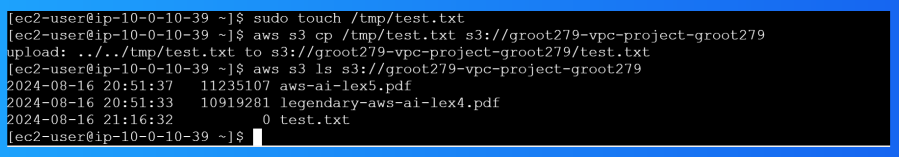
NextWork.org

Access S3 from a VPC

TA tahirgroot@gmail.com



TA
tahirgroot@gmail.com
NextWork Student
NextWork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPc is AWS foundational networking service that allows us to create our isolated networks within an AWS region and control network traffic and security

How I used Amazon VPC in this project

We launched a VPC with a public subnet and EC2 instance and directly accessed/managed an Amazon S3 bucket through the EC2 instance using AWS CLI

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

n/a

This project took me...

45 minutes

In the first part of my project...

Step 1 - Architecture set up

in this step , I Launch a vpc with a public subnet, I also launch an EC2 instance inside that public subnet.

Step 2 - Connect to my EC2 instance

In this step I directly access an EC2 instance using EC2 instance connect

Step 3 - Set up access keys

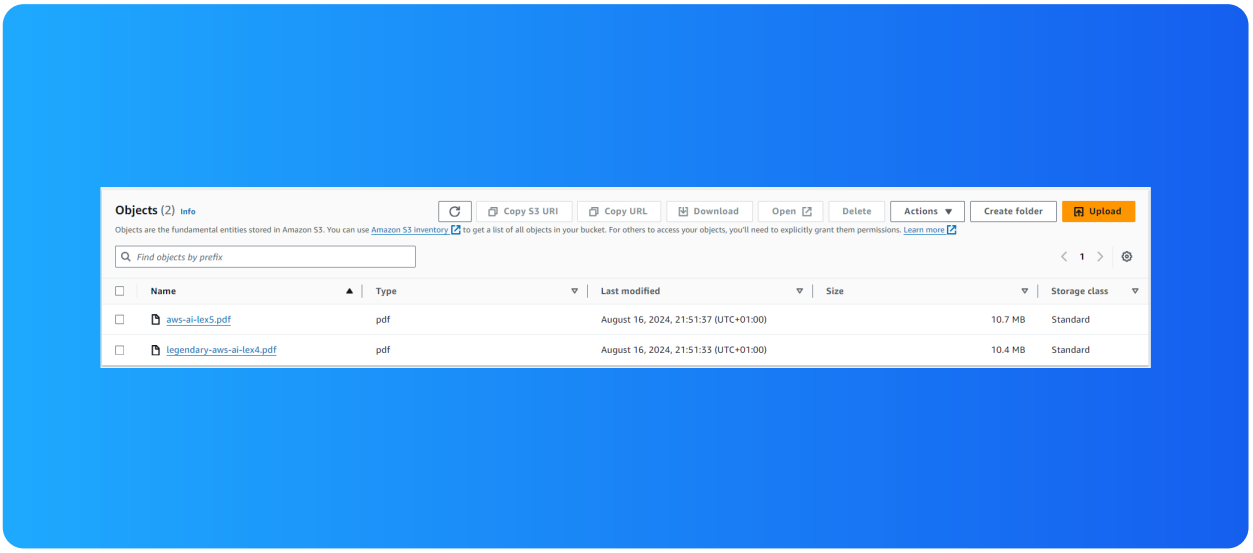
In this step, we create access keys so that our EC2 instance can access our AWS environment, specifically, interact with an S3 bucket.

TA
tahirgroot@gmail.com
NextWork Student

Architecture set up

i started my project by launching a VPC with a public subnet and an EC2 instance inside the public subnet

In this step, we use AWS CLI commands to control/ manage our s3 bucket. This means we're interacting with our s3 buckets through our EC2 instance/vpc instead of the AWS Management console

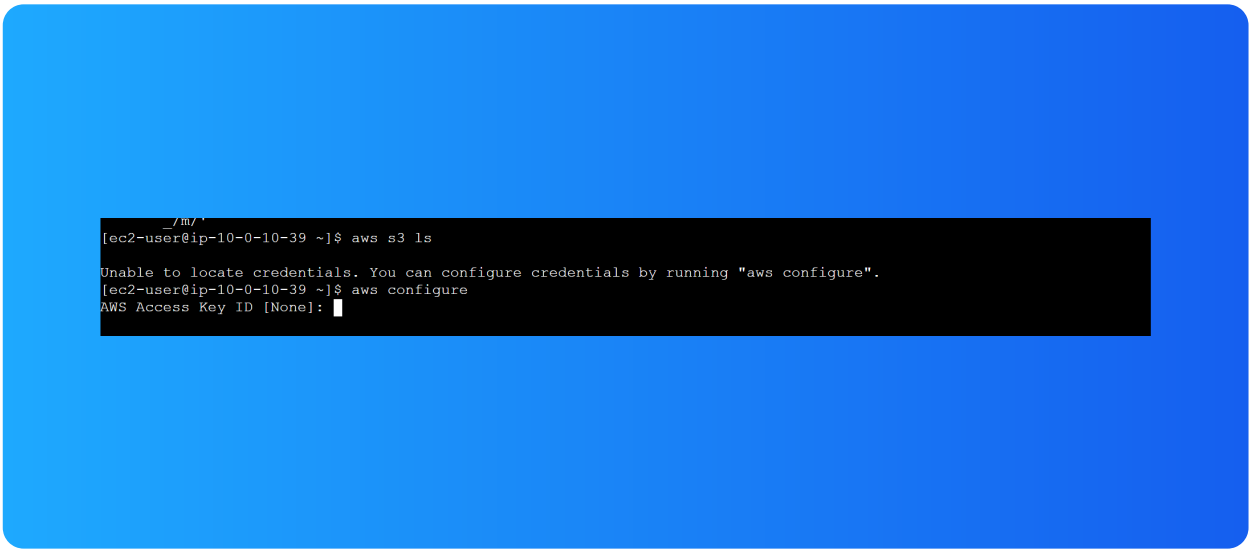


Running CLI commands

AWS CLI is a powerful command-line tool for managing AWS resources. I have access to AWS CLI because it's a widely used and supported tool for interacting with AWS services, providing flexibility and automation capabilities.

The first command I ran was aws s3 ls. This command is used to list all s3 buckets inside the AWS account (the Ec2 instance/application) that have access to

'The second command I ran was 'aws configure' This command is used to set up my EC2 instance credentials to access my AWS environment



Access keys

Credentials

To set up my EC2 instance to interact with my AWS environment, I configured an access key ID, secret Access Key, Default region and a default output format

Access keys are digital credentials that grant programmatic access to my AWS account. They consist of an Access Key ID and a Secret Access Key, which work together to authenticate requests made to AWS services.

Secret access keys are the confidential counterpart to access key IDs. They are used with the access key ID to authenticate requests to AWS services. Think of them as the password to my AWS account's digital front door.

Best practice

Although I'm using access keys in this project, a best practice alternative is using IAM roles with attached permissions. This is a more secure way to access an EC2 instance because tracking, attaching, and detaching IAM policies is much easier.

TA
tahirgroot@gmail.com
NextWork Student
NextWork.org

In the second part of my project...

Step 4 - Set up an S3 bucket

In this step, I launch an Amazon S3 bucket with two files inside. Later in the project, my EC2 instance will access this S3 bucket to test whether my keys have successfully given my EC2 instance access to my AWS resources.

