NextWork.org

Access S3 from a VPC

DI dineshrajdhanapathy@gmail.com

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a virtual network within AWS where you can launch resources in an isolated environment. It’s useful for controlling network traffic, securing resources, and integrating with other AWS services.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to create a secure and isolated network for my EC2 instance. This ensured proper network configuration and allowed me to manage access to resources like S3 efficiently.

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

One thing I didn’t expect was the complexity of setting up permissions. I had to troubleshoot IAM roles and policies to grant the EC2 instance proper access to the S3 bucket and other resources.

This project took me...

This project took me about 1-2 hours to complete. The majority of the time was spent configuring the VPC, launching the EC2 instance, setting permissions, and troubleshooting S3 access issues.

In the first part of my project...

Step 1 - Architecture set up

In this step, I'll create a custom VPC to define a secure, isolated network for AWS resources. Then, I'll launch an EC2 instance within it, ensuring it has controlled access to the internet and other resources.

Step 2 - Connect to my EC2 instance

In this step, I'll connect directly to my EC2 instance using SSH to access and manage it. This is essential for configuring the instance, installing software, and verifying its ability to interact with other AWS services.

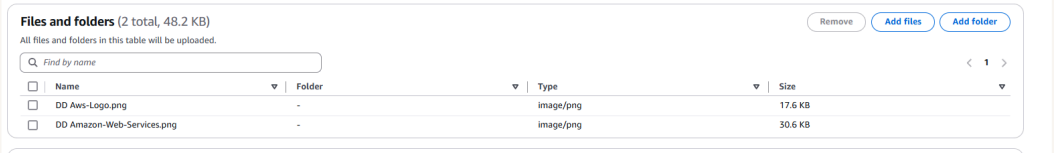
Step 3 - Set up access keys

In this step, I'll assign an IAM role to my EC2 instance, granting it permissions to access AWS resources securely without using access keys. This enhances security and simplifies resource management.

Architecture set up

I started my project by launching a VPC with subnets, an internet gateway, and a route table for internet access. Then, I launched an EC2 instance within the VPC to serve as a compute resource.

I also set up an S3 bucket in this step, creating a secure container for storing and organizing data. The bucket will be used to manage files, backups, and other application-related objects efficiently.

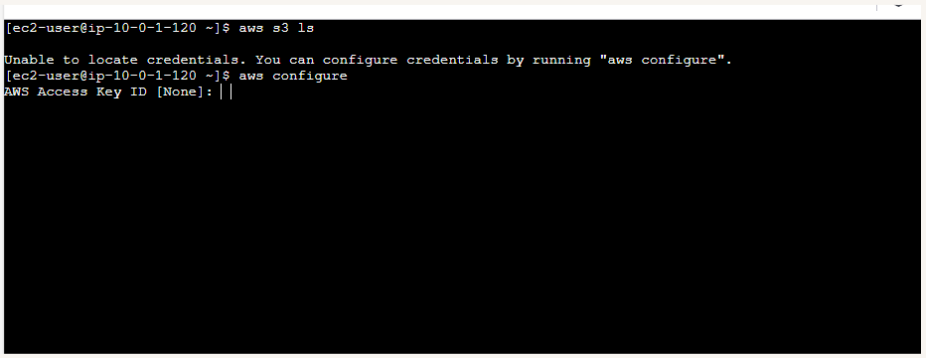


Running CLI commands

AWS CLI is a command-line tool to interact with AWS services efficiently. I have access to AWS CLI because it's installed on my local system or EC2 instance, enabling easy automation and management.

The first command I ran was aws s3 ls, but it returned "unable to locate credentials." This error means AWS CLI lacks access keys or an IAM role, so it can't authenticate to list S3 buckets.

The second command I ran was aws configure. This command is used to set up access credentials, default region, and output format, ensuring my AWS CLI is properly configured to interact with AWS services.



Access keys

Credentials

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

Access keys are credential pairs consisting of an access key ID and a secret access key. They authenticate AWS CLI or SDK requests, enabling secure programmatic access to AWS resources.

Secret access keys are confidential strings that pair with an access key ID to authenticate and authorize requests to AWS services. They should be kept secure and never exposed to maintain security.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM roles. They provide temporary credentials to EC2 instances, ensuring secure and automated access to AWS resources.

In the second part of my project...

Step 4 - Set up an S3 bucket

In this step, I'll create an S3 bucket to store data in AWS. S3 provides scalable, durable object storage, and the bucket will act as a container for managing files, backups, or application data securely.

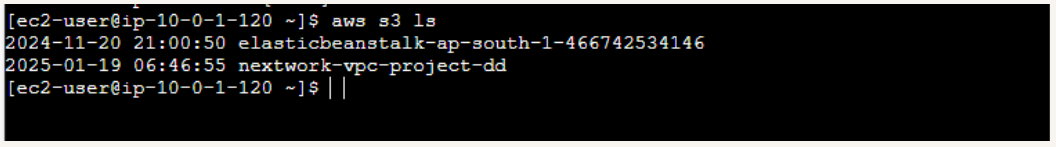
Step 5 - Connecting to my S3 bucket

In this step, I'll configure my EC2 instance to interact with the S3 bucket. This allows the instance to upload, download, or manage data stored in S3, streamlining data handling in the AWS environment.

Connecting to my S3 bucket

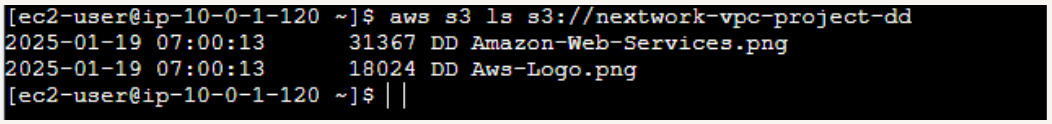
The first command I ran was aws s3 ls, but it returned "unable to locate credentials." This error means AWS CLI lacks access keys or an IAM role, so it can't authenticate to list S3 buckets.

When I ran the command to interact with the S3 bucket again, the terminal responded with the correct project contents in the EC2 terminal.



Connecting to my S3 bucket

Another CLI command I ran was aws s3 ls s3://<bucket-name>, which returned the S3 project name, and I was able to see the uploaded files. This showed that my EC2 instance have the proper IAM role or permissions to access the S3 bucket.

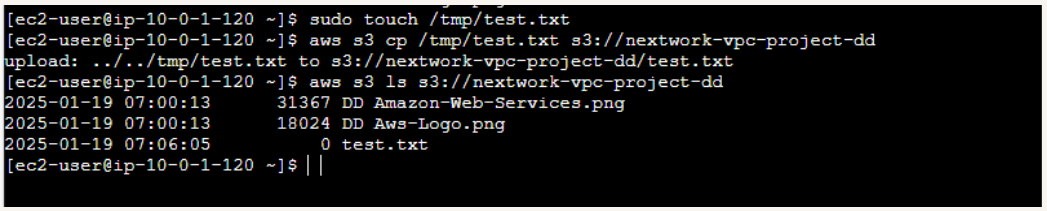


Uploading objects to S3

To upload a new file to my bucket, I first ran the command "sudo touch /tmp/test.txt" to create a blank .txt file in your EC2 instance.This command creates a text of the specified file from my EC2 instance to the S3 bucket.

The second command I ran was the aws s3 cp <file-path> s3://<bucket-name>/. This command creates a copy of the specified file from my EC2 instance to the S3 bucket.

The third command I ran was aws s3 ls s3://<bucket-name>/, which validated that the file was successfully uploaded to the S3 bucket by listing the contents of the specified bucket.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects