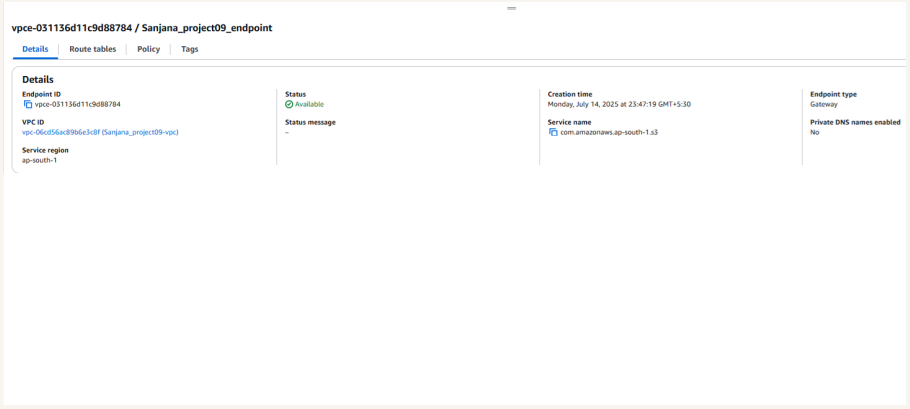
nextwork.org

VPC Endpoints

Sanjana Tripathy



Sanjana Tripathy
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) lets you create a logically isolated network within AWS. It allows full control over networking, including IP ranges, subnets, route tables, and gateways—enabling secure, customizable, and scalable cloud infrastruct

How I used Amazon VPC in this project

In today’s project, I used Amazon VPC to create an isolated network environment where I launched an EC2 instance and securely connected it to an S3 bucket using a VPC endpoint, ensuring all traffic remained within the AWS private network.

One thing I didn't expect in this project was...

One thing I didn’t expect in this project was how straightforward yet powerful VPC endpoints are for enabling secure access to S3. I also didn’t anticipate needing to troubleshoot issues like route table misconfiguration.

Sanjana Tripathy
NextWork Student

This project took me...

This project took me approximately 1 hour.

In the first part of my project...

Step 1 - Architecture set up

In this step I will set up the foundations by: 1.Creating a VPC from scratch. 2.Launching an EC2 instance and later connecting with it using "instance connect". 3.Creating a S3 bucket.

Step 2 - Connect to EC2 instance

In this step, I'll connect to my EC2 instance and access the S3 bucket over the public internet. This serves as a baseline to demonstrate how S3 access works without a VPC endpoint, highlighting the need for secure, private connectivity.

Step 3 - Set up access keys

In this step, I'll grant my EC2 instance access to the AWS environment by creating access keys. These credentials will allow the instance to authenticate and securely interact with AWS services like S3 during the testing phase.

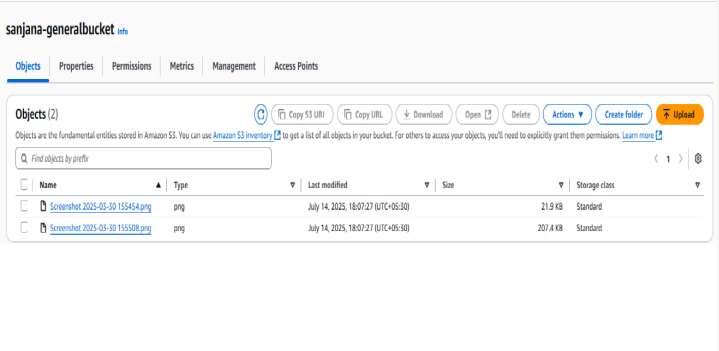
Step 4 - Interact with S3 bucket

In this step, I'll enable interaction between my EC2 instance and the S3 bucket. This involves using the AWS CLI on the instance to check if it can successfully access and list the contents of the bucket—verifying connectivity and permissions.

Architecture set up

I began the project by quickly creating a new VPC using the VPC Wizard, which set up the essential components in under 2 minutes. After that, I launched an EC2 instance within the VPC .

I set up an S3 bucket and uploaded two objects to it. This bucket will be used to test secure access from my VPC using a VPC endpoint, ensuring data transfer remains within the AWS network and doesn't traverse the public internet.



Sanjana Tripathy
NextWork Student

Access keys

Credentials

To enable my EC2 instance to interact with the AWS environment,I ran aws configure,which sets up the AWS CLI by prompting for the Access Key,Secret Key,and region.Using the previously generated credentials, the configuration was successfully complete

Access keys are a set of security credentials (Access Key ID and Secret Access Key) used to authenticate programmatic access to AWS services. In this project, they enable my EC2 instance to interact with S3 during testing and validation.

