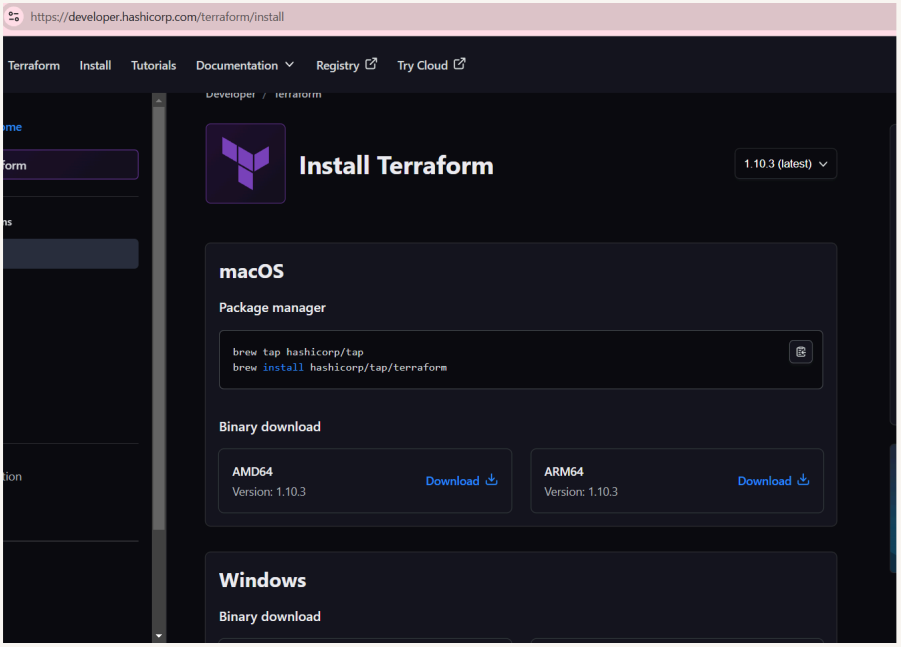
nextwork.org

Create S3 Buckets with Terraform

MA Marzena Pugo



MA
Marzena Pugo
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use Terraform to launch S3 bucket. The goal is to install and set up Terraform, troubleshoot any errors (i.e. installing the CLI) and then succesully apply my Terraform configuration to launch a bucket.

Tools and concepts

Services I used were Amazon S3, AWS IAM, Terraform and AWS CLI. Key concepts I learnt include infrastructure as code, creating configuration files, modularity of code,using providers, plugins, state files, lock files (Terraform concepts).

Project reflection

This project took me approximately 1.5 hours. The most challenging part was customizing my S3 configuration using the official Terraform documentation. It was most rewarding to see a local image uploaded into the S3 bucket.

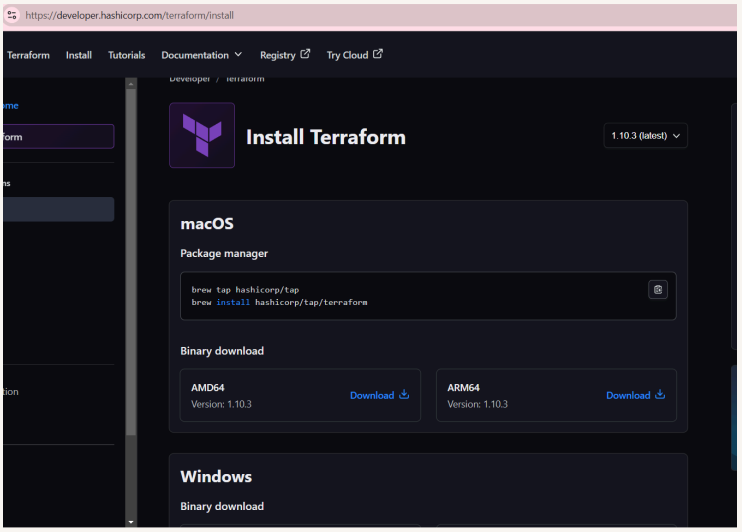
I did this project today because I am a beginner to Terraform and want to make a start. This project met my goals because I could install and set up Terraform and use it to launch resources in my AWS environment.

