Hosting a Static Website Using EC2 and S3

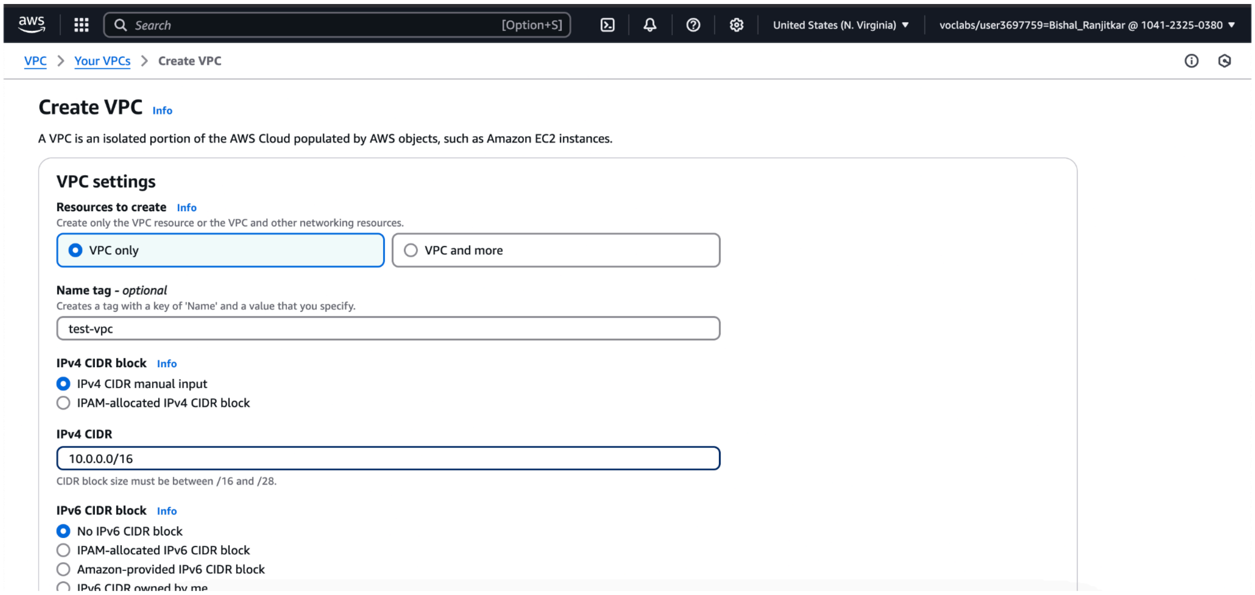
Project Overview :

In this project, I successfully hosted a static website using AWS services, specifically EC2 and S3. This involved setting up the necessary networking components, deploying an EC2 instance with Nginx, and configuring an S3 bucket for static website hosting.

Steps Involved

1. Setting Up the Network Infrastructure

- Created a **Virtual Private Cloud (VPC)**.
- Configured **Subnets** to allow public and private traffic.
- Set up a **Route Table** and associated it with the subnets.
- Created and attached an **Internet Gateway** to enable internet access.


aws
Search [Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

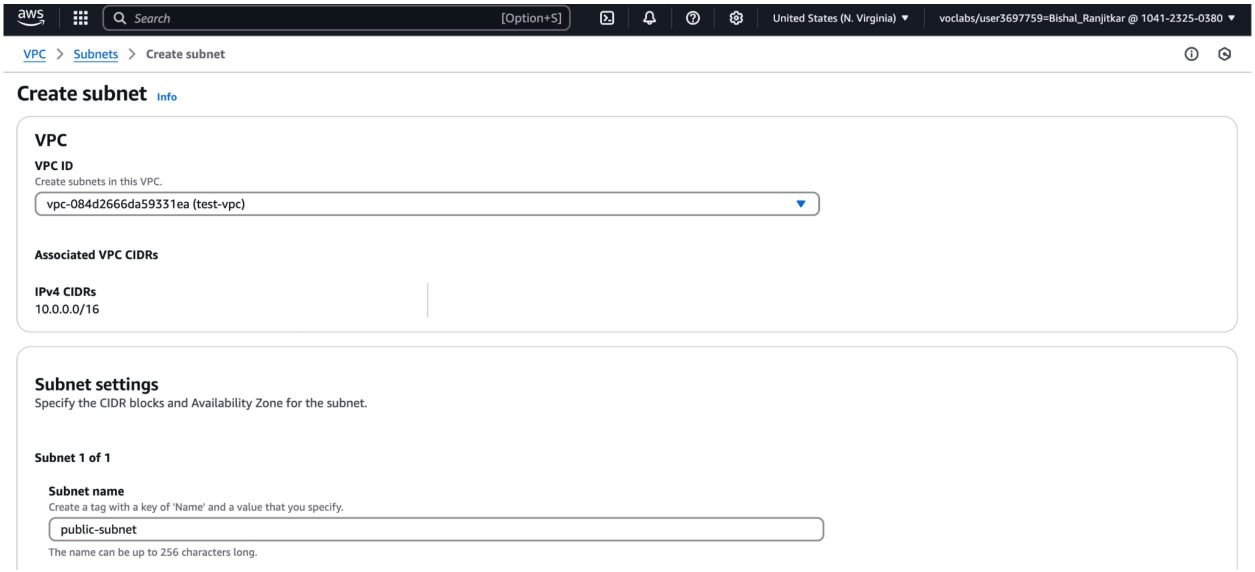
Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
● VPC only ○ VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
test-vpc

IPv4 CIDR block Info
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me


aws
Search
[Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
VPC > Subnets > Create subnet
Info

Create subnet

VPC

VPC ID
Create subnets in this VPC.
vpc-084d2666da59331ea (test-vpc)

Associated VPC CIDRs

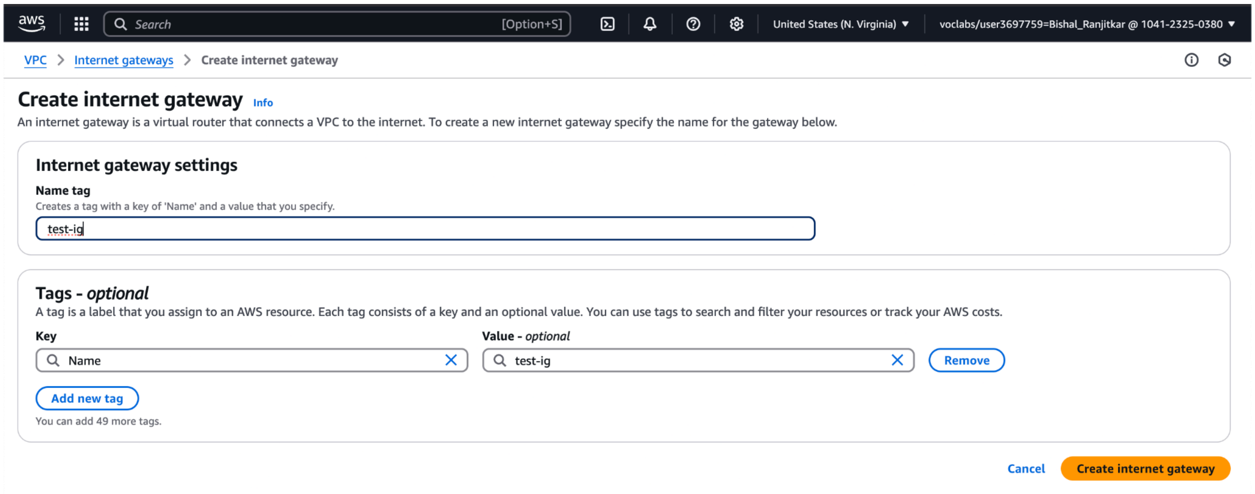
IPv4 CIDRs
10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

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


aws
Search
[Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
VPC > Internet gateways > Create internet gateway
Info
Remove
Add new tag
Cancel
Create internet gateway

Create internet gateway

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

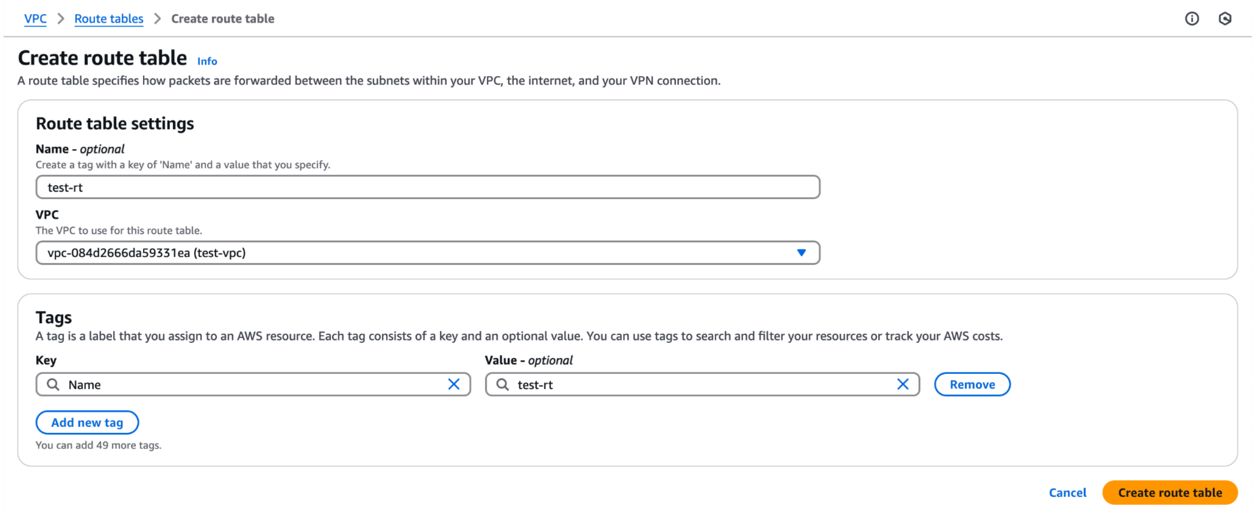
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
test-ig

