



# VPC Endpoints



nikhil7\_94@hotmail.com

The screenshot shows the AWS VPC Endpoints service interface. At the top, there's a search bar and a 'Create endpoint' button. Below that is a table with one row:

Name	VPC endpoint ID	Endpoint type	Status
NextWork VPC Endpoint	vpce-0ade15c5444903bae	Gateway	Available

Below the table, the endpoint details are shown:

**vpce-0ade15c5444903bae / NextWork VPC Endpoint**

**Details** **Route tables** **Policy** **Tags**

**Details**

Endpoint ID vpce-0ade15c5444903bae	Status Available	Creation time Sunday, November 3, 2024 at 18:51:52 GMT	Endpoint type Gateway
VPC ID vpc-0641ec5cbcd1cfa2e (NextWork-vpc)	Status message -	Service name com.amazonaws.eu-west-3.s3	Private DNS names enabled No

# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC is a private isolated network in the cloud where you can securely run resources like servers and databases. It's useful because it gives full control over network settings, security and connectivity to the internet.

## How I used Amazon VPC in this project

Today I learned how to set up direct, private access to S3 from my VPC.

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

One thing I didnt expect was there are different ways to allow or deny access to S3.

## This project took me...

I took around 1 and half hour with this project.

# In the first part of my project...

## Step 1 - Architecture set up

In this step I'm going to create a VPC from scratch, launch an EC2 instance which I will connect to using EC2 Instance Connect later and set up an S3 bucket.

## Step 2 - Connect to EC2 instance

Now I'm going to connect directly to my EC2 instance.

## Step 3 - Set up access keys

In this step I will give my EC2 instance access to your AWS environment so I can access to the S3 bucket list.

## Step 4 - Interact with S3 bucket

Now I'm going to get my EC2 instance to access my S3 bucket.

# Architecture set up

I started my project by launching a VPC and an EC2 instance.

I also set up an S3 bucket.

Files and folders (2 Total, 4.6 MB)						
<input type="text"/> Find by name						
Name	Folder	Type	Size	Status	Error	⋮
ex1.png	-	image/png	2.3 MB	<span>⌚ Succeeded</span>	-	< 1 >
ex2.png	-	image/png	2.3 MB	<span>⌚ Succeeded</span>	-	

# Access keys

## Credentials

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

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

Secret access keys are like a password of the access key. It is needed to log in with the credentials.

## Best practice

Although I'm using access keys in this project, a best practice alternative is to use an IAM role with the necessary permissions and then attaching that role to my EC2 instance.

# 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.

The terminal responded with the list of S3 buckets in my account. This indicated that the access keys I set up correctly.

```
[ec2-user@ip-10-0-2-104 ~]$ aws s3 ls
2024-11-03 18:15:57 nextwork-vpc-endpoints-nikhil
[ec2-user@ip-10-0-2-104 ~]$ █
```

# Connecting to my S3 bucket

I also tested the command "aws s3 ls s3://nextwork-vpc-endpoints-nikhil" which returned the list of objects inside that S3 bucket indicated.

```
[ec2-user@ip-10-0-2-104 ~]$ aws s3 ls s3://nextwork-vpc-endpoints-nikhil
2024-11-03 18:16:59    2431554 ex1.png
2024-11-03 18:17:02    2399812 ex2.png
[ec2-user@ip-10-0-2-104 ~]$ █
```

# 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 .txt file.

The second command I ran was "aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-nikhil". This command will add the empty .txt file to my S3 bucket.

The third command I ran was "aws s3 ls s3://nextwork-vpc-endpoints-nikhil" which validated that the .txt file was added to the S3 bucket correctly.

```
[ec2-user@ip-10-0-2-104 ~]$ aws s3 ls s3://nextwork-vpc-endpoints-nikhil
2024-11-03 18:16:59      2431554 ex1.png
2024-11-03 18:17:02      2399812 ex2.png
2024-11-03 18:45:07          0 nextwork.txt
```

# 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

Now I will limit my S3 bucket access's to only traffic from your endpoint.