Introducing Terraform

Terraform is a tool for managing our IT resources using code. You use Terraform to process a configuration file you've prepared, that details the desired state of your infrastructure. Terraform then creates/updates to set up that desired state.

Terraform is one of the most popular tools used for infrastructure as code (IaC), which is a way to manage your IT infrastructure instead of manually managing resources using the console/text commands in CLI. IaC automates that process with code.

Terraform uses configuration files to understand the desired state of my infrastructure. main.tf is the main configuration file that I use in Terraform to describe that desired state.

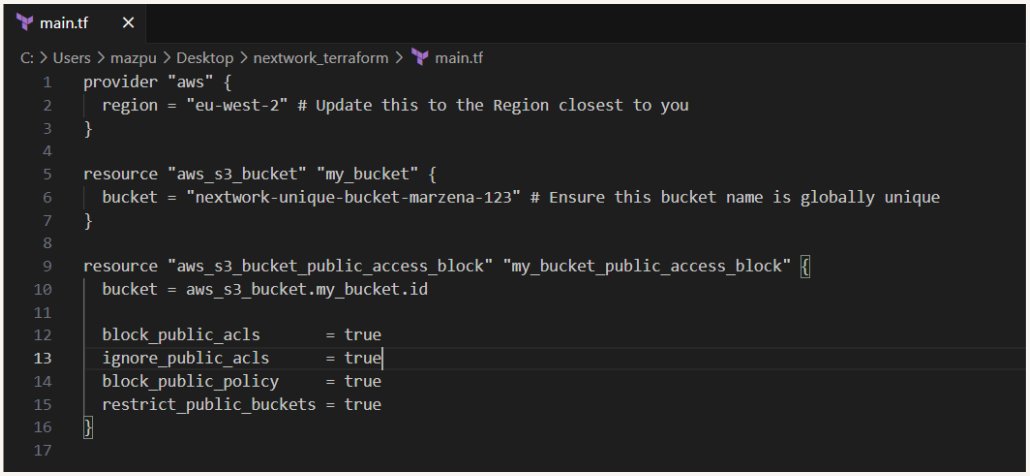

https://developer.hashicorp.com/terraform/install
Terraform
Install
Tutorials
Documentation
Registry
Try Cloud
Install Terraform
1.10.3 (latest)
macOS
Package manager
brew tap hashicorp/tap
brew install hashicorp/tap/terraform
Binary download
AMD64
Version: 1.10.3
Download
ARM64
Version: 1.10.3
Download
Windows
Binary download

Configuration files

The configuration is structured in blocks. Instead of a single block of code.The advantage of doing this is (i.e.modularity) is the ability to work on/update different parts of main.tf without affecting other parts. Great for teams collaborating too.

My main.tf configuration has three blocks

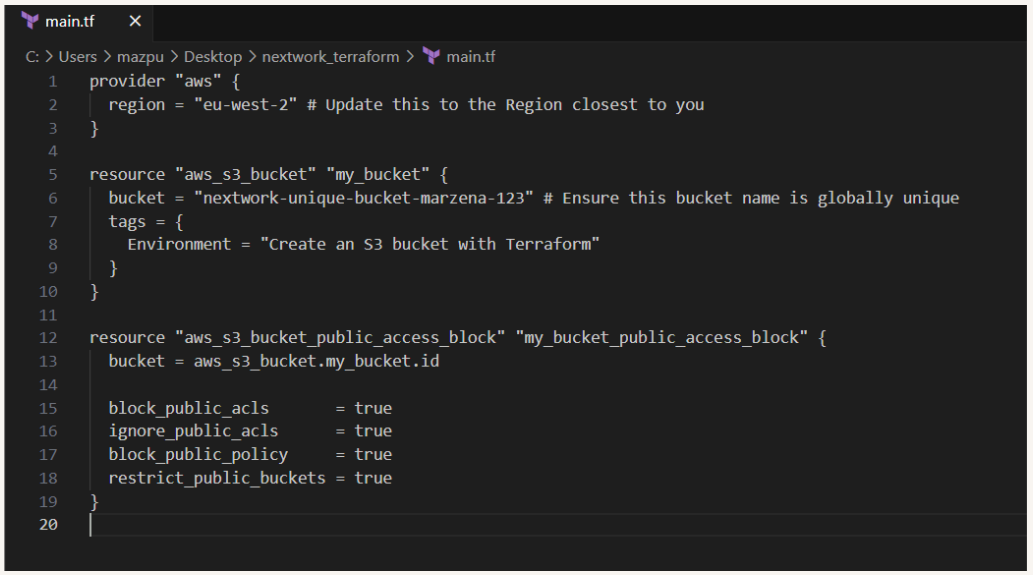
The first block indicates that I am using AWS as my provider for this infrastructure. The second block provisions an Amazon S3 bucket and gives it a unique name. The third block manages the bucket's permission i.e. blocking all public access.


