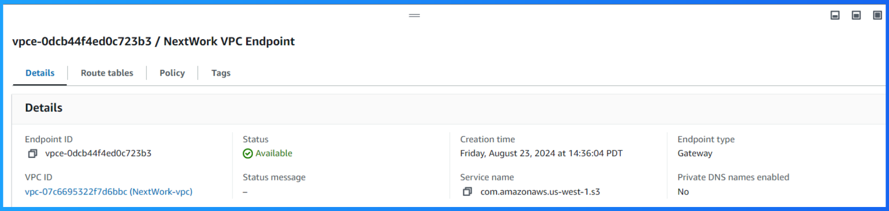
NextWork.org

VPC Endpoints

mmoorejr24@gmail.com


vpce-0dcb44f4ed0c723b3 / NextWork VPC Endpoint

Details

Endpoint ID	Status	Creation time	Endpoint type
vpce-0dcb44f4ed0c723b3	Available	Friday, August 23, 2024 at 14:36:04 PDT	Gateway
VPC ID	Status message	Service name	Private DNS names enabled
vpc-07c6695322f7d6bbc (NextWork-vpc)	–	com.amazonaws.us-west-1.s3	No

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service that lets you launch AWS resources into a virtual network environment, including resource placement, connectivity, and security. It also gives you full control in a logically isolated section of the AWS cloud.

How I used Amazon VPC in this project

I used AWS VPC in launching an EC2 instance and S3 bucket. I also set up an endpoint so that I could restrict access to the public internet and connect through the endpoint. I also setup policies on the S3 bucket and Endpoint to make it more granular

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

I didn't expect how granular S3 buckets and Endpoints can be, which is powerful when making secure connections.

This project took me...

This project took me about 1 hour and 50 minutes.

In the first part of my project...

Step 1 - Architecture set up

In this step, we are setting up the foundations of this project - i.e. launching a VPC, EC2 instance and S3 bucket so that we can set up an endpoint architecture and test that set up in the last step of this project.

Step 2 - Connect to EC2 instance

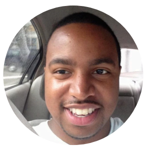
In this step, we connected directly to our EC2 instance using EC2 Instance Connect. Connecting to an instance will help us with accessing S3 and running commands later in this project.

Step 3 - Set up access keys

In this step, we will set up an access key so that our EC2 instance will have access to our AWS environment. We can think of access keys almost like "login details" for EC2 instances/applications (non humans) to interact with our AWS services.

Step 4 - Interact with S3 bucket

In this step, we applying our access key credentials to our EC2 instance, and then we are using AWS CLI and our EC2 instance to access Amazon S3.

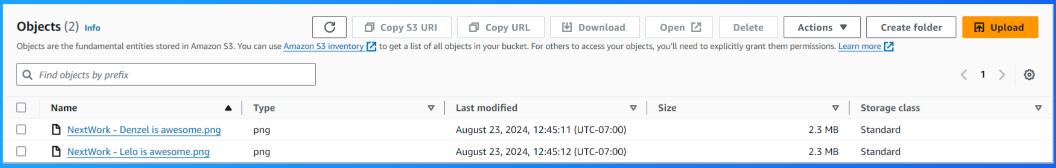


mmoorejr24@gmail.com
NextWork Student

Architecture set up

I started my project by launching three key resources - a VPC, EC2, and an S3 bucket.

I also set up an S3 bucket with two files in the bucket.

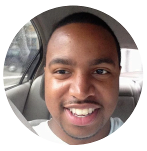


Objects (2) Info

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
NextWork - Denzel is awesome.png	png	August 23, 2024, 12:45:11 (UTC-07:00)	2.3 MB	Standard
NextWork - Lelo is awesome.png	png	August 23, 2024, 12:45:12 (UTC-07:00)	2.3 MB	Standard



Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured the AWS access key ID, secret access key that matches the that key ID, the default region type, and the default output type.

