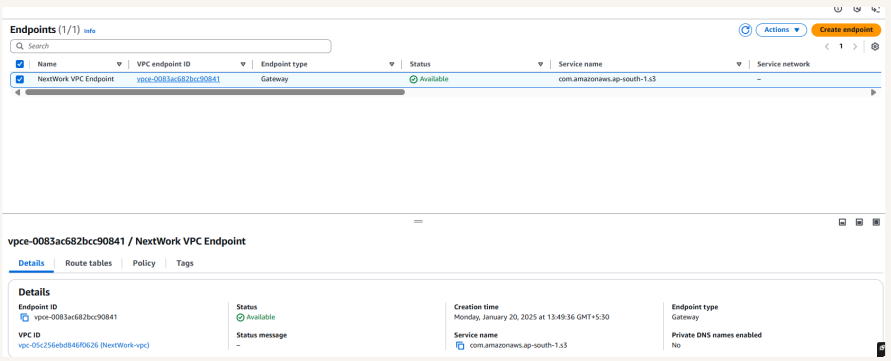
NextWork.org

VPC Endpoints

DI dineshrajdhanapathy@gmail.com



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a network that lets you create isolated resources in AWS. It’s useful because it allows secure, private connections to services like S3 via VPC endpoints, ensuring data privacy.

How I used Amazon VPC in this project

In this project, I used Amazon VPC to create a secure network for my EC2 instance and S3 bucket. By setting up VPC endpoints, I enabled private, direct communication between EC2 and S3, enhancing security.

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

One thing I didn’t expect was the need to update the route tables and endpoint policies multiple times. These steps were crucial for ensuring secure, direct communication between EC2 and S3 via the VPC.

This project took me...

This project took me about 1 - 2 hours to complete. The majority of the time was spent setting up the VPC, configuring endpoints, and troubleshooting connectivity to ensure secure communication with S3.

In the first part of my project...

Step 1 - Architecture set up

We're creating a VPC, launching an EC2 instance, and setting up an S3 bucket. This sets the foundation for private communication between resources in the VPC and S3, enhancing security and control.

Step 2 - Connect to EC2 instance

We'll connect to the EC2 instance via EC2 Instance Connect and test accessing S3 through the public internet. This verifies the default setup before enabling private communication via the VPC endpoint.

Step 3 - Set up access keys

In this step, we're assigning an IAM role to the EC2 instance. This allows it to securely access AWS services like S3 without requiring manual credentials, enhancing security and ease of use.

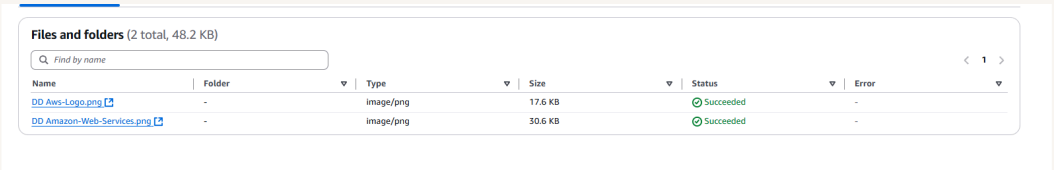
Step 4 - Interact with S3 bucket

In this step, we're configuring our EC2 instance's IAM role to allow access to the S3 bucket. This enables the EC2 instance to securely interact with S3 for storage tasks without using access keys.

Architecture set up

I started my project by launching a new VPC, an EC2 instance within it for compute tasks, and an S3 bucket for storage. These resources set the stage for secure, private communication and data handling

I also set up subnets, an internet gateway, route tables for the VPC, and security groups to ensure proper networking and access for the EC2 instance while preparing for private connectivity to S3.



Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured an IAM role with S3 access, attached the role to the EC2 instance, verified permissions, and ensured proper security groups.

Access keys are credential pairs consisting of an access key ID and a secret access key. They authenticate requests to AWS services from applications, CLI tools, or SDKs. Secure them carefully

Secret access keys are private parts of an access key pair used to sign AWS API requests. They work with access key IDs to authenticate and authorize actions in your AWS account. Keep them secure!

