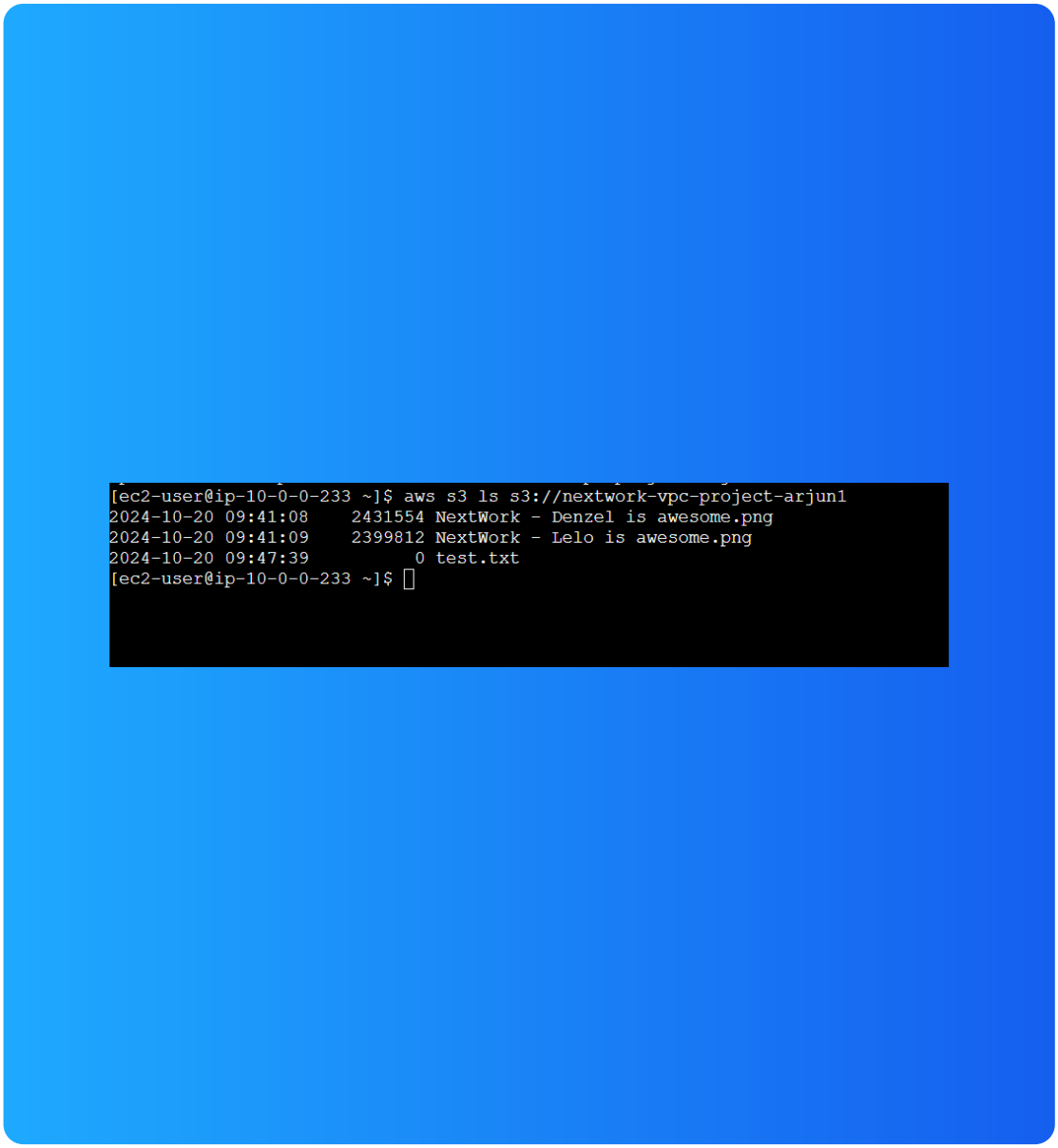
NextWork.org

Access S3 from a VPC

AJ ajv.singh3107@gmail.com



AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC allows you to create a secure, isolated network within the AWS cloud. It's useful for controlling network settings, subnets, and security, enabling you to deploy resources with customizable configurations and security policies.

How I used Amazon VPC in this project

I utilized a VPC endpoint to establish a direct connection between the EC2 instance and S3, thereby enhancing security by eliminating the need to publicly expose any services.

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

I didn't expect the complexity of configuring the security settings for the VPC endpoint. Ensuring the right permissions and access controls took longer than anticipated, but it ultimately enhanced the overall security of the project.

This project took me...

The project took me approximately 1.5 hours to complete, including configuration and troubleshooting.

AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

In the first part of my project...

Step 1 - Architecture set up

We're going to create a virtual private cloud (VPC) in AWS and launch an EC2 instance into it. This will give us a secure and isolated environment to run our application.

Step 2 - Connect to my EC2 instance

In this instance, I am going to connect to my instance directly.

Step 3 - Set up access keys

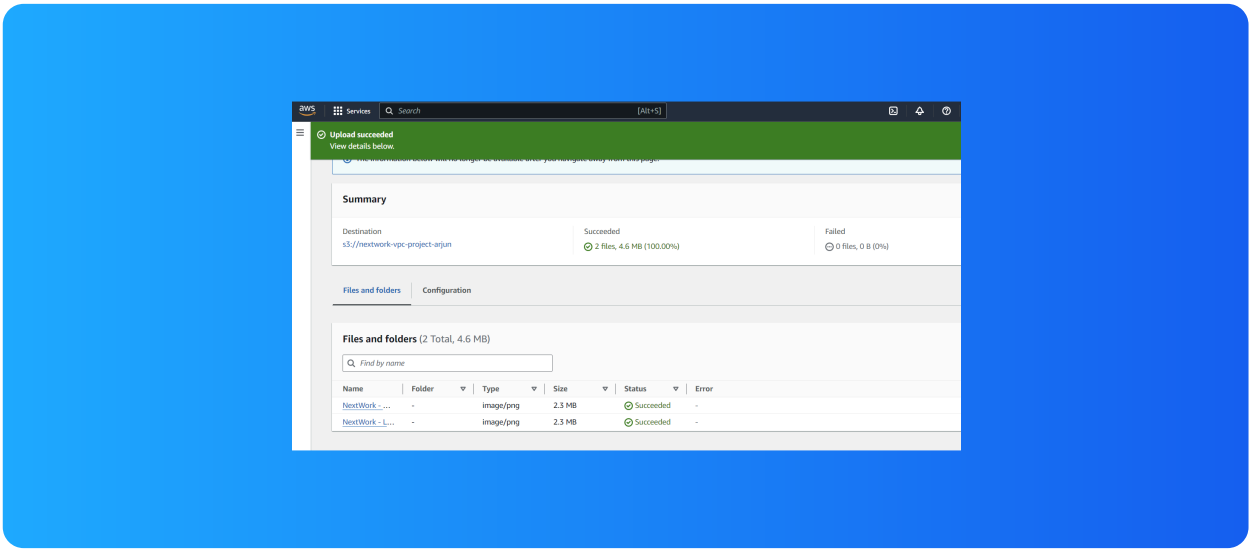
In this step, I am going to give access of my AWS environment to my EC2 instance.

AJ
ajv.singh3107@gmail.com
NextWork Student

Architecture set up

I started my project by launching a VPC, which is a virtual private cloud that provides a secure and isolated environment for my resources. Then, I launched an EC2 instance into the VPC.

I set up an S3 bucket titled **nextwork-vpc-project-arjun**, where I uploaded two files, "NextWork - Denzel is awesome.png" and "NextWork - Lelo is awesome.png." These files were saved locally and uploaded to the S3 bucket as part of the setup.



