



VPC Endpoints



Louis Moyo

vpce-03c289549230414b7 / LouisMoyo VPC Endpoint

Details Route tables Policy Tags

Details	Status	Creation time	Endpoint type
Endpoint ID vpce-03c289549230414b7	Available	Saturday 16 August 2025 at 06:18:51 BST	Gateway
VPC ID vpc-007b5d70ff25d7f13 (LouisMoyo-vpc)	Status message -	Service name com.amazonaws.us-east-1.s3	Private DNS names enabled No
Service region us-east-1			



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a private virtual network in AWS where you can launch resources like EC2 and connect them securely. It's useful because it gives control over IP ranges, subnets, routing, and connectivity to other AWS services.

How I used Amazon VPC in this project

I used Amazon VPC to create a secure environment with an EC2 instance, then connected it to S3 using a VPC endpoint. This let my instance access S3 privately without sending traffic over the internet.

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

One thing I didn't expect in this project was that once I applied a restrictive bucket policy, even the AWS Management Console was blocked until the VPC endpoint was used.

This project took me...

This project took me about 75 minutes to complete, including creating the VPC, setting up the EC2 instance, creating the S3 bucket, and configuring the VPC endpoint with bucket policies.



In the first part of my project...

Step 1 - Architecture set up

In this step I'm setting up the foundations of my project by creating a new VPC, launching an EC2 instance that I can later connect to with EC2 Instance Connect, and creating an S3 bucket for storage.

Step 2 - Connect to EC2 instance

In this step I'm connecting directly to my EC2 instance using EC2 Instance Connect. This lets me open the terminal on the instance so I can run AWS CLI commands and test S3 access over the public internet.

Step 3 - Set up access keys

In this step I will create access keys to give my EC2 instance credentials, allowing it to securely connect to my AWS environment and interact with services like S3 using the CLI.

Step 4 - Interact with S3 bucket

In this step I'm heading back to my EC2 instance and using the AWS CLI to access my S3 bucket. This lets me confirm that the instance can securely connect and list objects in the bucket.



Architecture set up

I started my project by launching a new VPC named LouisMoyo with the default CIDR block 10.0.0.0/16, one public subnet, and an EC2 instance called Instance – LouisMoyo VPC Endpoints with a security group for SSH access.

I also set up an S3 bucket named lfblm-vpc-endpoints-louis and uploaded two files into it. This bucket will be used to practise accessing and listing objects from my EC2 instance.

The screenshot shows the AWS S3 console interface. At the top, a green success message box displays "Upload succeeded" and "For more information, see the Files and folders table." Below this, a summary table provides upload details: Destination is "s3://lfblm-vpc-endpoints-louis", Status is "Succeeded" (with 2 files, 4.6 MB total), and Failed files are listed as "0 files, 0 B (0%)". The main content area shows a table titled "Files and folders (2 total, 4.6 MB)". The table includes columns for Name, Folder, Type, Size, Status, and Error. Two files are listed: "NextWork - Denzel is awesome.png" and "NextWork - Leo is awesome.png", both of which have a status of "Succeeded".

Name	Folder	Type	Size	Status	Error
NextWork - Denzel is awesome.png	-	image/png	2.3 MB	Succeeded	-
NextWork - Leo is awesome.png	-	image/png	2.3 MB	Succeeded	-



Access keys

Credentials

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

Access keys are a pair of credentials (an access key ID and a secret access key) that allow secure programmatic access to AWS services through the CLI or SDKs.

Secret access keys are like the password paired with an access key ID (the username). Together they provide programmatic access to AWS services and must be kept secure.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM roles with temporary credentials, which are automatically managed and more secure.

Connecting to my S3 bucket

The command I ran was aws s3 ls. This command is used to list all the S3 buckets in my account and confirm that my EC2 instance can connect to AWS services through the CLI.

The terminal responded with a list of my S3 buckets. This indicated that the access keys I set up were working correctly and my EC2 instance was successfully connected to my AWS environment.

```
[ec2-user@ip-10-0-10-14 ~]$ aws s3 ls
2025-08-05 06:56:25 config-bucket-756716632322
2025-08-05 07:44:00 config-bucket-756716632322-us-east-1
2025-08-04 13:23:51 cspm-demo-eor-20250804
2025-08-04 13:24:34 cspmdemo1ouis
2025-06-02 07:33:50 l-bucket-2
2025-08-16 04:44:02 lfblm-vpc-endpoints-louis
2025-08-16 03:36:04 lfblm-vpc-project-louis
2025-06-01 06:23:29 louis1-lab-bucket-1234
2025-08-05 08:42:31 my-config-bucket-logs
2025-08-10 02:57:45 soc2-insecure-bucket-1754794663
2025-08-10 03:10:54 soc2-trail-logs-1754795452
2025-08-10 03:16:48 soc2-trail-logs-1754795806
2025-08-10 03:23:58 soc2-trail-logs-839131973
[ec2-user@ip-10-0-10-14 ~]$ █
```

Connecting to my S3 bucket

I also tested the command `aws s3 ls s3://lfblm-vpc-endpoints-louis` which returned the objects inside my S3 bucket, confirming that my EC2 instance could access the files.

```
[ec2-user@ip-10-0-10-14 ~]$ aws s3 ls s3://lfblm-vpc-endpoints-louis
2025-08-16 04:45:11    2431554 NextWork - Denzel is awesome.png
2025-08-16 04:45:12    2399812 NextWork - Lelo is awesome.png
[ec2-user@ip-10-0-10-14 ~]$ █
```

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 text file in the /tmp directory to be uploaded to S3.