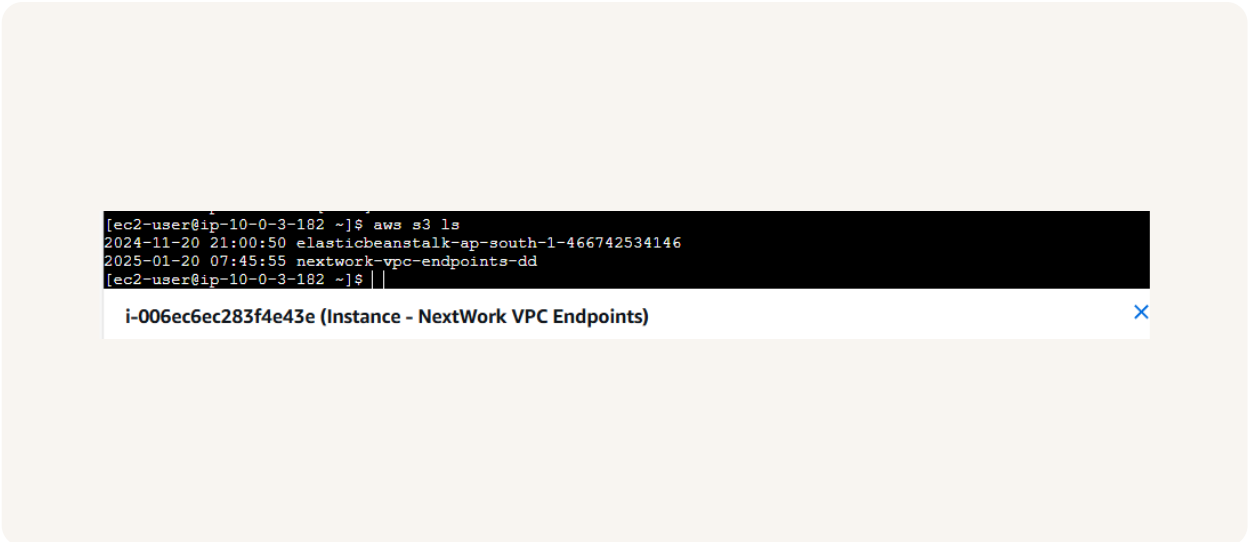
Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM roles. They provide temporary credentials automatically, improving security and reducing the risk of key exposure.

Connecting to my S3 bucket

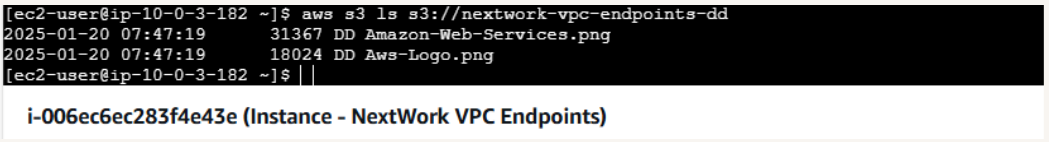
The command I ran was aws s3 ls. This command is used to list the S3 buckets in your AWS account. However, it failed because no AWS credentials were configured on the EC2 instance.

The terminal responded with a list of S3 buckets or an error message indicating access permissions. This indicated that the access keys I set up were either correctly configured.



Connecting to my S3 bucket

I also tested the command aws s3 ls, which returned a list of S3 buckets successfully. This confirmed whether my EC2 instance had S3 access.

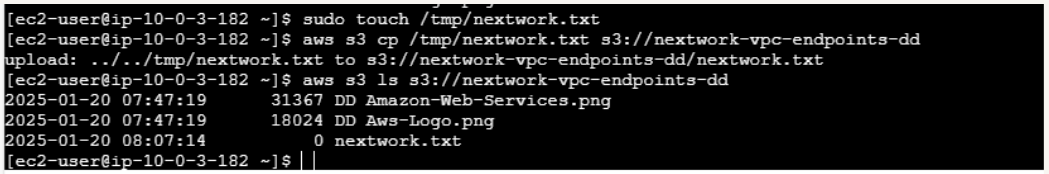


Uploading objects to S3

To upload a new file to my bucket, I first ran the command sudo touch new_text_file_name. This command creates an empty file in the EC2 instance, which can then be uploaded to S3.

The second command I ran was cp new_text_file_name s3://your-bucket-name/. This command will copy the newly created file from the EC2 instance to the specified S3 bucket folder for storage.

The third command I ran was aws s3 ls s3://your-bucket-name/. This validated that the file was successfully uploaded to the S3 bucket by listing its contents, confirming the file was in the right location.



In the second part of my project...

