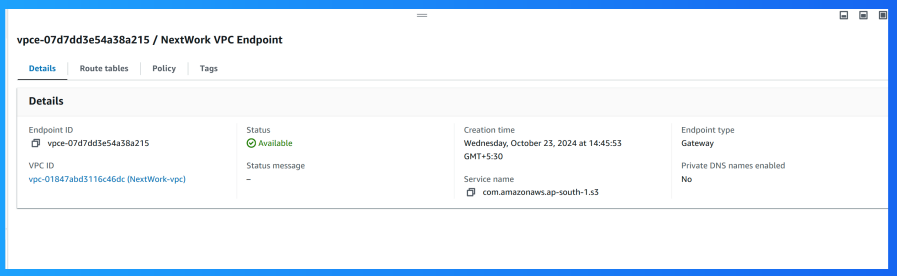
NextWork.org

VPC Endpoints

MA mallangiharinathreddy727@gmail.com



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is networking service provided by AWS and gives us the ability to isolate our resources from public internet, setup secure connections between our resources and control traffic flow/security.

How I used Amazon VPC in this project

I used Amazon VPC today to set up VPC endpoint, specifically an S3 gateway. This provides our VPC with direct, private access to another AWS service!

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

one thing I didn't expect in this project was to see my own access to the S3 bucket getting blocked once I saved my bucket's new ploicy to block all access/traffic except traffic from my Endpoint.

This project took me...

it takes nearly 2 and half hour with documentation.

In the first part of my project...

Step 1 - Architecture set up

In this step , we are setting up the foundations of this project - i.e. Launching a VPC, EC2 instance and S3 bucket so that we can set up an Endpoint architecture and test that setup in the last step of this project.

Step 2 - Connect to EC2 instance

In this step, we are connecting directly to our EC2 instance using EC2 instance connect. Connecting to our EC2 instance will help us with accessing S3 and running commands later in this project.

Step 3 - Set up access keys

In this step, we will setup an access key so that our EC2 instance will have access to our AWS environment.we can think of access keys almost like "login details" for EC2 instances/applications to interact with our AWS services.

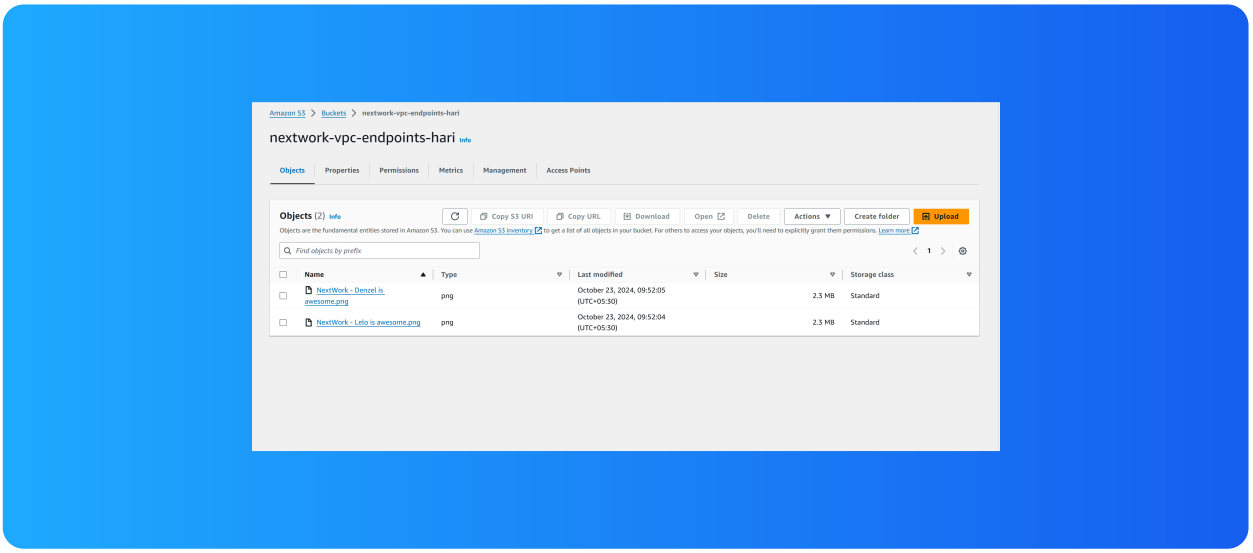
Step 4 - Interact with S3 bucket

In this step, we are applying our access key credentials to our EC2 instance, and then we are using AWS CLI and our EC2 Instance to access Amazon S3.