main.tf
C: > Users > mazpu > Desktop > nextwork_terraform > main.tf
provider "aws" {
region = "eu-west-2" # Update this to the Region closest to you
}
resource "aws_s3_bucket" "my_bucket" {
bucket = "nextwork-unique-bucket-marzena-123" # Ensure this bucket name is globally unique
}
resource "aws_s3_bucket_public_access_block" "my_bucket_public_access_block" {
bucket = aws_s3_bucket.my_bucket.id
block_public_acls = true
ignore_public_acls = true
block_public_policy = true
restrict_public_buckets = true
}

Customizing my S3 Bucket

For my project extension, I visited the official Terraform documentation to look for ways I can customize my bucket's configurations. The documentation shows example configurations, all the available parameters for a resource, and usage rules.

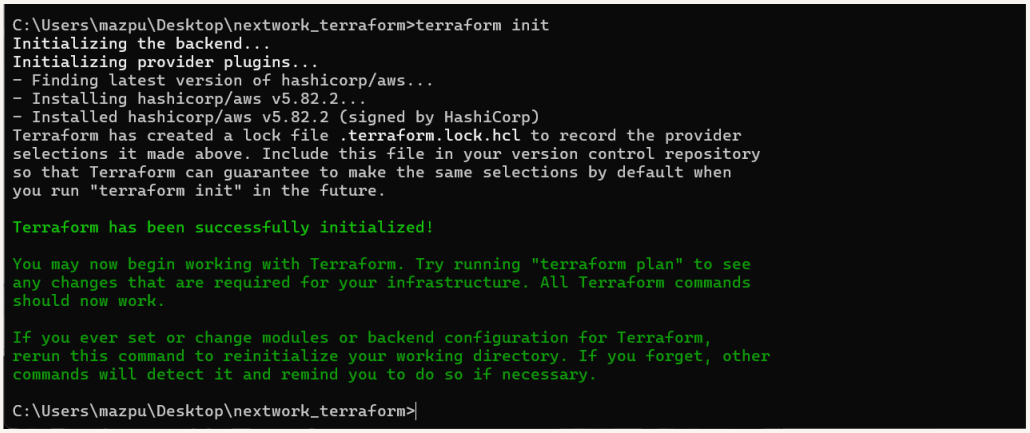
I chose to customise my bucket by adding tags, because it lets me identify the project that I launched it for later on. When I launch my bucket, I can verify my customization by visiting the "Tags" panel in the S3 console for this bucket.



Terraform commands

'I ran 'terraform init' to initialize Terraform. This means getting it set up the backend to store state files, and install the necessary plugins i.e. the AWS provider plugin.