Tags - *optional*

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	test-ig

You can add 49 more tags.



Create route table

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.
test-rt

VPC
The VPC to use for this route table.
vpc-084d2666da59331ea (test-vpc)

Tags

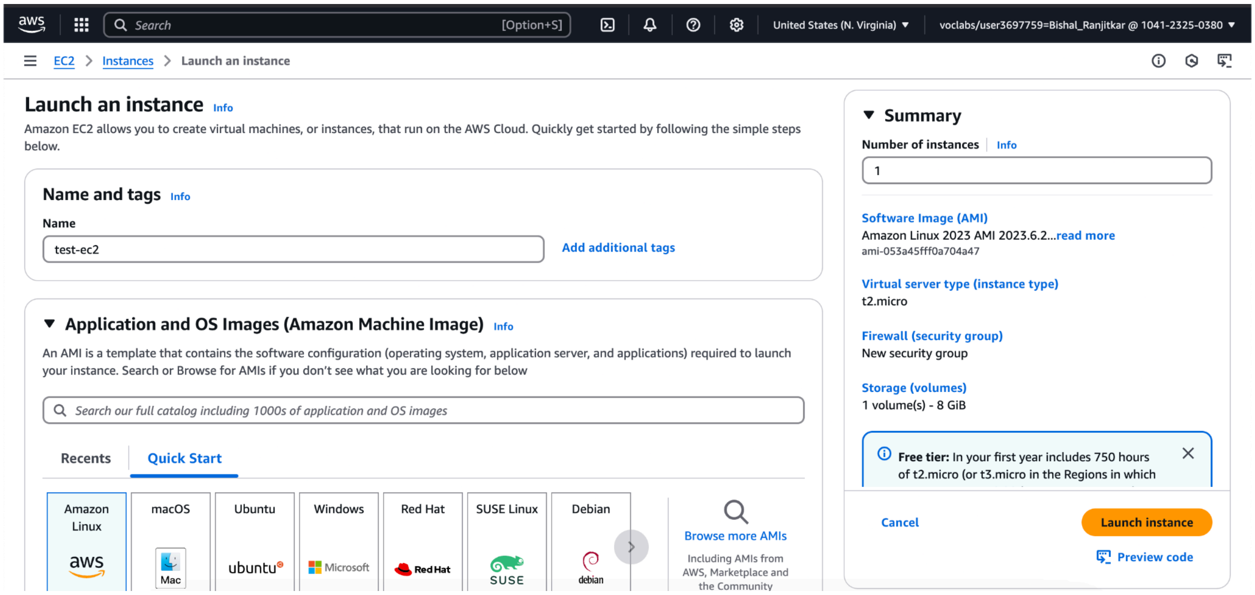
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	test-rt

You can add 49 more tags.

2. Hosting a Static Website on EC2

- Launched an **EC2 instance** with an Amazon Linux.
- Connected to the EC2 instance via **SSH**.
- Installed **Nginx** as the web server using:
- sudo yum install nginx -y # For Amazon Linux
- cd /usr/share/nginx/html
- nano index.html



aws
Search
[Option+S]
United States (N. Virginia) ▾
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▾
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: test-ec2

▼ 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
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

