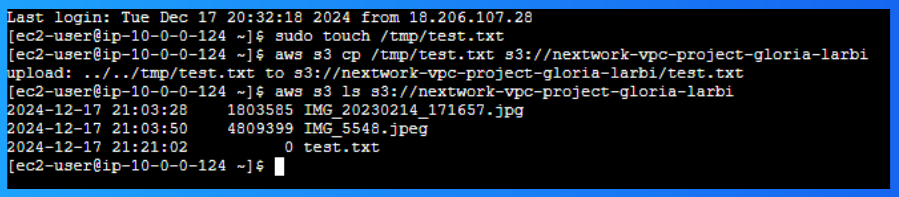
NextWork.org

Access S3 from a VPC

Gloria Iarbi



Gloria Iarbi
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a secure, isolated network in AWS where you can launch resources like EC2 instances. It’s useful because it gives control over networking, such as IP ranges, subnets and security.

How I used Amazon VPC in this project

I used Amazon VPC to create an isolated network and launched an EC2 instance inside it. The EC2 instance securely accessed my S3 bucket using access keys, showing how resources within a VPC can interact with AWS services outside the VPC like S3.

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

One thing I didn’t expect in this project was how easily I could use the AWS CLI to interact with S3 from my EC2 instance. It was straightforward to upload and list files in the bucket.

This project took me...

This project took me about 30 minutes to complete, including setting up the VPC, launching the EC2 instance, configuring access keys, and interacting with the S3 bucket using the AWS CLI.

G
Gloria Iarbi
NextWork Student
NextWork.org

In the first part of my project...

Step 1 - Architecture set up

In this step, I will create a VPC from scratch and launch an EC2 instance into it. This setup forms the foundation for today's project, allowing me to interact with AWS services like S3 securely from within my VPC.

Step 2 - Connect to my EC2 instance

In this step, I will connect directly to my EC2 instance using Instance Connect.This connection allows me to access the instance's terminal, where I can perform tasks like installing software or interacting with AWS services from within the instance.

Step 3 - Set up access keys

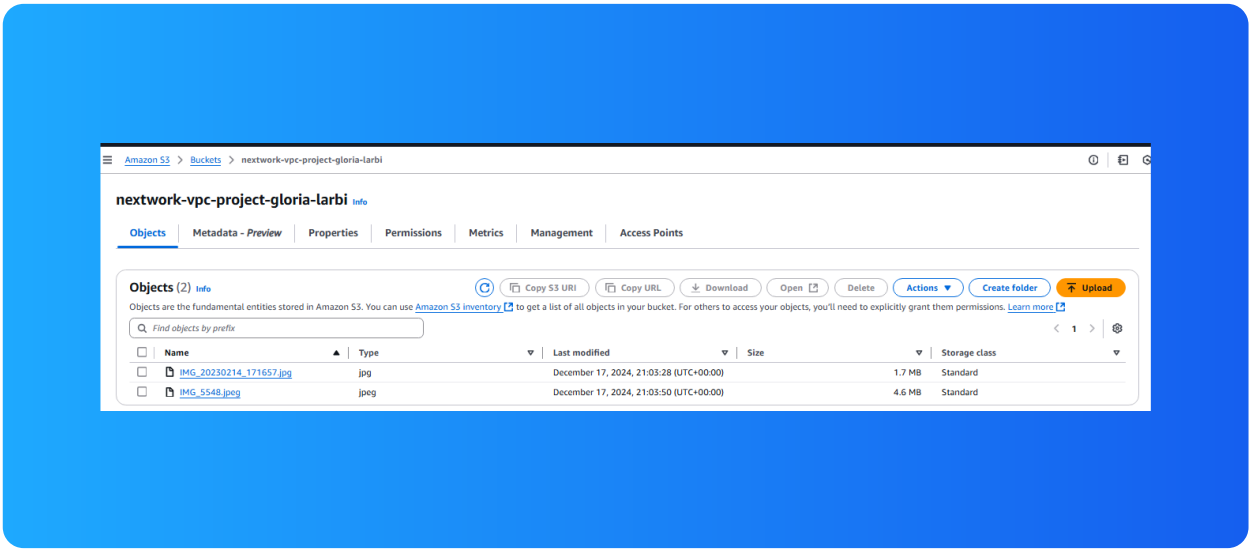
In this step, I am creating access keys to give my EC2 instance credentials to access AWS services. These keys will allow the instance to securely interact with AWS resources, like S3, through the AWS CLI.

