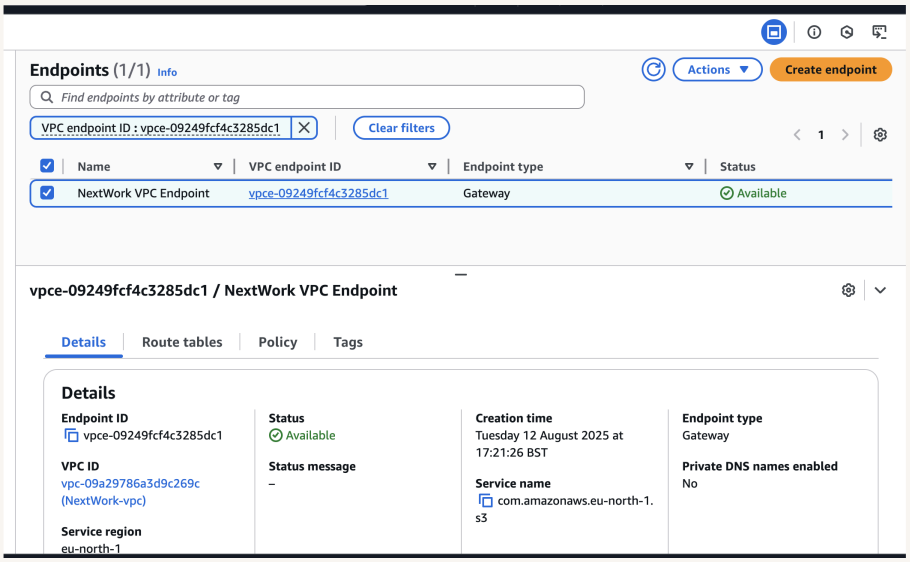
nextwork.org

VPC Endpoints

K Kehinde Abiuwa



Kehinde Abiuwa
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a secure, isolated virtual network within AWS that lets you control networking resources such as IP ranges, subnets, and routing so you can run AWS resources in a customizable and private environment.

How I used Amazon VPC in this project

I used Amazon VPC to create a vpc, and then created an EC2 instance inside the VPC, Then I created a S3 bucket and uploaded some files into the bucket. I then went back into my VPC and went to Endpoints and then created a VPC endpoint for my S3 bucket and updated my route table so that traffic from my EC2 instance to my S3 bucket travels over the private AWS network instead of the public internet.

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

Nothing really. It was an interesting project overall

Kehinde Abiuwa
NextWork Student

This project took me...

I spent around 40 minutes on this project

In the first part of my project...

Step 1 - Architecture set up

In this step I am going to Create a VPC from scratch, Launch an EC2 instance, which I'll connect to using EC2 Instance Connect later and Set up an S3 bucket.

Step 2 - Connect to EC2 instance

In this step I am going to connect to my EC2 instance.

Step 3 - Set up access keys

In this step I am going to create an IAM role that enables my EC2 instance to access my S3 buckets

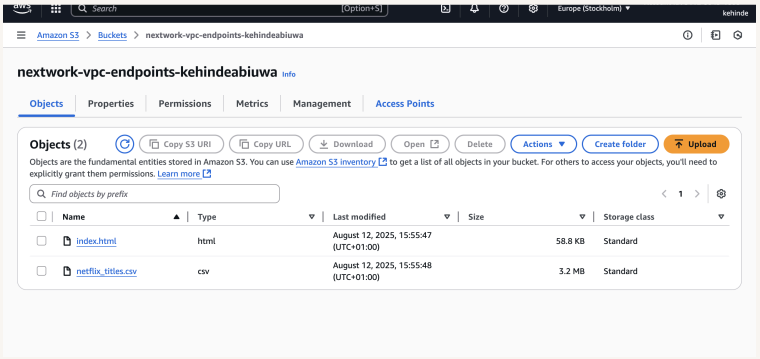
Step 4 - Interact with S3 bucket

In this step I am going to get my EC2 instance to access my S3 bucket.

Architecture set up

I started my project by launching a VPC and then launching and instance inside my VPC

I also set up sn S3 bucket and then uploaded two files (index.html & netflix_titles.csv) into it



Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured the following four things: Access Key ID – A unique identifier associated with my AWS credentials. Secret Access Key – The confidential key paired with the access key ID to securely sign requests. IAM Role or User Permissions – The specific permissions allowing the EC2 instance or user to access AWS resources. AWS CLI or SDK Configuration – The setup of AWS command-line tools or software development kits on the instance to use the credentials for API calls.

Access keys are a pair of security credentials, consisting of an access key ID and a secret access key, that allow you to authenticate and make programmatic requests to AWS services through tools like the AWS CLI

Secret access keys are confidential credentials used in AWS to securely sign programmatic requests to AWS services. They work together with the access key ID to authenticate and authorize API calls, ensuring that only authorized users or applications can access AWS resources. Secret access keys should be kept private and never shared or exposed publicly, as they grant access to your AWS account's resources.

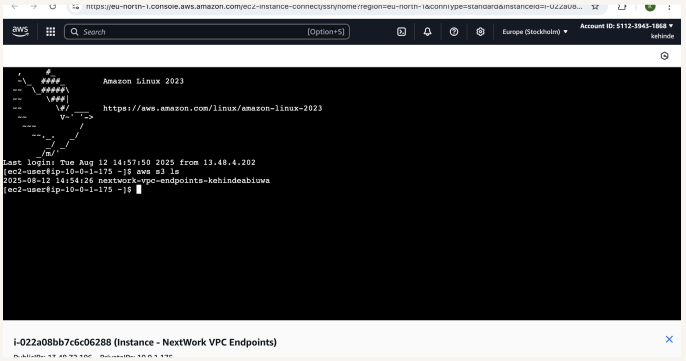
Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM roles with attached policies, which provide temporary security credentials to AWS resources without needing to store long-term keys.

Connecting to my S3 bucket

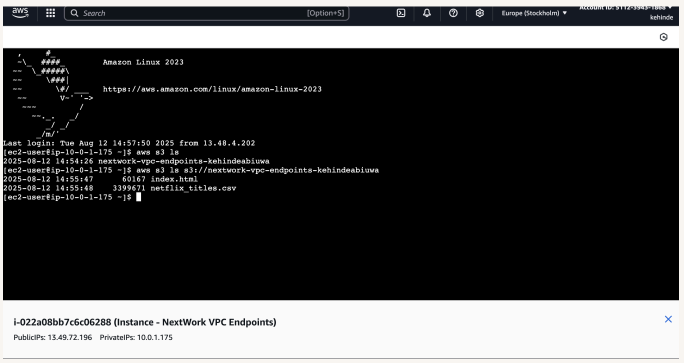
The command I ran was 'aws s3 ls'. Thhis command is used to list all S3 buckets in my AWS account, and if I specify a bucket name, it lists the objects inside that bucket.

The terminal responded with a list of all my S3 buckets. This indicated that the access keys I set up are working correctly



Connecting to my S3 bucket

I also tested the command 'aws s3 ls s3://nextwork-vpc-endpoints-kehindeabiuwa'. Which returned a list of all the files inside my S3 bucket

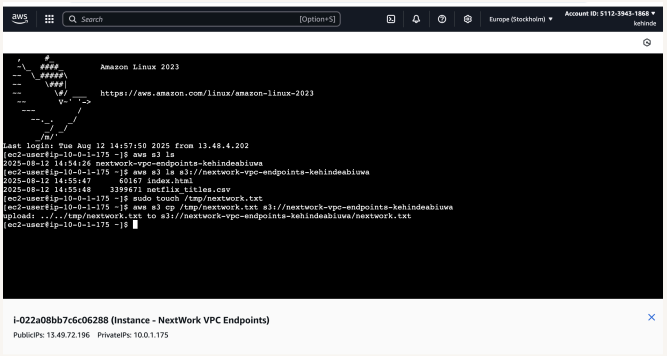


Uploading objects to S3

To upload a new file to my bucket, I first ran the command 'sudo touch /tmp/nextwork.txt'. This command creates an empty file called nextwork.txt in a temporary folder on my EC2 instance

The second command I ran was 'aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-kehindeabiuwa'. This command will upload the just created nextwork.txt file into my S3 bucket

The third command I ran was 'aws s3 ls s3://nextwork-vpc-endpoints-kehindeabiuwa'. Which validated that the nextwork.txt file has been succesfully uploaded into my S3 bucket



In the second part of my project...

Step 5 - Set up a Gateway

In this step I am going to setup a VPC endpoint so that my EC2 instance and my S3 bucket can interact without using a public network.

Step 6 - Bucket policies

In this step I am going to Limit my S3 bucket access's to only traffic from my endpoint. I am doing this to validate my endpoint connection

Step 7 - Update route tables

In this step I am going to test my VPC endpoint set up & Troubleshoot a connectivity issue. I want to know if my EC2 instancee can still access my S3 bucket now that I have limited access to my S3 bucket

Step 8 - Validate endpoint conection

In this step, I am going to test my VPC endpoint setup and restrict my VPC's access to my AWS environment. The purpose is to validate that my EC2 instance can communicate with my S3 bucket over the private network, ensuring that the connection works as intended and does not rely on the public internet.