Secret access keys are sensitive credentials used with access key IDs to securely authenticate API requests to AWS services. They must be kept confidential, as anyone with access can potentially control your AWS resources.

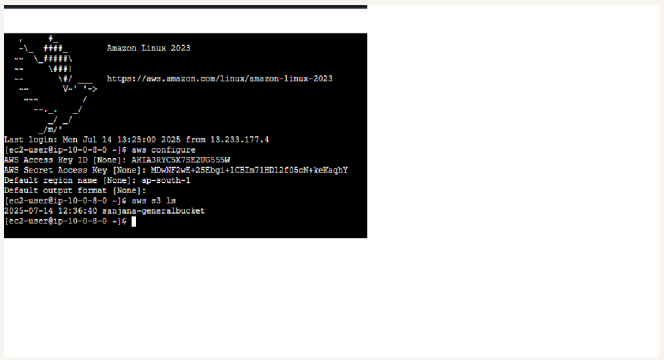
Best practice

Although I'm using access keys in this project, a best practice alternative is to create an IAM role with necessary permissions and attach it to the EC2 instance—this enhances security by avoiding hardcoded credentials.

Connecting to my S3 bucket

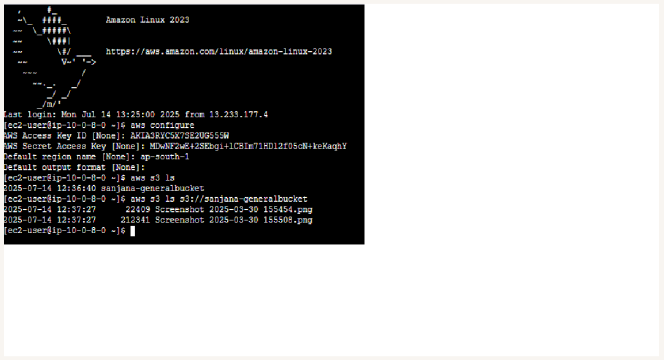
The command I ran was "AWS S3 ls". This command is used to list all the S3 buckets in a account

The terminal responded with my S3 bucket name after I wrote the command "aws s3 ls".This indicated that the AWS CLI was correctly configured and my EC2 instance successfully connected to the S3 service using the access credentials I provided.



Connecting to my S3 bucket

I also tested another command: aws s3 ls s3://<my-s3-bucket-name>, where I replaced the placeholder with my actual bucket name, which returned a list of all objects stored in my S3 bucket, confirming successful access.

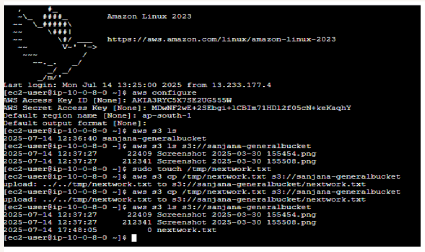


Uploading objects to S3

To upload a new file to my bucket, I first ran the command sudo touch /tmp/test.txt. This command creates an empty file named test.txt in the /tmp directory, preparing it for upload to the S3 bucket in the next step.

The second command I ran was aws s3 cp /tmp/test.txt s3://nextwork-vpc-projectyourname, replacing the placeholder with my actual S3 bucket name. This command uploads the test.txt file from my EC2 instance to the specified S3 bucket.

The third command I ran was aws s3 ls s3://nextwork-vpc-project-yourname, replacing the placeholder with my actual bucket name.This command validated that the file was successfully uploaded by listing the contents (objects)of the specified S3 bucket.



In the second part of my project...

Step 5 - Set up a Gateway

In this step, I'll create a VPC endpoint to establish a secure, private connection between my VPC and Amazon S3. This setup ensures that all traffic stays within the AWS network, eliminating the need for public internet access.

Step 6 - Bucket policies

In this step,I'm securing the S3 bucket by blocking all access except through the VPC endpoint using bucket policy to ensure that only traffic originating from my VPC can reach the bucket,verifying that communication occurs privatelly.

Step 7 - Update route tables

In this step, I'll test whether my EC2 instance can still access the S3 bucket after applying the restrictive bucket policy. This validates the VPC endpoint setup and helps identify any misconfigurations that may be blocking private connectivity.

Step 8 - Validate endpoint conection

In this step, I'll re-test the VPC endpoint by accessing the S3 bucket from my EC2 instance to confirm private connectivity. Additionally, I'll begin restricting my VPC's access to the broader AWS environment for enhanced security.