Architecture set up

In this step, I started my project by launching two key resources - a VPC and EC2 instance.

I also set up an S3 bucket with two files inside.



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 that matches that key ID, the default region type and then the default output format.

Access keys are credentials that an EC2 instance/other server/application would need in order to get access to our AWS environment. eg. creating resources, reading what's inside our AWS account etc.

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

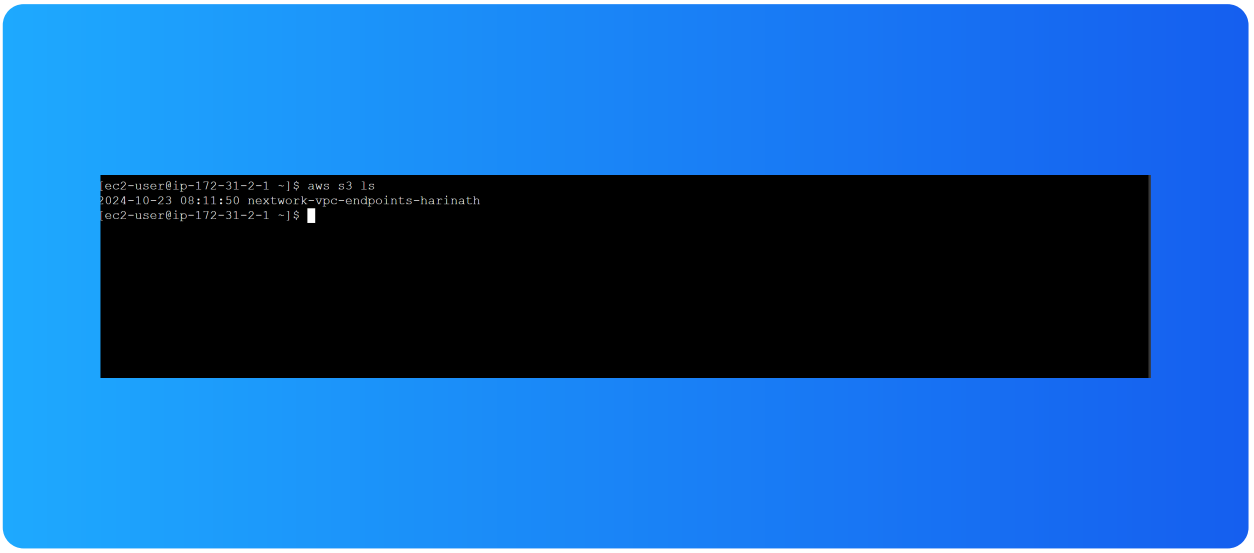
Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM amin roles instead! This means any necessary permissions will be attached to an IAM role.Then, the role will be associated with the relavant resources.

Connecting to my S3 bucket

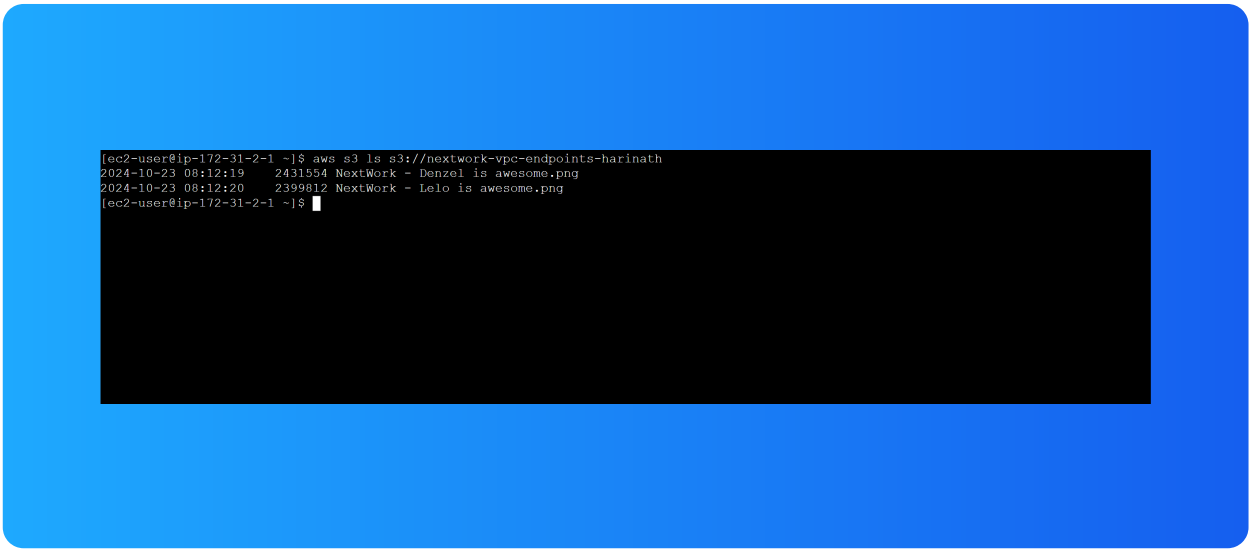
The command I ran was ' aws s3 ls '. This command is used to list all buckets in an AWS account.