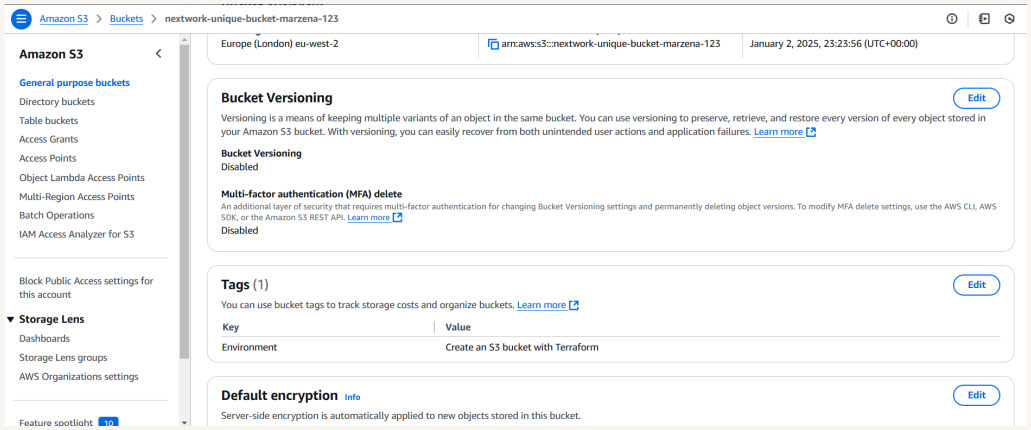
Next, I ran 'terraform plan' to get Terraform compare the infrastructure/ resources in main.tf, against my infrastructure's current state. Then share back the execution plan i.e. how running main.tf will impact my infrastucture e.g. create resources.



Lanching the S3 Bucket

I ran 'terraform apply' to get Terraform to appy the changes/ updates it showed us when I ran "terraform apply". Running 'terraform apply' will affect my AWS account by creating to resources- an S3 bucket, and its set of permissions settings.

The sequence of running terraform init, plan, and apply is crucial because I need to initialize Terraform first before I get to 1) make any comparison between main.tf and my current infrastructure or 2) connect to AWS in the first place.

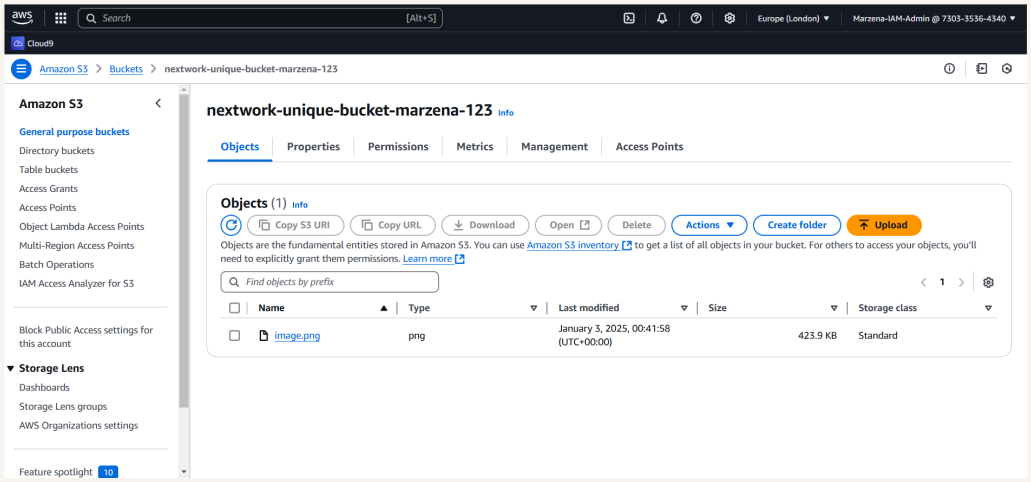


Uploading an S3 Object

I created a new resource block to upload an image called "image.png" into the S3 bucket. The image will be uploaded straight away without being nested in a subfolder in the bucket.

I need to run terraform apply again because I have updated my Terraform configration file, which means there are changes that Terraform needs to review and compare with my current infrastructure again. This time only one new resource is created image

To validate that I've updated my configuration successfully, I checked the S3 bucket and confirmed that a new image is inside. I also downloaded the imagee back me and confirmed that it was a correct image that was initially uploaded.


Search
[Alt+S]
Europe (London)
Marzena-IAM-Admin @ 7303-3536-4340
Cloud9
Amazon S3 > Buckets > nextwork-unique-bucket-marzena-123
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 10
nextwork-unique-bucket-marzena-123 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
image.png
png
January 3, 2025, 00:41:58 (UTC+00:00)
423.9 KB
Standard

The place to learn & showcase your skills

Check out nextwork.org for more projects