



# VPC Endpoints

N

Nchindo Boris

The screenshot shows the AWS Management Console interface for VPC Endpoints. A success message at the top indicates "Successfully created VPC endpoint vpce-0327f068c5b41bb98". The main table lists one endpoint:

Name	VPC endpoint ID	Endpoint type	Status	Service name
NextWork VPC Endpoint	vpce-0327f068c5b41bb98	Gateway	Available	com.amazonaws.eu-north-1.svc

The "Details" tab is selected, displaying the following information:

Details	Status	Creation time	Endpoint type
Endpoint ID: vpce-0327f068c5b41bb98	Available	Friday 8 August 2025 at 12:01:28 GMT+1	Gateway
VPC ID: vpc-0b0075b2f5fc99d36 (NextWork-vpc)	Status message: -	Service name: com.amazonaws.eu-north-1.svc	Private DNS names enabled: No
Service region: -	-	-	-



# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC is a service that lets you create a custom, isolated virtual network within the AWS cloud.

## How I used Amazon VPC in this project

I used Amazon VPC in this project to create a VPC Endpoint in order to access my S3 Bucket securely and not through the public internet

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

I did not expect this project to be this easy to do

## This project took me...

This project took me 45 minutes to complete



# 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 will connect to using EC2 Instance Connect later.
- Set up an S3 bucket.

## Step 2 - Connect to EC2 instance

In this step, I am going to Connect directly to your EC2 instance.

## Step 3 - Set up access keys

In this step, I am going to Give your EC2 instance access to your AWS environment.

## Step 4 - Interact with S3 bucket

In this step, I am going to:

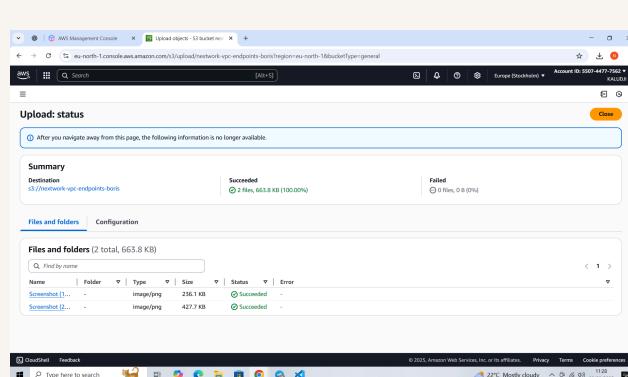
- Head back to your EC2 instance.
- Get your EC2 instance to access your S3 bucket.



# Architecture set up

I started my project by launching a VPC and an EC2 Instance

I also set up an S3 bucket and uploaded 2 files into it





# Access keys

## Credentials

To set up my EC2 instance to interact with my AWS environment, I configured; -  
Access key ID - Secret key - Default region - Default output format

Access keys are a set of security credentials that allow access to your AWS account

The secret access key is like the password that pairs with your access key ID (your username).