## Step 7 - Update route tables

In this step I will test my VPC endpoint set up.

## Step 8 - Validate endpoint connection

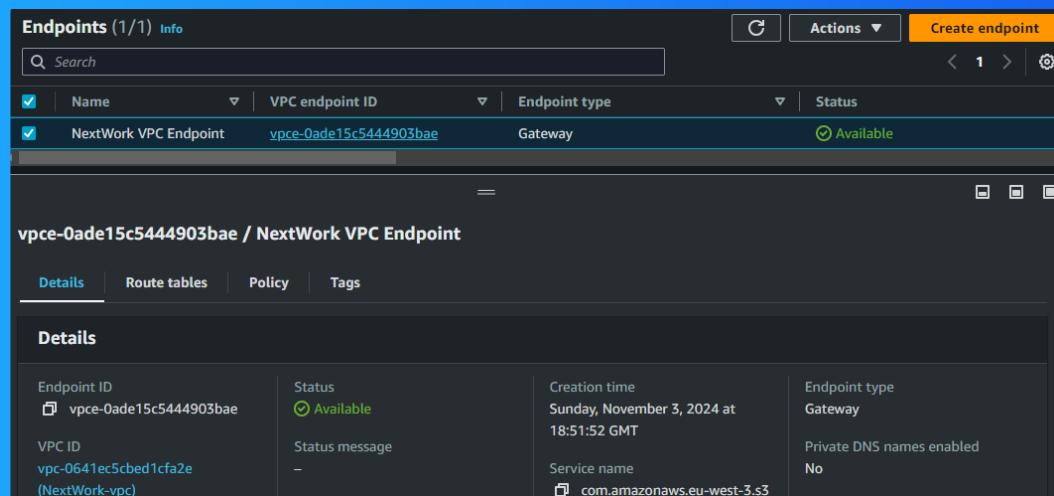
Now I will test my VPC endpoint set up again and restrict my VPC's access to my AWS environment.

# Setting up a Gateway

I set up an S3 Gateway, which is a type of endpoint used specifically for Amazon S3 and DynamoDB. It works by simply adding a route to your VPC route table that directs traffic bound for S3 to head straight for the Gateway instead of the internet.

## What are endpoints?

An endpoint is a service that allows private connections between my 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.

My bucket policy will deny 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:sourceVpce.