Step 5 - Connecting to my S3 bucket

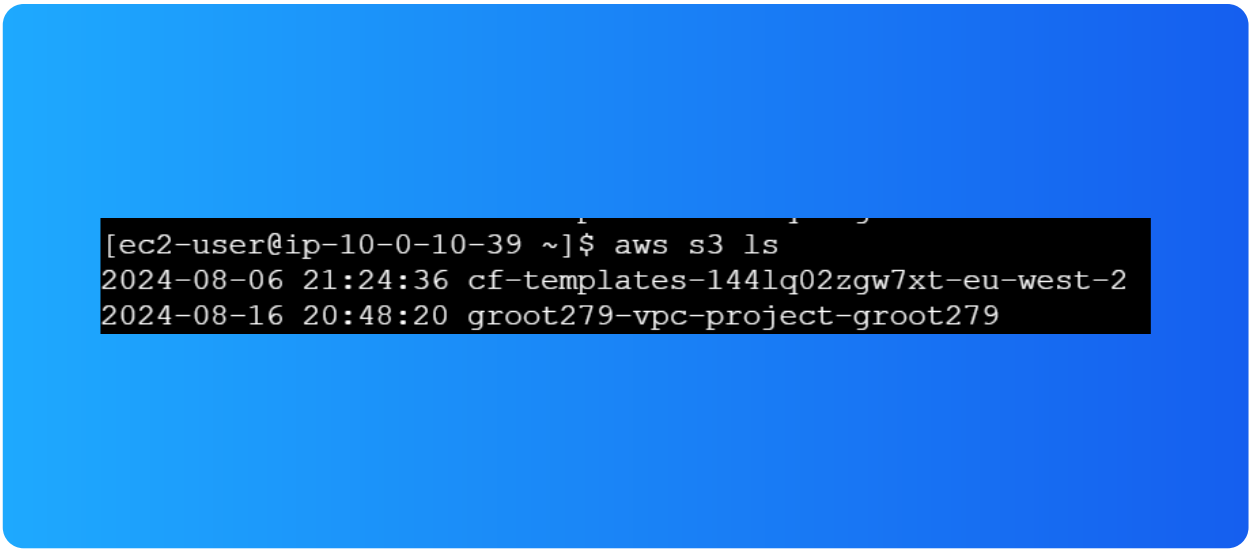
Connect to the S3 bucket

TA
tahirgroot@gmail.com
NextWork Student

Connecting to my S3 bucket

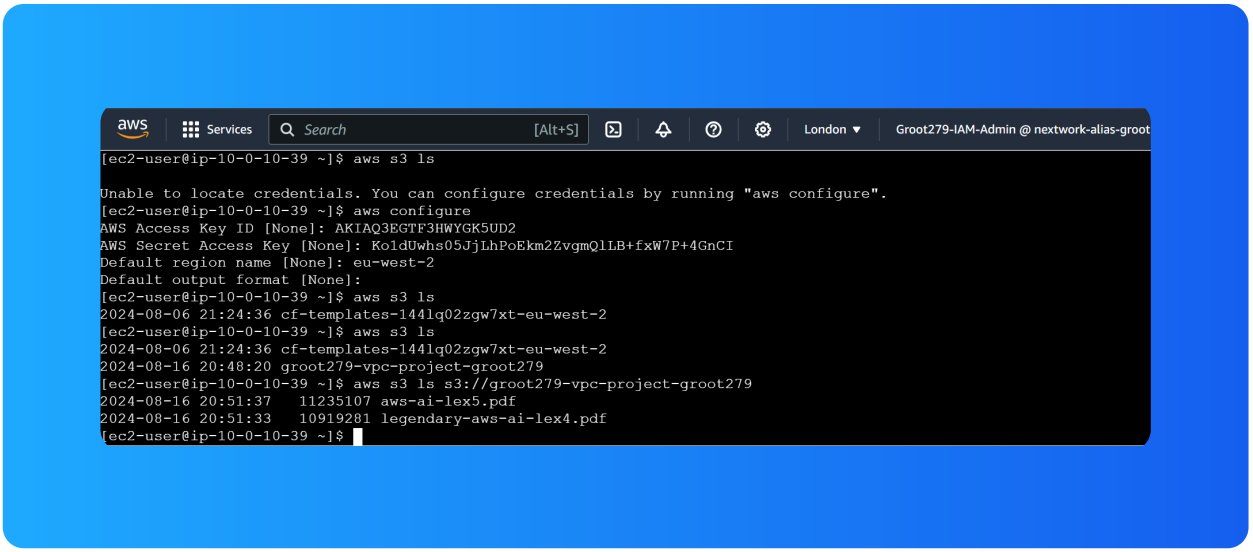
The first command I ran was aws s3 ls. This command is used to list all s3 buckets inside the AWS account (the Ec2 instance/application) that have access to

When I ran the command 'aws s3 ls' again, the terminal responded with a list of my s3 buckets. This indicated that my access keys works! i.e. my EC2 instance



Connecting to my S3 bucket

Another CLI command I ran was 'aws s3 ls s3://groot279-vpc-project-groot279, ' which returned a list of objects inside my S3 bucket.



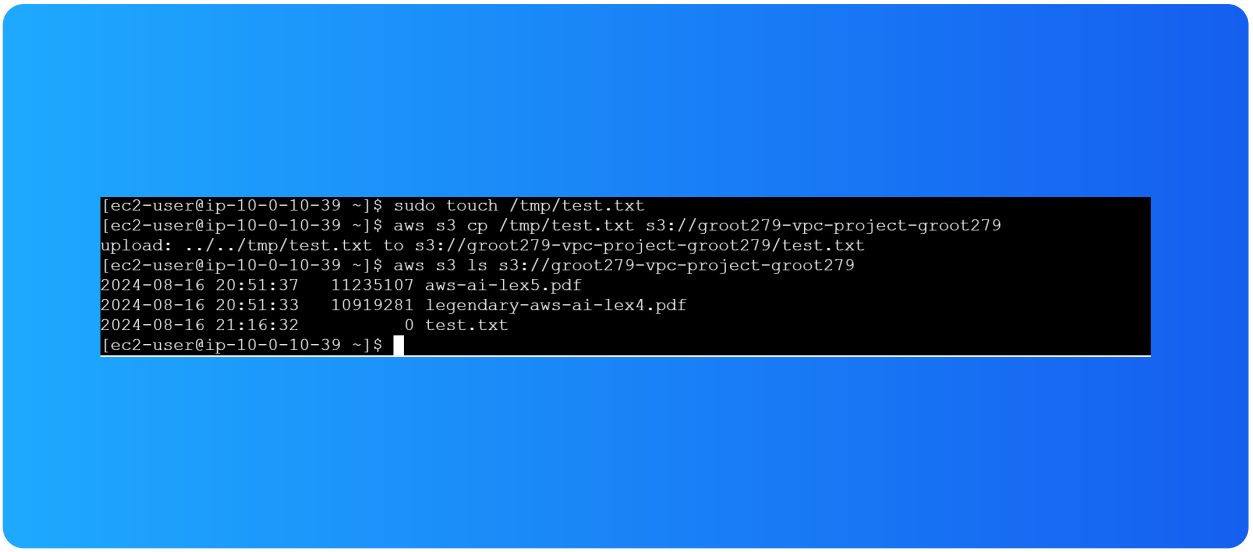
TA
tahirgroot@gmail.com
NextWork Student

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 blank file called test.txt in my EC2 instance's local directory.

My second command was 'aws s3 cp /tmp/test.txt s3://groot279-vpc-project-groot279'. This command will 'copy', i.e., upload the blank file created into my S3 bucket.

The third command I ran was aws s3 ls s3://groot279-vpc-project-groot279, which returned a list of all objects in my s3 bucket - including test.txt. This validated my EC2 instance through AWS CLI commands, and can get access to other AWS services(s3)



Everyone should be in a job they love.

Check out nextwork.org for more projects