Step 5 - Set up a Gateway

In this step, we're setting up a VPC endpoint to allow direct communication between our VPC and S3. This eliminates the need for internet traffic, improving security and performance for S3 access.

Step 6 - Bucket policies

In this step, we're configuring an S3 bucket policy to restrict access so that only traffic from the VPC endpoint can access the bucket. This ensures that no outside internet traffic can interact with it.

Step 7 - Update route tables

In this step, we're testing the VPC endpoint setup by attempting to access the S3 bucket from the EC2 instance. If there's a connectivity issue, we'll troubleshoot to ensure the endpoint is properly configured.

Step 8 - Validate endpoint conection

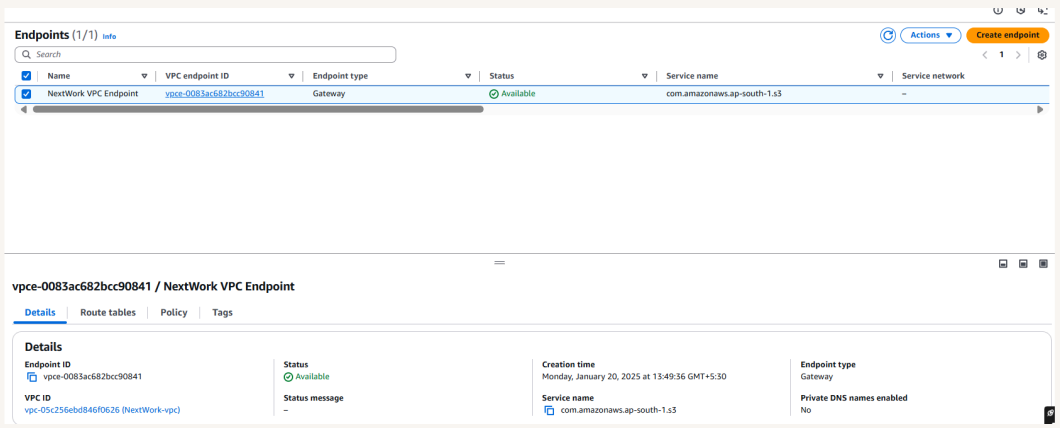
In this step, we're testing the VPC endpoint setup again to confirm that communication with S3 is functioning. Additionally, we'll restrict the VPC's access to AWS resources to enhance security.

Setting up a Gateway

I set up an S3 Gateway, which is a VPC endpoint that allows private, secure communication between my VPC and S3 without routing traffic over the internet, improving both security and performance.

What are endpoints?

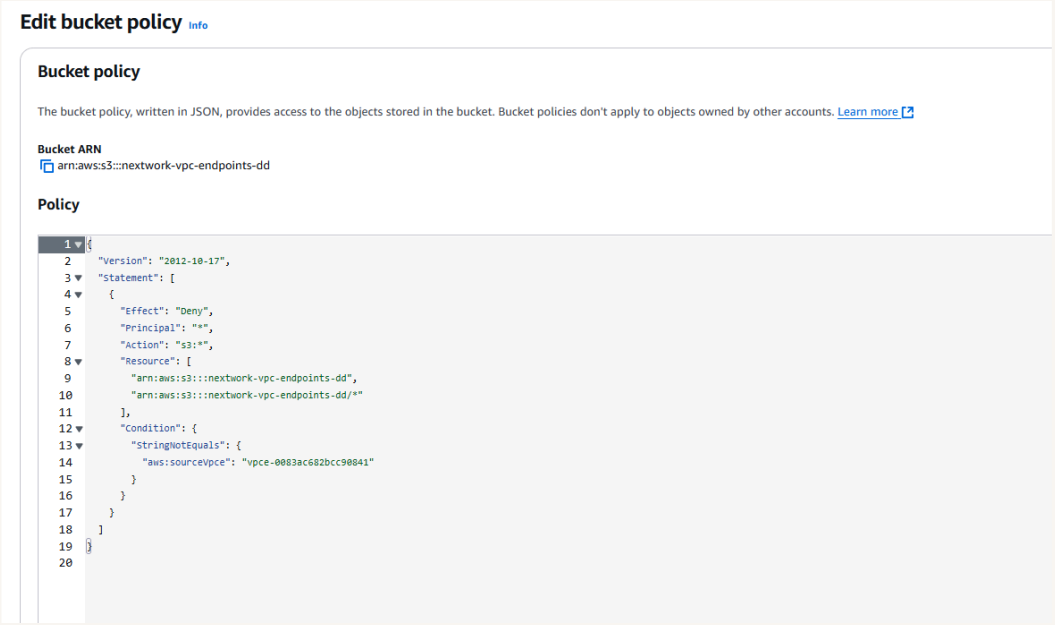
An endpoint is a network gateway that allows communication between your VPC and other AWS services, like S3. It enables private connections without internet traffic, enhancing security and performance.