▼ Summary

Number of instances Info: 1

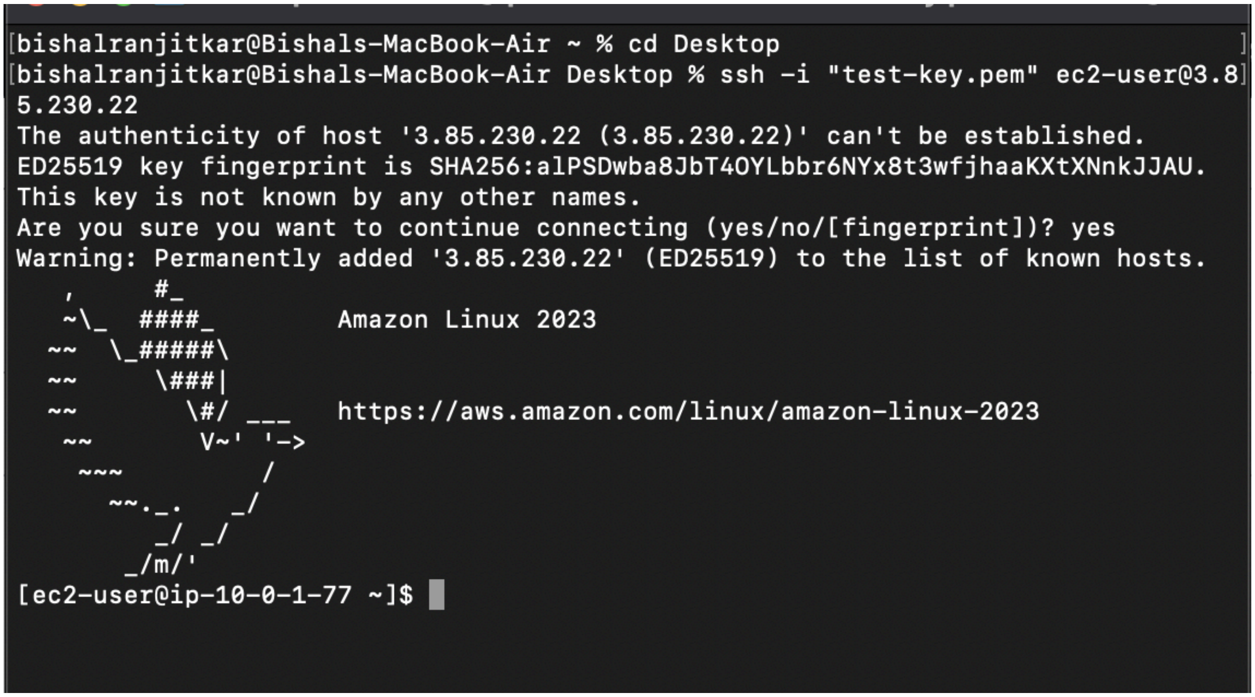
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-053a45fff0a704a47

Virtual server type (instance type)
t2.micro

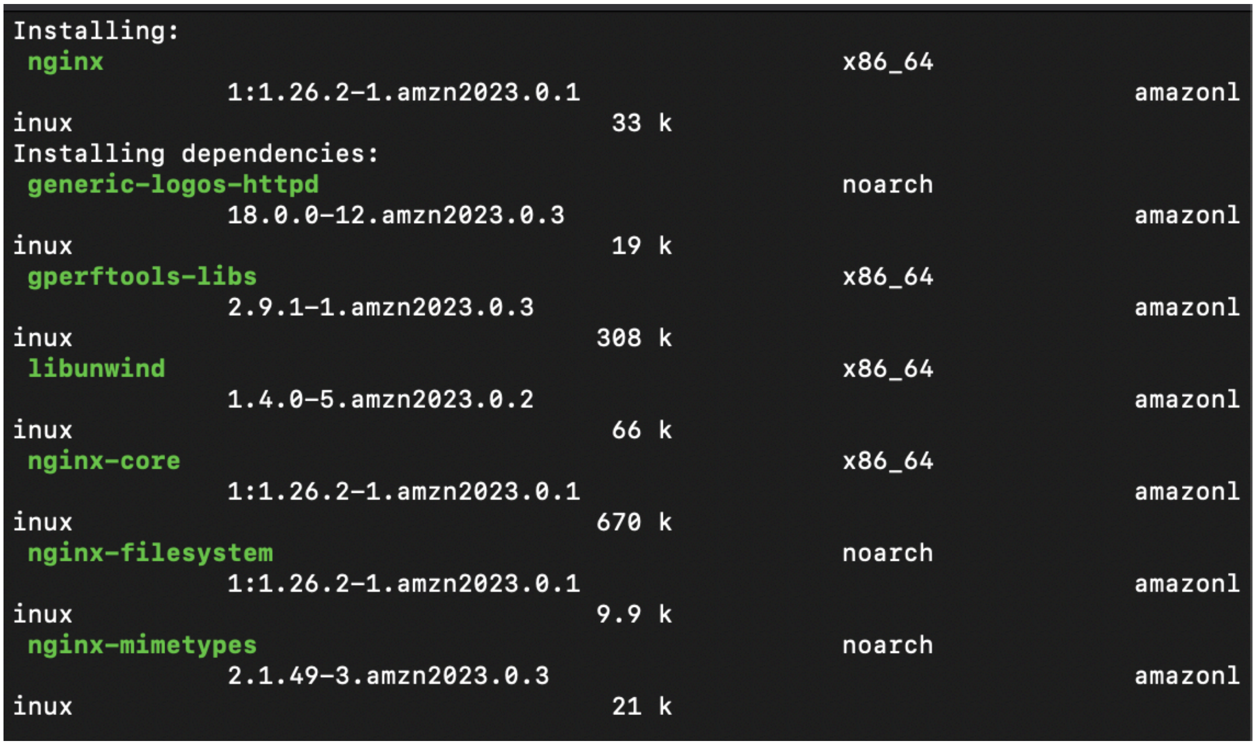
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

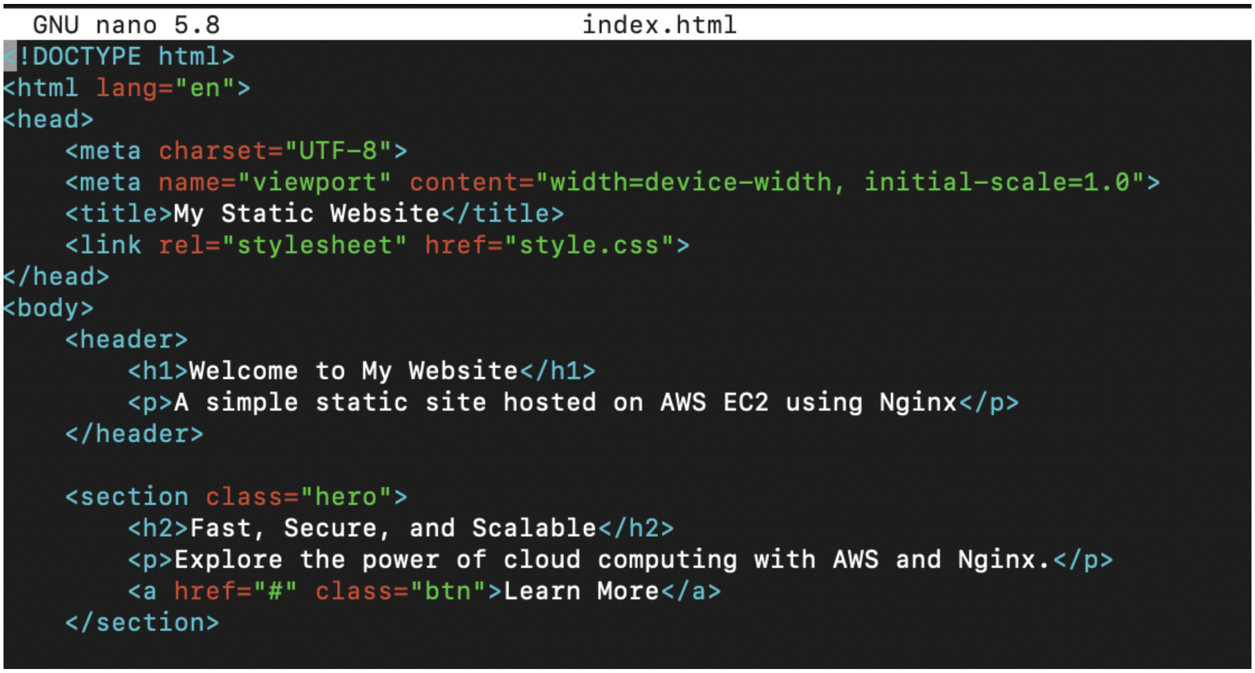
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which



```
[bishalranjitkar@Bishals-MacBook-Air ~ % cd Desktop
[bishalranjitkar@Bishals-MacBook-Air Desktop % ssh -i "test-key.pem" ec2-user@3.8
5.230.22
The authenticity of host '3.85.230.22 (3.85.230.22)' can't be established.
ED25519 key fingerprint is SHA256:alPSDwba8JbT4OYLbbr6NYx8t3wfjhaaKXtXNnkJJAU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.85.230.22' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-1-77 ~]$ 
```

