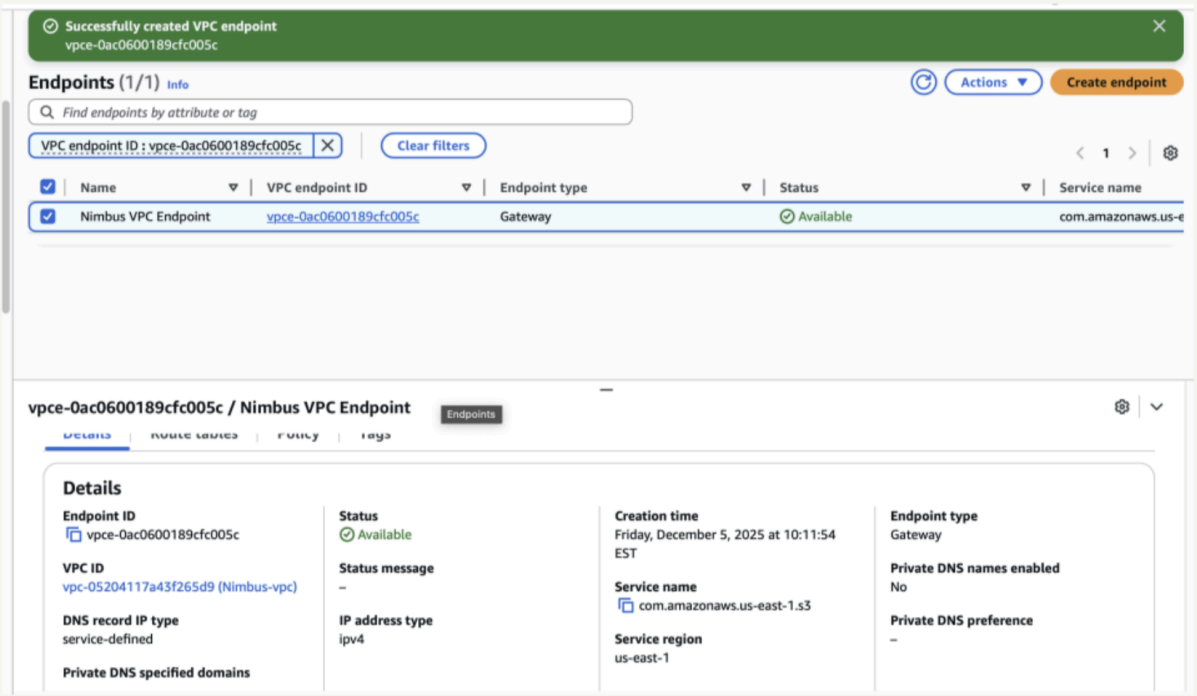
VPC Endpoints

Erik Gonzalez



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a tool in AWS that allows users to segment and isolate their network from the public internet to monitor and control their own private sector.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to create an EC2 instance, subnets, edit policies, communicate between networks, insert commands in CLI.

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

One thing I didn't expect in this project was the amountof errors I would get when editing polices. It was just a huge wall of red!

This project took me...

This project took me about two hours to complete while also taking a lunch break!

In the first part of my project...

Step 1 - Architecture set up

In this step, I will create a VPC from scratch, launch an EC2 instance, and set up an S3 bucket.

Step 2 - Connect to EC2 instance

In this step, I will connect directly to your EC2 instance.