Bucket policies

A bucket policy is a set of permissions attached to an S3 bucket that defines who can access the bucket and what actions they can perform. It can restrict access by IP, VPC, or other conditions.

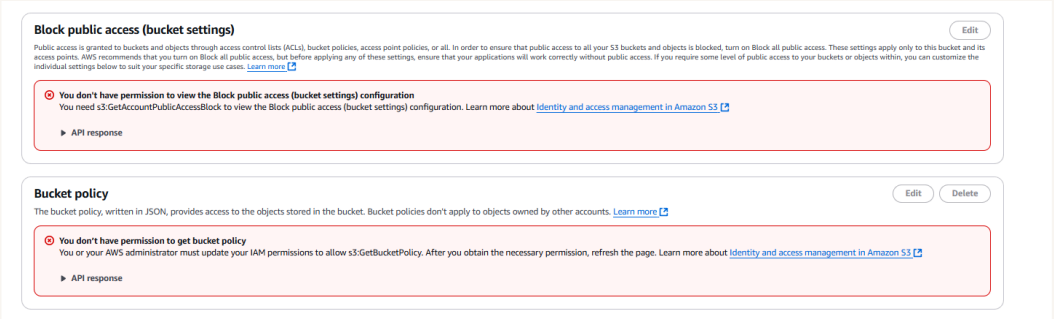
My bucket policy will restrict access to the S3 bucket, allowing only traffic from the VPC endpoint. This ensures that no external internet traffic can reach the bucket, enhancing security and control.



Bucket policies

Right after saving my bucket policy, my S3 bucket page showed "denied access" warnings. This was because the policy restricted access to only traffic from the VPC endpoint, blocking other sources.

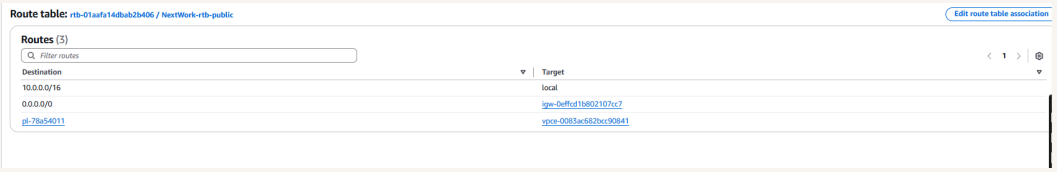
I also had to update my route table because the VPC needed a route to the S3 bucket through the VPC endpoint. Without this, traffic wouldn't be directed properly, causing connectivity issues.



Route table updates

To update my route table, I added a new route that directs traffic destined for S3 to the VPC endpoint. This ensures that requests to S3 are routed securely and privately through the endpoint.

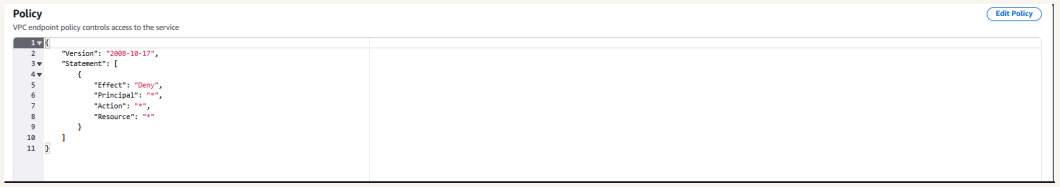
After updating my public subnet's route table, my terminal could return a successful list of S3 bucket contents. This confirmed that the VPC endpoint was properly routing traffic to S3.



Endpoint policies

An endpoint policy is a set of permissions attached to a VPC endpoint that defines which actions can be performed on the connected AWS service. It controls access to resources like S3 from the endpoint.

I updated my endpoint's policy by adding permissions to allow access only to my S3 bucket. I could see the effect of this right away, because traffic was restricted and access was granted only through the endpoint.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects