VPC Endpoints | S3 Gateways, Bucket Policies

By Haroon Zaman | December 2025

Introduction

In this project, I connected my VPC **directly to Amazon S3**, allowing resources inside the VPC to interact with S3 securely and efficiently.

Instead of sending traffic through the public internet, this setup allows communication to stay within AWS's internal network. This improves **security**, **performance**, and **cost efficiency**, since traffic takes a shorter and more controlled path to its destination.

What I Did in This Project

- Set up a **VPC with an EC2 instance**
- Configured **IAM access keys** to allow AWS CLI access from the instance
- Created an **Amazon S3 bucket**
- Uploaded and listed S3 objects using **AWS CLI commands**
- Verified that the EC2 instance inside the VPC could interact with S3 successfully

---

Why This Matters

By enabling direct access between a VPC and S3:

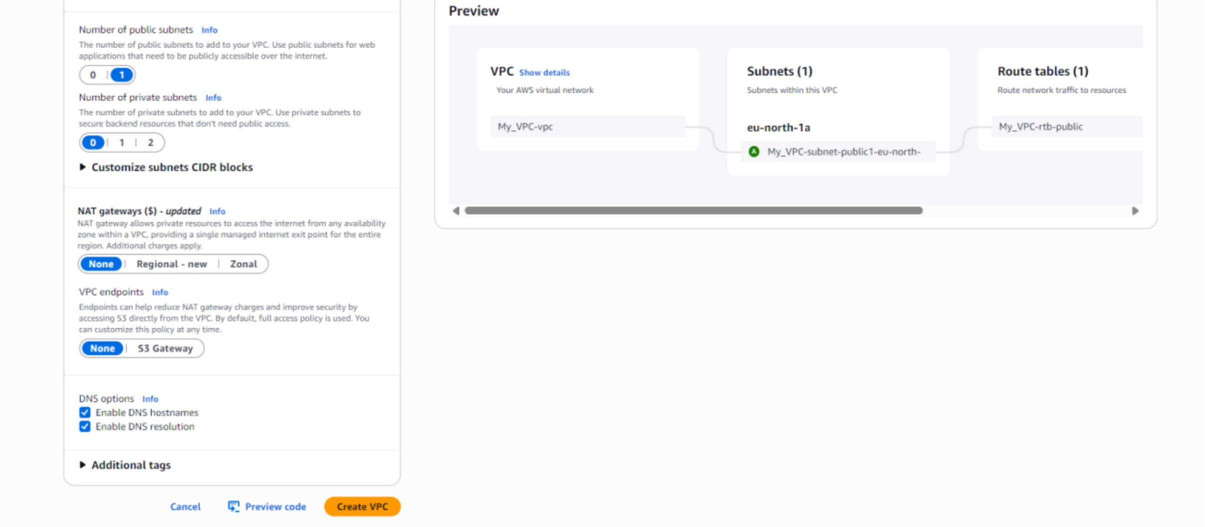
- Network traffic stays **private and secure**
- Requests reach their destination **faster**
- The setup is **cheaper**, as it avoids unnecessary internet routing
- Permissions are tightly controlled using **IAM**

This project tied together everything I learned throughout the AWS networking series — VPCs, EC2, IAM, and S3 — into a single, real-world use case.

Creating the VPC

- First, I created a new VPC and named it **My_VPC_1**.
- I chose **one Availability Zone**, since this project does not require multi-AZ availability.
- I created **one public subnet** and **no private subnet**.

    - This is because the EC2 instance in this project needs to be publicly reachable for testing S3 access, and a private subnet is not required at this stage.



Number of public subnets Info

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | [✓ 1]

Number of private subnets Info

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

[✓ 0] | 1 | 2

▶ Customize subnets CIDR blocks

NAT gateways ($) - *updated* Info

NAT gateway allows private resources to access the internet from any availability zone within a VPC, providing a single managed internet exit point for the entire region. Additional charges apply.

[✓ None] | Regional - new | Zonal

VPC endpoints Info

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

[✓ None] | S3 Gateway

DNS options Info

☑ Enable DNS hostnames

☑ Enable DNS resolution

▶ Additional tags

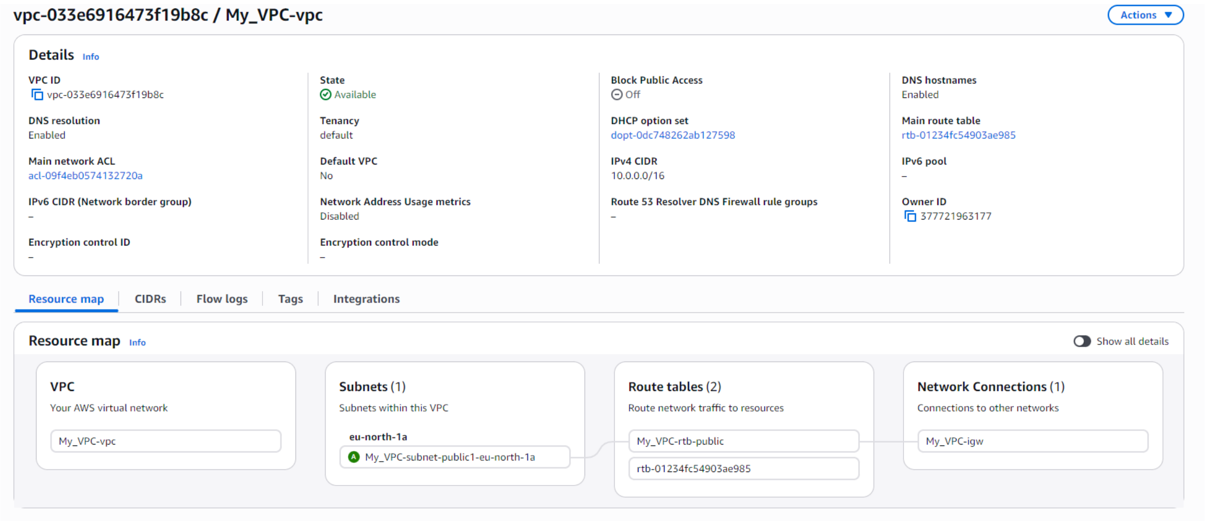
Preview

VPC Show details
Your AWS virtual network
My_VPC-vpc

Subnets (1)
Subnets within this VPC
eu-north-1a
My_VPC-subnet-public1-eu-north-

Route tables (1)
Route network traffic to resources
My_VPC-rtb-public

- I selected **No NAT Gateway**.
    - A NAT Gateway is only needed when **private subnets** require outbound internet access.
    - Since I did not create a private subnet, a NAT Gateway was not necessary.
- I selected **No S3 Gateway Endpoint**.
    - This is because I wanted to create and configure S3 access **separately and in detail** later in the project.
- After reviewing the configuration, I clicked **Create VPC**, and the VPC was created successfully.



vpc-033e6916473f19b8c / My_VPC-vpc

Details Info

VPC ID vpc-033e6916473f19b8c	State Available	Block Public Access Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0dc748262ab127598	Main route table rtb-01234fc54903ae985
Main network ACL acl-09f4eb0574132720a	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –
IPv6 CIDR (Network border group) –	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 377721963177
Encryption control ID –	Encryption control mode –		

Resource map Info

VPC
Your AWS virtual network
My_VPC-vpc

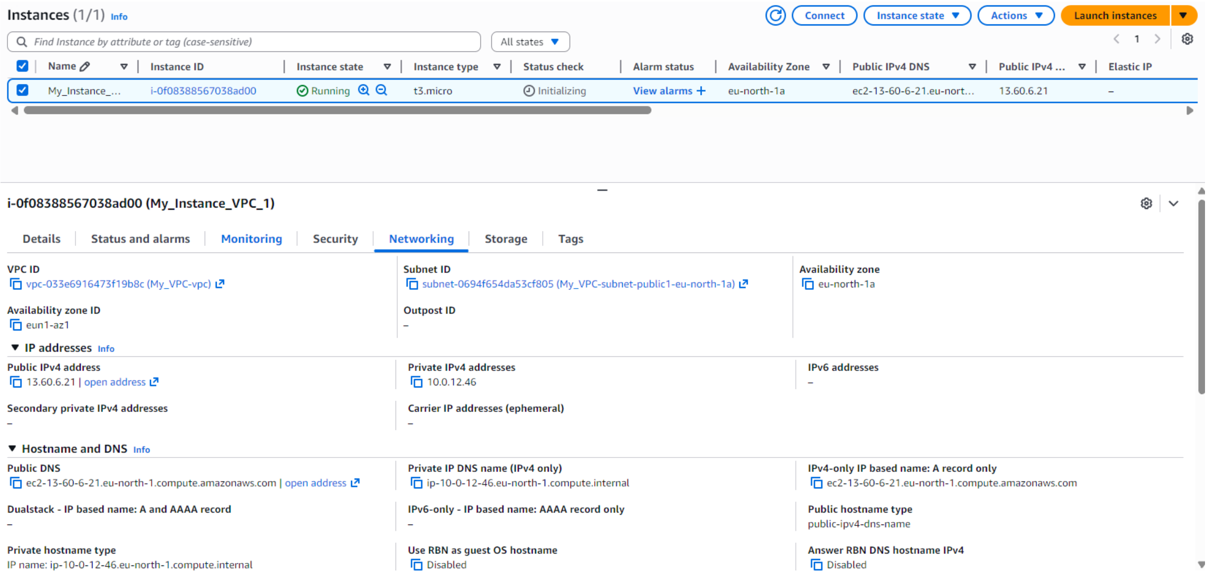
Subnets (1)
Subnets within this VPC
eu-north-1a
My_VPC-subnet-public1-eu-north-1a

Route tables (2)
Route network traffic to resources
My_VPC-rtb-public
rtb-01234fc54903ae985

Network Connections (1)
Connections to other networks
My_VPC-igw

Creating the EC2 Instance

- Next, I created an EC2 instance and named it **My_Instance_VPC_1**.
- I selected **Amazon Linux** as the operating system.
- I chose the **t3 instance type**.



Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	My_Instance_...	i-0f08388567038ad00	Running	t3.micro	Initializing	View alarms +	eu-north-1a	ec2-13-60-6-21.eu-nort...	13.60.6.21	–

i-0f08388567038ad00 (My_Instance_VPC_1)

VPC ID
vpc-033e6916473f19b8c (My_VPC-vpc)

Subnet ID
subnet-0694f654da53cf805 (My_VPC-subnet-public1-eu-north-1a)

Availability zone
eu-north-1a

Availability zone ID
eun1-az1

Outpost ID
–

▼ IP addresses Info

Public IPv4 address
13.60.6.21 | open address

Private IPv4 addresses
10.0.12.46

IPv6 addresses
–

Secondary private IPv4 addresses
–

Carrier IP addresses (ephemeral)
–

▼ Hostname and DNS Info

Public DNS
ec2-13-60-6-21.eu-north-1.compute.amazonaws.com | open address

Private IP DNS name (IPv4 only)
ip-10-0-12-46.eu-north-1.compute.internal

IPv4-only IP based name: A record only
ec2-13-60-6-21.eu-north-1.compute.amazonaws.com

Dualstack - IP based name: A and AAAA record
–

IPv6-only - IP based name: AAAA record only
–

Public hostname type
public-ipv4-dns-name

Private hostname type
IP name: ip-10-0-12-46.eu-north-1.compute.internal

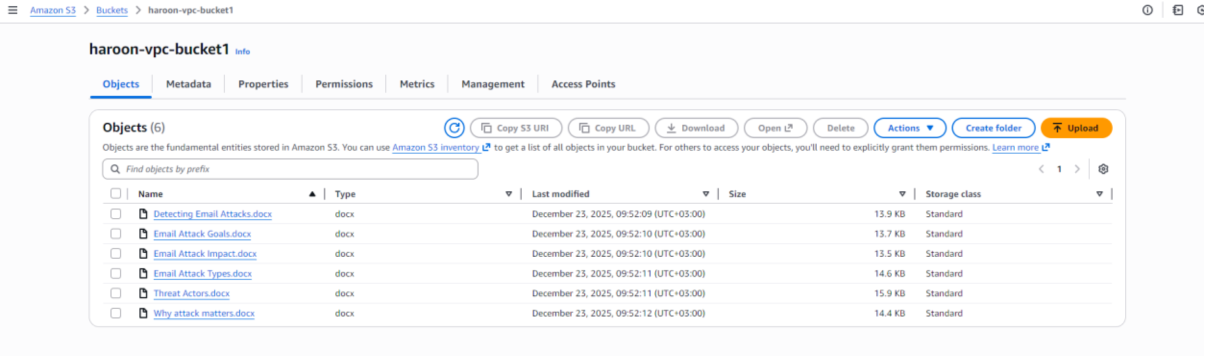
Use RBN as guest OS hostname
Disabled

Answer RBN DNS hostname IPv4
Disabled

- In the **Networking** section, I selected **My_VPC_1**.
- I selected the **public subnet** of the VPC.
- I enabled **Auto-assign Public IP**.
- In the **Firewall (Security Group)** section, I created a **new security group**.
- After reviewing the settings, I clicked **Create instance**, and the instance was launched successfully.

Creating the S3 Bucket

- Next, I created an Amazon S3 bucket.
- I named the bucket **haroon-vpc-bucket1**.



haroon-vpc-bucket1 Info

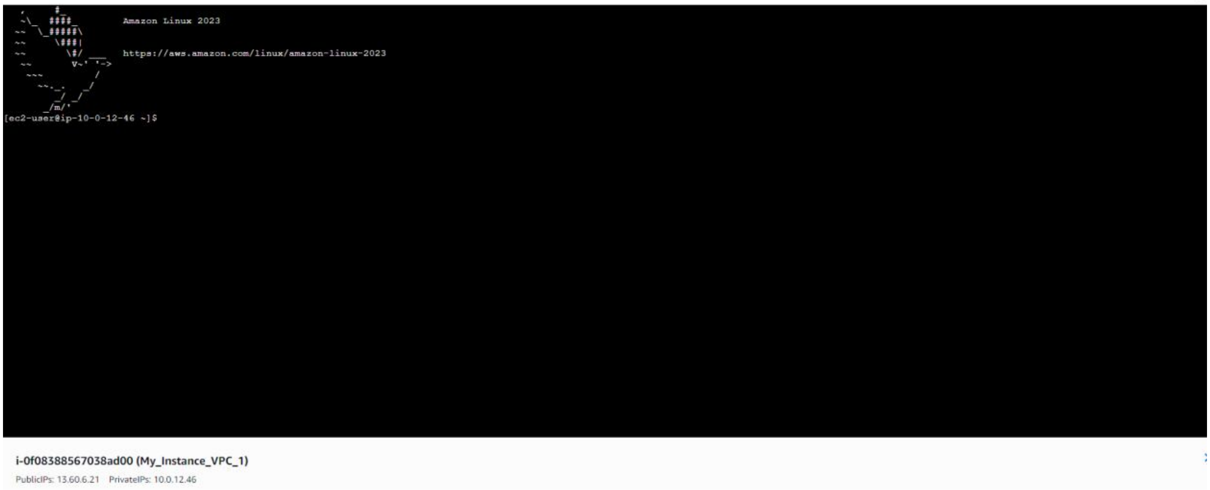
Objects (6)

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

□	Name	Type	Last modified	Size	Storage class
□	Detecting Email Attacks.docx	docx	December 23, 2025, 09:52:09 (UTC+03:00)	13.9 KB	Standard
□	Email Attack Goals.docx	docx	December 23, 2025, 09:52:10 (UTC+03:00)	13.7 KB	Standard
□	Email Attack Impact.docx	docx	December 23, 2025, 09:52:10 (UTC+03:00)	13.5 KB	Standard
□	Email Attack Types.docx	docx	December 23, 2025, 09:52:11 (UTC+03:00)	14.6 KB	Standard
□	Threat Actors.docx	docx	December 23, 2025, 09:52:11 (UTC+03:00)	15.9 KB	Standard
□	Why attack matters.docx	docx	December 23, 2025, 09:52:12 (UTC+03:00)	14.4 KB	Standard

- I left all the settings on **default**.
- After reviewing the configuration, I clicked **Create bucket**, and the bucket was created successfully.



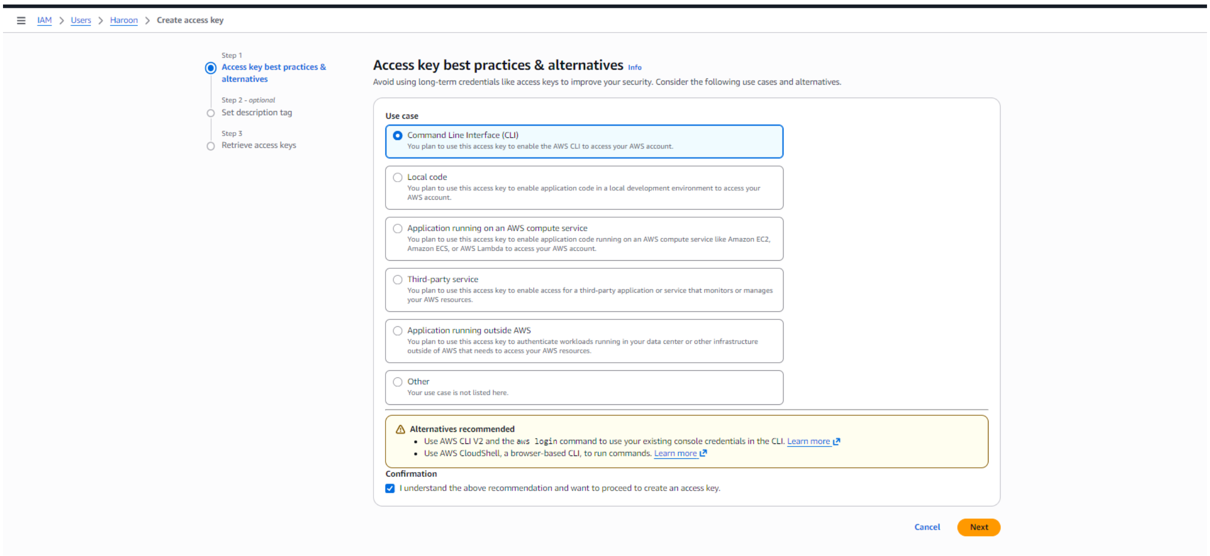
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-12-46 ~]$
```

i-0f08388567038ad00 (My_Instance_VPC_1)

PublicIPs: 13.60.6.21   PrivateIPs: 10.0.12.46

Connecting to the EC2 Instance

- After creating the instance, I connected to **My_Instance_VPC_1** using the **Connect** option from the EC2 console.
- The connection was established successfully.



☰ IAM > Users > Haroon > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

○ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

○ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

○ Other
Your use case is not listed here.

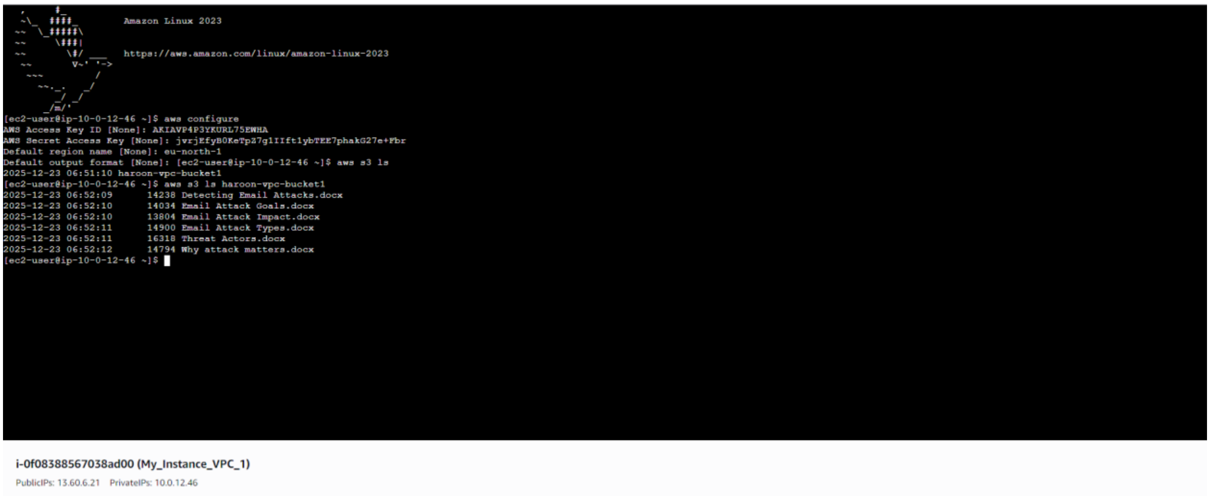
⚠ Alternatives recommended
- Use AWS CLI V2 and the `aws login` command to use your existing console credentials in the CLI. Learn more
- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more

Confirmation

☑ I understand the above recommendation and want to proceed to create an access key.

Creating the Access Key in IAM

- Next, I created an **access key** for my IAM user.
- In the access key setup, I selected the **CLI** option as the use case.
- I added a **description** to identify the key.
- After that, I clicked **Create access key**, and the access key was generated successfully.



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-12-46 ~]$ aws configure
AWS Access Key ID [None]: AKIAVP4P3YKURL75EWHA
AWS Secret Access Key [None]: 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
Default region name [None]: eu-north-1
Default output format [None]: [ec2-user@ip-10-0-12-46 ~]$ aws s3 ls
2025-12-23 06:51:10 haroon-vpc-bucket1
[ec2-user@ip-10-0-12-46 ~]$ aws s3 ls haroon-vpc-bucket1
2025-12-23 06:52:09      14238 Detecting Email Attacks.docx
2025-12-23 06:52:10      14034 Email Attack Goals.docx
2025-12-23 06:52:10      13804 Email Attack Impact.docx
2025-12-23 06:52:11      14900 Email Attack Types.docx
2025-12-23 06:52:11      16318 Threat Actors.docx
2025-12-23 06:52:12      14794 Why attack matters.docx
[ec2-user@ip-10-0-12-46 ~]$
```

i-0f08388567038ad00 (My_Instance_VPC_1)

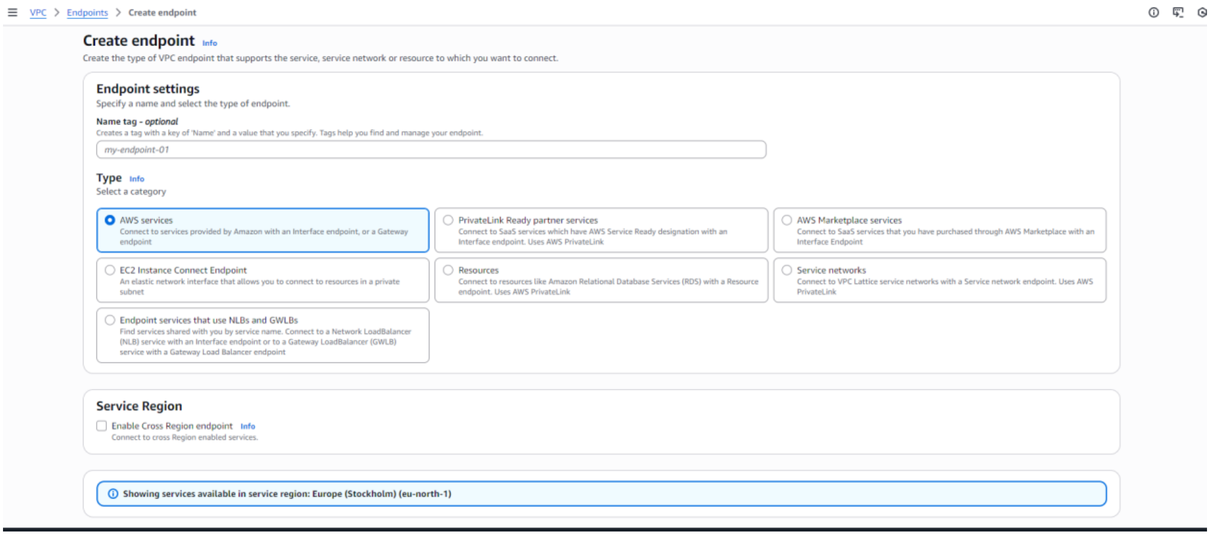
PublicIPs: 13.60.6.21   PrivateIPs: 10.0.12.46

Configuring Credentials and AWS CLI Settings

- After creating the access key, I **copied the credentials** (Access Key ID and Secret Access Key).
- I went back to the connected EC2 instance and entered the **Access Key ID** and **Secret Access Key** when prompted.
- Next, I was asked for the **Default region name**.
    - I entered **eu-north-1**, which is the region where my resources are created.
- Then I was asked for the **Default output format**.
    - The **default output format** controls **how AWS CLI command results are displayed** in the terminal.
    - Available options include:
        - **json** – structured output (default)
        - **text** – plain text output
        - **table** – formatted table output
        - **yaml** – human-readable structured format
- I left this field **empty**.
    - When left empty, AWS CLI automatically uses **JSON** as the default output format.
- After completing these steps, the AWS CLI was fully configured.
- I then tested access to the S3 bucket by running:
  `aws s3 ls s3://haroon-vpc-bucket1`
- The command returned the **list of files already uploaded** in the bucket, confirming successful access.

Creating an S3 VPC Endpoint

- An **endpoint** in AWS allows resources inside a VPC to connect to AWS services **privately**, without sending traffic over the public internet.
- This improves **security**, **performance**, and reduces exposure to external networks.
- A **Gateway Endpoint** is a special type of endpoint used specifically for **Amazon S3** and **DynamoDB**.
- Instead of using the internet gateway, traffic to S3 is routed internally through the AWS network using the VPC route table.



VPC > Endpoints > Create endpoint

Create endpoint Info

Create the type of VPC endpoint that supports the service, service network or resource to which you want to connect.

Endpoint settings

Specify a name and select the type of endpoint.

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify. Tags help you find and manage your endpoint.

my-endpoint-01

Type Info

Select a category

● AWS services
Connect to services provided by Amazon with an Interface endpoint, or a Gateway endpoint

○ PrivateLink Ready partner services
Connect to SaaS services which have AWS Service Ready designation with an Interface endpoint. Uses AWS PrivateLink

○ AWS Marketplace services
Connect to SaaS services that you have purchased through AWS Marketplace with an Interface Endpoint

○ EC2 Instance Connect Endpoint
An elastic network interface that allows you to connect to resources in a private subnet

○ Resources
Connect to resources like Amazon Relational Database Services (RDS) with a Resource endpoint. Uses AWS PrivateLink

○ Service networks
Connect to VPC Lattice service networks with a Service network endpoint. Uses AWS PrivateLink

○ Endpoint services that use NLBs and GWLBs
Find services shared with you by service name. Connect to a Network LoadBalancer (NLB) service with an Interface endpoint or to a Gateway LoadBalancer (GWLB) service with a Gateway Load Balancer endpoint

Service Region

□ Enable Cross Region endpoint Info
Connect to cross Region enabled services.

Showing services available in service region: Europe (Stockholm) (eu-north-1)

Creating the Endpoint

- I went to the **VPC console**.
- From the left-side panel, I selected **Endpoints**.
- I clicked on **Create endpoint**.
- I named the endpoint **My_S3_EndPoint**.

Endpoint Type Selection

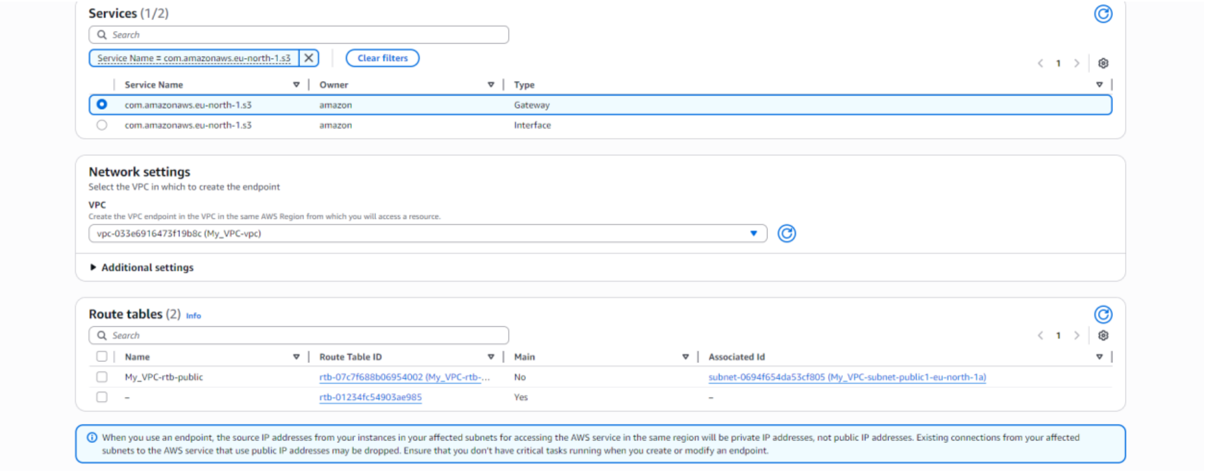
- I selected **AWS services** as the endpoint type because:
  - I wanted to connect my VPC to an AWS-managed service (Amazon S3).
  - This option supports **Gateway Endpoints** for S3.

Other options available were:

- **PrivateLink Ready partner services** – used for third-party SaaS services.
- **AWS Marketplace services** – used for services purchased from AWS Marketplace.
- **EC2 Instance Connect Endpoint** – used for SSH access to private EC2 instances.
- **Resources** – used to connect to services like RDS using PrivateLink.
- **Service networks** – used with VPC Lattice for service-to-service communication.

Why This Matters

- My EC2 instance can now access S3 **privately**.
- Traffic stays inside the AWS network.
- This follows AWS best practices for **secure VPC design**.



Services (1/2)

Service Name = com.amazonaws.eu-north-1.s3

	Service Name	Owner	Type
●	com.amazonaws.eu-north-1.s3	amazon	Gateway
○	com.amazonaws.eu-north-1.s3	amazon	Interface

Network settings

Select the VPC in which to create the endpoint

VPC

Create the VPC endpoint in the VPC in the same AWS Region from which you will access a resource.

vpc-033e6916473f19b8c (My_VPC-vpc)

▶ Additional settings

Route tables (2) Info

	Name	Route Table ID	Main	Associated Id
□	My_VPC-rtb-public	rtb-07c7f688b06954002 (My_VPC-rtb-...	No	subnet-0694f654da53cf805 (My_VPC-subnet-public1-eu-north-1a)
□	-	rtb-01234fc54903ae985	Yes	-

When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.

Selecting the S3 Service for the Endpoint

- In the **Service** section, I searched for **S3**.
- From the list, I selected the service name that **ends with `s3` and has the type `Gateway`**.

What "Gateway" Means

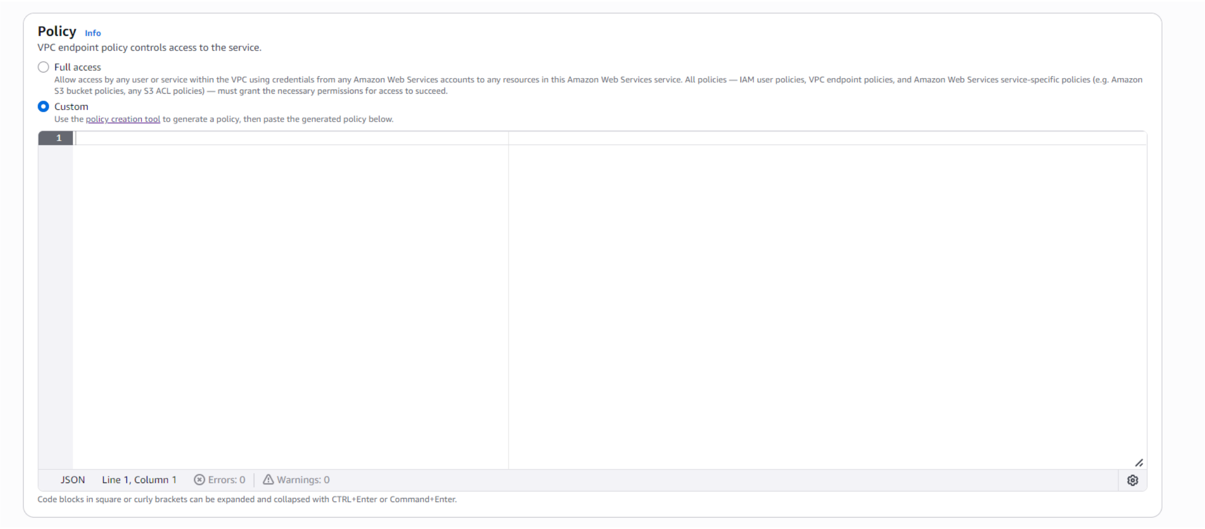
- A **Gateway Endpoint** is a special endpoint type used for **Amazon S3** (and DynamoDB).
- It allows traffic from the VPC to reach S3 **without using the internet**.
- Instead of creating network interfaces, it works by adding routes to the **VPC route table**.
- This means S3 traffic stays **inside the AWS network**, improving security and reliability.

Network Settings

- In the **Network settings** section, I selected the **VPC I created earlier**.
- I then selected the **route table that was automatically created with the VPC**.
- This ensures that traffic destined for S3 is routed through the **Gateway Endpoint** instead of the internet.

Endpoint Policy (Optional)

- While creating the endpoint, we can also attach an **endpoint policy**.
- An endpoint policy controls **what actions are allowed or denied** when accessing S3 through the endpoint.
- It works like an IAM policy but applies **only to traffic using this endpoint**.
- For this project, I left the policy as **Full access**, which allows all S3 actions from the VPC.
- In a real production setup, this can be restricted to specific buckets or actions for better security.



Policy Info

VPC endpoint policy controls access to the service.

○ Full access

Allow access by any user or service within the VPC using credentials from any Amazon Web Services accounts to any resources in this Amazon Web Services service. All policies — IAM user policies, VPC endpoint policies, and Amazon Web Services service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must grant the necessary permissions for access to succeed.

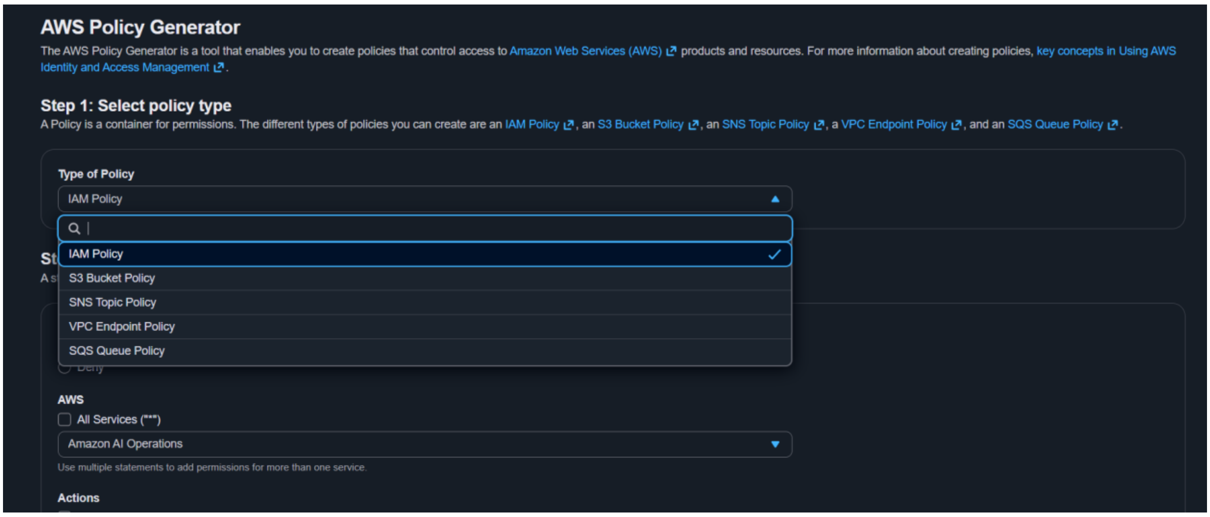
● Custom

Use the policy creation tool to generate a policy, then paste the generated policy below.

JSON Line 1, Column 1 Errors: 0 Warnings: 0

Code blocks in square or curly brackets can be expanded and collapsed with CTRL+Enter or Command+Enter.

- We can also use the **Policy Generator** while creating the endpoint.
- The policy generator helps create endpoint policies **without writing JSON manually**.
- It lets us select allowed actions, services, and resources step by step.
- This reduces mistakes and makes policy creation easier, especially for beginners.
- The generated policy is automatically attached to the endpoint.



AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, key concepts in Using AWS Identity and Access Management.

Step 1: Select policy type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Type of Policy: IAM Policy

- IAM Policy ✓
- S3 Bucket Policy
- SNS Topic Policy
- VPC Endpoint Policy
- SQS Queue Policy

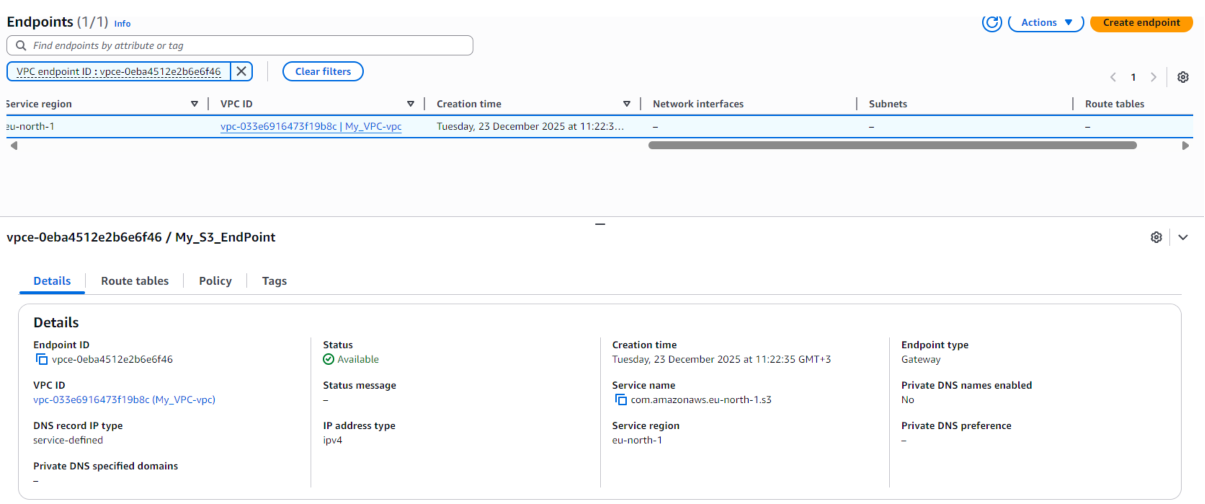
AWS

□ All Services ("*")

Amazon AI Operations

Use multiple statements to add permissions for more than one service.

Actions



Endpoints (1/1) Info

VPC endpoint ID : vpce-0eba4512e2b6e6f46

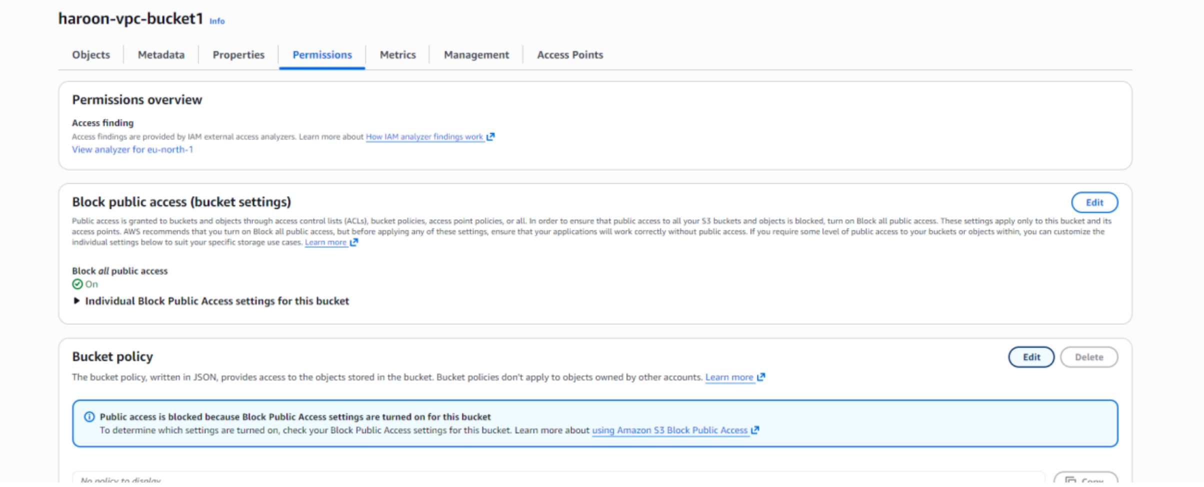
Service region	VPC ID	Creation time	Network interfaces	Subnets	Route tables
eu-north-1	vpc-033e6916473f19b8c | My_VPC-vpc	Tuesday, 23 December 2025 at 11:22:3...	–	–	–

vpce-0eba4512e2b6e6f46 / My_S3_EndPoint

Details

Field	Value
Endpoint ID	vpce-0eba4512e2b6e6f46
VPC ID	vpc-033e6916473f19b8c (My_VPC-vpc)
DNS record IP type	service-defined
Private DNS specified domains	–
Status	Available
Status message	–
IP address type	ipv4
Creation time	Tuesday, 23 December 2025 at 11:22:35 GMT+3
Service name	com.amazonaws.eu-north-1.s3
Service region	eu-north-1
Endpoint type	Gateway
Private DNS names enabled	No
Private DNS preference	–

- After reviewing all the settings, I clicked **Create endpoint**.
- The endpoint was created successfully and linked to my VPC.
- This allowed resources inside the VPC to access **Amazon S3 privately**, without using the public internet.



haroon-vpc-bucket1

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)

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

Block *all* public access

On

▶ Individual Block Public Access settings for this bucket

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

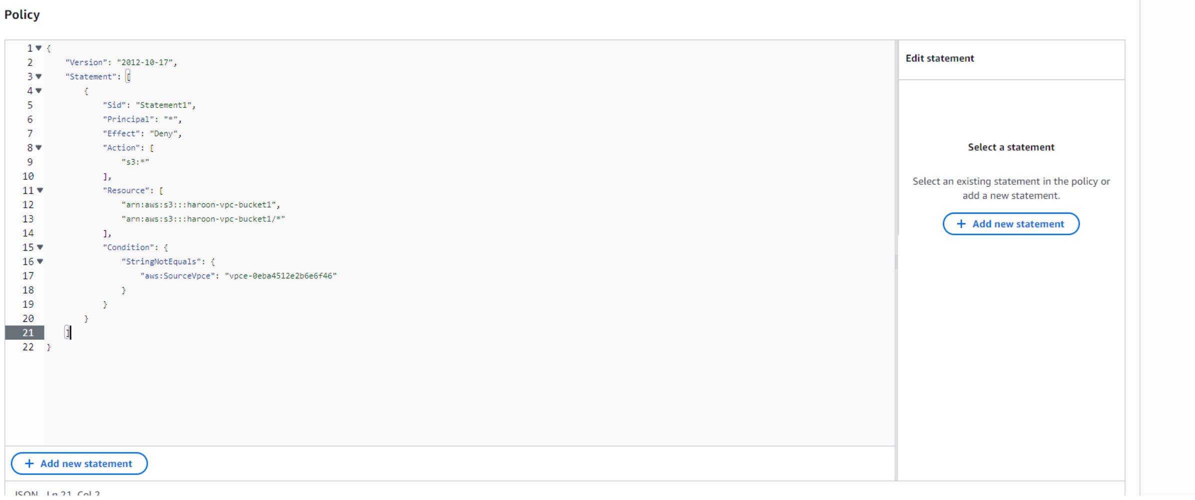
Public access is blocked because Block Public Access settings are turned on for this bucket

To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about using Amazon S3 Block Public Access

No policy to display

Creating a Super Secure S3 Bucket

- After creating the VPC endpoint, I wanted to make sure my S3 bucket could **only** be accessed through that endpoint and **not from the public internet**.
- A "super secure" S3 bucket means **all access is blocked by default**, except traffic coming from a trusted source—in this case, my **VPC endpoint**.
- To do this, I went to the **S3 console** and selected my bucket.
- I opened the **Permissions** tab.
- In the **Bucket policy** section, I clicked **Edit**.



Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Principal": "*",
            "Effect": "Deny",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::haroon-vpc-bucket1",
                "arn:aws:s3:::haroon-vpc-bucket1/*"
            ],
            "Condition": {
                "StringNotEquals": {
                    "aws:SourceVpce": "vpce-0eba4512e2b6e6f46"
                }
            }
        }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

JSON Ln 21, Col 2

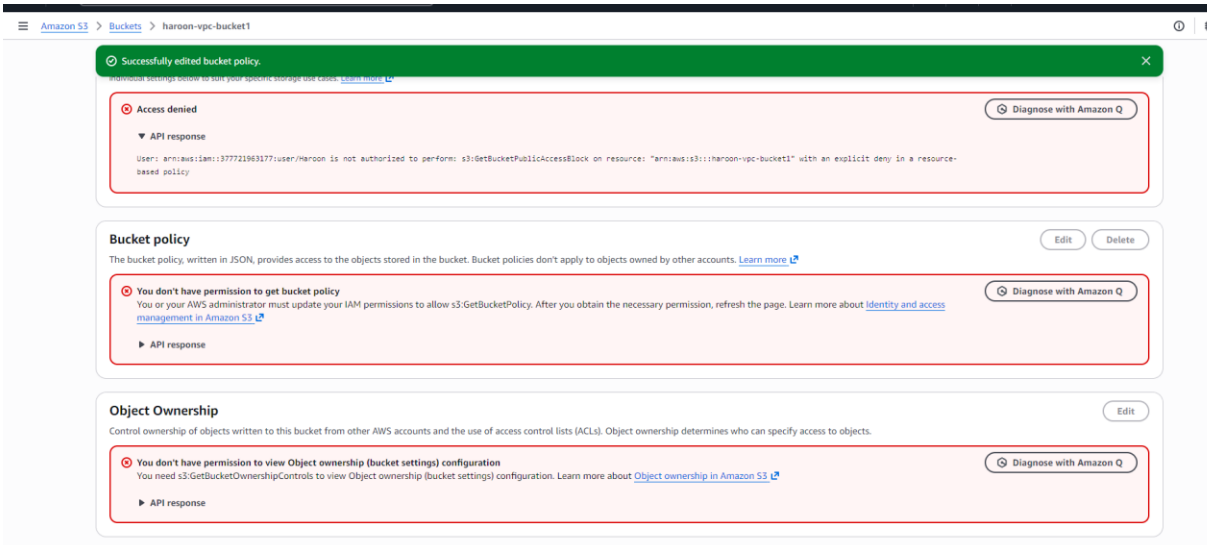
- In the next window, I added this **bucket policy**.
- This policy **denies all S3 actions (`s3:*`)** on the bucket and its objects for **everyone** by default.
- The `Principal: "*"` means the rule applies to **any user or service**.
- The condition `StringNotEquals` checks the source of the request.

- If the request **does NOT come from my VPC endpoint (`aws:SourceVpce`)**, access is denied.
- Only traffic that **originates from the specified S3 VPC endpoint** is allowed.
- Any request coming from:
  - The public internet
  - Another VPC
  - A local machine

  will be blocked.

- This confirms that:

  - My S3 bucket is **not accessible publicly**
  - My EC2 instance can access S3 **only through the VPC endpoint**
  - The data never travels over the public internet



Successfully edited bucket policy.

individual settings below to suit your specific storage use cases. Learn more

Access denied

▼ API response

```
User: arn:aws:iam::377721963177:user/Haroon is not authorized to perform: s3:GetBucketPublicAccessBlock on resource: "arn:aws:s3:::haroon-vpc-bucket1" with an explicit deny in a resource-based policy
```

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

Object Ownership

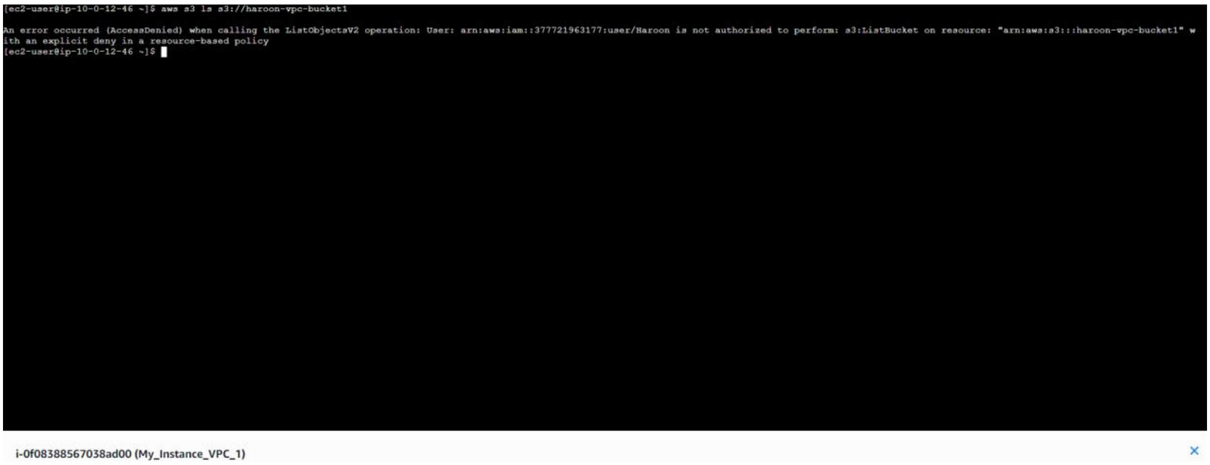
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

You don't have permission to view Object ownership (bucket settings) configuration
You need s3:GetBucketOwnershipControls to view Object ownership (bucket settings) configuration. Learn more about Object ownership in Amazon S3

▶ API response

- After applying this bucket policy, **all access from the AWS Console (GUI)** was denied.
- This is expected behavior because the **AWS Console accesses S3 over the public internet**, not through the VPC endpoint.
- Since the policy allows access **only when `aws:SourceVpce` matches my VPC endpoint**, any GUI-based access is blocked.
- The EC2 instance inside the VPC can still access the bucket because its traffic goes **through the S3 VPC endpoint**.
- This confirms the policy is working exactly as intended:

  - ❌ GUI / public access blocked
  - ✅ VPC endpoint access allowed

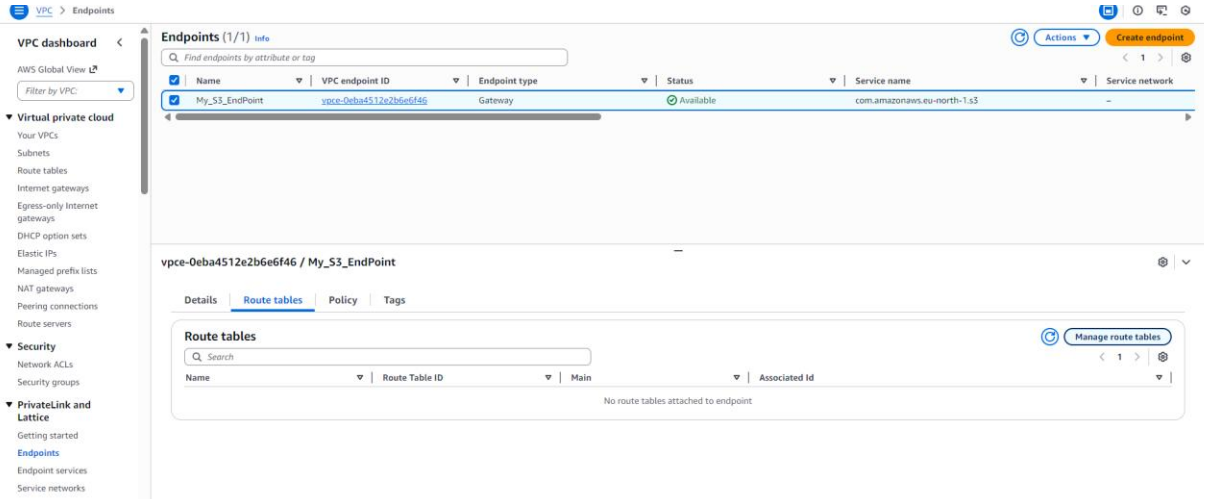


```
[ec2-user@ip-10-0-12-46 ~]$ aws s3 ls s3://haroon-vpc-bucket1

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User: arn:aws:iam::377721963177:user/Haroon is not authorized to perform: s3:ListBucket on resource: "arn:aws:s3:::haroon-vpc-bucket1" w
ith an explicit deny in a resource-based policy
[ec2-user@ip-10-0-12-46 ~]$ 
```

i-0f08388567038ad00 (My_Instance_VPC_1)

- Next, I tried to access the S3 bucket from the EC2 instance.
- Access was denied because the bucket policy allows requests **only through the S3 VPC endpoint**, and the request did not meet that condition at that time.



Endpoints (1/1) Info

Find endpoints by attribute or tag

☑	Name	VPC endpoint ID	Endpoint type	Status	Service name	Service network
☑	My_S3_EndPoint	vpce-0eba4512e2b6e6f46	Gateway	Available	com.amazonaws.eu-north-1.s3	–

vpce-0eba4512e2b6e6f46 / My_S3_EndPoint

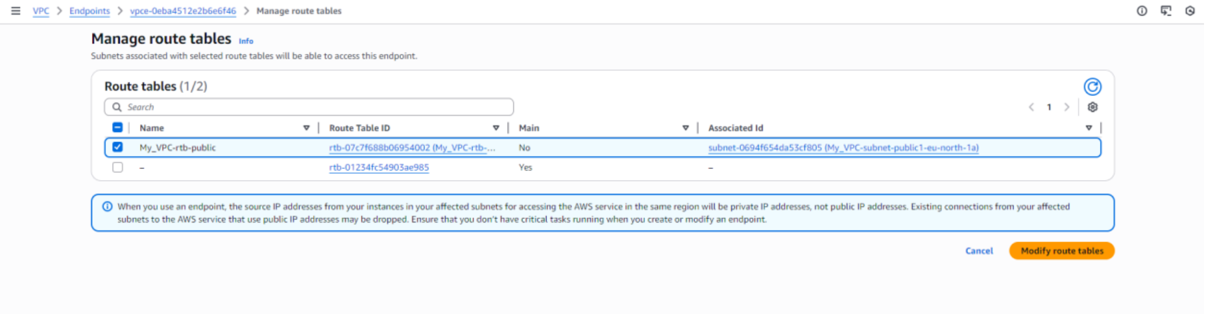
Route tables

Search

Name	Route Table ID	Main	Associated Id

No route tables attached to endpoint

- The problem was that the subnet’s **route table was not routing traffic to the S3 VPC endpoint**.
- Because there was no route pointing S3 traffic to the endpoint, the EC2 instance was trying to reach S3 **through the public internet instead**.
- Since the bucket policy blocks all public internet access, those requests were denied.
- Once the route table was updated to send S3-bound traffic to the **VPC endpoint**, the EC2 instance was able to access the bucket successfully.



VPC > Endpoints > vpce-0eba4512e2b6e6f46 > Manage route tables
Cancel
Modify route tables

Manage route tables Info

Subnets associated with selected route tables will be able to access this endpoint.

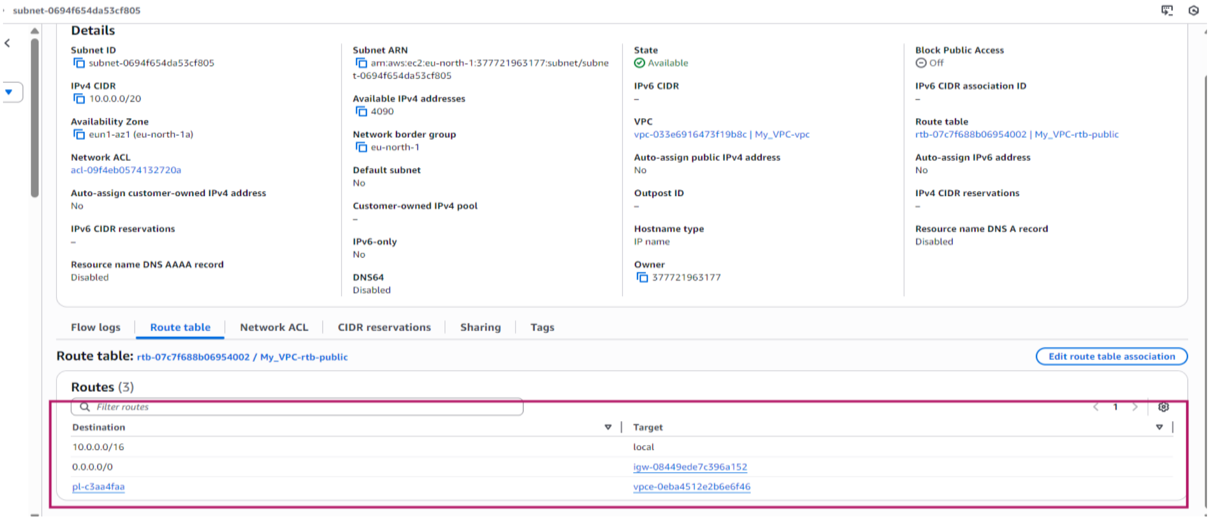
Route tables (1/2)

Search

☐	Name	Route Table ID	Main	Associated Id
☑	My_VPC-rtb-public	rtb-07c7f688b06954002 (My_VPC-rtb-...	No	subnet-0694f654da53cf805 (My_VPC-subnet-public1-eu-north-1a)
☐	–	rtb-01234fc54903ae985	Yes	–

When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.

