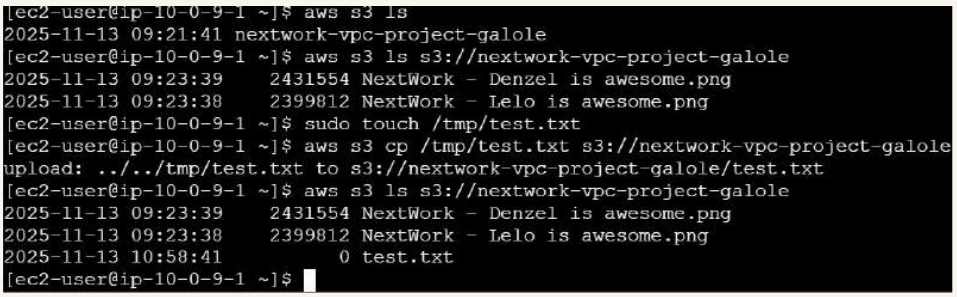
Access S3 from a VPC

GA Mohamed Galole



GA
Mohamed Galole
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a service that lets you create a private, isolated network within the AWS cloud, where you can launch and manage AWS resources like EC2 instances.

How I used Amazon VPC in this project

In today’s project, I used Amazon VPC to create a custom network for my EC2 instance. It allowed me to control the network configuration, secure my instance with a security group, and manage how it connects to the internet and other AWS services.

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

One thing I didn’t expect in this project was how quickly I could connect my EC2 instance to S3 using the AWS CLI once the access keys were configured. It was surprising to see how seamlessly compute and storage services can interact in AWS.

Mohamed Galole
NextWork Student

This project took me...

This project took me two hours

GA
Mohamed Galole
NextWork Student

In the first part of my project...

Step 1 - Architecture set up

In this step, I will create a VPC and launch an EC2 instance inside it. I’m doing this to understand how networking works in AWS, including subnets, route tables, and internet gateways, and to learn how to securely deploy and connect compute resource

Step 2 - Connect to my EC2 instance

In this step, I will connect directly to my EC2 instance using SSH from my terminal. I'm doing this to get direct access to the server to install tools, configure the AWS CLI, and verify that my instance can interact with other AWS services like S3

Step 3 - Set up access keys

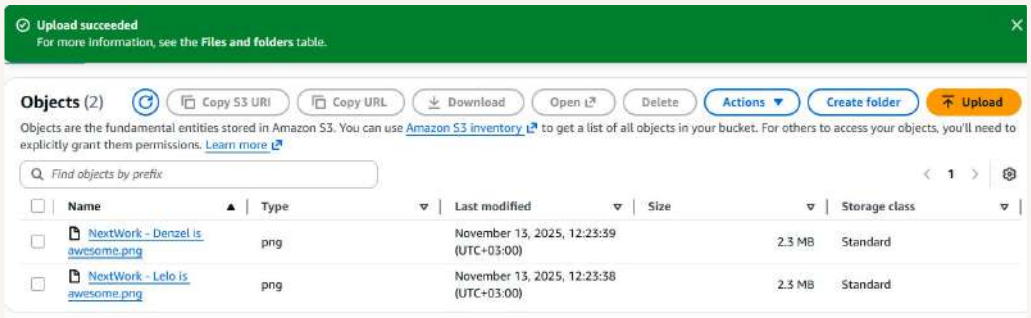
In this step, I will configure my EC2 instance with access keys so it can interact with AWS services like S3. I’m doing this because I want to enable the instance to securely perform actions in my AWS account without manually logging into the console

Mohamed Galole
NextWork Student

Architecture set up

I started my project by launching VPC and launched EC2 instance inside my custom VPC. I also created a Security Group to control inbound and outbound traffic to the instance.

I also set up I set up an Amazon S3 bucket and uploaded two files to it.



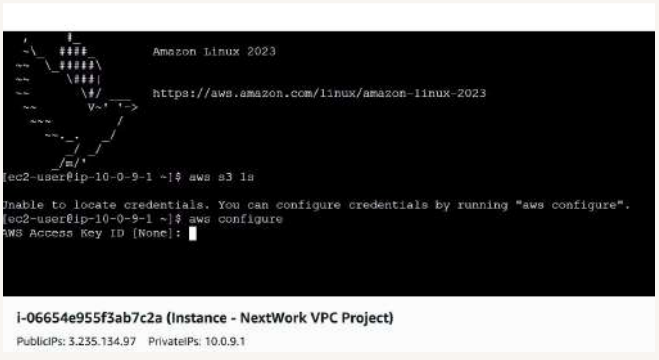
Mohamed Galole
NextWork Student

Running CLI commands

AWS CLI is a tool that lets me interact with AWS services directly from the command line instead of using the AWS Management Console.I have access to AWS CLI because I installed it on my EC2 instance, which allows me to run AWS commands.