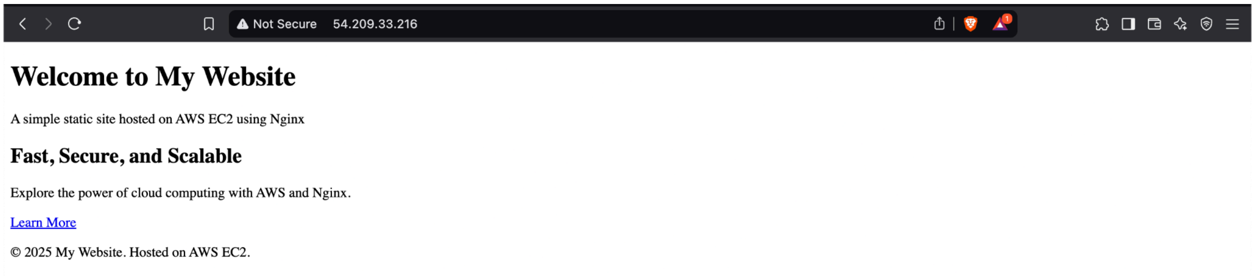


```
Installing:
 nginx                                                 x86_64
              1:1.26.2-1.amzn2023.0.1                                    amazonl
inux                                     33 k
Installing dependencies:
 generic-logos-httpd                                   noarch
              18.0.0-12.amzn2023.0.3                                     amazonl
inux                                     19 k
 gperftools-libs                                       x86_64
              2.9.1-1.amzn2023.0.3                                       amazonl
inux                                    308 k
 libunwind                                             x86_64
              1.4.0-5.amzn2023.0.2                                       amazonl
inux                                     66 k
 nginx-core                                            x86_64
              1:1.26.2-1.amzn2023.0.1                                    amazonl
inux                                    670 k
 nginx-filesystem                                      noarch
              1:1.26.2-1.amzn2023.0.1                                    amazonl
inux                                    9.9 k
 nginx-mimetypes                                       noarch
              2.1.49-3.amzn2023.0.3                                      amazonl
inux                                     21 k
```



```
  GNU nano 5.8                                index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Static Website</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Welcome to My Website</h1>
        <p>A simple static site hosted on AWS EC2 using Nginx</p>
    </header>

    <section class="hero">
        <h2>Fast, Secure, and Scalable</h2>
        <p>Explore the power of cloud computing with AWS and Nginx.</p>
        <a href="#" class="btn">Learn More</a>
    </section>
```



Welcome to My Website

A simple static site hosted on AWS EC2 using Nginx

Fast, Secure, and Scalable

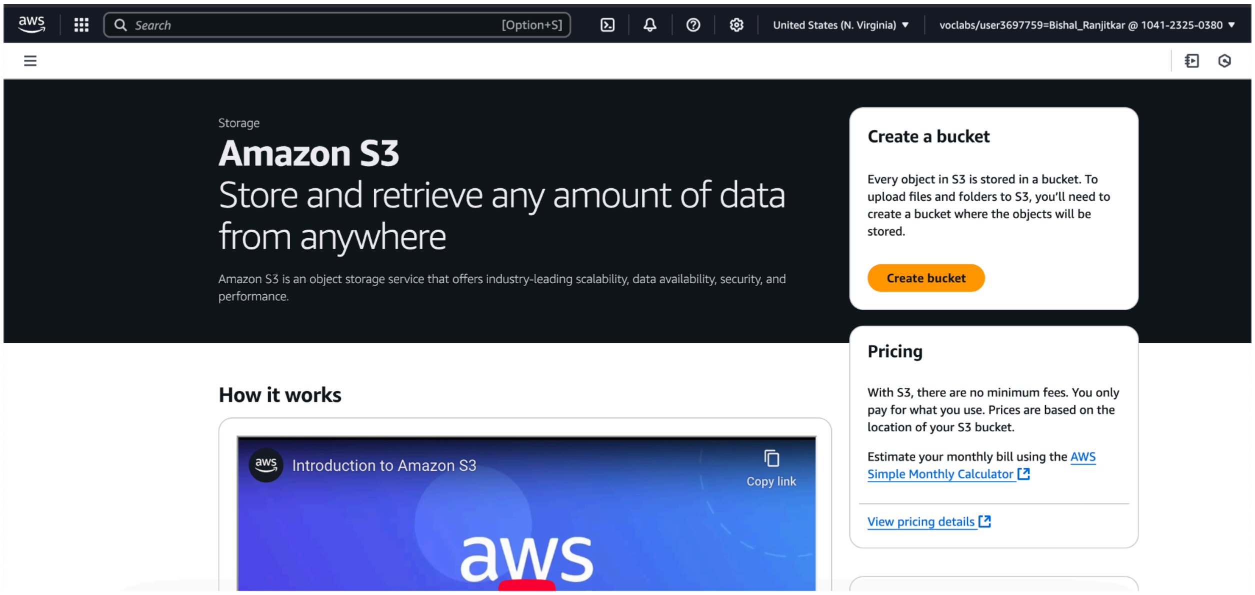
Explore the power of cloud computing with AWS and Nginx.

Learn More

© 2025 My Website. Hosted on AWS EC2.