Access keys are credentials that an EC2 instance/other server/application would need to get access to our AWS environment e.g. creating resources, reading what's inside our AWS account etc.

Secret access keys are like passwords in the context of access keys/credentials for our EC2 instance to get access to our AWS services/environment.

Best practice

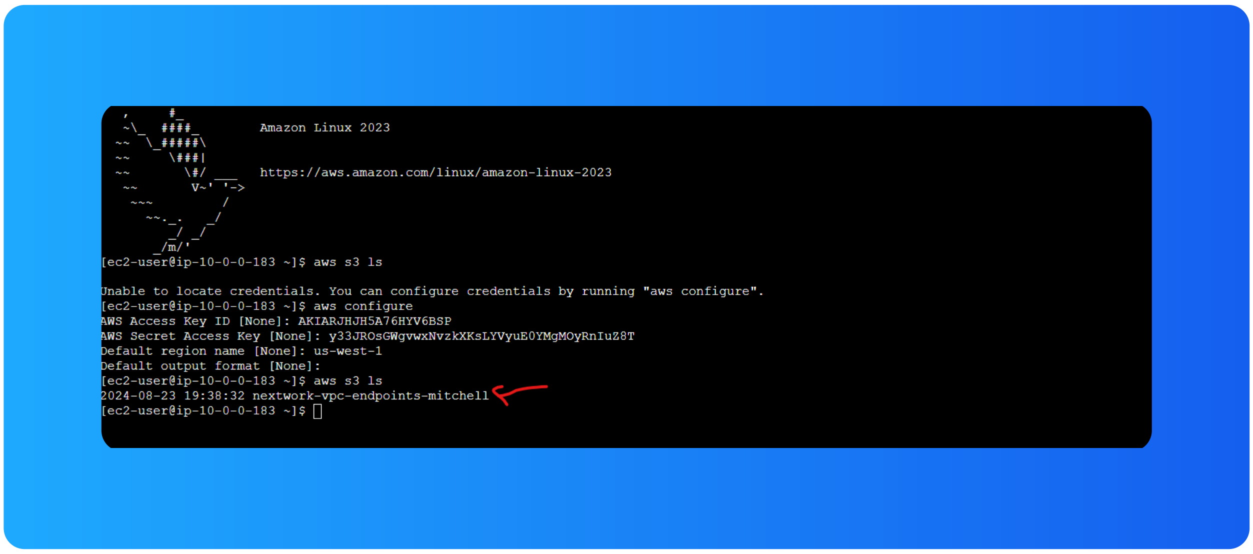
Although I'm using access keys in this project, a best practice alternative is to use IAM admin roles instead! This means the necessary permissions will be attached to an IAM role, then the role will be associated with the relevant resources.

mmoorejr24@gmail.com
NextWork Student

Connecting to my S3 bucket

The command I ran was 'aws s3 ls'. This command is used to list all buckets in an AWS account.

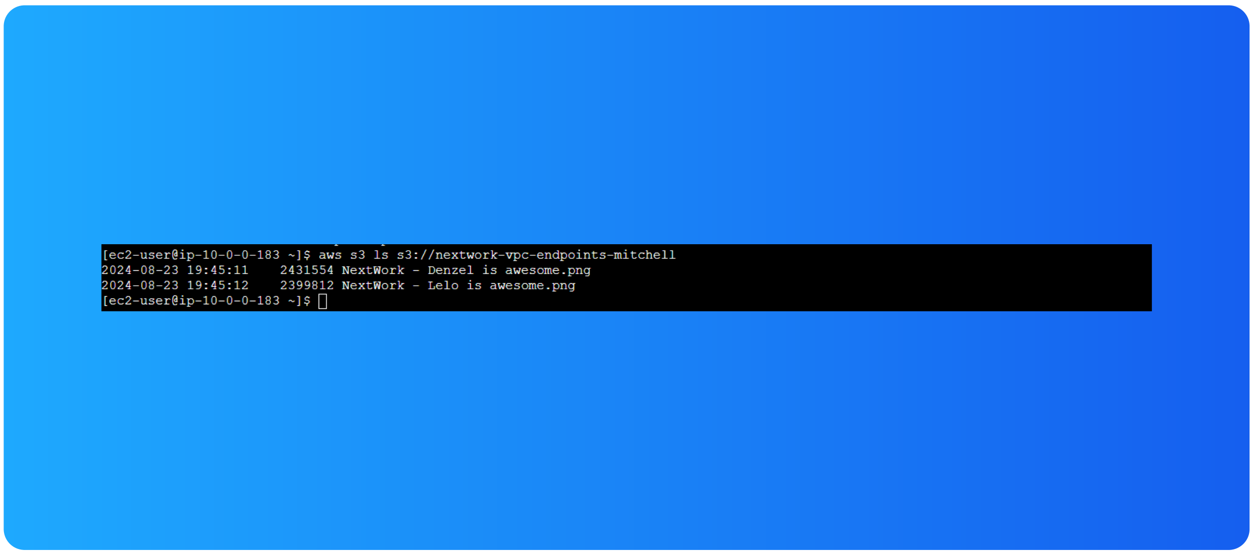
The terminal responded with a list of my account's S3 buckets. This indicated that the access keys I set up correctly and can give the EC2 instant access to my AWS account and environment.



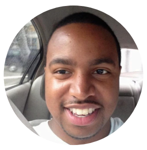
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-0-183 ~]$ aws s3 ls

Unable to locate credentials. You can configure credentials by running "aws configure".
[ec2-user@ip-10-0-0-183 ~]$ aws configure
AWS Access Key ID [None]: AKIARJHJH5A76HYV6BSP
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-1
Default output format [None]:
[ec2-user@ip-10-0-0-183 ~]$ aws s3 ls
2024-08-23 19:38:32 nextwork-vpc-endpoints-mitchell
[ec2-user@ip-10-0-0-183 ~]$ 
```

Connecting to my S3 bucket

I also tested the command 'aws s3 ls s3://nextwork-vpc-endpoints-mitchell' which returned a list of all of the objects inside that S3 bucket.



