



Access S3 from a VPC



Jonathan Nutsugah

```
[ec2-user@ip-10-0-4-242 ~]$ sudo touch /tmp/test.txt
[ec2-user@ip-10-0-4-242 ~]$ ls
[ec2-user@ip-10-0-4-242 ~]$ ls -l
total 0
[ec2-user@ip-10-0-4-242 ~]$ aws s3 cp /tmp/test.txt s3://sacnet-vpc-project-jonathan
upload: ../../tmp/test.txt to s3://sacnet-vpc-project-jonathan/test.txt
[ec2-user@ip-10-0-4-242 ~]$ aws s3 ls s3://sacnet-vpc-project-jonathan
2025-05-28 15:11:41      127226 rt2.png
2025-05-28 15:11:42      55845 success.png
2025-05-28 16:16:34          0 test.txt
[ec2-user@ip-10-0-4-242 ~]$ ||
```

i-0ee4b025afdbb2680 (Instance - Sacnet VPC Project)

PublicIPs: 54.175.61.253 PrivateIPs: 10.0.4.242



Jonathan Nutsugah

NextWork Student

nextwork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) lets you create a secure, isolated network in AWS where you control IP ranges, subnets, and traffic. It's useful for customizing network setup, enhancing security, and supporting hybrid cloud deployments.

How I used Amazon VPC in this project

Through the management console, I used it to virtually isolate a space to keep my resources away from external access.

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

How easy it is to navigate AWS.

This project took me...

1 hour 30 minutes



In the first part of my project...

Step 1 - Architecture set up

In this step, I'm going to create a new VPC and launch an EC2 instance inside it. This gives me a private, isolated network where I can safely run and manage my virtual server.

Step 2 - Connect to my EC2 instance

In this step, I'm going to connect directly to my EC2 instance so I can access its command line and start working with it. This connection is important because it lets me run commands, install software, and manage the instance from inside.

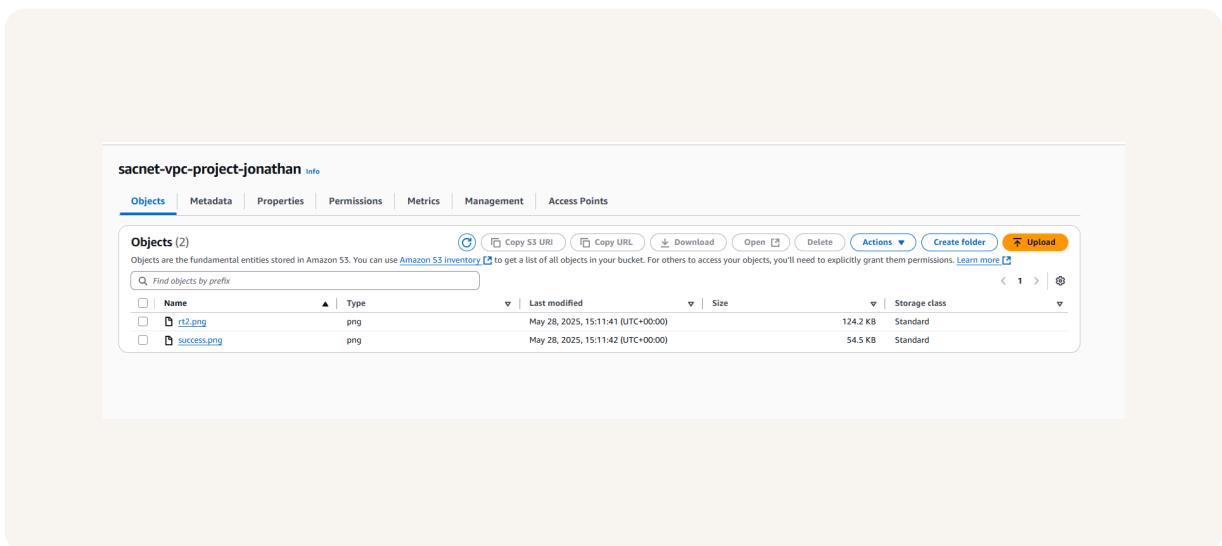
Step 3 - Set up access keys

In this step, I'm going to give my EC2 instance access to my AWS environment by attaching credentials. This is important so the instance can securely interact with AWS services like S3 without needing me to manually enter keys each time.

Architecture set up

I started my project by launching an EC2 instance inside my custom VPC, along with the necessary subnet, route table, internet gateway, and security group to make sure the instance is properly connected and accessible.

I also set up a general purpose S3 bucket, which I'll use to store and manage files from my EC2 instance.