Step 3 - Set up access keys

In this step, I will give my EC2 instance access to your AWS environment

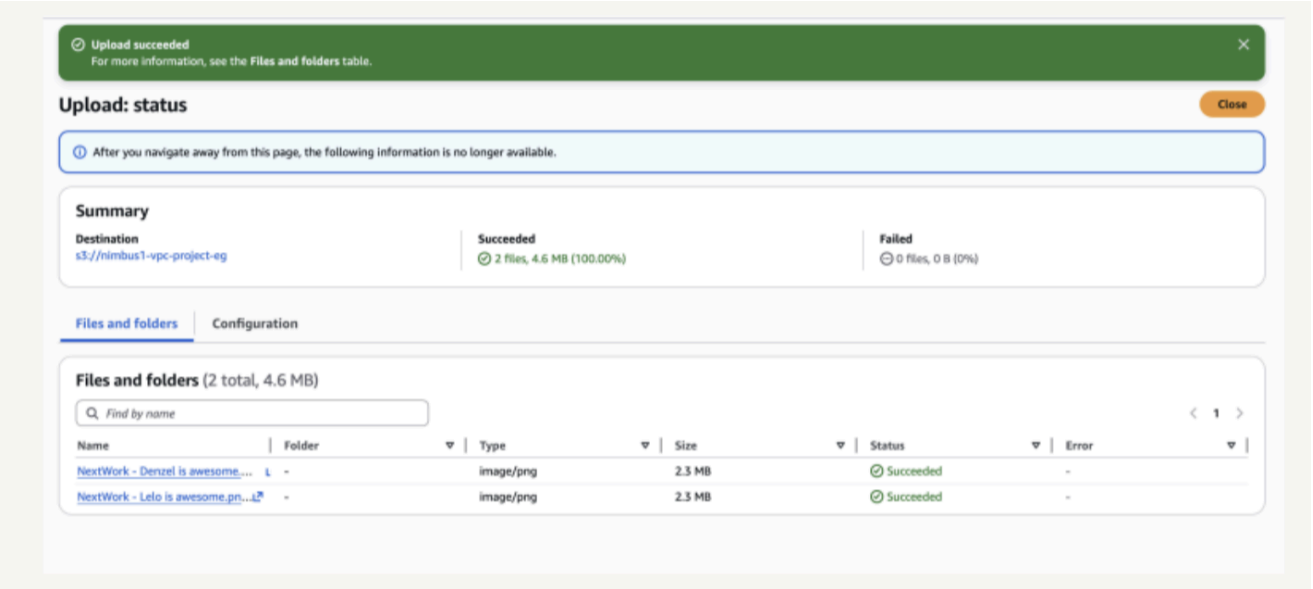
Step 4 - Interact with S3 bucket

In this step, I will head back to the EC2 instance and get the EC2 instance to access the S3 bucket.

Architecture set up

I started my project by launching VPC to start my VPC and EC2 to start a brand new Instance!

I also set up an S3 bucket with two file uploads.



Access Keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured the AWS access key ID, secret access key, default region type, and default output format.

Access keys are credentials for your applications and other servers to log into AWS and talk to your AWS services/resources.

The secret access key is like the password that pairs with your access key ID (your username). You need both to access AWS services.

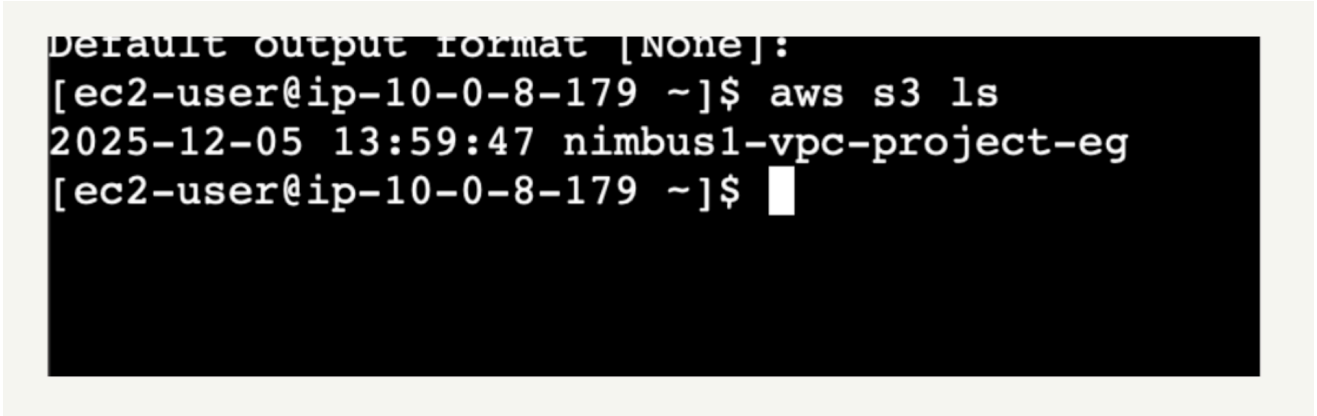
Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM admin roles. This means necessary permission will be attached to an IAM role, and then the role will be associated with the relevant resources.

Connecting to my S3 Bucket

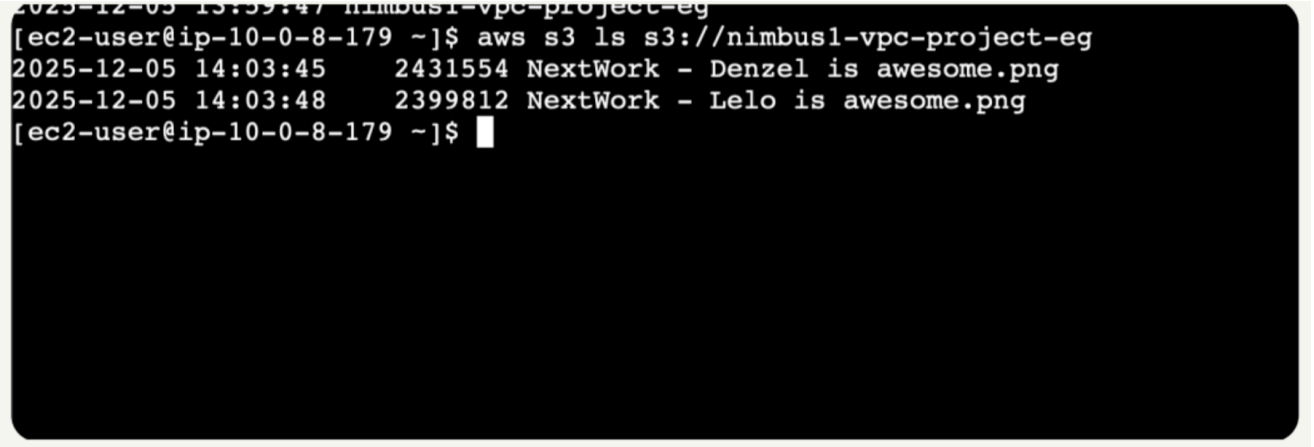
The command I ran was 'aws s3 ls' . This command is used to list all buckets in an AWS account.

The terminal responded with a list of my account's S3 bucket. This indicated that the access keys I set up correctly give my EC2 instance access to my AWS account and environment.



Connecting to my S3 Bucket

I also tested the command 'aws s3 ls s3://nimbus1-vpc-project-eg' , which returned a list of files I uploaded to my S3 bucket.



Uploading Objects to S3

To upload a new file to my bucket, I first ran the command 'sudo touch /tmp/nextwork.txt'. This command creates a .txt file.

The second command I ran was 'aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc project-eg' This command will upload the .txt file to my S3 bucket.

The third command I ran was 'aws s3 ls s3://nextwork-vpc-project-eg' which validated that the .txt file was successfully uploaded.

In the second part of my project...

Step 5 - Set up a Gateway

In this step, I will set up a way for your VPC and S3 to communicate directly.

Step 6 - Bucket policies

In this step, I will be testing my endpoint connection by blocking off all traffic to my S3 bucket, except for traffic coming from my endpoint.

Step 7 - Update route tables

In this step, I will test the VPC endpoint set up and troubleshoot a connectivity issue.

Step 8 - Validate endpoint connection

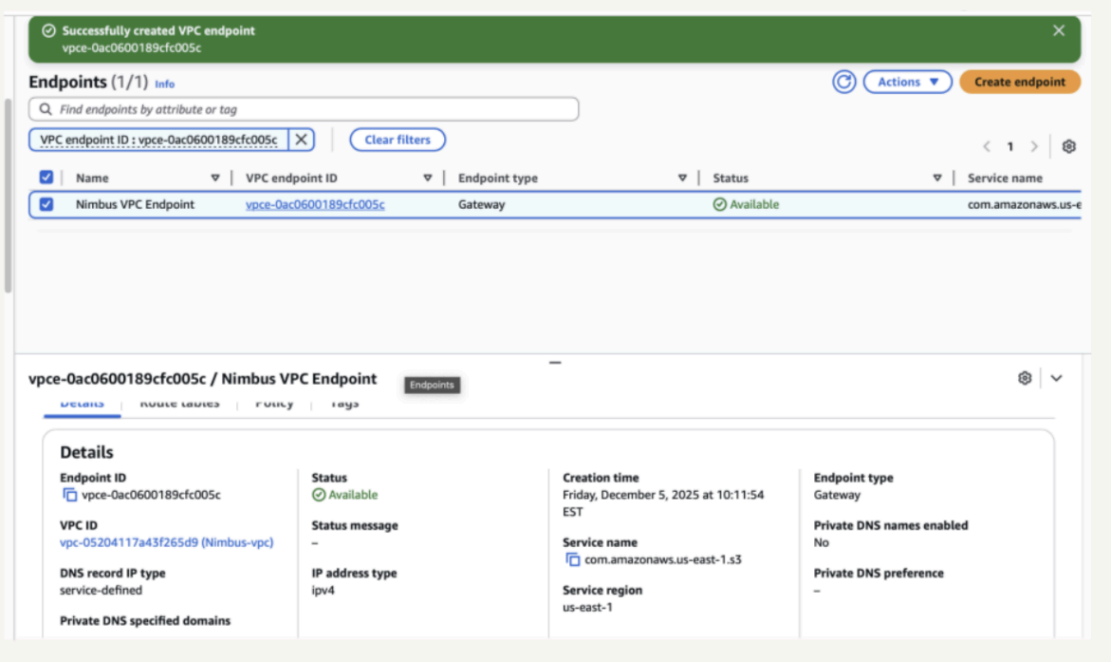
In this step, I will test my VPC endpoint set up (again) and restrict my VPC's access to my AWS environment.

Setting up a Gateway

I set up an S3 Gateway, which is a type of endpoint used specifically for Amazon S3 and DynamoDB (DynamoDB is an AWS database service). Gateways work by simply adding a route to your VPC route table that directs traffic bound for S3 or DynamoDB to head straight for the Gateway instead of the internet.

What are endpoints?

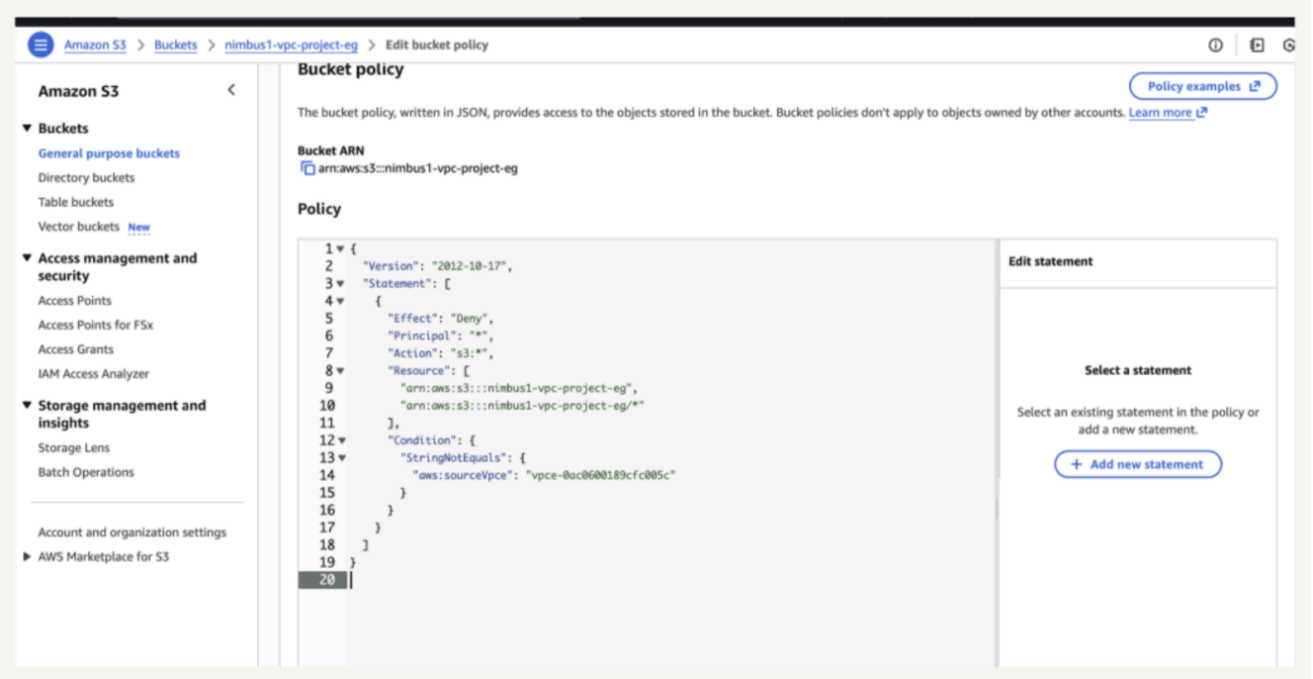
An endpoint in AWS is a service that allows private connections between your VPC and other AWS services without needing the traffic to go over the internet.



Bucket Policies

A bucket policy is a type of IAM policy designed for setting access permissions to an S3 bucket. Using bucket policies, you get to decide who can access the bucket and what actions they can perform with it.

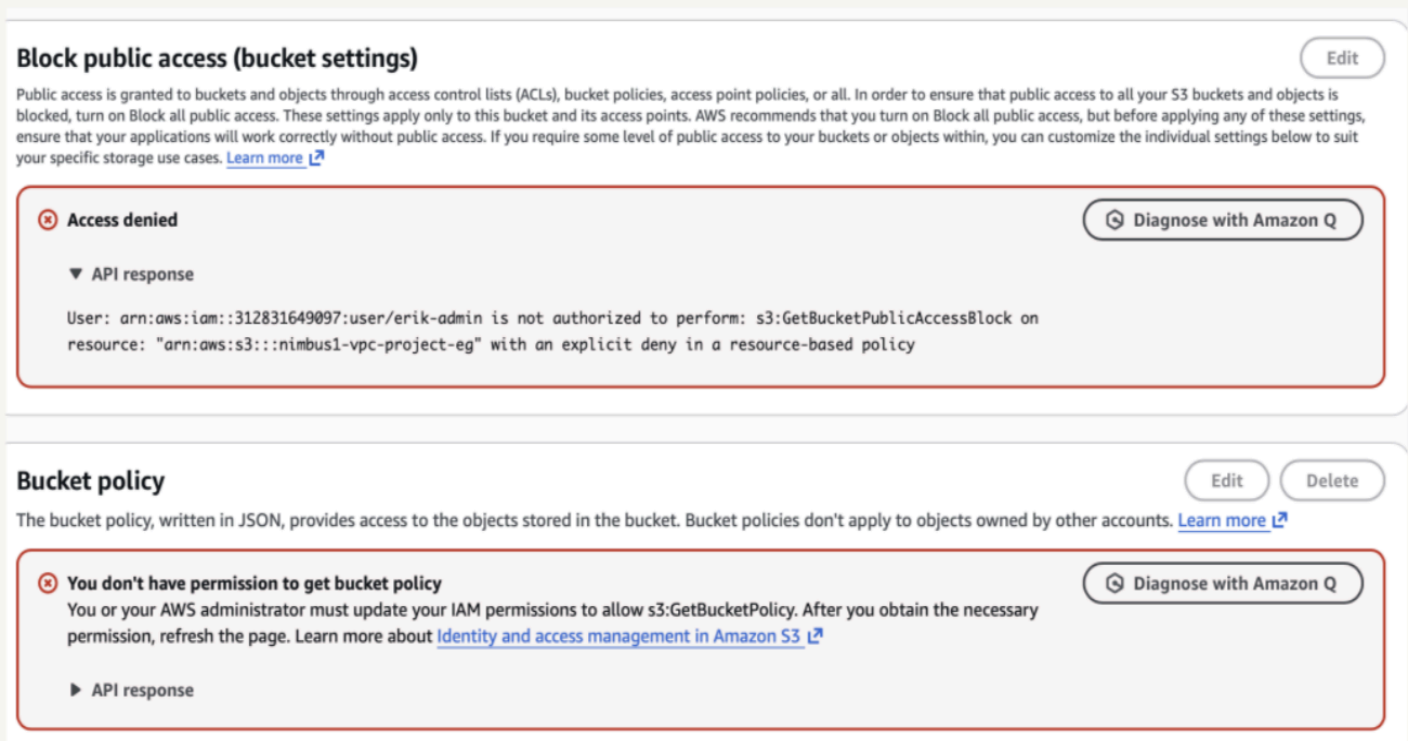
My bucket policy will deny all traffic EXCEPT from the VPC.