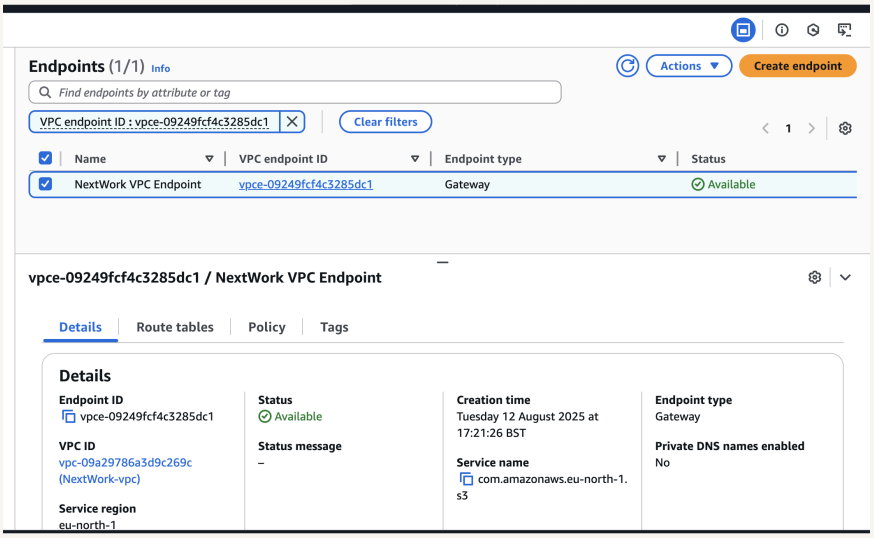
Kehinde Abiuwa
NextWork Student

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).

What are endpoints?

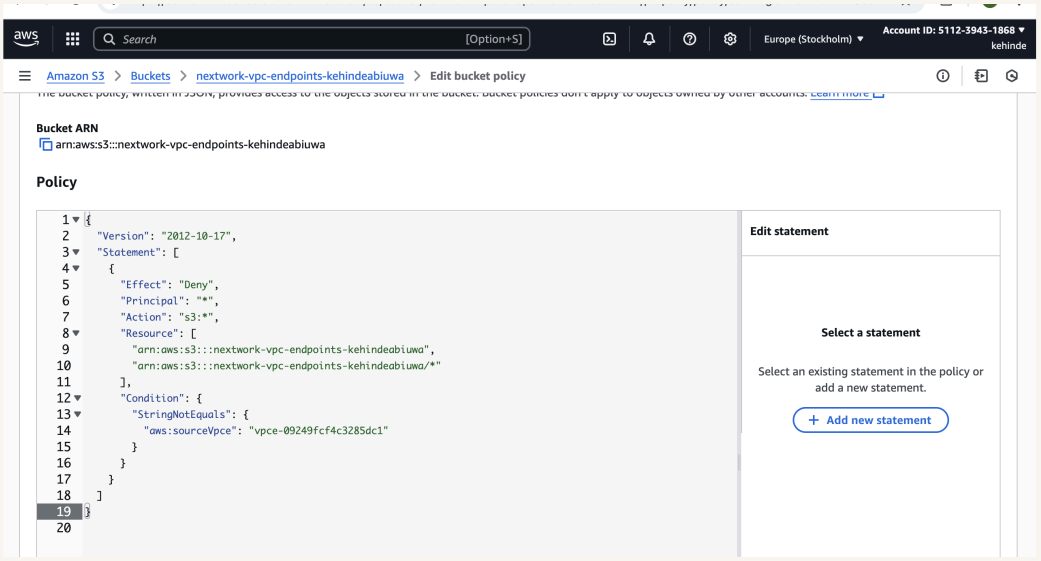
An endpoint is a service in AWS that allows private connections between a VPC and other AWS services without needing the traffic to go over the internet but rather through a private network.



Bucket policies

A bucket policy is a JSON-based access control policy attached directly to an S3 bucket that defines who can access the bucket and what actions they are allowed to perform. It lets you specify permissions for users, roles, or AWS services, controlling access to the bucket and its contents at a granular level. Bucket policies are used to manage security and ensure that only authorized entities can read, write, or manage objects in the bucket.