The terminal responded with a list of my account's S3 buckets.This indicated that the access keys I set up correctly and can 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://nextwork-vpc-endpoints-harinath ' which returned the list of all objects inside my S3 bucket.

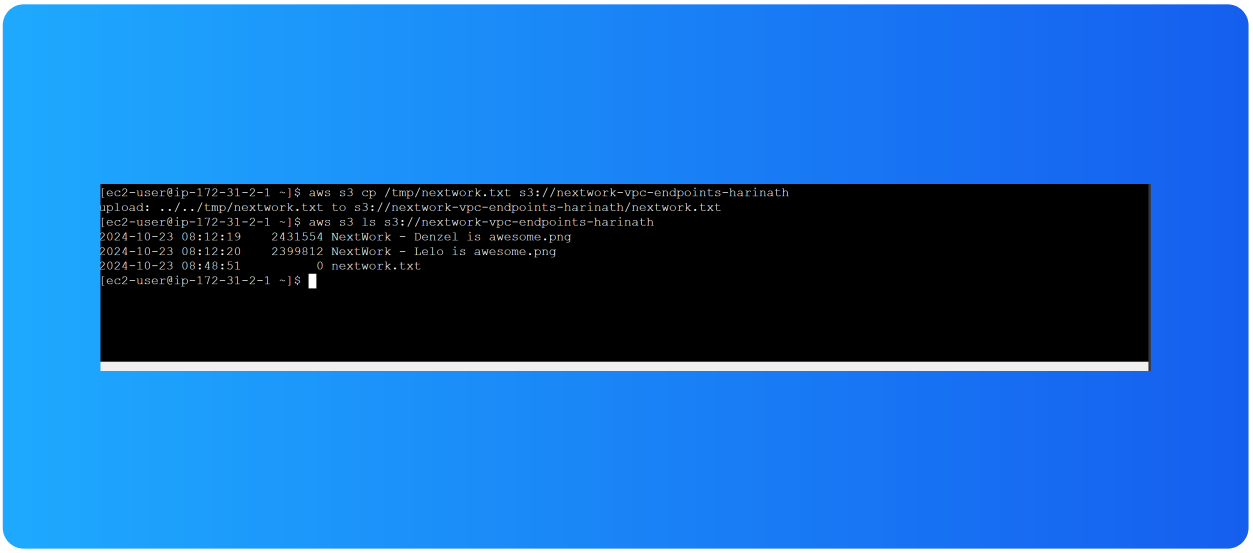


Uploading objects to S3

To upload a new file to my bucket, I first ran the command ' sudo touch /tmp/nextwork.txt ' This command creates an empty file named ' nextwork.txt ' and saves it locally in the EC2 instance.

The second command I ran was ' aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-harinath ' This command will copy the file I created i.e. nextwork.txt and upload that to my S3 bucket!

The third command I ran was ' aws s3 ls s3://nextwork-vpc-endpoints-harinath ' which validated that a new file was created and uploaded into my S3 bucket!



In the second part of my project...

Step 5 - Set up a Gateway

In this step, we are setting up a VPC Enpoint so that communication between our VPC and other services(especially S3) is direct and secure.

Step 6 - Bucket policies

In this step, we are testing our Endpoint connection by blocking off all traffic to our S3 bucket, expect for traffic coming from our Endpoint.

Step 7 - Update route tables

In this step, we are testing our endpoint connection between our bucket and EC2 instance.