3. Hosting a Static Website on S3

- Created an S3 bucket with a unique name.
- Uploaded the index .html file to the bucket.
- Enabled static website hosting in the bucket settings.
- Configured the Permissions:
- Used the Bucket Policy Generator to create a policy allowing public access.
- Applied the policy to the bucket.
- Obtained the S3 website endpoint URL and accessed the static site in the browser.


aws
Search
[Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
Introduction to Amazon S3
Copy link
aws

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

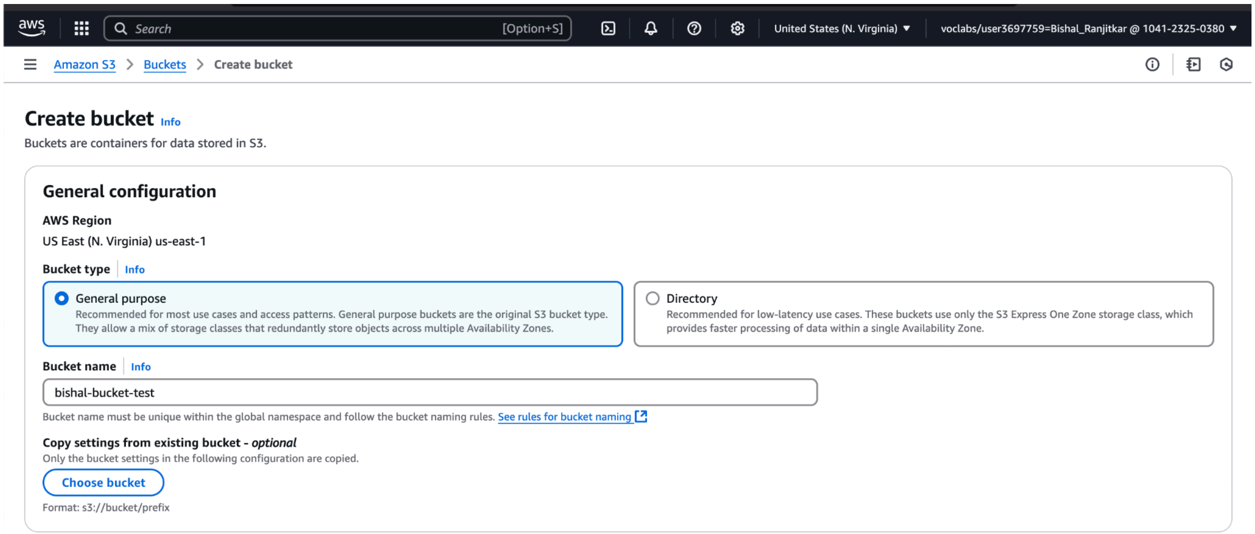
Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

View pricing details

How it works


aws
Search [Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
Amazon S3 > Buckets > Create bucket

Create bucket

Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type | Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name | Info

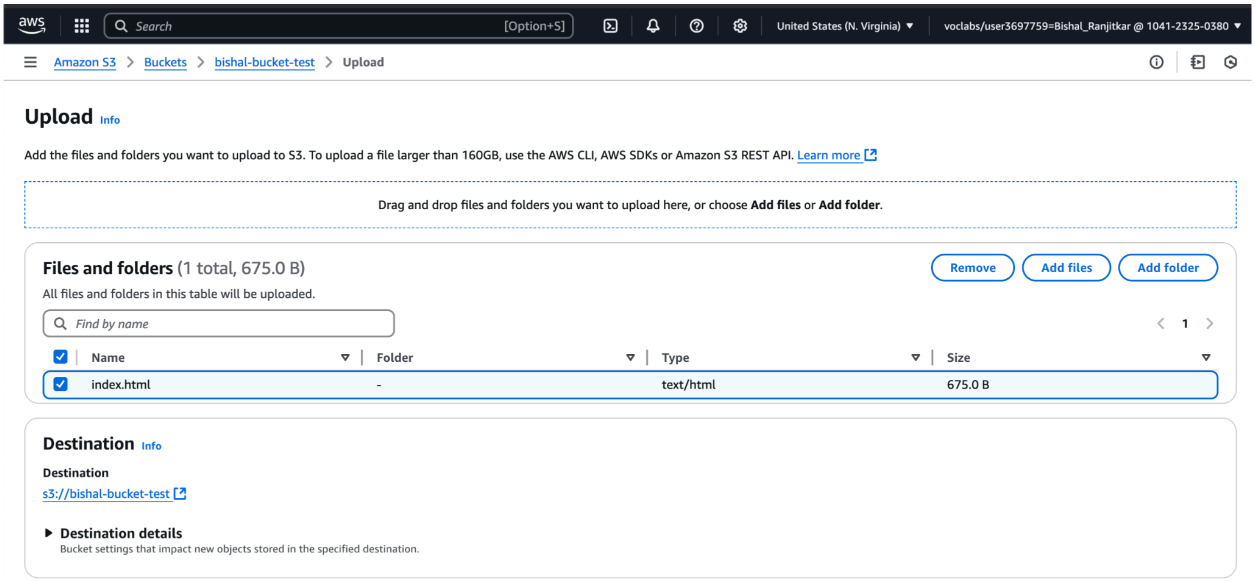
bishal-bucket-test

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix


aws
Search [Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
Amazon S3 > Buckets > bishal-bucket-test > Upload

Upload

Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 total, 675.0 B)

Remove
Add files
Add folder

All files and folders in this table will be uploaded.