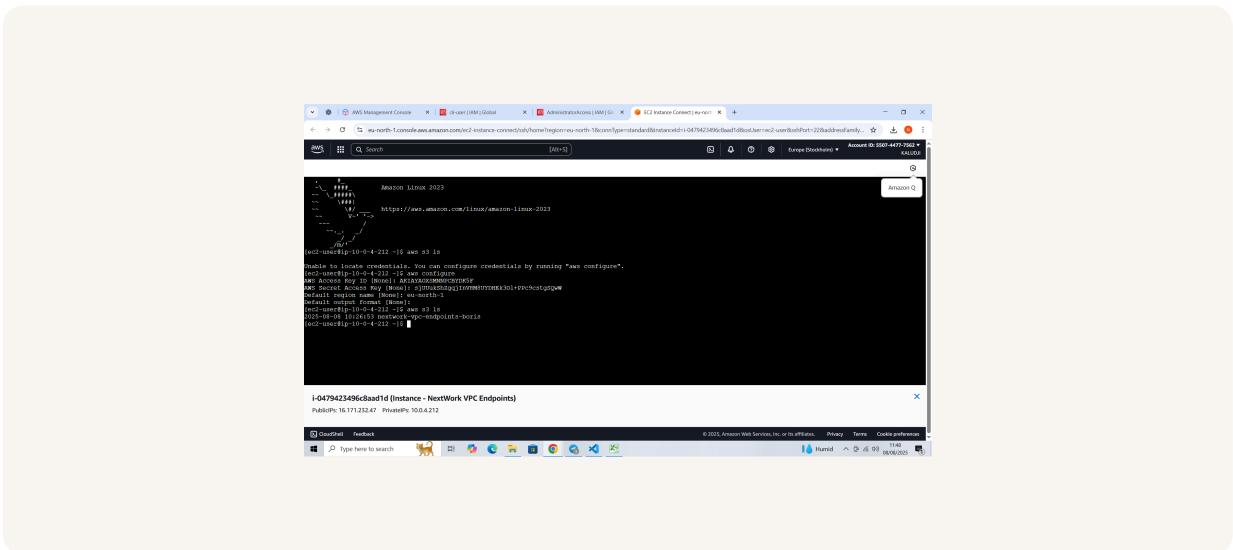
## Best practice

The best practice is to avoid long-term access keys whenever possible and instead use IAM Roles (with Temporary Credentials)

# Connecting to my S3 bucket

The command I ran was aws s3 ls . This command is used to list s3 buckets in my account

The terminal responded with 2025-08-08 10:26:53 nextwork-vpc-endpoints-boris. This indicated that the access keys I set up have connected me with my AWS S3 Bucket.





# Connecting to my S3 bucket

I also tested the command `aws s3 ls s3://nextwork-vpc-endpoints-boris .` which returned the list of objects in my s3 bucket.

```
Amazon Linux 2023
https://www.amazon.com/linux/amazon-linux-2023

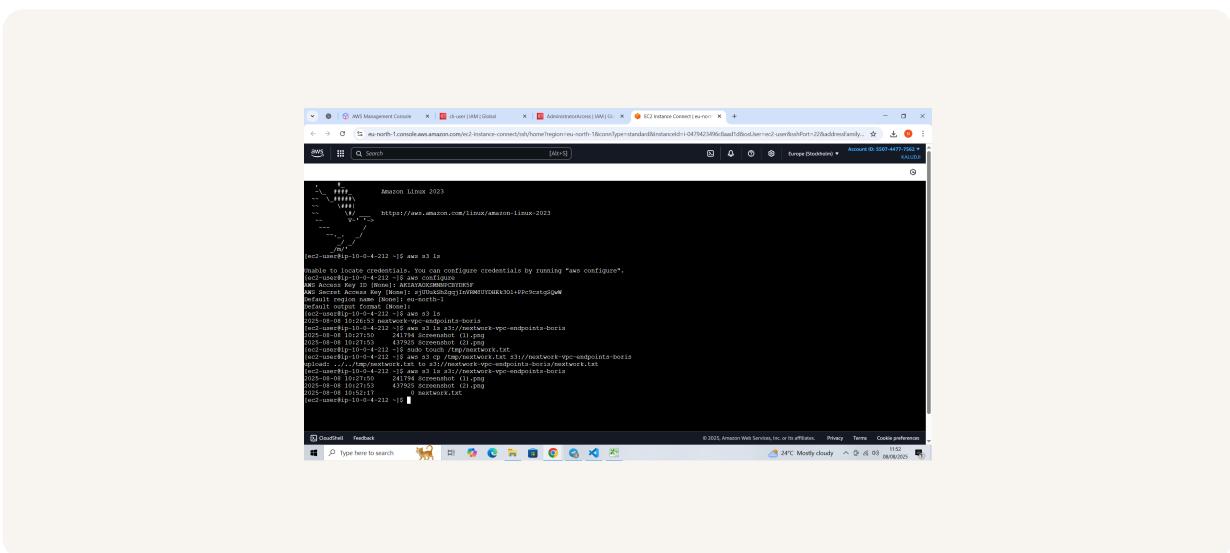
[ec2-user@ip-10-0-4-212 ~]$ aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
AWS Access Key ID in [config]: AKIAVXKQHNNBZC5F
AWS Secret Access Key in [config]: 1A234567890123456789012345678901234567890
Default region name in [config]: us-east-1
Default output format in [config]: json
[ec2-user@ip-10-0-4-212 ~]$ aws s3 ls
[ec2-user@ip-10-0-4-212 ~]$ aws s3 ls s3://nextwork-vpc-endpoints-boris
2025-08-08 10:29:50    241974 Screenshot_03.png
[ec2-user@ip-10-0-4-212 ~]$
```