Bucket policies

Right after saving my bucket policy, my S3 bucket page showed 'denied access' warnings. This was because my policy denies all actions unless they come from my VPC endpoint.

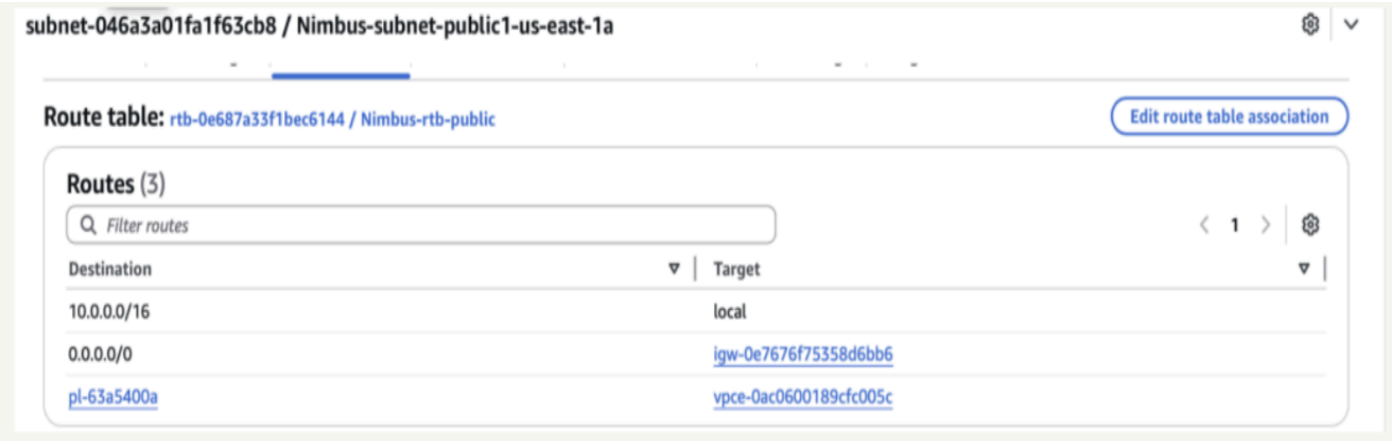
I also had to update my route table because my route table, by default, didn't provide a route for traffic in my public subnet to the VPC endpoint.



Route table updates

To update my route table, I visited the 'Endpoints' page of my VPC console and I modified the route table from there to associate my VPC's Public subnet.

After updating my public subnet's route table, my terminal could return my S3 bucket files that were uploaded.



Endpoint policies

An endpoint policy is a type of policy designed for specifying the range of resources and actions permitted by an endpoint.

I updated my endpoint's policy by changing the effect from 'allow' to 'deny'. I could see the effect of this right away, because my EC2 instance was again denied access to S3 when I tried to run another AWS S3 command.