☑	Name	Folder	Type	Size
☑	index.html	-	text/html	675.0 B

Destination

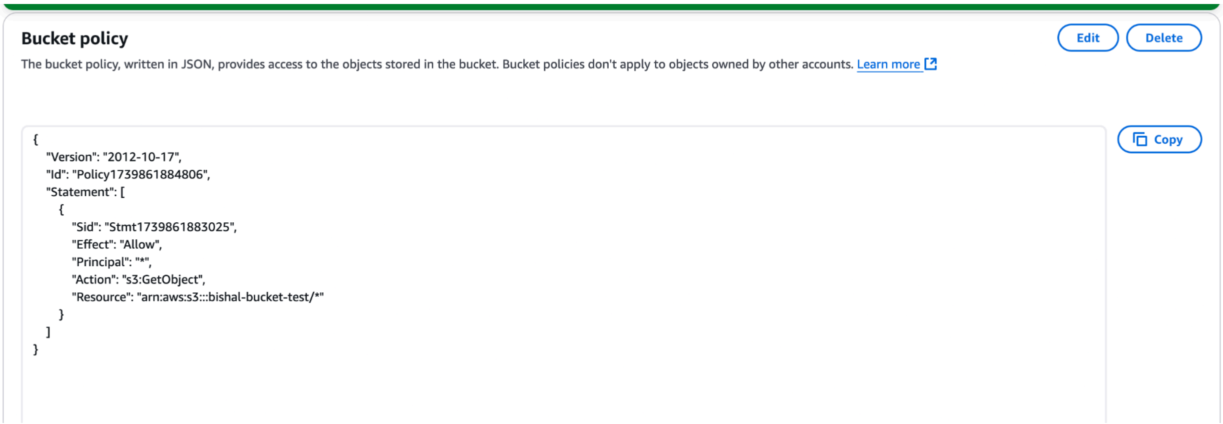
Info

Destination

s3://bishal-bucket-test

▶ **Destination details**

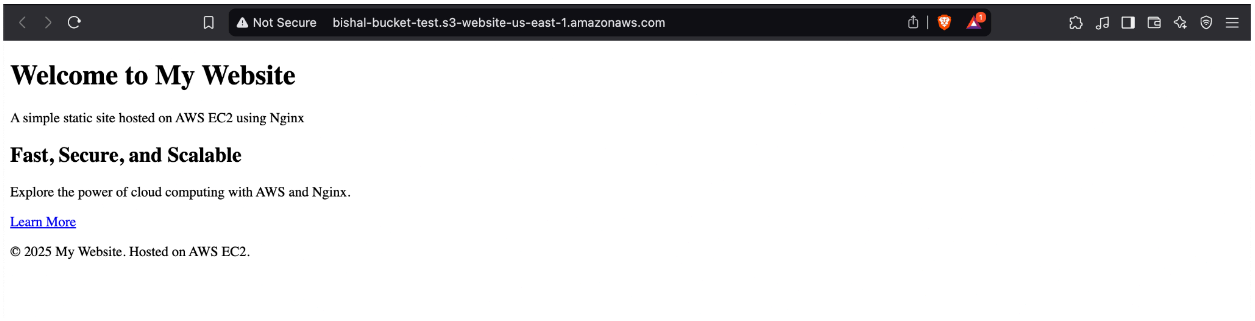
Bucket settings that impact new objects stored in the specified destination.



Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

```
{
  "Version": "2012-10-17",
  "Id": "Policy1739861884806",
  "Statement": [
    {
      "Sid": "Stmt1739861883025",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::bishal-bucket-test/*"
    }
  ]
}
```



Welcome to My Website

A simple static site hosted on AWS EC2 using Nginx

Fast, Secure, and Scalable

Explore the power of cloud computing with AWS and Nginx.

Learn More

© 2025 My Website. Hosted on AWS EC2.

Outcome

By the end of this project, I successfully deployed a static website on both EC2 and S3. The EC2 setup used Nginx as a web server, while the S3 setup leveraged AWS's built-in static website hosting feature. This project helped me understand networking, security configurations, and web hosting on AWS.

Future Improvements

- Implement **CloudFront** for content delivery optimization.
- Use **Route 53** to set up a custom domain.
- Automate deployment using **Terraform** or **AWS CLI**.