Step 8 - Validate endpoint conection

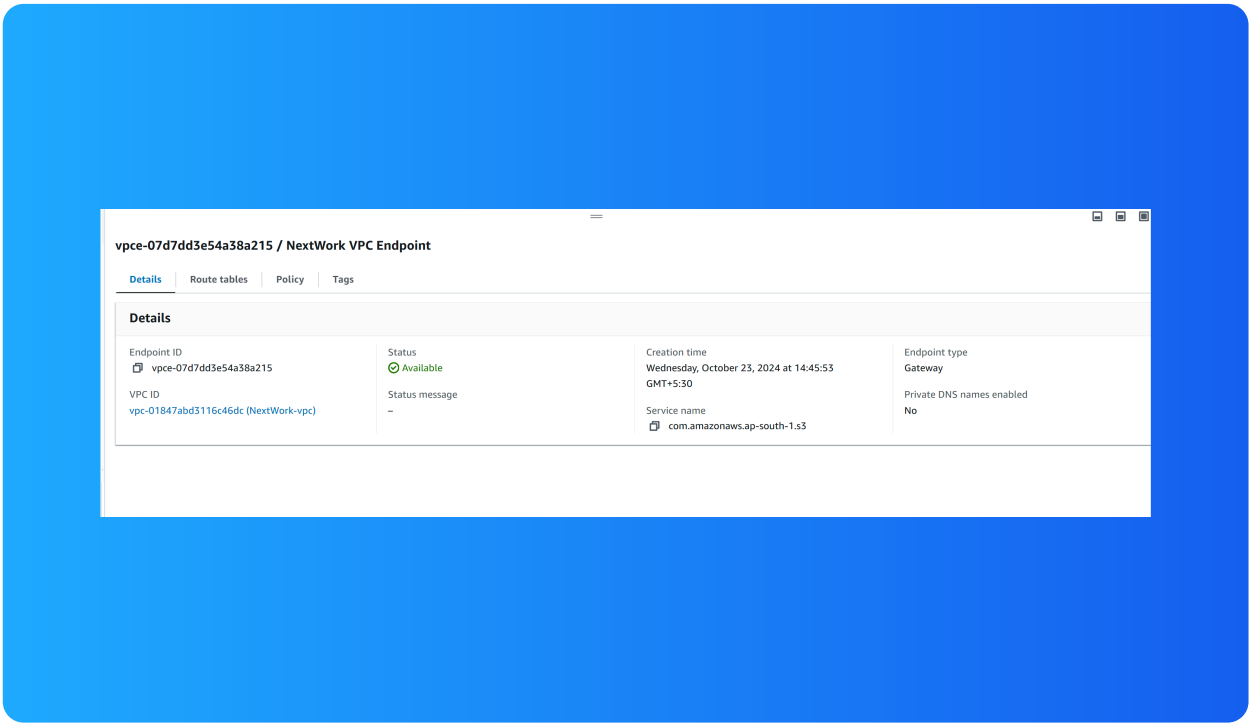
In this step, I am going to validate our VPC endpoint set up one more time. we are also going to use endpoint policies to restrict my EC2 instance's access to our AWS environment.

Setting up a Gateway

I set up an S3 Gateway, which is a type of endpoint used specifically for Amazon S3. Gateways work by simply adding a route to your VPC route table that directs traffic bound for S3.