```
[ec2-user@ip-10-0-0-183 ~]$ aws s3 ls s3://nextwork-vpc-endpoints-mitchell
2024-08-23 19:45:11    2431554 NextWork - Denzel is awesome.png
2024-08-23 19:45:12    2399812 NextWork - Lelo is awesome.png
[ec2-user@ip-10-0-0-183 ~]$ 
```

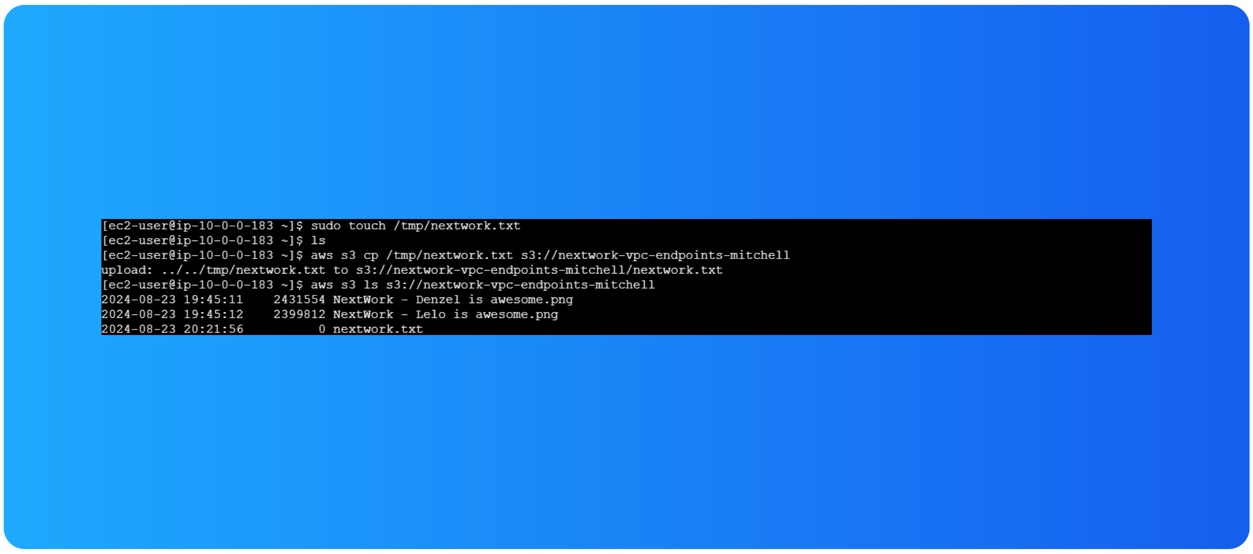


Uploading objects to S3

To upload a new file to my bucket, I first ran the command 'sudo touch /tmp/nextwork.txt'. This command creates an empty file called nextwork.txt and saves it locally in the EC2 instance.

The second command I ran was 'aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-mitchell'. This command will copy the file I created and upload that to the S3 bucket.

The third command I ran was 'aws s3 ls s3://nextwork-vpc-endpoints-mitchell'. which validated that a new file was created and uploaded into our S3 bucket.