The second command I ran was aws s3 cp /tmp/nextwork.txt s3://lfblm-vpc-endpoints-louis/nextwork.txt. This command will copy my new text file into the S3 bucket.

The third command I ran was aws s3 ls s3://lfblm-vpc-endpoints-louis. This validated that the new file nextwork.txt was successfully uploaded and appeared in my S3 bucket.

```
[ec2-user@ip-10-0-10-14 ~]$ sudo touch /tmp/nextwork.txt
[ec2-user@ip-10-0-10-14 ~]$ aws s3 cp /tmp/nextwork.txt s3://lfblm-vpc-endpoints-louis
upload: ../../tmp/nextwork.txt to s3://lfblm-vpc-endpoints-louis/nextwork.txt
[ec2-user@ip-10-0-10-14 ~]$ aws s3 ls s3://lfblm-vpc-endpoints-louis
2025-08-16 04:45:11    2431554 NextWork - Denzel is awesome.png
2025-08-16 04:45:12    2399812 NextWork - Lelo is awesome.png
2025-08-16 05:07:36      0 nextwork.txt
[ec2-user@ip-10-0-10-14 ~]$
```



In the second part of my project...

Step 5 - Set up a Gateway

I'm creating a VPC endpoint for S3 so my VPC talks to S3 privately instead of the public internet.

Step 6 - Bucket policies

I'm locking down my S3 bucket with a policy that only allows access via my S3 VPC endpoint and blocks everything else. This proves traffic stays within AWS.

Step 7 - Update route tables

I'm testing my VPC endpoint setup by trying to access my S3 bucket again from my EC2 instance. This confirms that my instance can still reach S3 privately through the endpoint, even though public access is blocked.

Step 8 - Validate endpoint connection

I'm validating my setup by accessing my S3 bucket from my EC2 instance one last time. This confirms that my VPC endpoint is working correctly and that bucket access is fully restricted to the private AWS network.



Setting up a Gateway

I set up an S3 Gateway, which is a type of VPC endpoint that routes traffic between my VPC and Amazon S3 directly over the AWS network.

What are endpoints?

An endpoint is a private connection that lets resources in a VPC securely access AWS services without using the public internet.

The screenshot shows the AWS VPC Endpoint Details page. The endpoint ID is vpce-03c289549230414b7, it is associated with VPC ID vpc-007b5d70ff23df713 (LouisMoyo-vpc), and the service region is us-east-1. The status is Available, and the creation time is Saturday 16 August 2025 at 06:18:51 BST. The service name is com.amazonaws.us-east-1.s3. The endpoint type is Gateway, and Private DNS names enabled is No.

Details	Status	Creation time	Endpoint type
Endpoint ID vpce-03c289549230414b7	Available	Saturday 16 August 2025 at 06:18:51 BST	Gateway
VPC ID vpc-007b5d70ff23df713 (LouisMoyo-vpc)	Status message -	Service name com.amazonaws.us-east-1.s3	Private DNS names enabled No
Service region us-east-1			



Bucket policies

A bucket policy is a JSON-based resource policy that defines permissions and access rules for an S3 bucket and the objects inside it.

My bucket policy will block all access to my S3 bucket except traffic that comes through my VPC endpoint, ensuring secure private access only.

```
Policy

1 [ { 
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Deny",
6       "Principal": "*",
7       "Action": "s3:*",
8       "Resource": [
9         "arn:aws:s3::::1fb1m-vpc-endpoints-louis",
10        "arn:aws:s3::::1fb1m-vpc-endpoints-louis/*"
11      ],
12      "Condition": {
13        "StringNotEquals": {
14          "aws:sourceVpce": "vpce-03c289549230414b7"
15        }
16      }
17    }
18  ]
19 }
20 }
```



Bucket policies

Right after saving my bucket policy, my S3 bucket page showed 'denied access' warnings. This was because my policy denies all access unless it comes through my VPC endpoint, which blocks even the AWS Console.

I also had to update my route table because without a route to my VPC endpoint, my EC2 instance was trying to reach S3 over the public internet instead of through the private AWS network.

The screenshot shows three separate screenshots of the AWS S3 console, each displaying a 'denied access' error message:

- Block public access (bucket settings):** Shows a warning about public access being blocked. It includes a link to learn more about identity and access management in Amazon S3.
- Bucket policy:** Shows a warning about not having permission to view the bucket policy configuration. It includes a link to learn more about IAM permissions and a link to diagnose with Amazon Q.
- Object Ownership:** Shows a warning about not having permission to view object ownership configuration. It includes a link to learn more about object ownership in Amazon S3 and a link to diagnose with Amazon Q.