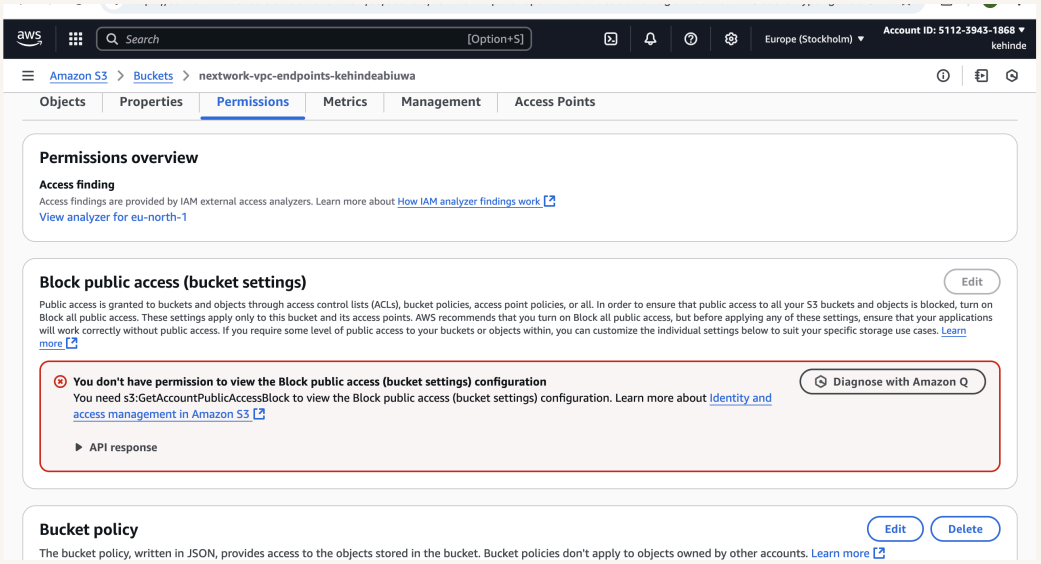
My bucket policy will deny all S3 actions from any principal unless the request originates from the specified VPC endpoint (my nextwork VPC with ID vpce-09249fcf4c3285dc1). This ensures that only traffic coming through that private VPC endpoint can access my bucket nextwork-vpc-endpoints-kehindeabiuwa, effectively blocking all other public or unauthorized access attempts.


Search
[Option+S]
Europe (Stockholm)
Account ID: 5112-3943-1868
kehinde
Amazon S3
Buckets
nextwork-vpc-endpoints-kehindeabiuwa
Edit bucket policy
Bucket ARN
arn:aws:s3:::nextwork-vpc-endpoints-kehindeabiuwa
Policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Deny",
"Principal": "*",
"Action": "s3:*",
"Resource": [
"arn:aws:s3:::nextwork-vpc-endpoints-kehindeabiuwa",
"arn:aws:s3:::nextwork-vpc-endpoints-kehindeabiuwa/*"
],
"Condition": {
"StringNotEquals": {
"aws:sourceVpce": "vpce-09249fcf4c3285dc1"
}
}
}
]
}
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
+ Add new statement

Bucket policies

Right after saving my bucket policy, my S3 bucket page showed 'denied access' warnings. This was because the bucket policy I applied restricts access to only requests coming through the specified VPC endpoint. Since this current access was likely from outside that VPC endpoint such as from the AWS Management Console over the public internet, it triggered the “denied access” warnings. The policy is working as intended by blocking any access that doesn’t come from the approved private network.

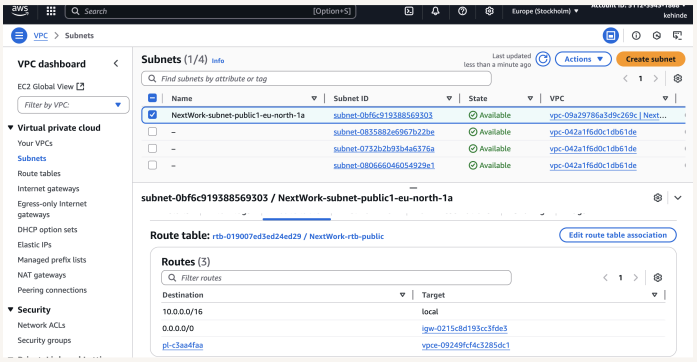
I also had to update my route table because if my route table doesn’t have a route directing traffic destined for S3 to my VPC endpoint, then traffic from my EC2 instance will try to reach the S3 bucket over the public internet instead of through the secure, private connection. This defeats the purpose of using the VPC endpoint for private communication.