Gloria Iarbi
NextWork Student

Architecture set up

I started my project by creating a VPC, which provides an isolated network for my resources. I added a public subnet, launched an EC2 instance an AMI, enabled a public IP for EC2 Instance Connect, and created a security group allowing SSH access.

I also set up an S3 bucket named nextwork-vpc-project-gloria-larbi and uploaded two files into it. This bucket will store data that I can later access from my EC2 instance, allowing me to interact with S3 and verify connectivity.



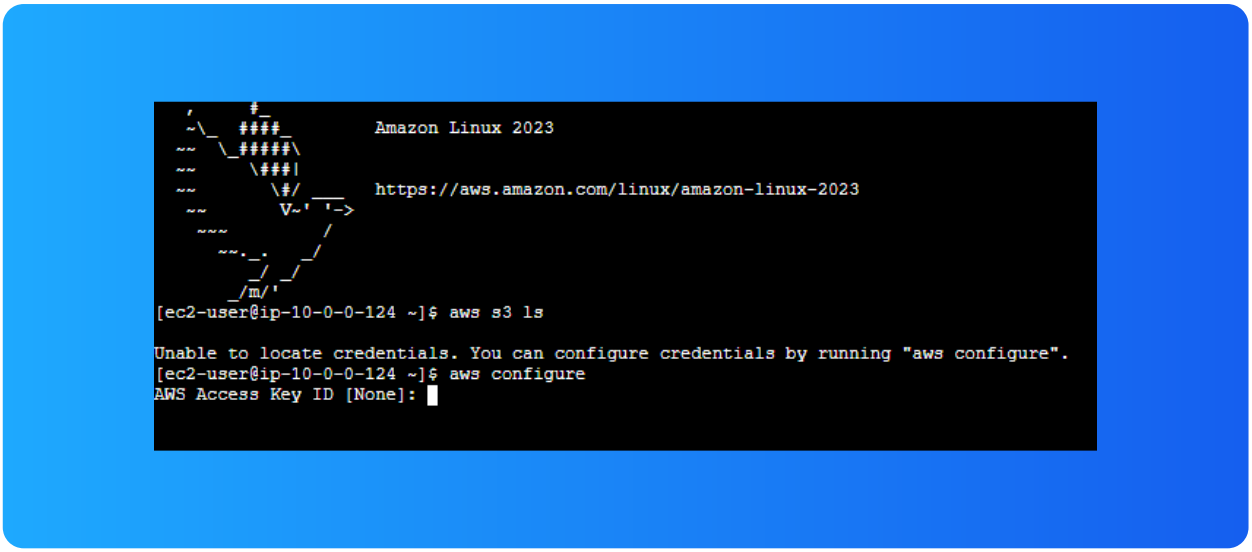
Gloria Iarbi
NextWork Student

Running CLI commands

AWS CLI is the **Amazon Web Services Command Line Interface**, a tool that allows me to interact with AWS services using commands in a terminal. I have access to AWS CLI because it can be installed on my EC2 instance already.

The first command I ran was `aws s3 ls`. This command is used to list all the Amazon S3 buckets in my AWS account. It shows the bucket names and their creation dates, but I couldn't access that I haven't configured my AWS credentials.

The second command I ran was `aws configure`. This command is used to set up the AWS CLI by providing the access key, secret key, default region, and output format, ensuring the CLI can interact with AWS services.



G
Gloria Iarbi
NextWork Student

Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured the AWS CLI using the aws configure command.This allowed me to provide my Access Key ID, Secret Access Key, Region, and output format so the instance can access AWS services.

Access keys are credentials that allow secure programmatic access to AWS services. They consist of an Access Key ID and a Secret Access Key and are used to authenticate and authorize requests made through tools like the AWS CLI.

Secret access keys are unique codes that work like a password, paired with an access key ID, to authenticate and securely access AWS services. They must be kept private to prevent unauthorized access to your AWS account.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use **IAM roles**. IAM roles provide temporary credentials to instances, making them more secure because they don’t require storing or managing long-term access keys.

Gloria Iarbi
NextWork Student

In the second part of my project...

Step 4 - Set up an S3 bucket

In this step, I am creating an S3 bucket to store files. This bucket will act as a storage space in the cloud, and I'll add files to it. Later, I'll access the bucket from my EC2 instance to check its contents and interact with the stored files.

Step 5 - Connecting to my S3 bucket

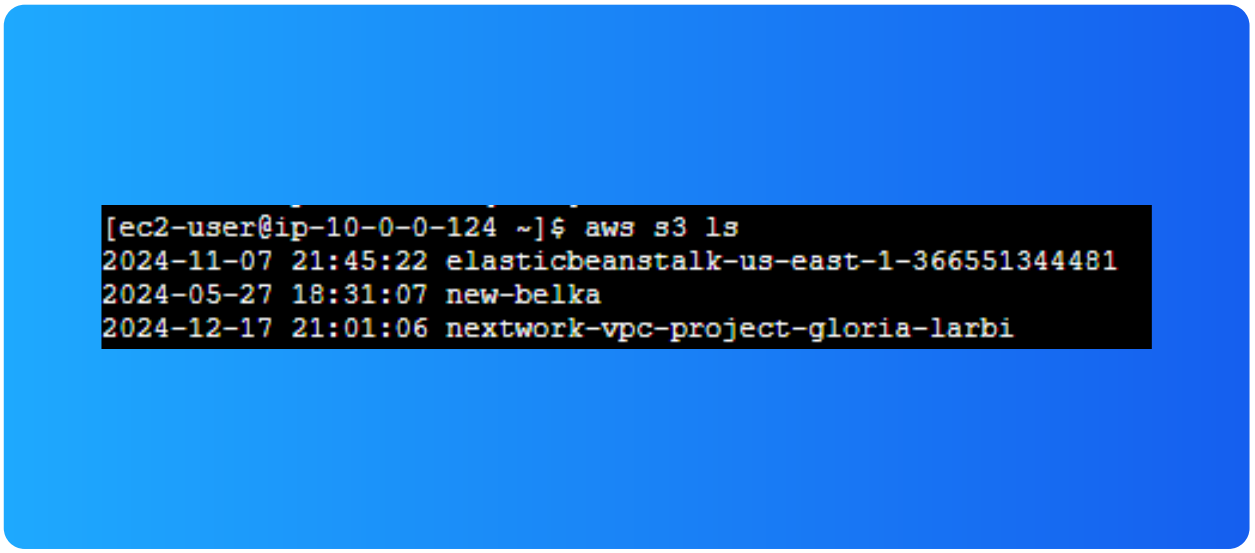
In this step, I will connect back to my EC2 instance and interact with the S3 bucket I created. This is important to verify that my EC2 instance can access and work with AWS services outside the VPC, like S3, using the AWS CLI.

Gloria Iarbi
NextWork Student

Connecting to my S3 bucket

The first command I ran was `aws s3 ls`. This command is used to list all the Amazon S3 buckets in my AWS account. It shows the bucket names and their creation dates, but I couldn't access that I haven't configured my AWS credentials.

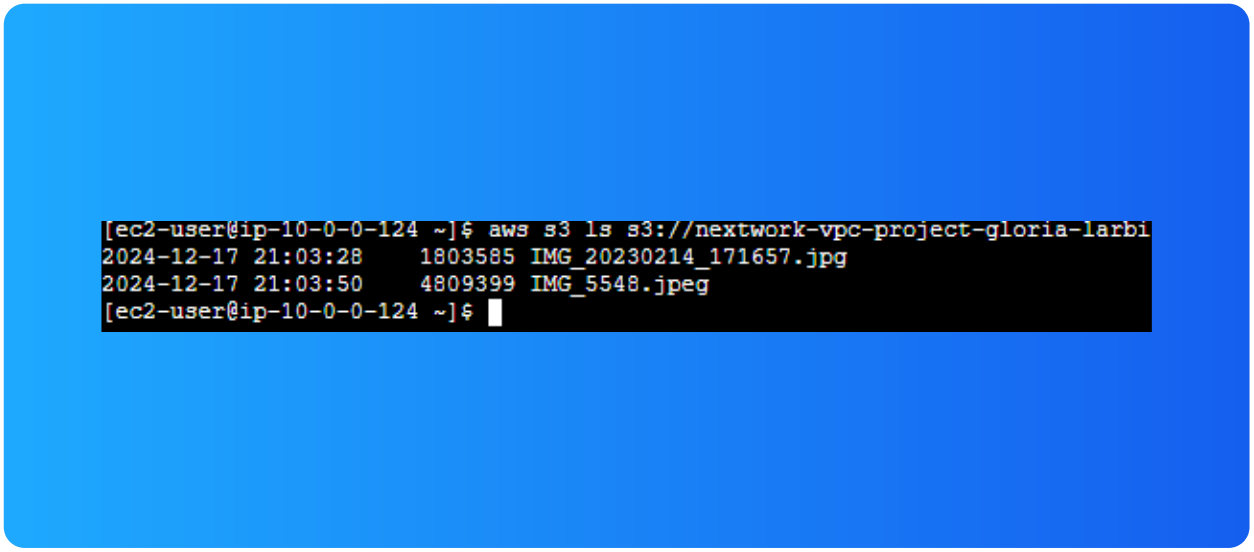
When I ran the command `aws s3 ls` again, the terminal responded with a list of my S3 buckets, including their names and creation dates. This indicated that my EC2 instance successfully connected to the S3 service and could access my AWS environment.



