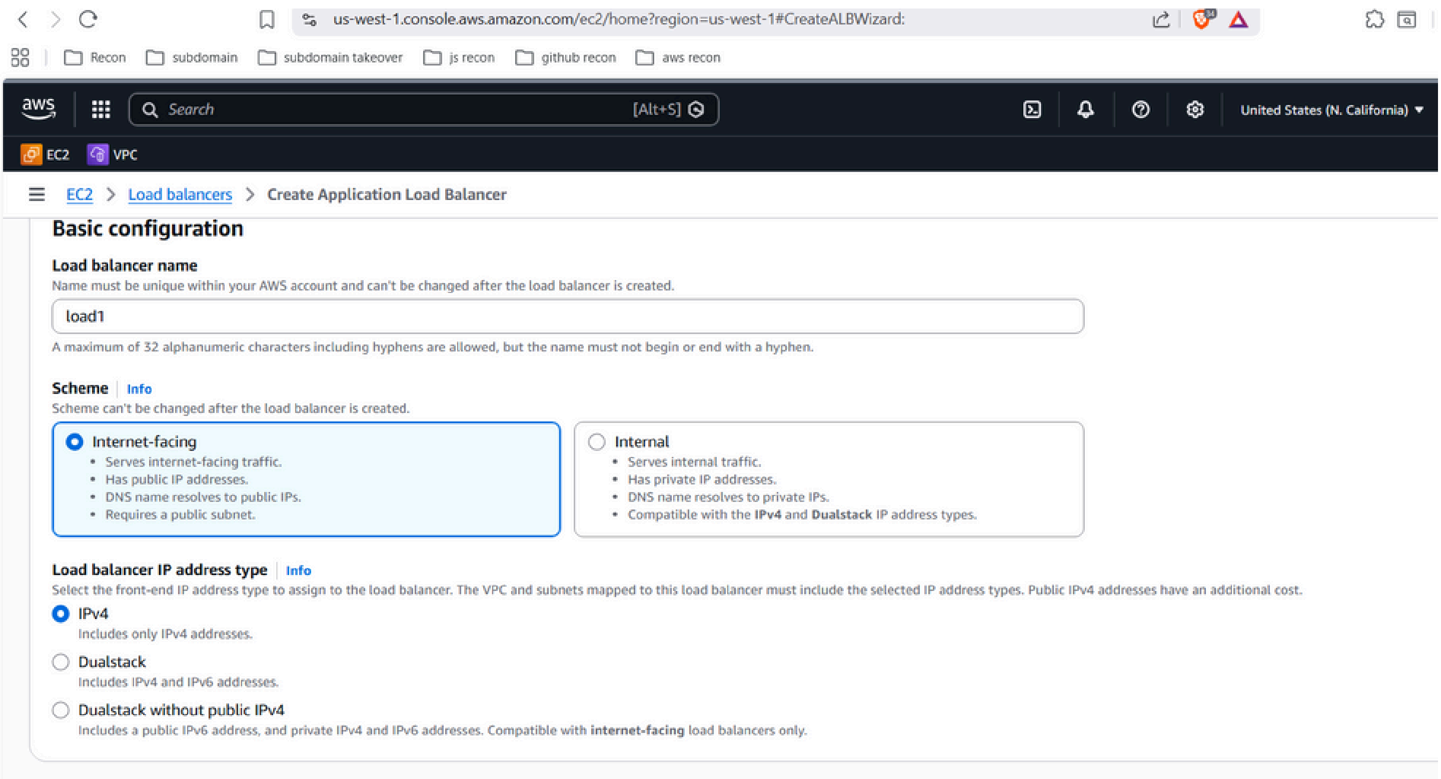
Include as pending below

# Step 10: Create load balancer

An Application Load Balancer was created to distribute incoming HTTP traffic across multiple EC2 instances.

ALB Configuration Details

- load balancer name: load1
- vpc : myvpc
- subnets : sub1 , sub2
- ip address type : ipv4


us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#CreateALBWizard:
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
aws
Search
[Alt+S]
United States (N. California)
EC2
VPC

EC2 > Load balancers > Create Application Load Balancer

**Basic configuration**

**Load balancer name**
Name must be unique within your AWS account and can't be changed after the load balancer is created.

load1

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.