```
[ec2-user@ip-10-0-0-183 ~]$ sudo touch /tmp/nextwork.txt
[ec2-user@ip-10-0-0-183 ~]$ ls
[ec2-user@ip-10-0-0-183 ~]$ aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-mitchell
upload: ../../tmp/nextwork.txt to s3://nextwork-vpc-endpoints-mitchell/nextwork.txt
[ec2-user@ip-10-0-0-183 ~]$ aws s3 ls s3://nextwork-vpc-endpoints-mitchell
2024-08-23 19:45:11    2431554 NextWork - Denzel is awesome.png
2024-08-23 19:45:12    2399812 NextWork - Lelo is awesome.png
2024-08-23 20:21:56          0 nextwork.txt
```

In the second part of my project...

Step 5 - Set up a Gateway

In this step we are setting up a VPC endpoint so that communication between our VPC and other services (especially S3) is direct and secure.

Step 6 - Bucket policies

In this step, we are testing our endpoint connection by blocking off all traffic to our S3 bucket, except for traffic coming from our endpoint.

Step 7 - Update route tables

In this step, we are testing our endpoint connection between our bucket and EC2 instance.

Step 8 - Validate endpoint conection

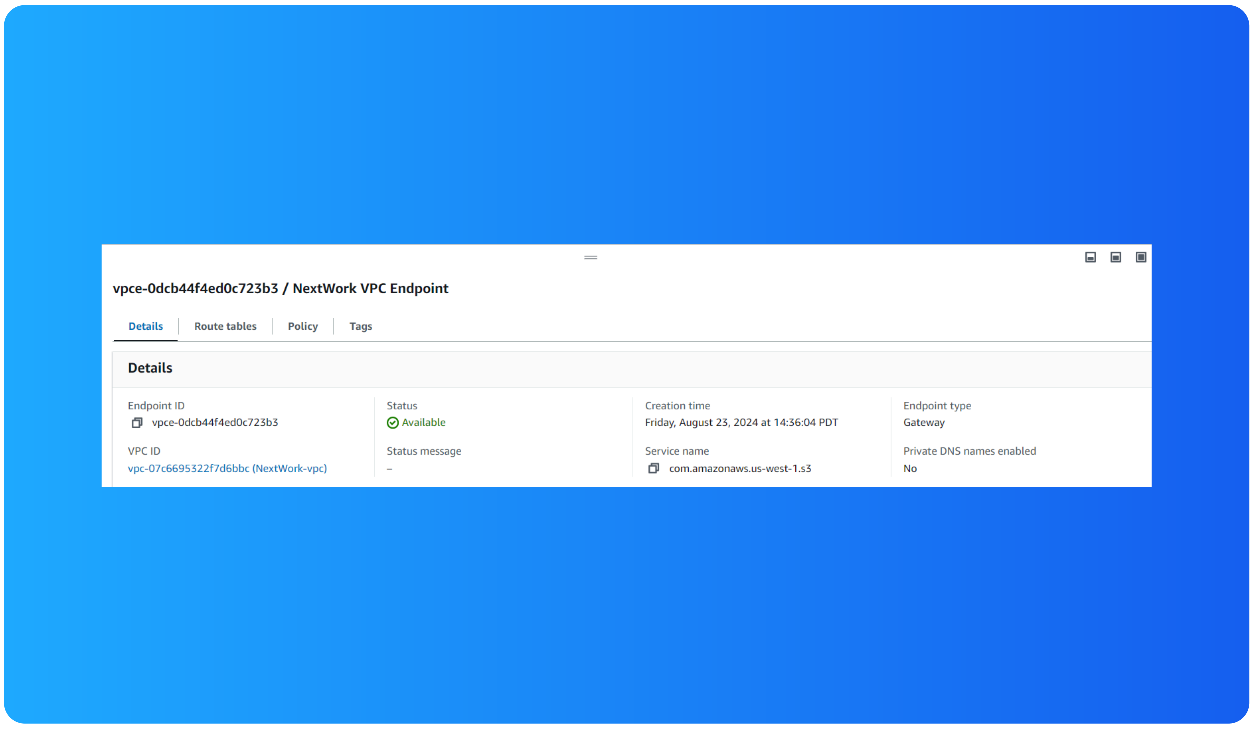
In this step, we are going to validate our VPC endpoint set up one more time. We are also going to use endpoint policies to restrict my EC2 instance's access to our AWS environment.