Gloria larbi
NextWork Student

Connecting to my S3 bucket

Another CLI command I ran was aws s3 ls s3://nextwork-vpc-project-gloria-larbi, which returned the list of objects inside my S3 bucket, file names, sizes, and upload timestamps. This confirmed that I could access and view the contents of my bucket.



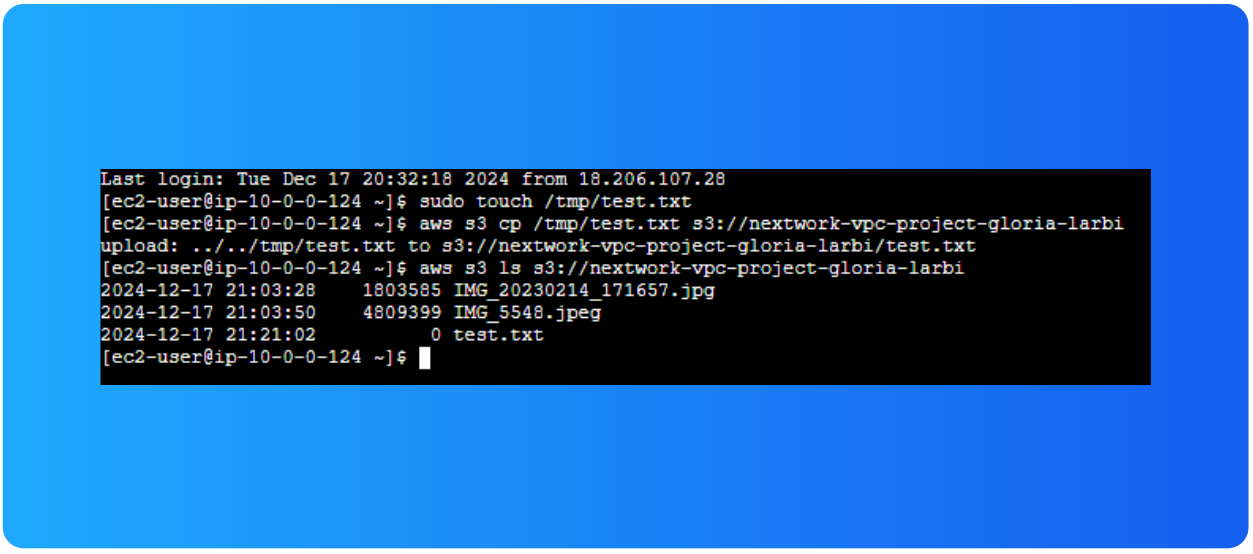
Gloria Iarbi
NextWork Student

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-gloria-larbi/test.txt`. This command will upload the `test.txt` file from my EC2 instance to my S3 bucket, allowing me to store it in the cloud.

The third command I ran was "aws s3 ls s3://nextwork-vpc-project-gloria-larbi", which validated that the file "test.txt" was successfully uploaded to my S3 bucket by listing its name, size, and upload timestamp alongside other objects in the bucket.



Everyone should be in a job they love.

Check out nextwork.org for more projects