The screenshot shows the AWS Lambda function editor interface. On the left, there's a sidebar with 'Bucket ARN' and a dropdown menu. The main area is titled 'Policy' and contains a JSON-based policy document. The policy document includes a condition that denies access to anyone except the specific VPC endpoint. On the right, there's a sidebar with tabs for 'Edit statement', 'Remove', 'Add actions', 'Choose a service', and dropdown menus for 'Included' (S3) and 'Available' services (AMP, API Gateway, API Gateway V2, ASC).

```
Bucket ARN
arn:aws:s3:::nextwork-vpc-endpoints-nikhil

Policy

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-nikhil",  
10        "arn:aws:s3:::nextwork-vpc-endpoints-nikhil/*"  
11      ],  
12      "Condition": {  
13        "StringNotEquals": {  
14          "aws:sourceVpce": "vpce-0ade15c5444903bae"  
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 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 didn't have a route that directs traffic bound for S3 to my VPC endpoint, traffic from my EC2 instance is actually trying to get to your S3 bucket through the public internet instead.

The screenshot shows two screenshots of the AWS S3 console. The top screenshot is titled 'Block public access (bucket settings)' and the bottom one is titled 'Bucket policy'. Both screenshots display a red error message box stating 'You don't have permission to view the Block public access (bucket settings) configuration' and 'You don't have permission to get bucket policy'. These messages indicate that the user lacks the necessary IAM permissions to view or edit these specific configurations. The error message also provides a link to 'Identity and access management in Amazon S3' for more information.

**Block public access (bucket settings)**

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

► API response

**Edit**

**Bucket policy**

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](#)

**You don't have permission to get bucket policy**  
You or your AWS administrator must update your IAM permissions to allow s3:GetBucketPolicy. After you obtain the necessary permission, refresh the page. Learn more about [Identity and access management in Amazon S3](#)

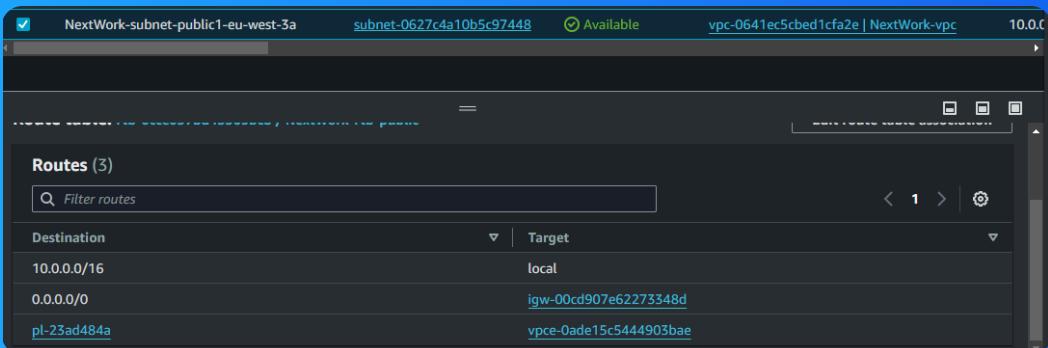
► API response

**Edit** **Delete**

# Route table updates

To update my route table, I modified the route table of the endpoint so it can access without the route of the public internet.

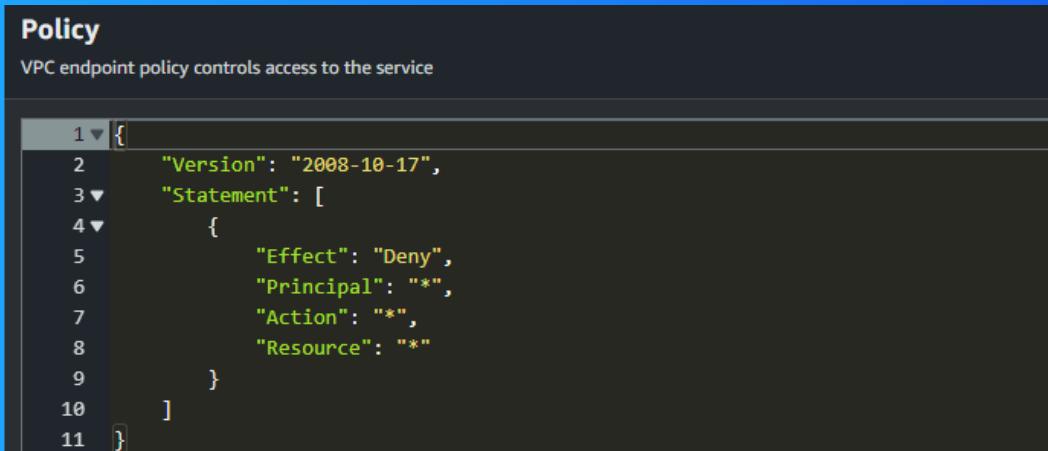
After updating my public subnet's route table, my terminal could return a response showing the list of objects in the S3 bucket.



# Endpoint policies

An endpoint policy is used to allow or deny access to the resources by permissions.

I updated my endpoint's policy by denying all access instead of allowing. I could see the effect of this right away, because I got access denied when I try to see my list of S3 bucket.



The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "Policy" and a subtitle says "VPC endpoint policy controls access to the service". The main content is a JSON document with line numbers 1 through 11 on the left:

```
1 {  
2   "Version": "2008-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Deny",  
6       "Principal": "*",  
7       "Action": "*",  
8       "Resource": "*"  
9     }  
10   ]  
11 }
```



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