# 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 create a blank .txt file in your EC2 instance..

The second command I ran was aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-boris. This command will upload the empty file into your bucket.

The third command I ran was aws s3 ls s3://nextwork-vpc-endpoints-boris. which validated that there is an empty file nextwork.txtwith no data, it has 0 bytes.



A screenshot of a terminal window within the AWS Management Console. The window title is 'Amazon Linux 2023'. The terminal output shows the following commands and their results:

```
[root@ip-10-0-4-212 ~]# sudo touch /tmp/nextwork.txt
[sudo] password for root:
[root@ip-10-0-4-212 ~]# ls -l /tmp/nextwork.txt
ls: /tmp/nextwork.txt: No such file or directory
[root@ip-10-0-4-212 ~]# aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-boris
[Amazon S3: Uploading: /tmp/nextwork.txt to s3://nextwork-vpc-endpoints-boris/nextwork.txt]
[root@ip-10-0-4-212 ~]# aws s3 ls s3://nextwork-vpc-endpoints-boris
2023-09-08 10:26:53 nextwork-vpc-endpoints-boris/
[root@ip-10-0-4-212 ~]# aws s3 ls s3://nextwork-vpc-endpoints-boris
2023-09-08 10:27:05 241794 Screenshot (1).jpg
[root@ip-10-0-4-212 ~]# aws s3 cp ./Screenshot (1).jpg s3://nextwork-vpc-endpoints-boris
[Amazon S3: Uploading: Screenshot (1).jpg to s3://nextwork-vpc-endpoints-boris/Screenshot (1).jpg]
[root@ip-10-0-4-212 ~]# aws s3 ls s3://nextwork-vpc-endpoints-boris
2023-09-08 10:27:05 241794 Screenshot (1).jpg
[root@ip-10-0-4-212 ~]# rm Screenshot (1).jpg
[root@ip-10-0-4-212 ~]# aws s3 ls s3://nextwork-vpc-endpoints-boris
2023-09-08 10:27:17 0 nextwork.txt
[root@ip-10-0-4-212 ~]#
```



# In the second part of my project...

## Step 5 - Set up a Gateway

In this step, I am going to Set up a way for your VPC and S3 to communicate directly.

## Step 6 - Bucket policies

In this step, I am going to Limit your S3 bucket access's to only traffic from your endpoint.

## Step 7 - Update route tables

In this step, I am going to: - Test your VPC endpoint set up. - Troubleshoot a connectivity issue.

## Step 8 - Validate endpoint connection

In this step, I am going to: - Test your VPC endpoint set up (again). - Restrict your VPC's access to your AWS environment.



# Setting up a Gateway

I set up an S3 Gateway, which is a type of endpoint which 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.

The screenshot shows the AWS Management Console interface for the VPC service. The main title bar indicates the URL is `eu-north-1.console.aws.amazon.com/vpcconsole/home?region=eu-north-1#EndpointsvpCEndpointId=vpce-0327f068c5b41bb98`. The top navigation bar includes links for AWS, Search, and various AWS services like IAM and Glue. The account information shows "Account ID: 5507-4477-7852" and "Region: Europe (Stockholm)".