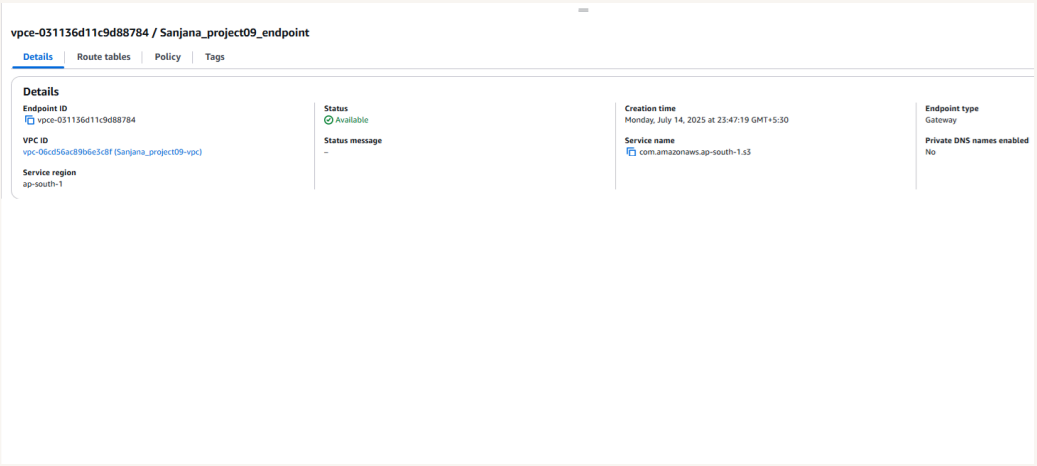
Sanjana Tripathy
NextWork Student

Setting up a Gateway

I set up an S3 Gateway,which is type of VPC endpoint designed for Amazon S3 and DynamoDB. It adds a route to the VPC route table, enabling secure, direct communication with these services without requiring an internet gateway or NAT device.

What are endpoints?

An endpoint in AWS enables private connectivity between your VPC and supported AWS services without traversing the public internet. It enhances security, performance, and control by routing traffic through the AWS network.

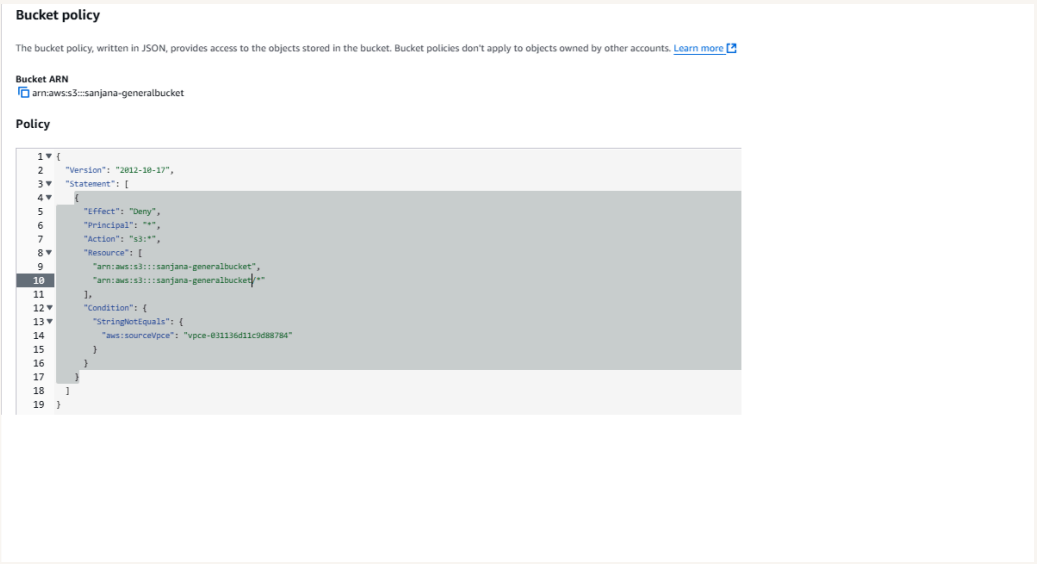


Sanjana Tripathy
NextWork Student

Bucket policies

A bucket policy is a JSON-based access control mechanism that defines permissions for an Amazon S3 bucket. It specifies who can access the bucket, what actions are allowed or denied, and under what conditions, enhancing security and control.

My bucket policy will deny all S3 access unless the request comes from the specific VPC endpoint vpce-031136d11c9d88784.This enforces private-only access to my S3 bucket via my VPC.

