



VPC Endpoints

J

Michael K.

⌚ Successfully created VPC endpoint
vpce-0d01c42208dddd80d

Endpoints (1/1) info

Find endpoints by attribute or tag

VPC endpoint ID : vpce-0d01c42208dddd80d | X | Clear filters

Name	VPC endpoint ID	Endpoint type	Status	Service name
VPC-Endpoint-test	vpce-0d01c42208dddd80d	Gateway	Available	com.amazonaws.us-east-2

vpce-0d01c42208dddd80d / VPC-Endpoint-test

Endpoint ID vpce-0d01c42208dddd80d	Status Available	Creation time Wednesday, January 7, 2026 at 00:44:24 EST	Endpoint type Gateway
VPC ID vpc-0256f7fc8c9066cea (VPC-1-vpc)	Status message -	Service name com.amazonaws.us-east-2.s3	Private DNS names enabled No
DNS record IP type service-defined	IP address type ipv4	Service region us-east-2	Private DNS preference -
Private DNS specified domains -			



Michael K.

NextWork Student

nextwork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a logically isolated virtual network within AWS where I can launch and manage AWS resources (such as EC2 instances, RDS databases, and load balancers) using my own defined IP address range, subnets, route tables, and network gateways.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to access Amazon S3 privately and securely without having to go through the Internet.

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

One thing I didn't expect in this project was the time I spent to understand how Endpoints work.

This project took me...

This project took me 4 hours.

In the first part of my project...

Step 1 - Architecture set up

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

Step 2 - Connect to EC2 instance

In this step, I will connect directly to my EC2 instance.

Step 3 - Set up access keys

In this step, I will give my EC2 instance access to my AWS environment.

Step 4 - Interact with S3 bucket

In this step, I will head back to my EC2 instance and get my EC2 instance to access my S3 bucket.

Architecture set up

I started my project by launching a VPC, it's resources and an Instance in the VPC for the test.

I also set up an S3 Bucket with 2 files in it.

test-test-159 info

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	NextWork - Denzel is awesome.png	png	January 6, 2026, 23:36:48 (UTC-05:00)	2.3 MB	Standard
<input type="checkbox"/>	NextWork - Lelo is awesome.png	png	January 6, 2026, 23:36:47 (UTC-05:00)	2.3 MB	Standard

Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured Access Key ID, Secret Access Key, the Region and the JSON output.

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

A Secret access keys is like the password that pairs with my access key ID (your username). I need both to access AWS services.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use an IAM role with the right permissions and attach it to the instance.

Connecting to my S3 bucket

The command I ran was <<aws s3 ls>>.

The terminal responded by listing the buckets found in my account after I re-typed the previous command. This indicated that the access keys I set up worked.

```
, #_ #
~\###_#_ Amazon Linux 2023n Linux 2023
~~ \####/#/#\
~~ \###|\#\#|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023x/amazon-linux-2023
~~ V~'-'> V~'-'>
~~~ / /
~~~ . / /
~~~ / / / /
~~~ /m/ /m/ ' Last login: Wed Jan 7 04:42:23 2026 from 3.16.146.3 from 3.16.146.3
[ec2-user@ip-10-0-1-153 ~]$ aws s3 ls aws s3 ls
2025-12-13 02:17:04 mjk-portfolio-bucket!io-bucket
2026-01-07 04:36:23 test-test-159 test-159
```



J

Michael K.

NextWork Student

nextwork.org

Connecting to my S3 bucket

I also tested the command <<aws s3 ls s3://test-test-159>> which returned the objects in my bucket.

```
[ec2-user@ip-10-0-1-153 ~]$ aws s3 ls s3://test-test-159 //test-test-159
2026-01-07 04:36:48    2431554 NextWork - Denzel is awesome.png is awesome.png
2026-01-07 04:36:47   2399812 NextWork - Lelo is awesome.png is awesome.png
[ec2-user@ip-10-0-1-153 ~]$ 53 ~ ]$
```

Uploading objects to S3

To upload a new file to my bucket, I first ran the command <<sudo touch /temp/test.txt>>. This command creates the file in my instance.