aws
Search
[Option+S]
Europe (Stockholm)
Account ID: 5112-3943-1868
kehinde
Amazon S3 > Buckets > nextwork-vpc-endpoints-kehindeabiuwa
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for eu-north-1
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
You don't have permission to view the Block public access (bucket settings) configuration
You need s3:GetAccountPublicAccessBlock to view the Block public access (bucket settings) configuration. Learn more about Identity and access management in Amazon S3
Diagnose with Amazon Q
API response
Bucket policy
Edit
Delete
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

Route table updates

To update my route table, I went to the VPC console, selected my VPC endpoint, and then associated the route table linked to my VPC so that traffic from my EC2 instance is routed through the VPC endpoint to access S3 privately.

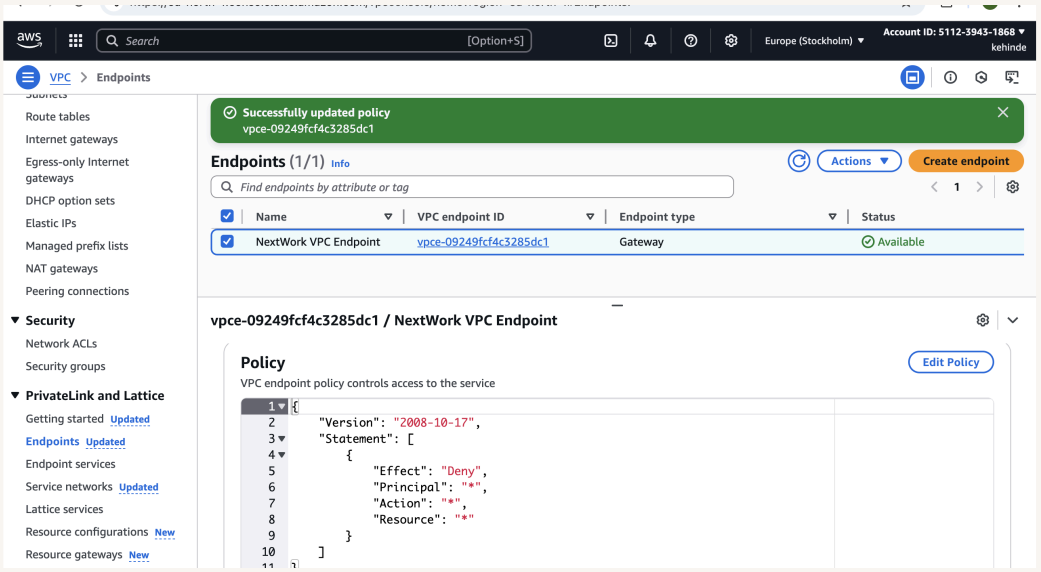
After updating my public subnet's route table, my terminal could return the list of files in my s3 bucket



Endpoint policies

An endpoint policy is a JSON-based resource policy attached to a VPC endpoint that controls which AWS services and resources can be accessed through that endpoint. It allows you to define fine-grained permissions, such as restricting access to specific S3 buckets or DynamoDB tables, and ensures that traffic through the endpoint follows your security and compliance requirements.

I updated my endpoint's policy by changing the line "Effect": "Allow" to "Effect": "Deny"... I could see the effect of this right away, because then I wasnt able to access my S3 bucket from my EC2 instance


aws
Search
[Option+S]
Europe (Stockholm)
Account ID: 5112-3943-1868
kehinde
VPC > Endpoints
Route tables
Internet gateways
Egress-only Internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Security
Network ACLs
Security groups
PrivateLink and Lattice
Getting started Updated
Endpoints Updated
Endpoint services
Service networks Updated
Lattice services
Resource configurations New
Resource gateways New
Successfully updated policy
vpce-09249fcf4c3285dc1
Endpoints (1/1) Info
Actions
Create endpoint
Find endpoints by attribute or tag
Name
VPC endpoint ID
Endpoint type
Status
NextWork VPC Endpoint
vpce-09249fcf4c3285dc1
Gateway
Available
vpce-09249fcf4c3285dc1 / NextWork VPC Endpoint
Policy
VPC endpoint policy controls access to the service
Edit Policy
{
"Version": "2008-10-17",
"Statement": [
{
"Effect": "Deny",
"Principal": "*",
"Action": "*",
"Resource": "*"
}
]

The place to learn & showcase your skills

Check out nextwork.org for more projects