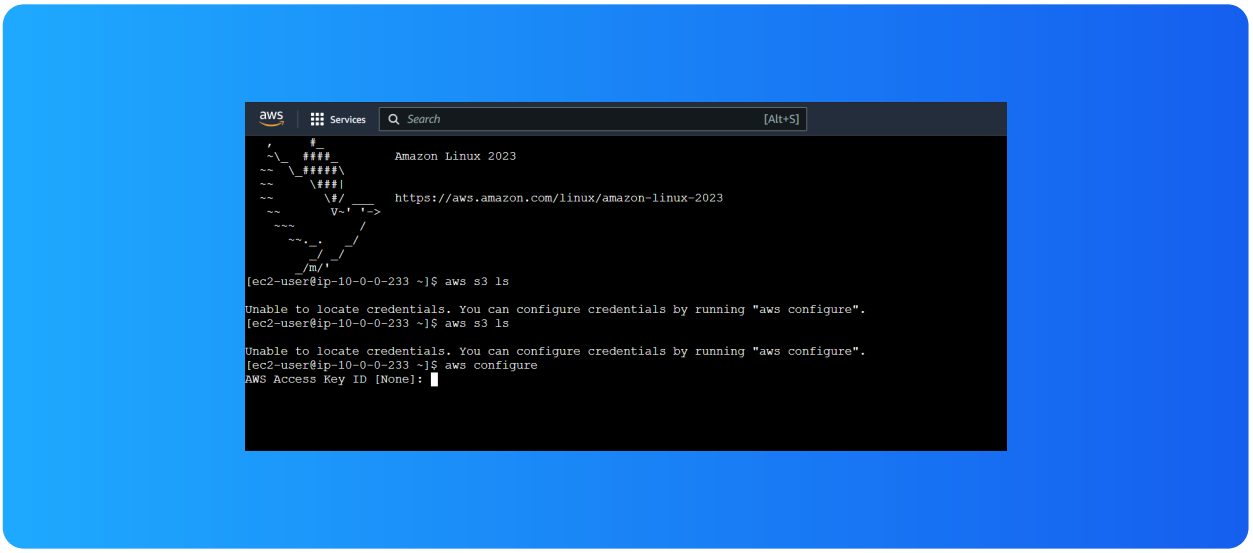
AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Running CLI commands

AWS CLI is a command-line interface that allows you to manage AWS resources from your terminal. I have access to AWS CLI because it is pre-installed on my Amazon Linux AMI EC2 instance.

The aws s3 ls command is a fundamental tool for interacting with Amazon S3. It allows you to list the contents of your S3 buckets, including objects and folders (prefixes).

The second command you ran was aws configure. This command is used to configure your AWS credentials and region for use with the AWS CLI.



AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Access keys

Credentials

To set up my EC2 instance to interact with AWS, I used the `aws configure` command to provide access keys, set the default region, and define output formats. This ensures secure communication between the instance and AWS services like S3.

Access keys are credentials for secure cloud access. They consist of an access key ID and a secret access key. The access key ID acts as a username, while the secret access key serves as a password.

Secret access keys are a type of credential used to authenticate users and applications to AWS services. They are unique codes that are assigned to individual users or applications.

Best practice

Although I'm using access keys in this project, a best practice alternative is to use IAM roles with temporary security credentials. IAM roles provide a more granular and secure way to grant access to AWS resources.

AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

In the second part of my project...

Step 4 - Set up an S3 bucket

In this instance, I am going to launch a bucket in S3. After that we are going to access it from the EC2 instance.

Step 5 - Connecting to my S3 bucket

I'm going to configure my EC2 instance to interact with the S3 bucket I created earlier. This allows the EC2 instance to access and manage the files in the bucket, enabling data transfers and other operations directly from the instance.

AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Connecting to my S3 bucket

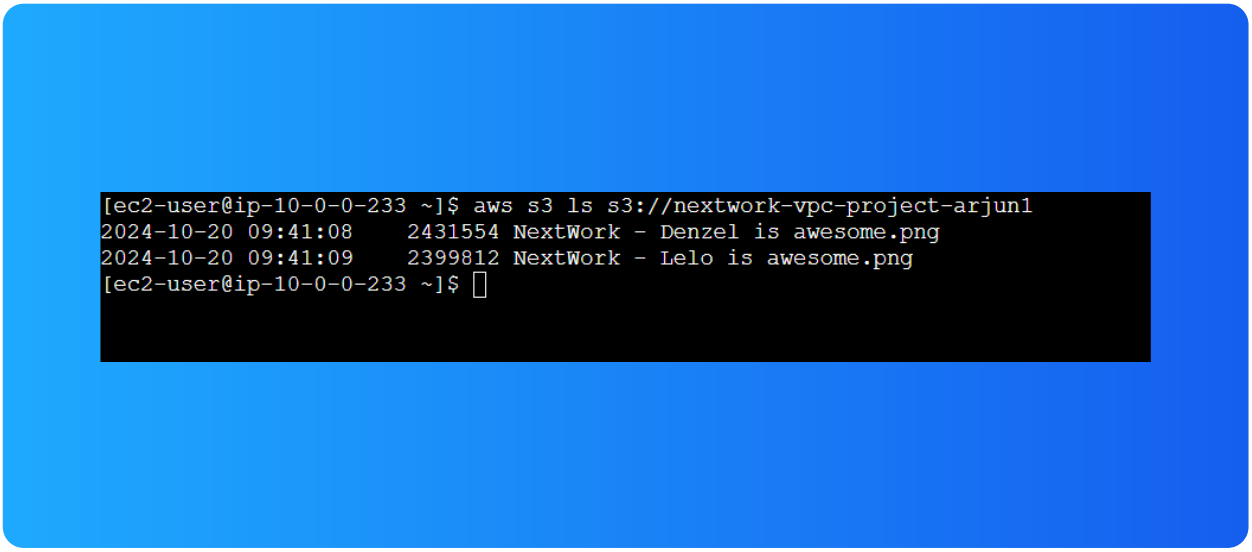
The aws s3 ls command is a fundamental tool for interacting with Amazon S3. It allows you to list the contents of your S3 buckets, including objects and folders (prefixes).