Setting up a Gateway

I set up an S3 Gateway, which is a type of endpoint specifically designed for Amazon S3. Gateways work by updating the route table of associated subnets, so that S3 bound traffic goes through 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 public internet.



vpce-0dcb44f4ed0c723b3 / NextWork VPC Endpoint

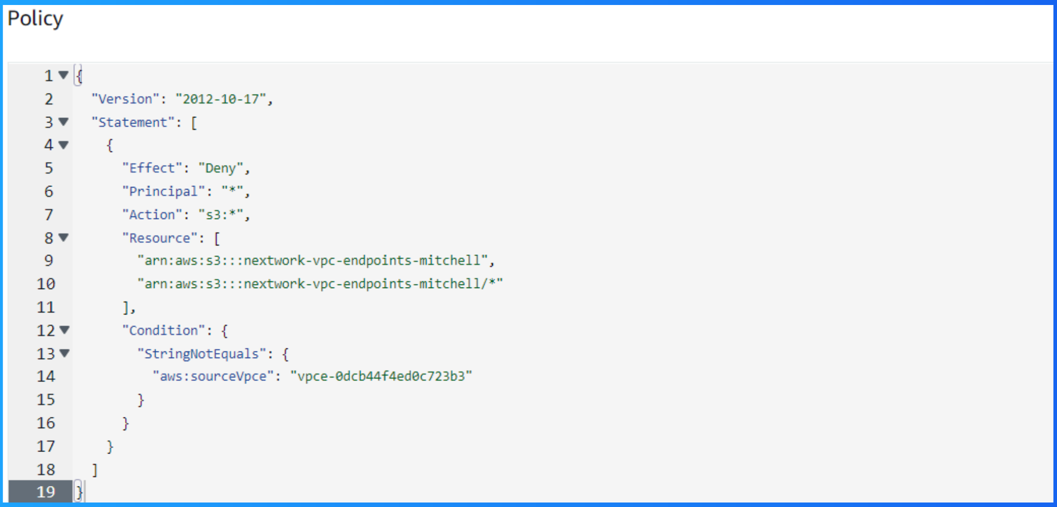
Details

Endpoint ID	Status	Creation time	Endpoint type
vpce-0dcb44f4ed0c723b3	✓ Available	Friday, August 23, 2024 at 14:36:04 PDT	Gateway
VPC ID	Status message	Service name	Private DNS names enabled
vpc-07c6695322f7d6bbc (NextWork-vpc)	–	com.amazonaws.us-west-1.s3	No

Bucket policies

A bucket policy is a type of policy that has granular control over who has access to an S3 bucket, and what are the actions they can perform.

My bucket policy will deny traffic from ALL sources - except for traffic coming from my VPC endpoint.


Policy

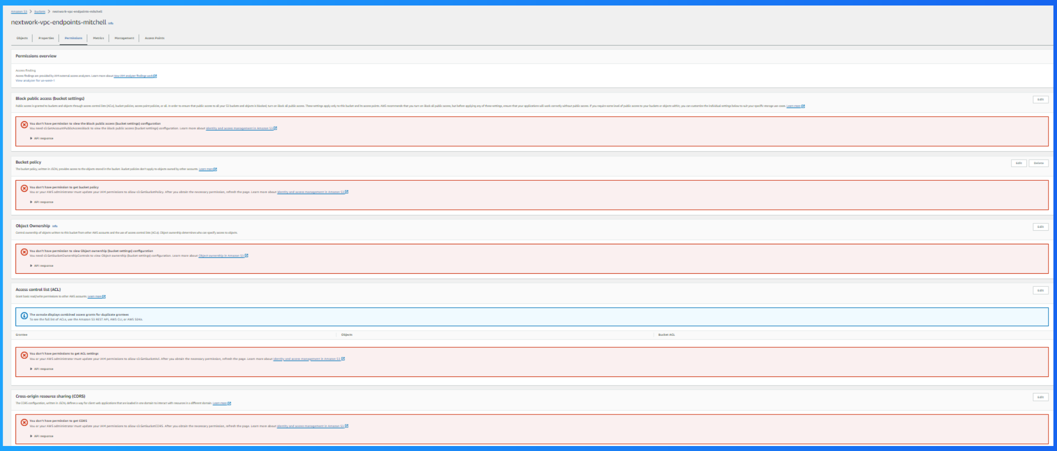
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::nextwork-vpc-endpoints-mitchell",
        "arn:aws:s3:::nextwork-vpc-endpoints-mitchell/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpce": "vpce-0dcb44f4ed0c723b3"
        }
      }
    }
  ]
}
```

mmoorejr24@gmail.com
NextWork Student

Bucket policies

Right after saving my bucket policy, my S3 bucket page showed 'denied access' warnings. This was because it deny all traffic coming from the public internet and only available to be accessed through the endpoint we set up.

I also had to update my route table because my route table by default, didn't provide a route for traffic in my public subnet to the VPC endpoint.



Amazon S3 > Buckets > nextwork-vpc-endpoints-mitchell

nextwork-vpc-endpoints-mitchell Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Permissions overview

Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for us-west-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

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

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 Info

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

Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. Learn more

The console displays combined access grants for duplicate grantees
To see the full list of ACLs, use the Amazon S3 REST API, AWS CLI, or AWS SDKs.

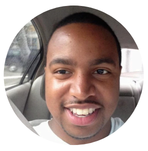
Grantee	Objects	Bucket ACL

You don't have permissions to get ACL settings
You or your AWS administrator must update your IAM permissions to allow s3:GetBucketAcl. After you obtain the necessary permission, refresh the page. Learn more about Identity and access management in Amazon S3
▶ API response

Cross-origin resource sharing (CORS)

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. Learn more

You don't have permission to get CORS
You or your AWS administrator must update your IAM permissions to allow s3:GetBucketCORS. After you obtain the necessary permission, refresh the page. Learn more about Identity and access management in Amazon S3
▶ API response

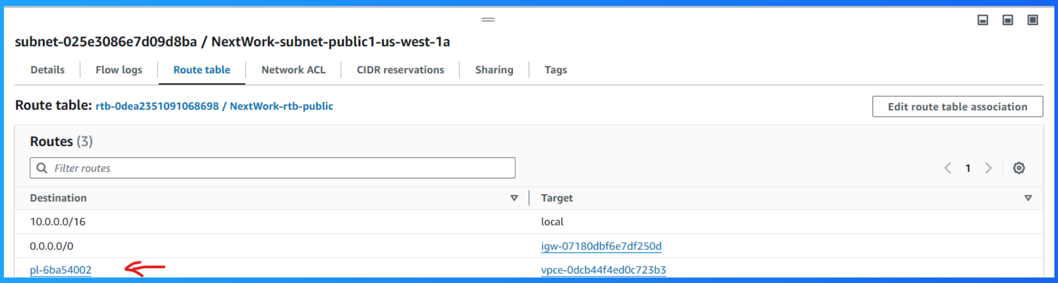


mmoorejr24@gmail.com
NextWork Student

Route table updates

To update my route table, I visited the Endpoints page of my VPC console, and we modified the route table from there to associate our VPC's public subnet.

After updating my public subnet's route table, my terminal could return the files in my S3 bucket. Access was no longer denied!!



subnet-025e3086e7d09d8ba / NextWork-subnet-public1-us-west-1a

Route table: rtb-0dea2351091068698 / NextWork-rtb-public

Routes (3)

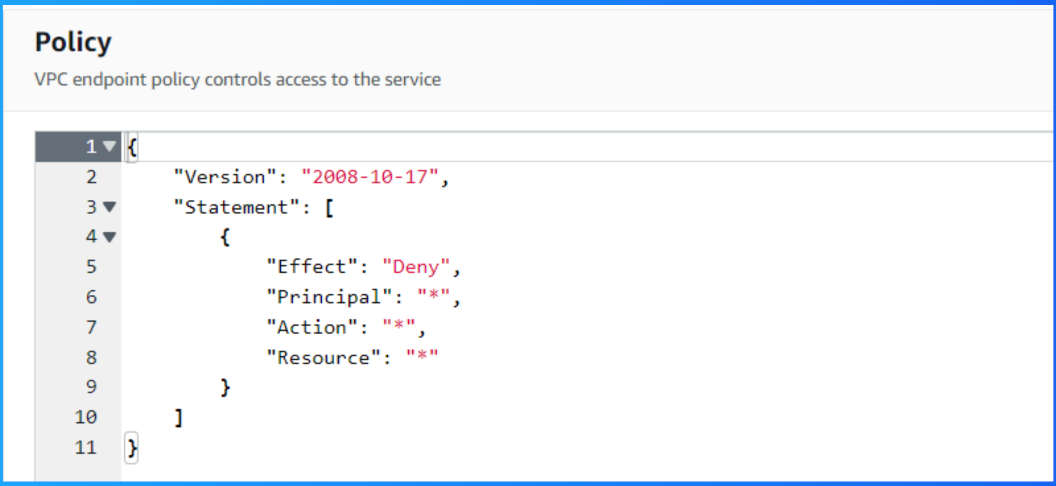
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-07180dbf6e7df250d
pl-6ba54002	vpce-0dcb44f4ed0c723b3

mmoorejr24@gmail.com
NextWork Student

Endpoint policies

An endpoint policy is a type of policy designed for specifying the range of resources and actions permitted by an 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 again denied access to S3 when I tried to run another 'aws S3' command.



Policy

VPC endpoint policy controls access to the service

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

Everyone should be in a job they love.

Check out nextwork.org for more projects