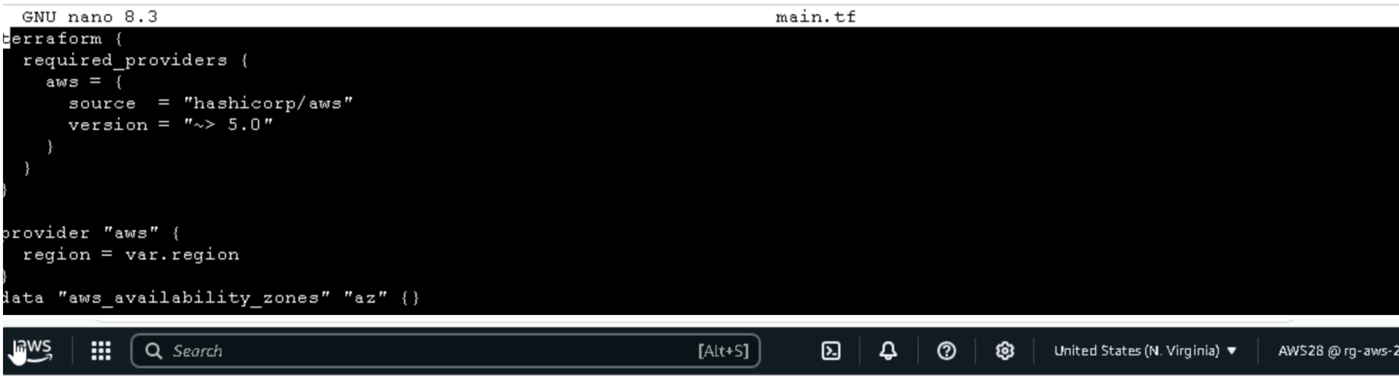
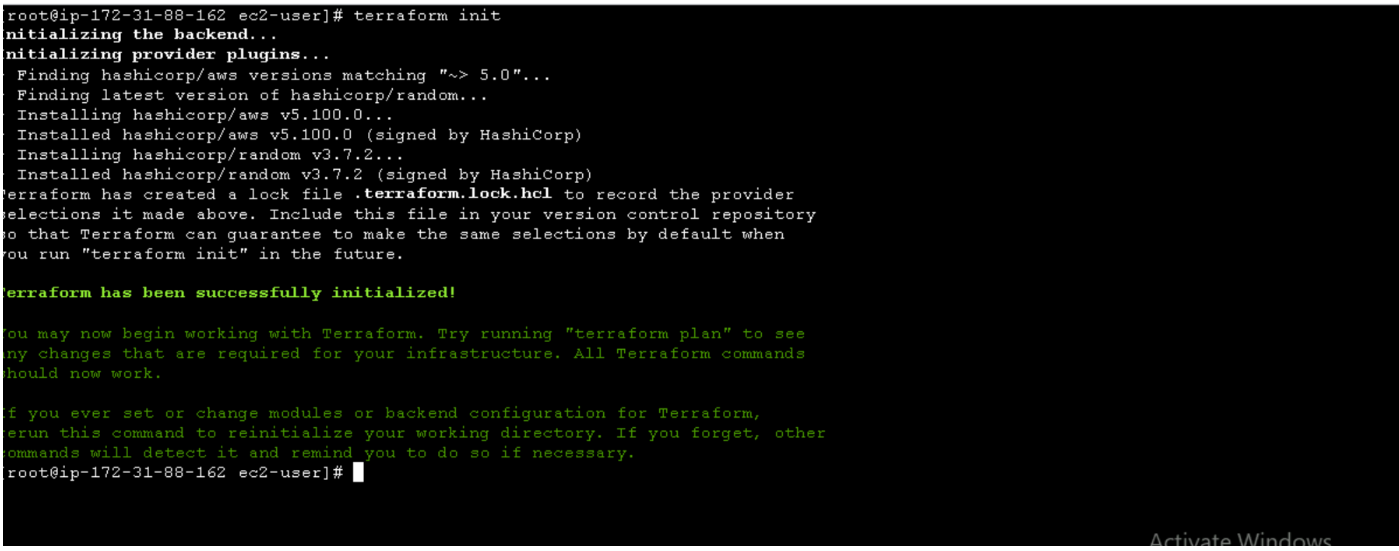
# Creating and maintaining s3 bucket using terraform

## 1.Initializing terraform and provider

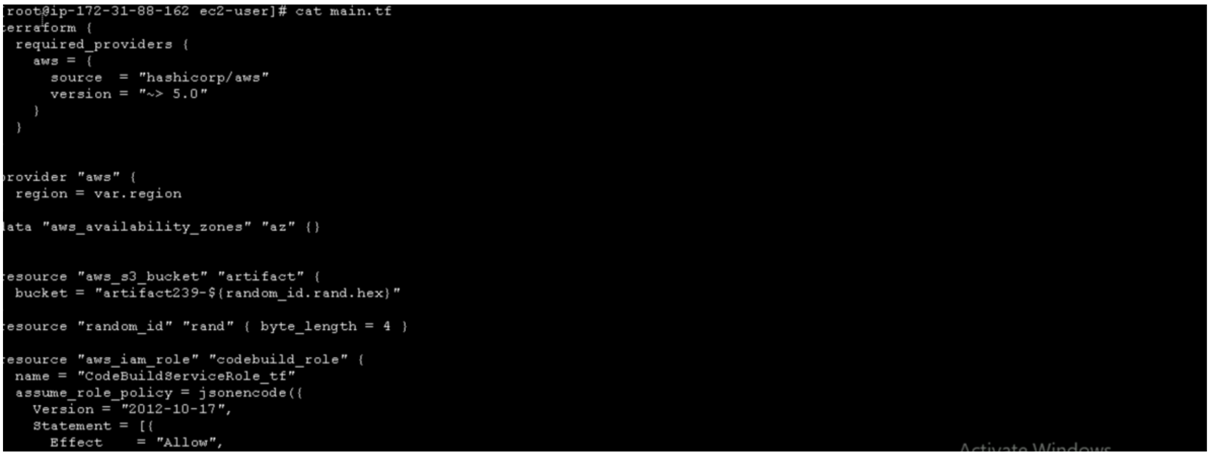


```
 GNU nano 8.3                                                         main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = var.region
}
data "aws_availability_zones" "az" {}
```

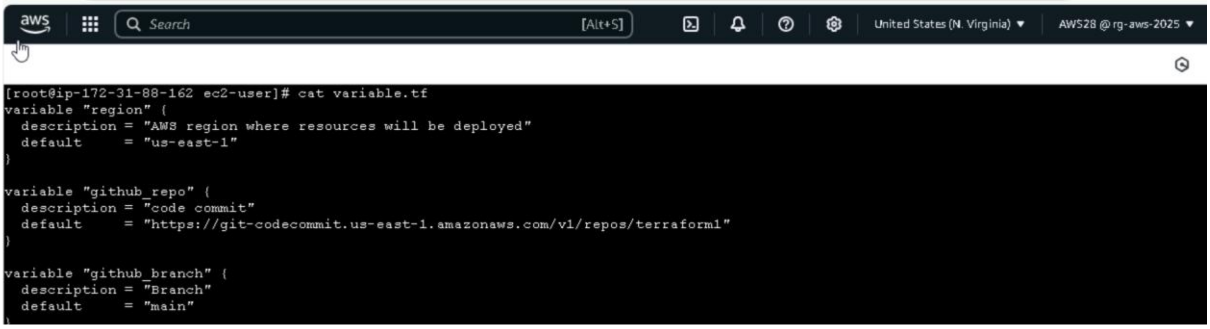


```
[root@ip-172-31-88-162 ec2-user]# terraform init
nitializing the backend...
nitializing provider plugins...
 Finding hashicorp/aws versions matching "~> 5.0"...
 Finding latest version of hashicorp/random...
 Installing hashicorp/aws v5.100.0...
 Installed hashicorp/aws v5.100.0 (signed by HashiCorp)
 Installing hashicorp/random v3.7.2...
 Installed hashicorp/random v3.7.2 (signed by HashiCorp)
erraform has created a lock file .terraform.lock.hcl to record the provider
elections it made above. Include this file in your version control repository
o that Terraform can guarantee to make the same selections by default when
ou run "terraform init" in the future.

erraform has been successfully initialized!

ou may now begin working with Terraform. Try running "terraform plan" to see
ny changes that are required for your infrastructure. All Terraform commands
hould now work.

f you ever set or change modules or backend configuration for Terraform,
erun this command to reinitialize your working directory. If you forget, other
ommands will detect it and remind you to do so if necessary.
[root@ip-172-31-88-162 ec2-user]#
```

## Main.tf and variable .tf files



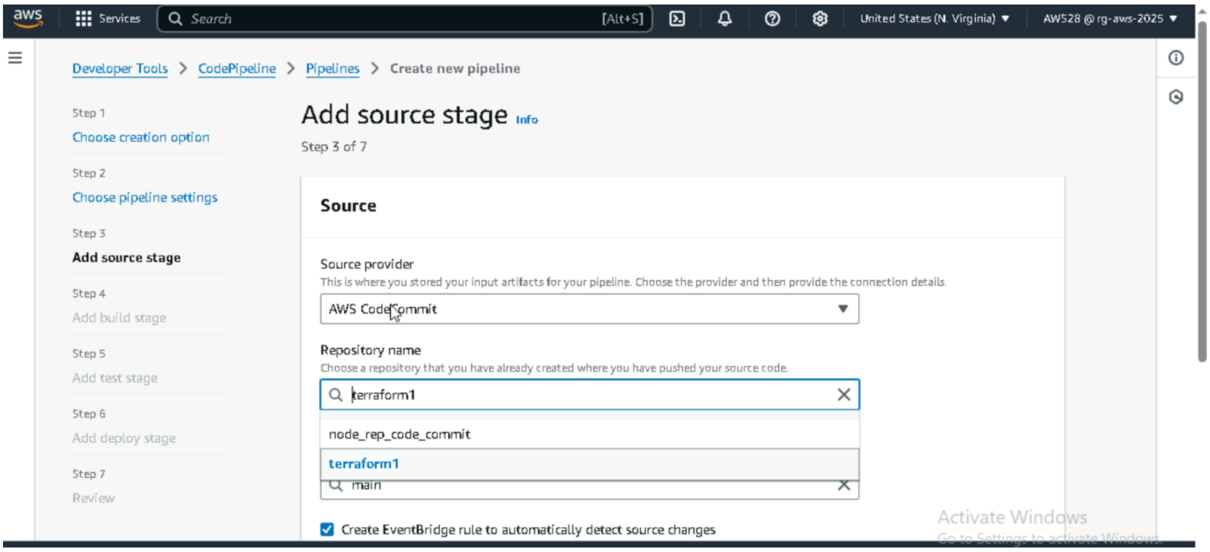
```
[root@ip-172-31-88-162 ec2-user]# cat main.tf
erraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }


rovider "aws" {
  region = var.region

ata "aws_availability_zones" "az" {}


esource "aws_s3_bucket" "artifact" {
  bucket = "artifact239-${random_id.rand.hex}"

esource "random_id" "rand" { byte_length = 4 }

esource "aws_iam_role" "codebuild_role" {
  name = "CodeBuildServiceRole_tf"
  assume_role_policy = jsonencode({
    Version = "2012-10-17",
    Statement = [{
      Effect    = "Allow",
```



```
[root@ip-172-31-88-162 ec2-user]# cat variable.tf
variable "region" {
  description = "AWS region where resources will be deployed"
  default     = "us-east-1"
}

variable "github_repo" {
  description = "code commit"
  default     = "https://git-codecommit.us-east-1.amazonaws.com/v1/repos/terraform1"
}

variable "github_branch" {
  description = "Branch"
  default     = "main"
```

## Aws code pipeline to run a terraform scripts:

Connection to code commit



Add source stage

Step 3 of 7

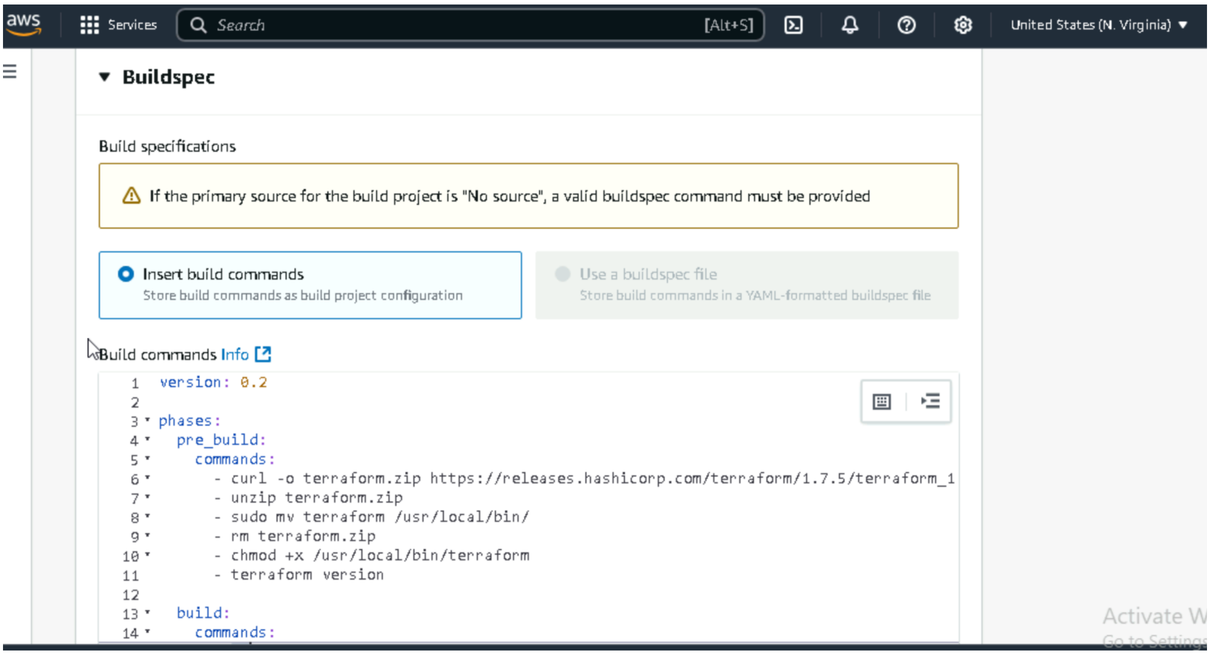
Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
AWS CodeCommit ▾