When I ran the command again, the terminal responded with "NextWork - Denzel is awesome.png" and "NextWork - Lelo is awesome.png." This confirmed the files were successfully uploaded to the S3 bucket and are accessible from the EC2 instance.

AJ
ajv.singh3107@gmail.com
NextWork Student

Connecting to my S3 bucket

Another CLI command I ran was `aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-arjun`, which returned a successful upload of the file "nextwork.txt" to my S3 bucket. This confirmed the file was copied from my EC2 instance to the bucket.



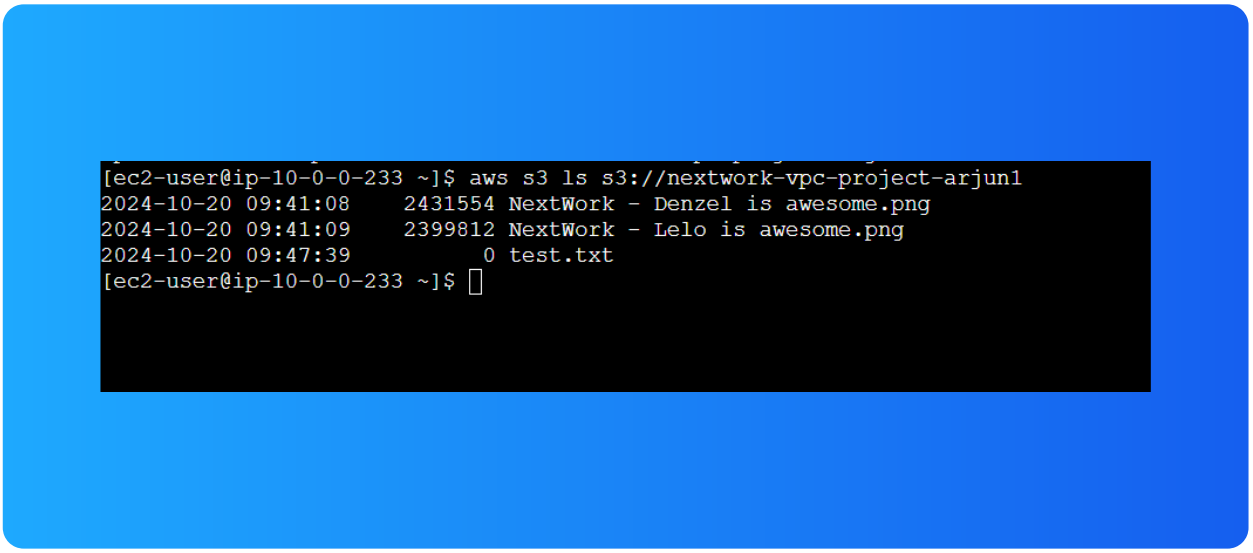
AJ
ajv.singh3107@gmail.com
NextWork Student

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 named "nextwork.txt" in the /tmp/ directory.

The second command I ran was aws s3 cp /tmp/nextwork.txt s3://nextwork-vpc-endpoints-arjun. This command will copy the file from the EC2 instance to the S3 bucket.

The third command I ran was aws s3 ls s3://nextwork-vpc-endpoints-arjun, which validated that "nextwork.txt" was successfully uploaded to the bucket.



Everyone should be in a job they love.

Check out nextwork.org for more projects