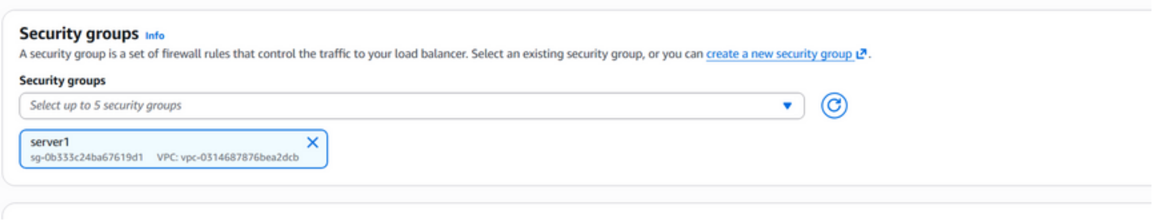
**Load balancer IP address type** Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

● 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.

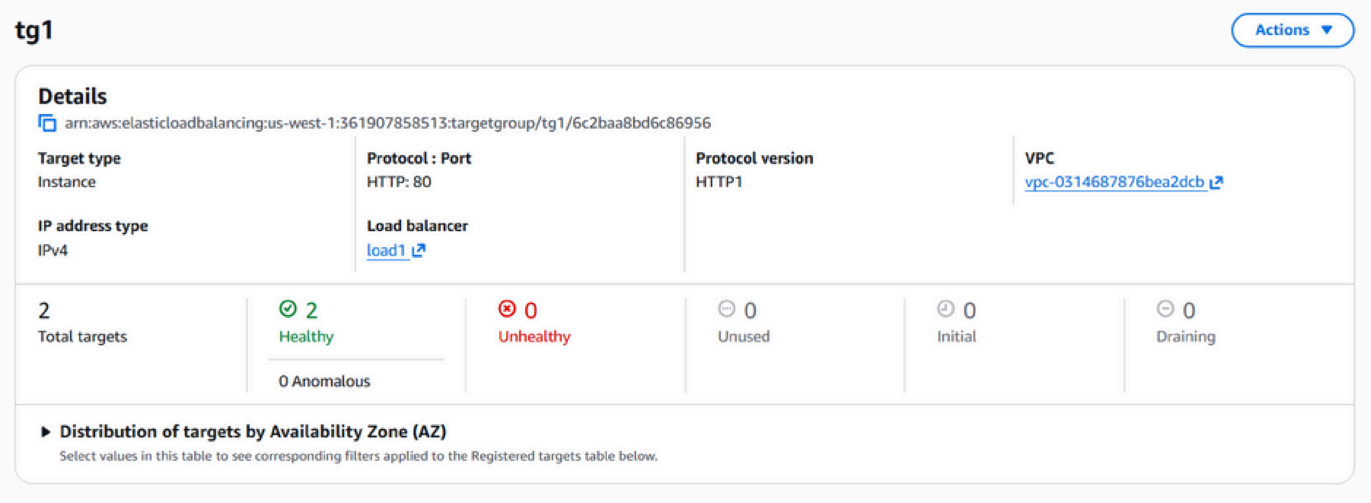
A security group was attached to the load balancer to allow HTTP traffic from the internet.



**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.

**Security groups**
Select up to 5 security groups

server1
sg-0b333c24ba67619d1 VPC: vpc-0314687876bea2dcb



tg1

**Details**

arn:aws:elasticloadbalancing:us-west-1:361907858513:targetgroup/tg1/6c2baa8bd6c86956

| | | | |
|---|---|---|---|
| **Target type**<br>Instance | **Protocol : Port**<br>HTTP: 80 | **Protocol version**<br>HTTP1 | **VPC**<br>vpc-0314687876bea2dcb |
| **IP address type**<br>IPv4 | **Load balancer**<br>load1 | | |

| | | | | | |
|---|---|---|---|---|---|
| 2<br>Total targets | ✓ 2<br>Healthy<br>0 Anomalous | ⊗ 0<br>Unhealthy | 0<br>Unused | 0<br>Initial | 0<br>Draining |

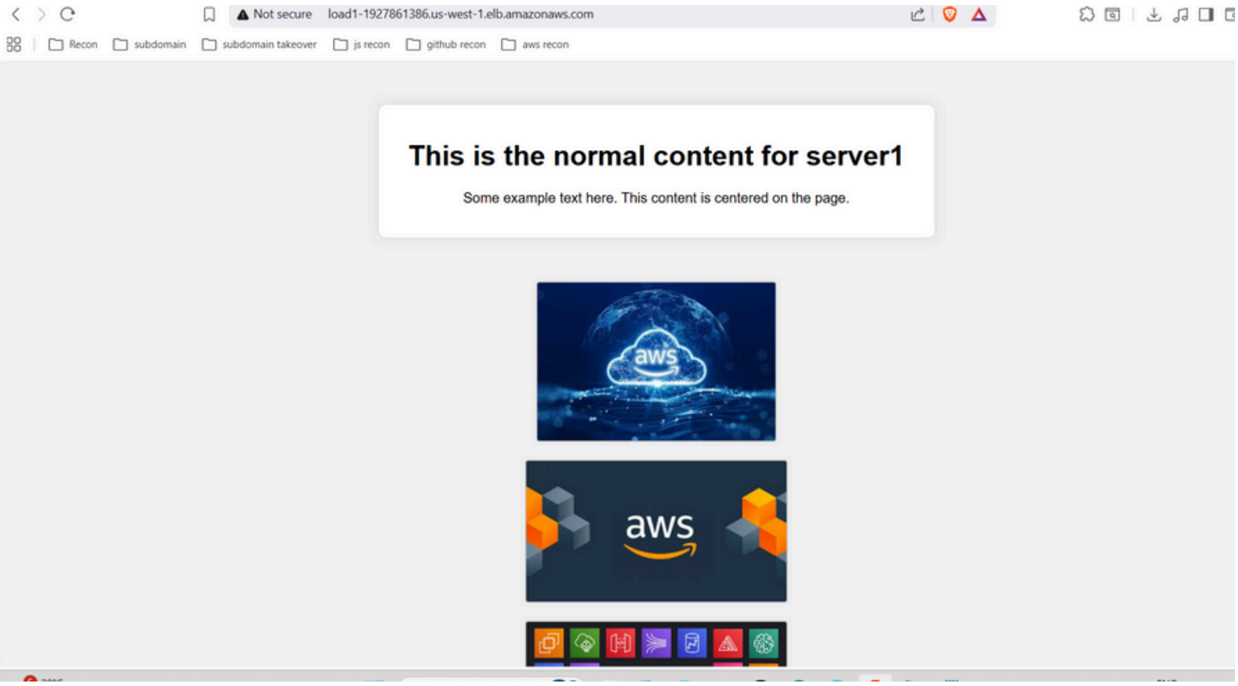
▶ **Distribution of targets by Availability Zone (AZ)**

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

## ◆ Step 11:Load Balancer Website Testing

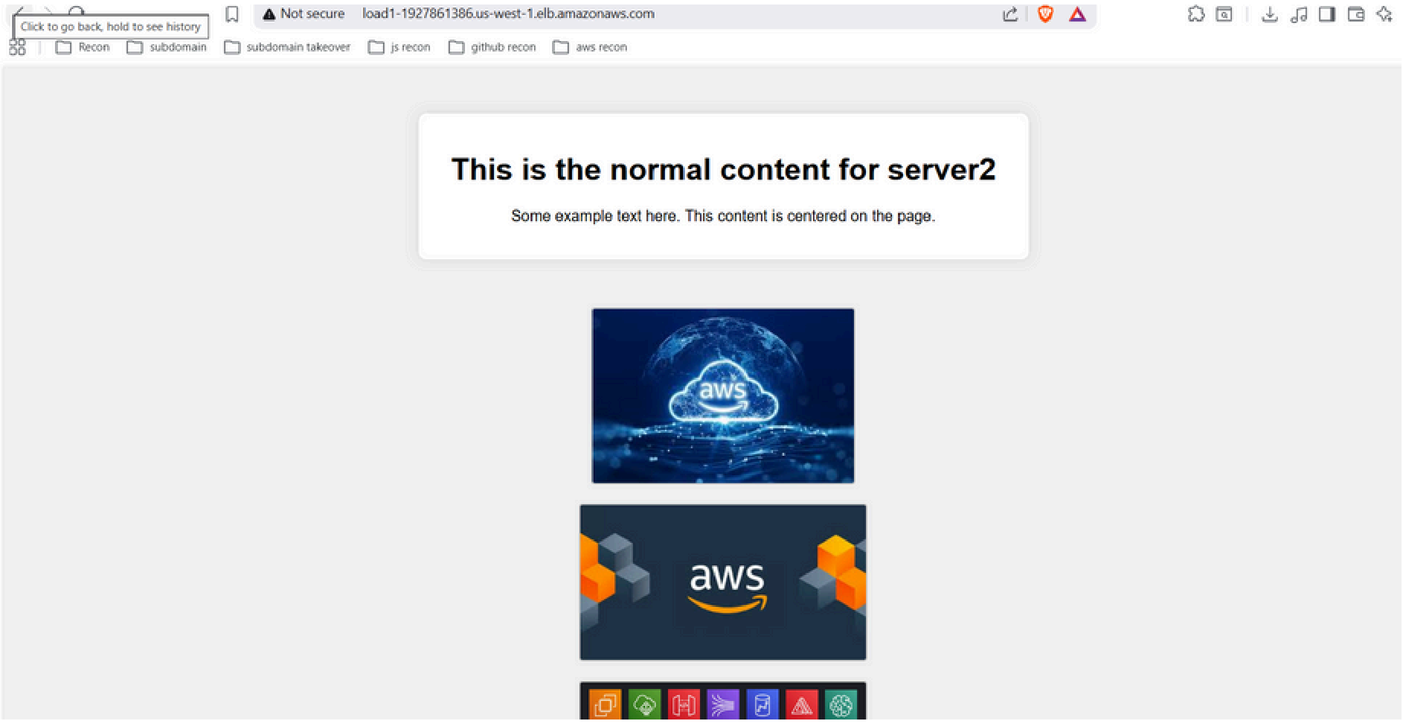
The Load Balancer DNS name was opened in a web browser.

Traffic was successfully distributed between server1 and server2, and the website displayed images stored in Amazon S3.



**This is the normal content for server1**

Some example text here. This content is centered on the page.

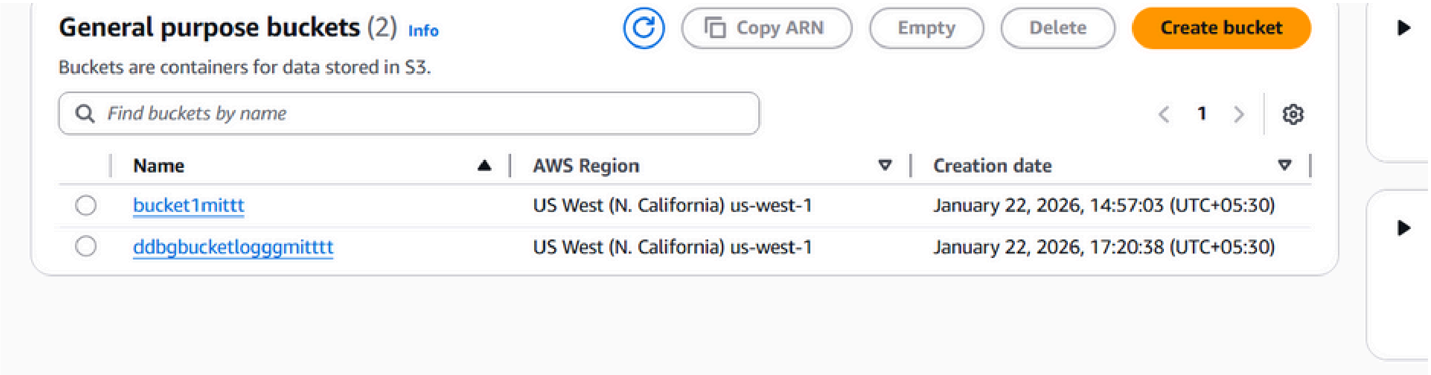

Click to go back, hold to see history
Not secure
load1-1927861386.us-west-1.elb.amazonaws.com
Recon
subdomain
subdomain takeover
js recon
github recon
aws recon
aws
aws

**This is the normal content for server2**

Some example text here. This content is centered on the page.

## ◆ Step 12 :Enable Logging and Store Logs in Amazon S3

A separate Amazon S3 bucket was created to store logs generated by the application and AWS services. Logging is important for monitoring, auditing, and security analysis.