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

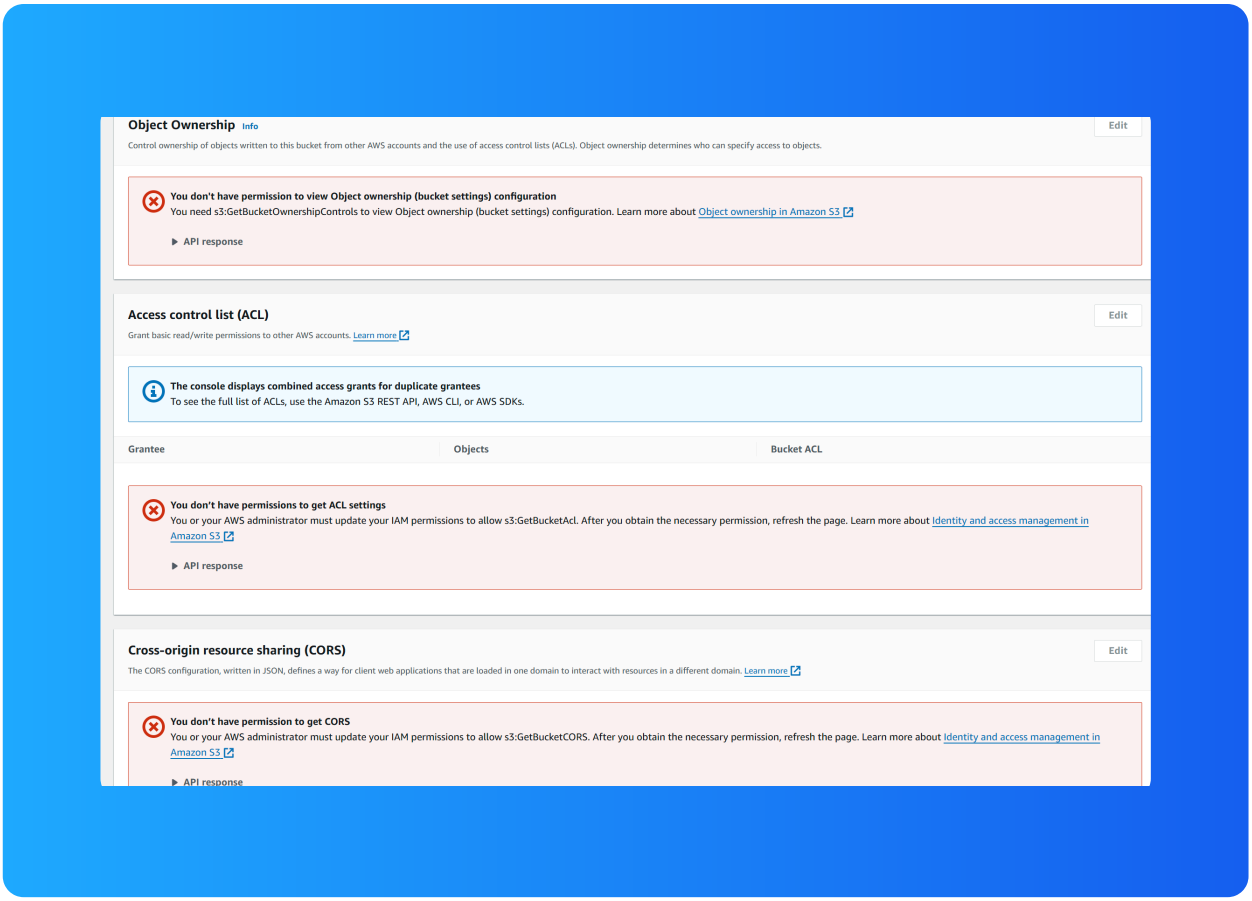
A bucket policy is a type of IAM policy designed for setting access permissions to an S3 bucket. Using bucket policies, you get to decide who can access the bucket and what actions they can perform with it.

My bucket policy will deny traffic from ALL sources- except for traffic coming from my VPC endpoint.

Bucket policies

Right after saving my bucket policy, my S3 bucket page showed 'denied access' warnings. This was because our policy denies all actions unless they come from your VPC endpoint. This means any attempt to access your bucket from other sources.

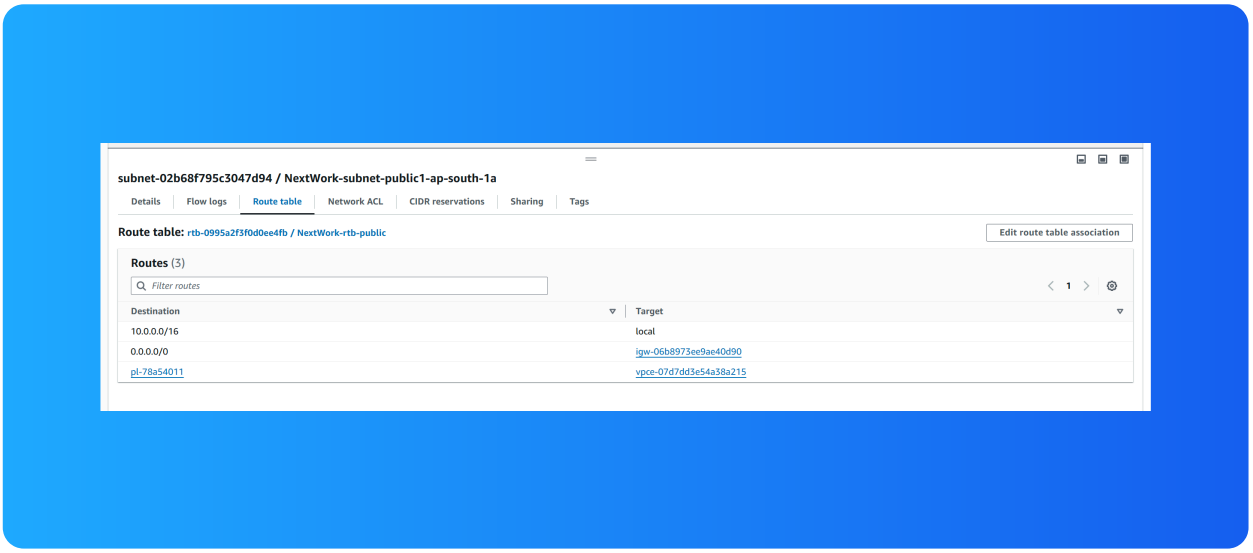
I also had to update my route table because my route table by default, didn't provide a route for traffic in my public subnet to the VPC endpoint.



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 our VPC's public subnet.

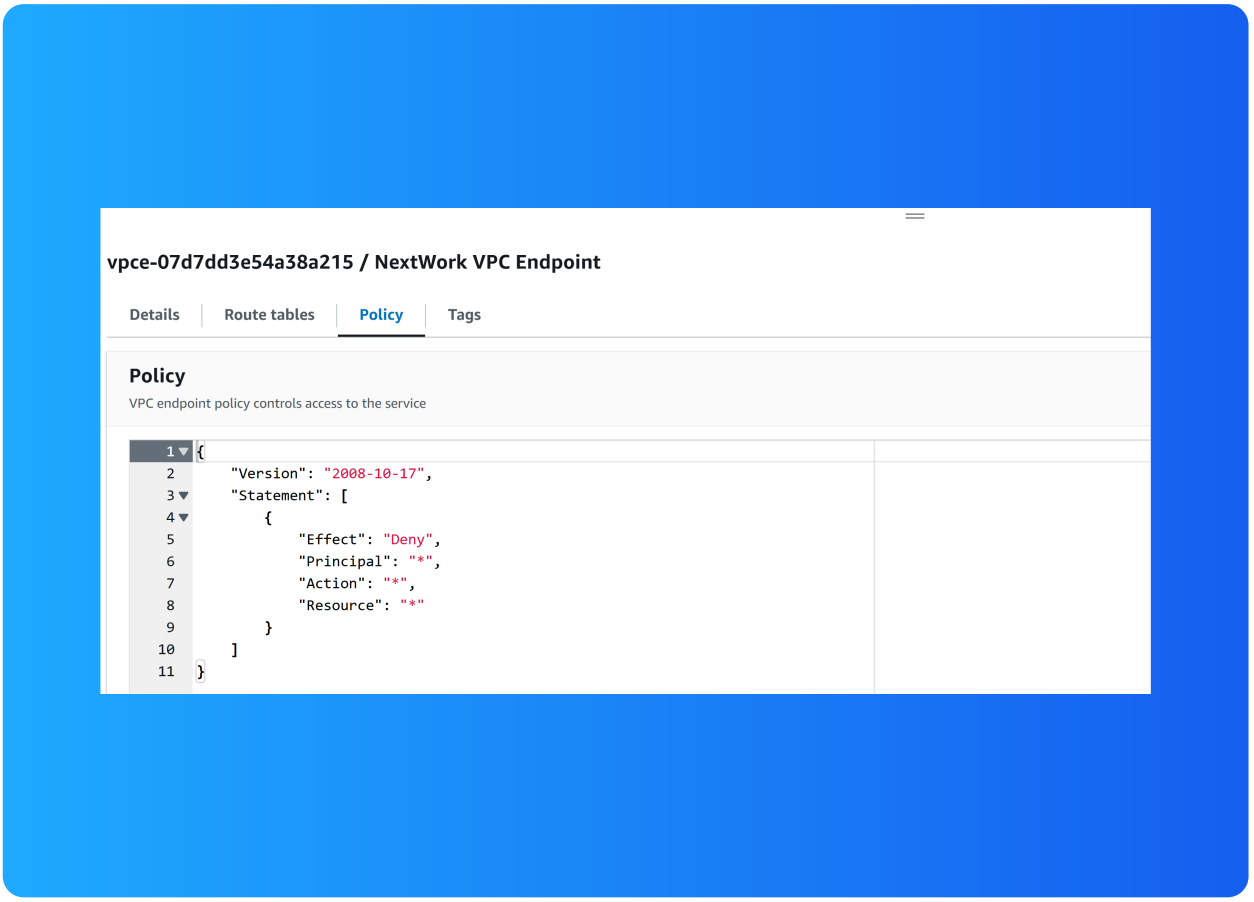
After updating my public subnet's route table, my EC2 instance could connect with my S3 bucket! Access was no longer denied!



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.



Everyone should be in a job they love.

Check out nextwork.org for more projects