Route table updates

To update my route table, I associated my public subnet's route table with the S3 VPC endpoint, so traffic from my EC2 instance is now directed securely to S3 via the endpoint.

After updating my public subnet's route table, my terminal could return the list of objects inside my S3 bucket, confirming that the VPC endpoint connection was working correctly.

Updated routes for rtb-042140fb2be96e70a / louismoyo-2-rtb-public successfully

Details

rtb-042140fb2be96e70a / louismoyo-2-rtb-public

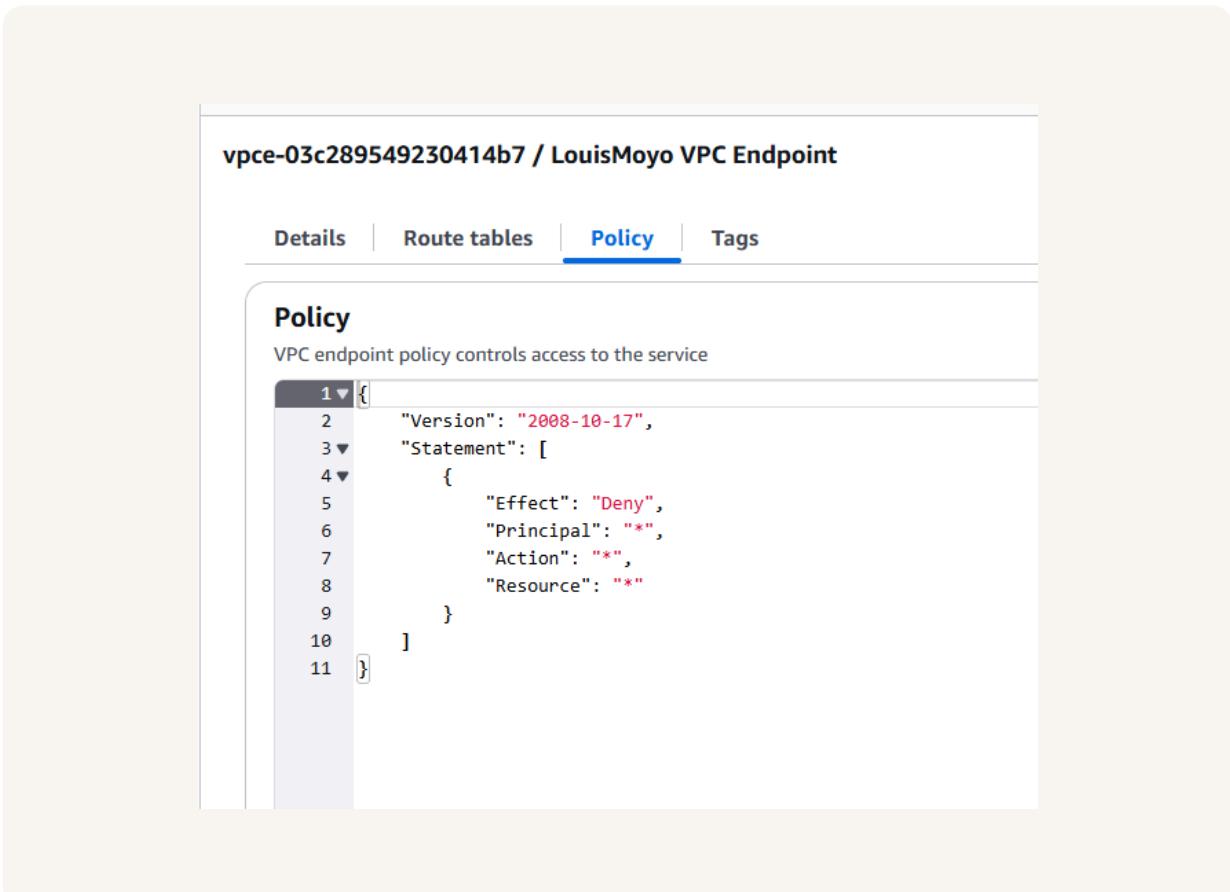
Actions

Details		Explicit subnet associations		Edge associations	
Route table ID	rtb-042140fb2be96e70a	Main	No	Owner ID	756716632322
VPC	vpc-0f9532cd5b429956d5 louismoyo-2-vpc	subnet-0f53de65410bed9316 / louismoyo-2-subnet-public1-us-east-1a			
Routes [3]		Edit routes			
Destination	Target	Status	Propagated	Route Origin	
0.0.0.0/0	igw-0ee9c4864634f0a09	Active	No	Create Route	
10.1.0.0/16	pcx-0fb9f41080eec18cf	Active	No	Create Route	
10.2.0.0/16	local	Active	No	Create Route Table	

Endpoint policies

An endpoint policy is a resource-based JSON policy attached to a VPC endpoint that controls which AWS services and actions can be accessed through that 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 no longer able to access my S3 bucket when I ran the aws s3 ls command.





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