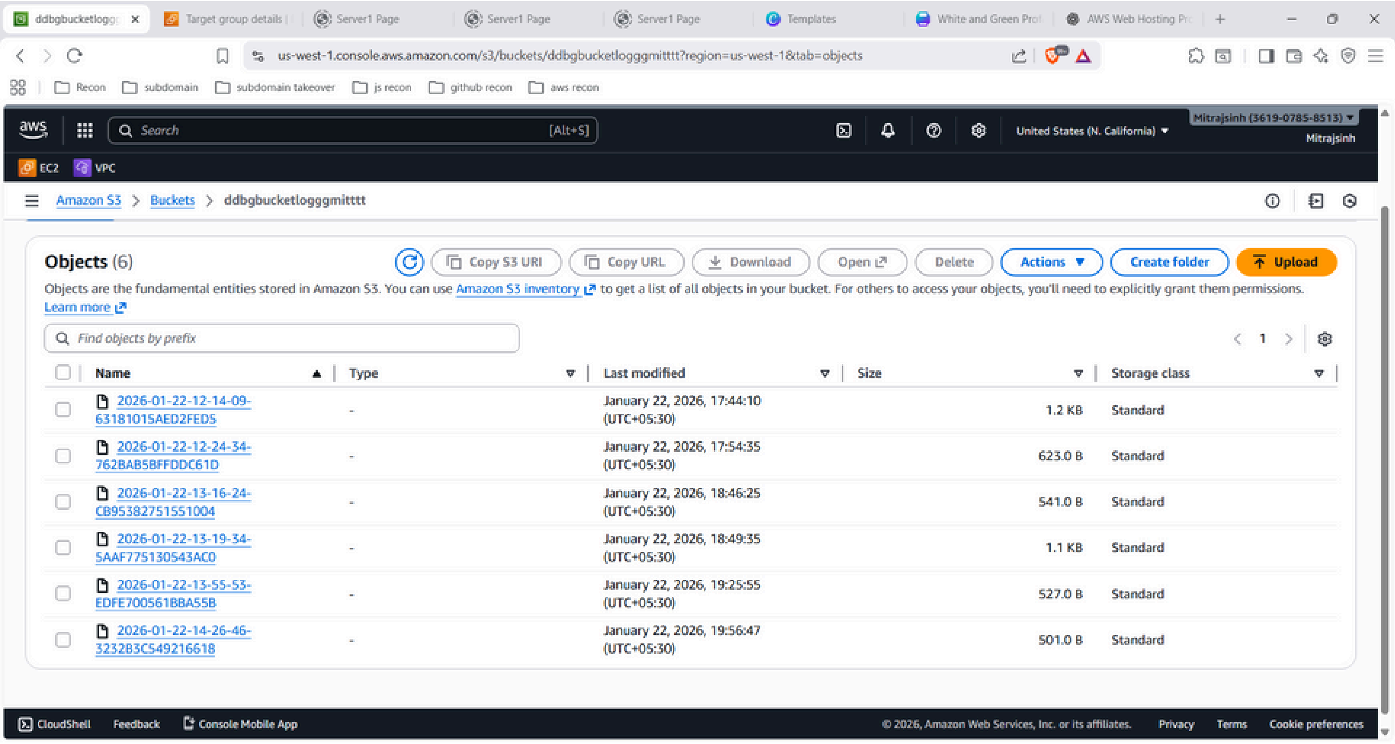
General purpose buckets (2)

Buckets are containers for data stored in S3.

| | Name ▲ | AWS Region ▽ | Creation date ▽ |
|---|---|---|---|
| ○ | bucket1mittt | US West (N. California) us-west-1 | January 22, 2026, 14:57:03 (UTC+05:30) |
| ○ | ddbgbucketlogggmitttt | US West (N. California) us-west-1 | January 22, 2026, 17:20:38 (UTC+05:30) |

# Step 13 :Upload and Store Log Files

Log files were uploaded and stored in the S3 bucket. These logs contain information such as request time, I address, and access details, which are useful for monitoring and troubleshooting.



us-west-1.console.aws.amazon.com/s3/buckets/ddbgbucketlogggmitttt?region=us-west-1&tab=objects
Amazon S3 > Buckets > ddbgbucketlogggmitttt
United States (N. California)
Mitrajsinh (3619-0785-8513)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Objects (6)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|
| 2026-01-22-12-14-09-63181015AED2FED5 | - | January 22, 2026, 17:44:10 (UTC+05:30) | 1.2 KB | Standard |
| 2026-01-22-12-24-34-762BAB5BFFDDC61D | - | January 22, 2026, 17:54:35 (UTC+05:30) | 623.0 B | Standard |
| 2026-01-22-13-16-24-CB95382751551004 | - | January 22, 2026, 18:46:25 (UTC+05:30) | 541.0 B | Standard |
| 2026-01-22-13-19-34-5AAF775130543AC0 | - | January 22, 2026, 18:49:35 (UTC+05:30) | 1.1 KB | Standard |
| 2026-01-22-13-55-53-EDFE700561BBA55B | - | January 22, 2026, 19:25:55 (UTC+05:30) | 527.0 B | Standard |
| 2026-01-22-14-26-46-3232B3C549216618 | - | January 22, 2026, 19:56:47 (UTC+05:30) | 501.0 B | Standard |

# Conclusion:

In this project, a scalable and highly available web hosting architecture was successfully deployed on AWS. A custom VPC was created to isolate the network environment. Two EC2 Windows instances were configured with IIS web server to host the website. Amazon S3 was used to store static assets and logs. An Application Load Balancer was implemented to distribute incoming traffic across multiple servers, ensuring fault tolerance and high availability.

This project demonstrates practical knowledge of cloud networking, compute, storage, load balancing, and logging in AWS.