The first command I ran was aws s3 ls This command is used to list all the S3 buckets in my AWS account.

The second command I ran was aws configure. This command is used to set up my AWS CLI credentials and default settings. It allows me to enter my Access Key ID, Secret Access Key, default region, and output format, so my EC2 instance can authenticate



GA
Mohamed Galole
NextWork Student

Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured the AWS CLI on the instance using my access keys. This allowed the EC2 instance to authenticate with AWS and perform actions such as listing S3 buckets, creating new buckets.

Access keys are a set of credentials (an Access Key ID and a Secret Access Key) that allow programmatic access to your AWS account.

Secret access keys are the confidential part of an AWS access key pair, used together with the Access Key ID to authenticate and authorize programmatic access to your AWS account.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use an IAM Role attached to the EC2 instance. IAM Roles allow the instance to access AWS services securely without storing long-term credentials.

In the second part of my project...

Step 4 - Set up an S3 bucket

In this step, I will create a new Amazon S3 bucket. I'm doing this because I want to store and manage files in the cloud and demonstrate how my EC2 instance can interact with S3 using the AWS CLI.

Step 5 - Connecting to my S3 bucket

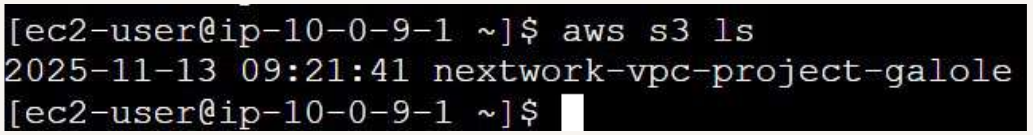
In this step, I will use my EC2 instance to access and interact with the S3 bucket I created. I'm doing this because I want to verify that my EC2 instance has the correct permissions and to practice managing cloud storage from compute resources.

GA
Mohamed Galole
NextWork Student

Connecting to my S3 bucket

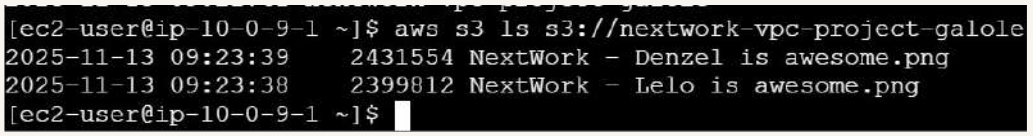
The first command I ran was aws s3 ls This command is used to list all the S3 buckets in my AWS account.

When I ran the command again, the terminal responded with S3 Bucket I created earlier This indicated EC2 instance was successfully connected to my AWS environment and had the necessary permissions to access S3.



Connecting to my S3 bucket

Another CLI command I ran was aws s3 ls s3://nextwork-vpc-project-galole which returned all the files stored in my S3 Bucket.



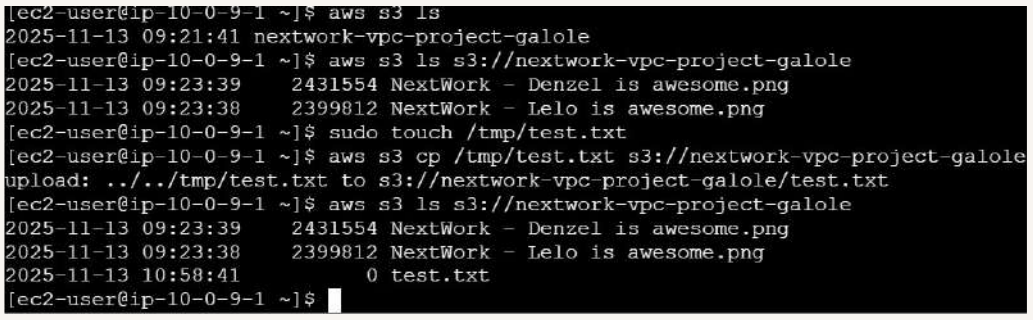
GA
Mohamed Galole
NextWork Student

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 a new, empty file named test.txt in the /tmp directory on the EC2 instance.

To upload a new file to my bucket, I first ran the command aws s3 cp /tmp/test.txt s3://nextwork-vpc-project-galole This command copies the file test.txt from my EC2 instance to the S3 bucket named nextwork-vpc-project-galole. It created new object

The second command I ran was aws s3 ls s3://nextwork-vpc-project-galole This command will lists all the objects in the specified S3 bucket. It allows me to verify that the file I uploaded from my EC2 instance was successfully stored in the bucket.



The place to learn & showcase your skills

Check out nextwork.org for more projects