The second command I ran was <<aws s3 cp /temp/test.txt s3://test-test-159>> This command will upload the file in my s3 bucket "test-test-159"

The third command I ran was <<aws s3 ls s3://test-test-159>> which validated that the file was now in my bucket.

```
ec2-user@ip-10-0-1-153 ~]$ aws s3 cp /tmp/test.txt s3://test-test-159
upload: ../../tmp/test.txt to s3://test-test-159/test.txt
ec2-user@ip-10-0-1-153 ~]$ aws s3 ls s3://test-test-159
2025-12-13 02:17:04 mjk-portfolio-bucket[1]o-bucket
2026-01-07 04:36:23 test-test-159[2]test-159
ec2-user@ip-10-0-1-153 ~]$ aws s3 ls s3://test-test-159 //test-test-159
2026-01-07 04:36:48 2431554 NextWork - Denzel is awesome.png is awesome.png
2026-01-07 04:36:47 2399812 NextWork - Lelo is awesome.png is awesome.png
2026-01-07 05:19:12 0 test.txt 0 test.txt
ec2-user@ip-10-0-1-153 ~]$ [3] $
```

In the second part of my project...

Step 5 - Set up a Gateway

In this step, I will set up a way for my VPC and S3 to communicate directly.

Step 6 - Bucket policies

In this step I'll limit my S3 bucket access's to only traffic from my endpoint.

Step 7 - Update route tables

In this step, I will Test your VPC endpoint set up and troubleshoot any connectivity issue if it pops up.

Step 8 - Validate endpoint connection

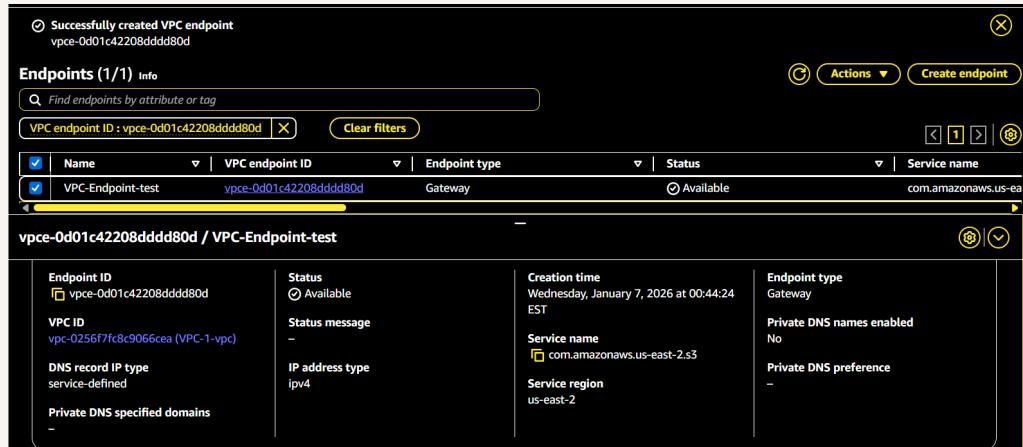
In this step, I will test my VPC endpoint set up (again) and restrict my VPC's access to my AWS environment.

Setting up a Gateway

I set up an S3 Gateway, which is a type of endpoint used specifically for Amazon S3 and DynamoDB (DynamoDB is an AWS database service). Gateways work by simply adding a route to my 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 is a service that allows private connections between my VPC and other AWS services without needing the traffic to go over the internet.



The screenshot shows the AWS VPC Endpoint console. At the top, a success message reads: "Successfully created VPC endpoint vpce-0d01c42208dddd80d". Below this, the "Endpoints (1/1)" section displays a single entry:

Name	VPC endpoint ID	Endpoint type	Status	Service name
VPCEndpoint-test	vpce-0d01c42208dddd80d	Gateway	Available	com.amazonaws.us-east-1.s3

Below the table, a detailed view for the endpoint "vpce-0d01c42208dddd80d / VPCEndpoint-test" is shown:

Endpoint ID vpce-0d01c42208dddd80d	Status Available	Creation time Wednesday, January 7, 2026 at 00:44:24 EST	Endpoint type Gateway
VPC ID vpc-0256f7fc8c9066cea (VPC-1-vpc)	Status message -	Service name com.amazonaws.us-east-1.s3	Private DNS names enabled No
DNS record IP type service-defined	IP address type IPv4	Service region us-east-1	Private DNS preference -
Private DNS specified domains -			

Bucket policies

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

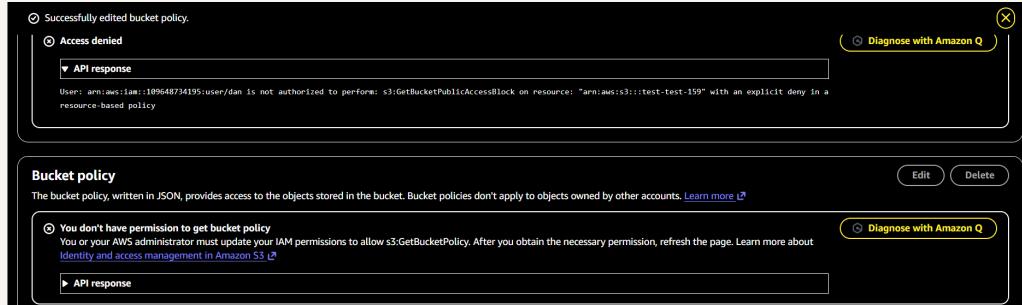
My bucket policy will deny all actions (s3:*) on my S3 bucket and its objects to everyone (Principal: "")... unless the access is from the VPC endpoint with the ID defined in aws:sourceVpce. In other words, only traffic coming from your VPC endpoint can get any access to your S3 bucket!

```
1 ▼ {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Effect": "Deny",
6             "Principal": "*",
7             "Action": "s3:*",
8             "Resource": [
9                 "arn:aws:s3:::test-test-159",
10                "arn:aws:s3:::test-test-159/*"
11            ],
12            "Condition": {
13                "StringNotEquals": {
14                    "aws:sourceVpce": "vpce-0d01c42208dd80d"
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 my VPC endpoint. This means any attempt to access my bucket from other sources, including the AWS Management Console, is blocked!

I also had to update my route table because i needed to add the gateway that will allow traffic to enter s3.



Route table updates

To update my route table, I added the route table in the Endpoints route table section.

After updating my public subnet's route table, my terminal could return the list of objects in my bucket after running <<aws s3 ls s3://test-test-159>>

The screenshot shows the AWS Route Tables page. At the top, there is a search bar and a table header with columns: Name, Route table ID, Explicit subnet associ..., Edge associations, Main, and VPC. Below the header, three route tables are listed:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0b61d1b880898ba7	-	-	Yes	vpc-0256f7fc8c9066cea VPC..
<input checked="" type="checkbox"/> VPC-1-rtb-public	rtb-0fe77a5a9d4849d39	subnet-05ddcf36369b2f...	-	No	vpc-0256f7fc8c9066cea VPC..
-	rtb-030cc0a239ee032	-	-	Yes	vpc-045b6eb9764bf4ca7

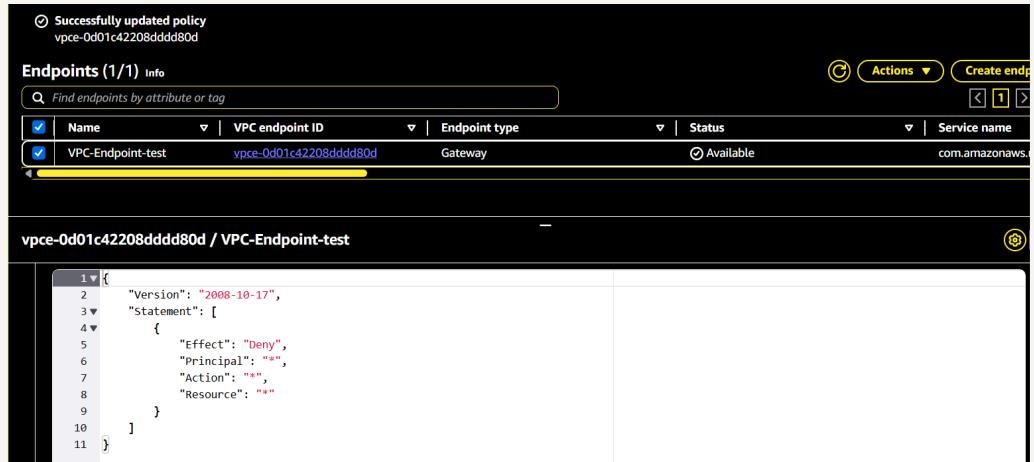
Below the table, a specific route table is selected: "rtb-0fe77a5a9d4849d39 / VPC-1-rtb-public". This table has five tabs: Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The Routes tab is active, showing three routes:

Destination	Target	Status	Propagated	Route Origin
pl-7ba54012	vpc-e-0d01c42208ddd80d	Active	No	Create Route
0.0.0.0/0	igw-08be40e7ab92bef34	Active	No	Create Route
10.0.0.0/16	local	Active	No	Create Route Table

Endpoint policies

An endpoint policy is one which determines and control access to s3 through the control of the Gateway.

I updated my endpoint's policy by changing the "Effect" to <<Deny>>. I could see the effect of this right away, because the gateway was blocked and i couldnt access s3 anymore.



The screenshot shows the AWS VPC Endpoint Policies interface. At the top, a success message says "Successfully updated policy vpce-0d01c42208ddd80d". Below it, a table lists endpoints. One row for "VPC-Endpoint-test" is selected, showing its VPC endpoint ID and status. A modal window displays the JSON policy for this endpoint:

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



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