Jonathan Nutsugah

NextWork Student

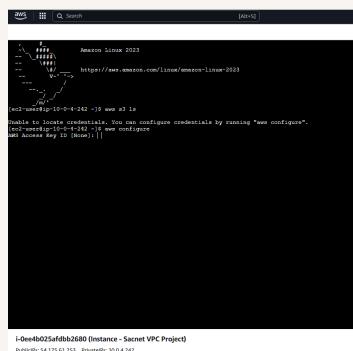
nextwork.org

Running CLI commands

AWS CLI is a command-line tool that lets me interact with AWS services using text commands instead of the web console. I have access to AWS CLI because it's pre-installed on my EC2 instance.

The first command I ran was aws s3 ls, which is a command used to list the S3 buckets in your account.

The second command I ran was aws configure. This command is used to set up my AWS credentials and default settings so the CLI can interact with my AWS account.





Jonathan Nutsugah
NextWork Student

nextwork.org

Access keys

Credentials

The purpose of the `aws configure` command is to set up the AWS CLI with your access key, secret access key, default region, and output format, so you can securely run AWS commands from your terminal.

Access keys are a pair of credentials (an access key ID and a secret access key) that allow secure programmatic access to AWS services, like when using the AWS CLI or SDKs.

Secret access keys are the private part of AWS access keys used to securely sign requests. They should be kept confidential to protect your AWS account from unauthorized access.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use AWS Cloudshell.



Jonathan Nutsugah
NextWork Student

nextwork.org

In the second part of my project...

Step 4 - Set up an S3 bucket

In this step, I'm going to create a new S3 bucket. This bucket will act as a storage space in the cloud where I can save and manage files that my EC2 instance or other services can access.

Step 5 - Connecting to my S3 bucket

In this step, I'm going back to my EC2 instance to test if it can interact with my S3 bucket. This is important to make sure the permissions and setup work, so the instance can upload, download, or list files in the bucket.



Jonathan Nutsugah
NextWork Student

nextwork.org

Connecting to my S3 bucket

The first command I ran was aws s3 ls, which is a command used to list the S3 buckets in your account.

When I ran the command aws s3 ls, the terminal responded with a list of my S3 buckets. This indicated that my EC2 instance was successfully connected to my AWS account and had the right permissions to access S3.

```
Default output format [None]:  
[ec2-user@ip-10-0-4-242 ~]$ aws s3 ls  
2025-02-12 16:12:27 kokuslbucket  
2025-05-28 15:10:16 sacnet-vpc-project-jonathan  
[ec2-user@ip-10-0-4-242 ~]$ ||
```



Jonathan Nutsugah
NextWork Student

nextwork.org

Connecting to my S3 bucket

Another CLI command I ran was aws s3 ls s3://nextwork-vpc-project-yourname, which returned the contents of my S3 bucket. This showed that my EC2 instance could access and read files from the specific bucket.

```
[ec2-user@ip-10-0-4-242 ~]$ aws s3 ls
2025-02-12 16:12:27 kokus1bucket
2025-05-28 15:10:16 sacnet-vpc-project-jonathan
[ec2-user@ip-10-0-4-242 ~]$ aws s3 ls s3://sacnet-vpc-project-jonathan
2025-05-28 15:11:41      127226 rt2.png
2025-05-28 15:11:42      55845 success.png
[ec2-user@ip-10-0-4-242 ~]$ ||
```



Jonathan Nutsugah
NextWork Student

nextwork.org

Uploading objects to S3

To upload a new file to my bucket, I first ran the command sudo touch /tmp/test.txt. This command creates an empty file named test.txt in the /tmp directory on my EC2 instance.

The second command I ran was aws s3 cp /tmp/test.txt s3://nextwork-vpc-project-yourname. This command will copy the test.txt file from my EC2 instance to my S3 bucket, uploading it to the cloud.

The third command I ran was aws s3 ls s3://nextwork-vpc-project-yourname, which validated that my file was successfully uploaded to the S3 bucket by listing it in the output.

```
[ec2-user@ip-10-0-4-242 ~]$ sudo touch /tmp/test.txt
[ec2-user@ip-10-0-4-242 ~]$ ls
[ec2-user@ip-10-0-4-242 ~]$ ls -l
total 0
[ec2-user@ip-10-0-4-242 ~]$ aws s3 cp /tmp/test.txt s3://sacnet-vpc-project-jonathan
upload: ./tmp/test.txt to s3://sacnet-vpc-project-jonathan/test.txt
[ec2-user@ip-10-0-4-242 ~]$ aws s3 ls s3://sacnet-vpc-project-jonathan
2025-05-28 15:11:41      127226 rt2.png
2025-05-28 15:11:42      55845 success.png
2025-05-28 16:16:34          0 test.txt
[ec2-user@ip-10-0-4-242 ~]$ ||
```

i-0ee4b025afdbb2680 (Instance - Sacnet VPC Project)

Public IPs: 54.175.61.253 Private IPs: 10.0.4.242



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