- Next, I selected the **public route table** of my VPC.
- From the **VPC endpoint route tables** section, I chose **Modify route tables** to associate the route table with the S3 endpoint.



subnet-0694f654da53cf805

Details

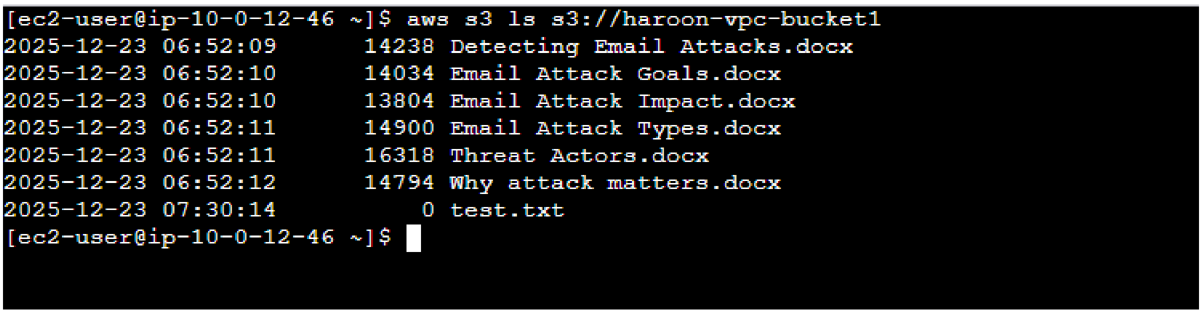
Subnet ID subnet-0694f654da53cf805	Subnet ARN arn:aws:ec2:eu-north-1:377721963177:subnet/subnet-0694f654da53cf805	State Available	Block Public Access Off
IPv4 CIDR 10.0.0.0/20	Available IPv4 addresses 4090	IPv6 CIDR –	IPv6 CIDR association ID –
Availability Zone eun1-az1 (eu-north-1a)	Network border group eu-north-1	VPC vpc-033e6916473f19b8c | My_VPC-vpc	Route table rtb-07c7f688b06954002 | My_VPC-rtb-public
Network ACL acl-09f4eb0574132720a	Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No
Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool –	Outpost ID –	IPv4 CIDR reservations –
IPv6 CIDR reservations –	IPv6-only No	Hostname type IP name	Resource name DNS A record Disabled
Resource name DNS AAAA record Disabled	DNS64 Disabled	Owner 377721963177	

Route table: rtb-07c7f688b06954002 / My_VPC-rtb-public

Routes (3)

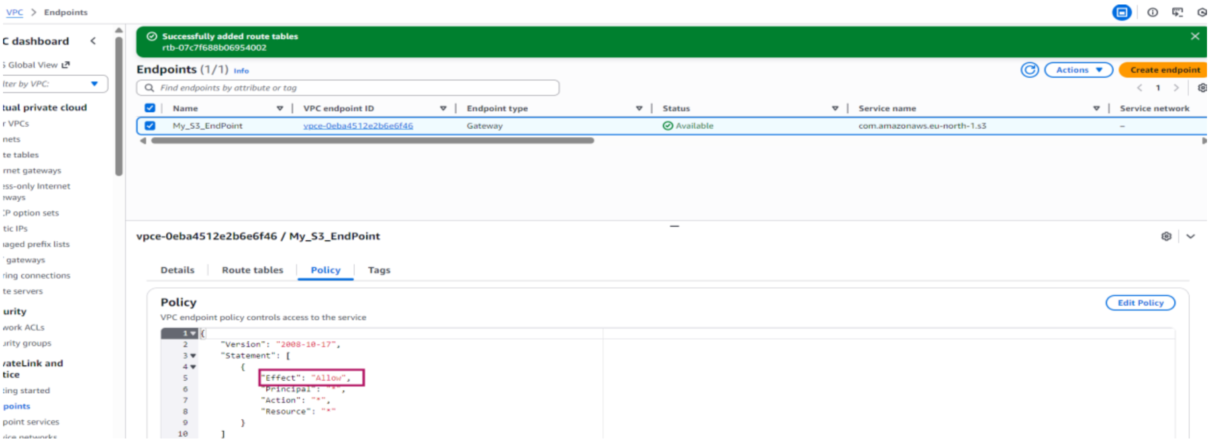
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-08449ede7c396a152
pl-c3aa4faa	vpce-0eba4512e2b6e6f46

- After associating the route table, all required routes were added successfully.
- This included the route that directs traffic from the **VPC endpoint to the S3 bucket**, ensuring private connectivity to S3.



```
[ec2-user@ip-10-0-12-46 ~]$ aws s3 ls s3://haroon-vpc-bucket1
2025-12-23 06:52:09      14238 Detecting Email Attacks.docx
2025-12-23 06:52:10      14034 Email Attack Goals.docx
2025-12-23 06:52:10      13804 Email Attack Impact.docx
2025-12-23 06:52:11      14900 Email Attack Types.docx
2025-12-23 06:52:11      16318 Threat Actors.docx
2025-12-23 06:52:12      14794 Why attack matters.docx
2025-12-23 07:30:14          0 test.txt
[ec2-user@ip-10-0-12-46 ~]$
```

- After updating the routes, I could access the S3 bucket **only from the EC2 instance inside the VPC**.
- All other access paths were blocked, confirming that S3 access was restricted to the private VPC endpoint only.



Successfully added route tables
rtb-07c7f688b06954002

Endpoints (1/1) Info

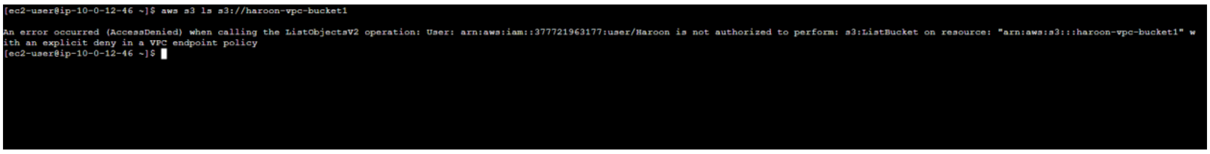
☑	Name	VPC endpoint ID	Endpoint type	Status	Service name	Service network
☑	My_S3_EndPoint	vpce-0eba4512e2b6e6f46	Gateway	Available	com.amazonaws.eu-north-1.s3	–

vpce-0eba4512e2b6e6f46 / My_S3_EndPoint

Policy

VPC endpoint policy controls access to the service

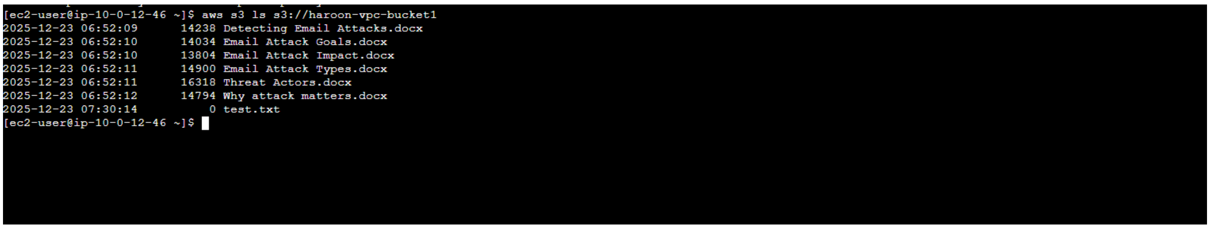
```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
```



```
[ec2-user@ip-10-0-12-46 ~]$ aws s3 ls s3://haroon-vpc-bucket1

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User: arn:aws:iam::377721963177:user/Haroon is not authorized to perform: s3:ListBucket on resource: "arn:aws:s3:::haroon-vpc-bucket1" w
ith an explicit deny in a VPC endpoint policy
[ec2-user@ip-10-0-12-46 ~]$ 
```

- In the endpoint policy, the effect is **Allow** by default, which permits access to the S3 bucket through the VPC endpoint.
- When I changed the effect to **Deny** in the edit policy, I could no longer access the S3 bucket.



```
[ec2-user@ip-10-0-12-46 ~]$ aws s3 ls s3://haroon-vpc-bucket1
2025-12-23 06:52:09      14238 Detecting Email Attacks.docx
2025-12-23 06:52:10      14034 Email Attack Goals.docx
2025-12-23 06:52:10      13804 Email Attack Impact.docx
2025-12-23 06:52:11      14900 Email Attack Types.docx
2025-12-23 06:52:11      16318 Threat Actors.docx
2025-12-23 06:52:12      14794 Why attack matters.docx
2025-12-23 07:30:14          0 test.txt
[ec2-user@ip-10-0-12-46 ~]$ 
```

- Then I changed the endpoint policy back to **Allow**, and access to the S3 bucket started working again.

Conclusion

In this project, I learned how to securely access Amazon S3 from within a VPC using a **VPC Gateway Endpoint**. I created a VPC, launched an EC2 instance, configured IAM credentials, and verified S3 access using the AWS CLI. By introducing an S3 VPC endpoint and applying a restrictive bucket policy, I ensured that the S3 bucket could only be accessed through the private AWS network.

I also learned how **route tables and endpoint policies** directly affect access to AWS services. When the route table was not associated with the endpoint or when the endpoint policy was set to deny, access to S3 was blocked. Once corrected, access was restored, confirming that the security controls were working as intended.

This project helped me understand how AWS networking, IAM, and S3 work together to provide **private, secure, and controlled access** to cloud storage without relying on the public internet.