The left sidebar has a tree view with "VPC dashboard" expanded, showing "Virtual private cloud" and "Security" sections. Under "Virtual private cloud", there are links for "Your VPCs", "Subnets", "Route tables", "Internet gateways", "Egress-only Internet gateways", "DHCP option sets", "Elastic IPs", "Managed prefix lists", "NAT gateways", and "Peering connections". Under "Security", there are links for "Network ACLs" and "Security groups".

The main content area shows a success message: "Successfully created VPC endpoint vpce-0327f068c5b41bb98". Below this, a table titled "Endpoints (1/1) Info" lists one endpoint:

Actions	Create endpoint
<input checked="" type="checkbox"/> Name	vpce-0327f068c5b41bb98
<input checked="" type="checkbox"/> NextWork VPC Endpoint	vpce-0327f068c5b41bb98
	Gateway
	Available
	Service name: com.amazonaws.eu-nx

Below the table, a detailed view for "vpce-0327f068c5b41bb98 / NextWork VPC Endpoint" is shown. The "Details" tab is selected, displaying the following information:

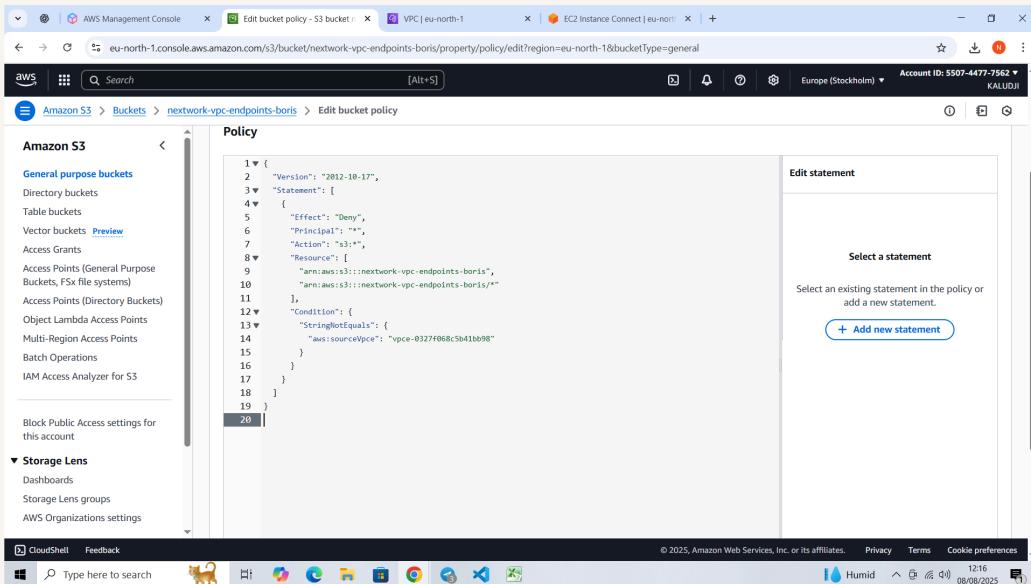
Endpoint ID	Status	Creation time	Endpoint type
vpce-0327f068c5b41bb98	Available	Friday 8 August 2025 at 12:01:28 GMT+1	Gateway
VPC ID	Status message	Service name	Private DNS names enabled
vpc-0b0075b2f5fc99d36 (NextWork-vpc)	-	com.amazonaws.eu-north-1.s3	No
Service region			

At the bottom of the screenshot, the Windows taskbar is visible with icons for CloudShell, Feedback, Start, Task View, File Explorer, Edge, Google Chrome, and File Explorer. The system tray shows the date and time as "08/08/2025" and "12:01".

# Bucket policies

A bucket policy is a type of IAM policy designed for setting access permissions to an S3 bucket

My bucket policy will This policy denies all actions (s3:\*) on your S3 bucket and its objects to everyone (Principal: "\*")... unless the access is from the VPC endpoint with the ID defined in aws:sourceVpc.

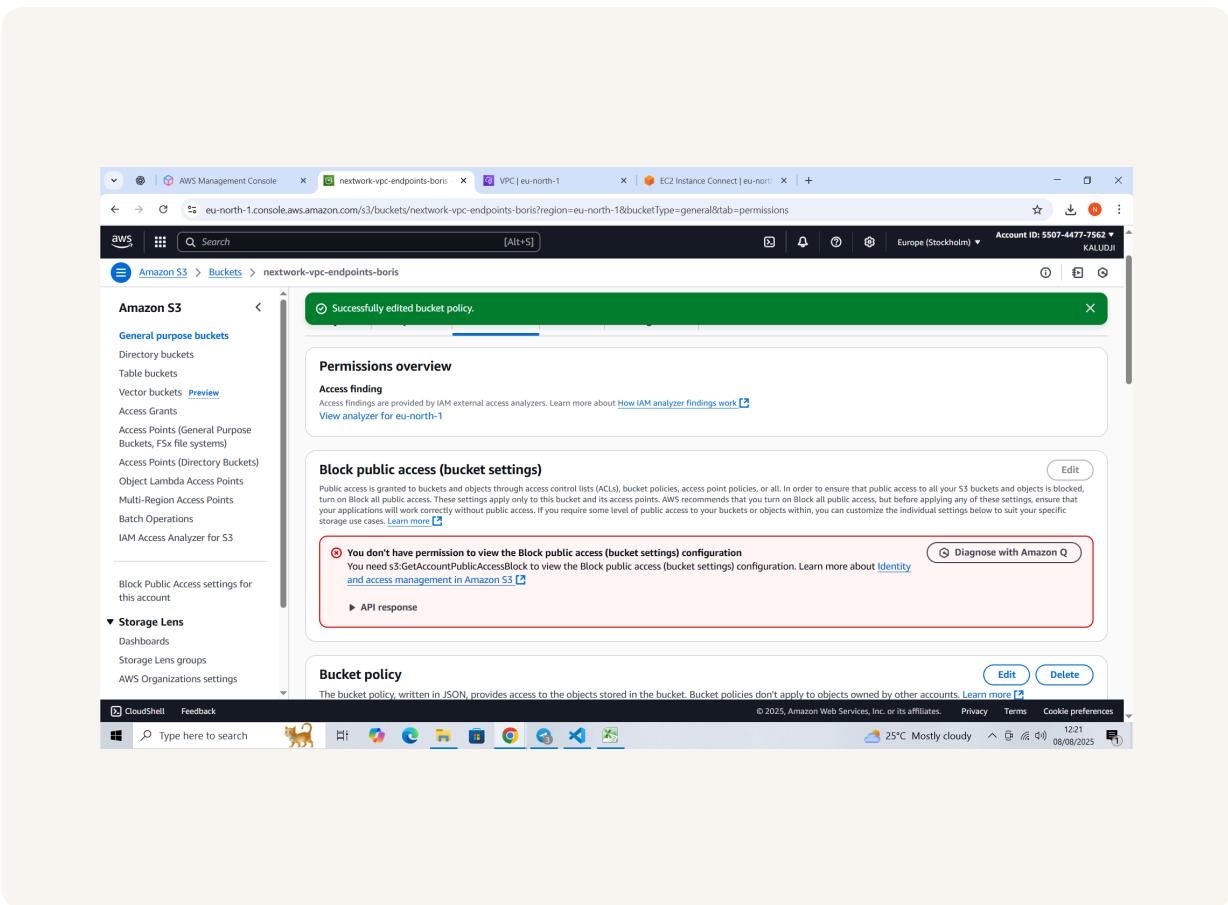


```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Deny",  
6             "Principal": "*",  
7             "Action": "s3:*",  
8             "Resource": [  
9                 "arn:aws:s3:::nextwork-vpc-endpoints-boris",  
10                "arn:aws:s3:::nextwork-vpc-endpoints-boris/*"  
11            ],  
12            "Condition": {  
13                "StringNotEquals": {  
14                    "aws:sourceVpc": "vpce-0327f068c5b41b698"  
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 Your policy denies all actions unless they come from your VPC endpoint. This means any attempt to access your bucket from other sources is blocked

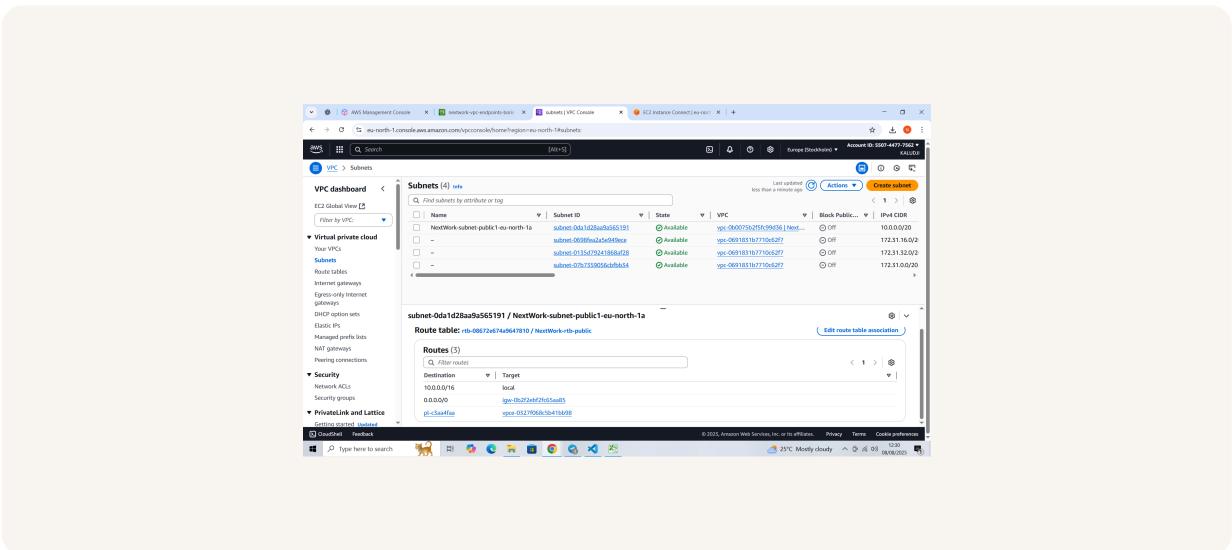
I also had to update my route table because the EC2 instance is trying to get to your S3 bucket through the public internet instead and so a route that directs traffic bound for S3 to your VPC endpoint is required to be added on the route table



# Route table updates

To update my route table, I; Select Endpoints Select the checkbox next to your endpoint, and select Route tables Select Manage route tables Select the checkbox next to your public route table. Select Modify route tables. Refresh the Subnets page

After updating my public subnet's route table, my terminal could return the content of my S3 bucket





# Endpoint policies

'An endpoint policy is a JSON policy document attached to a VPC Endpoint that controls and restricts access to the AWS service through that endpoint.

I updated my endpoint's policy by Select Edit policy. Change the line "Effect": "Allow" to "Effect": "Deny"!. I could see the effect of this right away, because access to my S3 bucket is denied on the CLI. .

The screenshot shows the AWS Management Console interface for managing VPC endpoints. The main window displays a success message: "Successfully updated policy vpce-0327f068c5b41bb98". Below this, the "Endpoints" section lists one endpoint named "NextWork VPC Endpoint" with ID "vpce-0327f068c5b41bb98", which is a "Gateway" type and currently "Available". The "Service name" is listed as "com.amazonaws.eu-north-1.s3". On the left sidebar, there are several navigation tabs including "VPC dashboard", "Virtual private cloud", "Security", "PrivateLink and Lattice", and "CloudShell". The "Policy" tab is selected in the "Details" section of the main content area, showing the JSON policy document:

```
1 {  
2     "Version": "2008-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Deny",  
6             "Principal": "*"  
7         }  
8     ]  
9 }
```



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