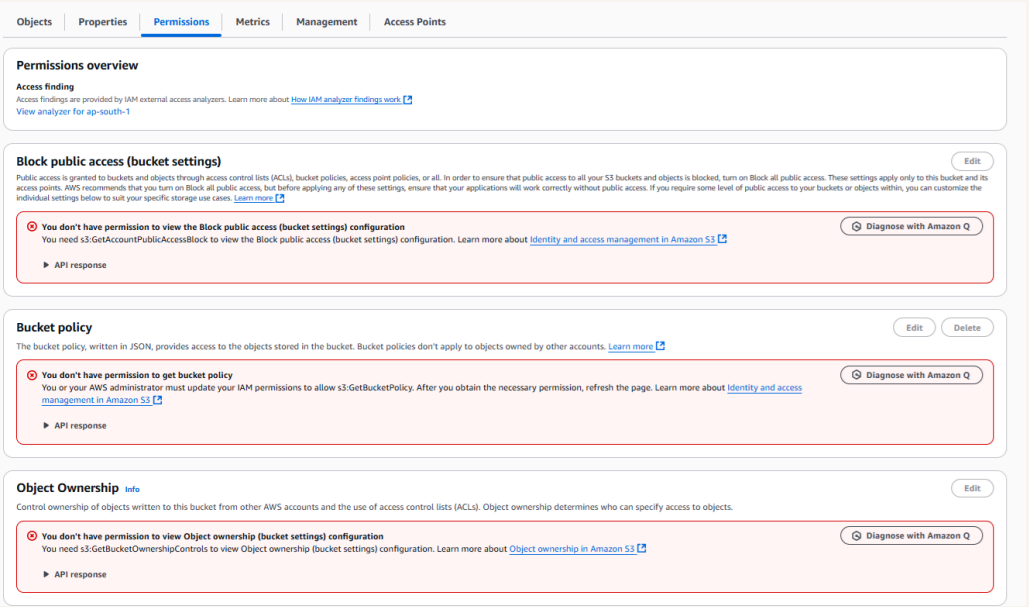


Sanjana Tripathy
NextWork Student

Bucket policies

After saving the bucket policy, the S3 console displayed an 'Access Denied' warning. This occurred because the policy blocks all access unless it originates from the specified VPC endpoint—blocking even console access outside the VPC.

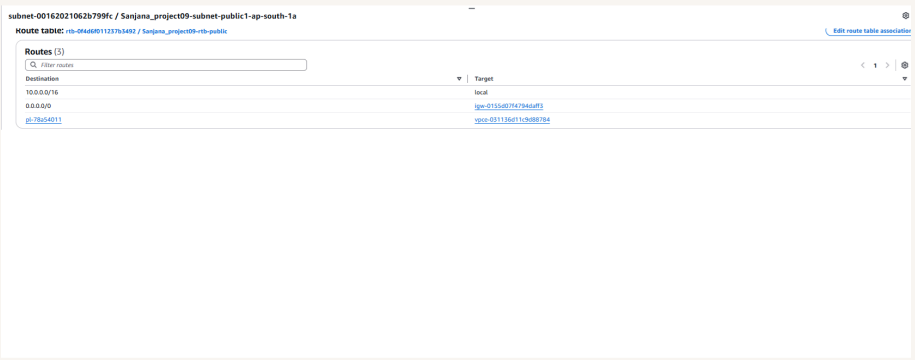
Adding this route was essential, as initial attempts to access S3 from the EC2 instance via the AWS CLI resulted in 'Access Denied' errors. This occurred because the route to the VPC endpoint was missing from the route table, preventing private acces



Route table updates

To update my route table, I selected Endpoints from the left navigation panel, chose my VPC endpoint, and clicked Manage route tables. I then attached my public route table, ensuring traffic from the subnet is routed through the endpoint to S3.

After updating my public subnet's route table to include the VPC endpoint, I ran the command aws s3 ls from my EC2 instance. The command returned my S3 bucket contents, confirming successful private connectivity through the endpoint.



Endpoint policies

An endpoint policy is a JSON-based resource policy attached to a VPC endpoint. It defines which AWS services and actions can be accessed through the endpoint, providing fine-grained control over traffic flowing from the VPC to supported AWS services.

I updated my endpoint's policy by editing it from the Endpoints tab and changing the effect from "Allow" to "Deny." I saw the impact immediately, as my EC2 instance could no longer access S3, confirming the policy change was successfully applied.

The place to learn & showcase your skills

Check out nextwork.org for more projects