Repository name
Choose a repository that you have already created where you have pushed your source code.
terraform1
node_rep_code_commit
terraform1
main

☑ Create EventBridge rule to automatically detect source changes

**Code build:**I have added a build stage andfor build terraforminit,plan,apply



▾ Buildspec

Build specifications

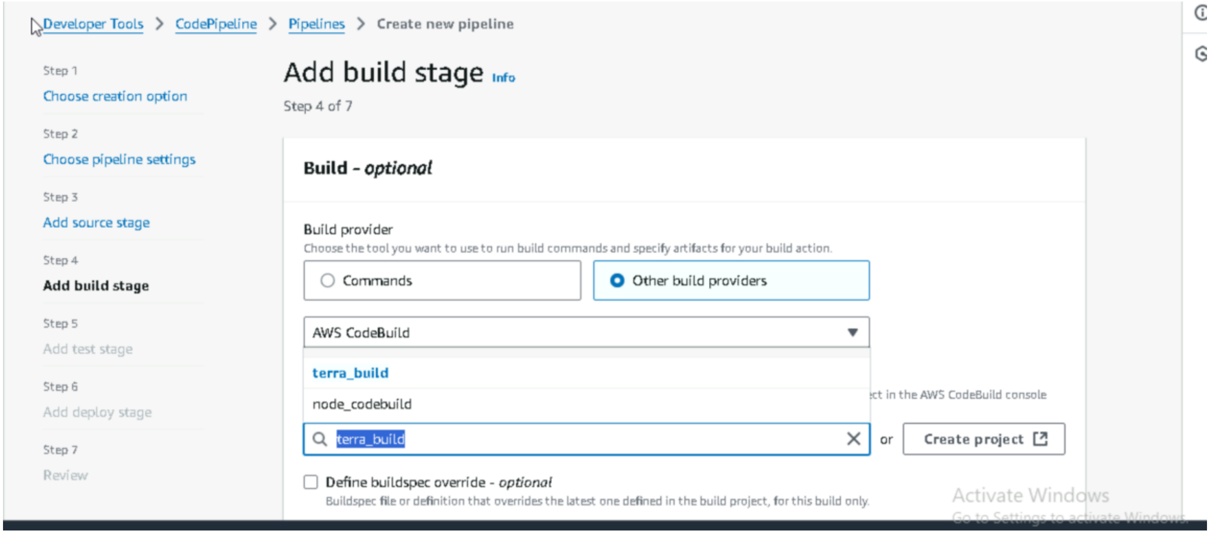
⚠ If the primary source for the build project is "No source", a valid buildspec command must be provided

● Insert build commands
Store build commands as build project configuration

○ Use a buildspec file
Store build commands in a YAML-formatted buildspec file

Build commands Info

```
version: 0.2

phases:
  pre_build:
    commands:
      - curl -o terraform.zip https://releases.hashicorp.com/terraform/1.7.5/terraform_1
      - unzip terraform.zip
      - sudo mv terraform /usr/local/bin/
      - rm terraform.zip
      - chmod +x /usr/local/bin/terraform
      - terraform version

  build:
    commands:
```



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose creation option
Step 2 Choose pipeline settings
Step 3 Add source stage
Step 4 **Add build stage**
Step 5 Add test stage
Step 6 Add deploy stage
Step 7 Review

Add build stage Info

Step 4 of 7

**Build - *optional***

Build provider
Choose the tool you want to use to run build commands and specify artifacts for your build action.

○ Commands ● Other build providers

AWS CodeBuild

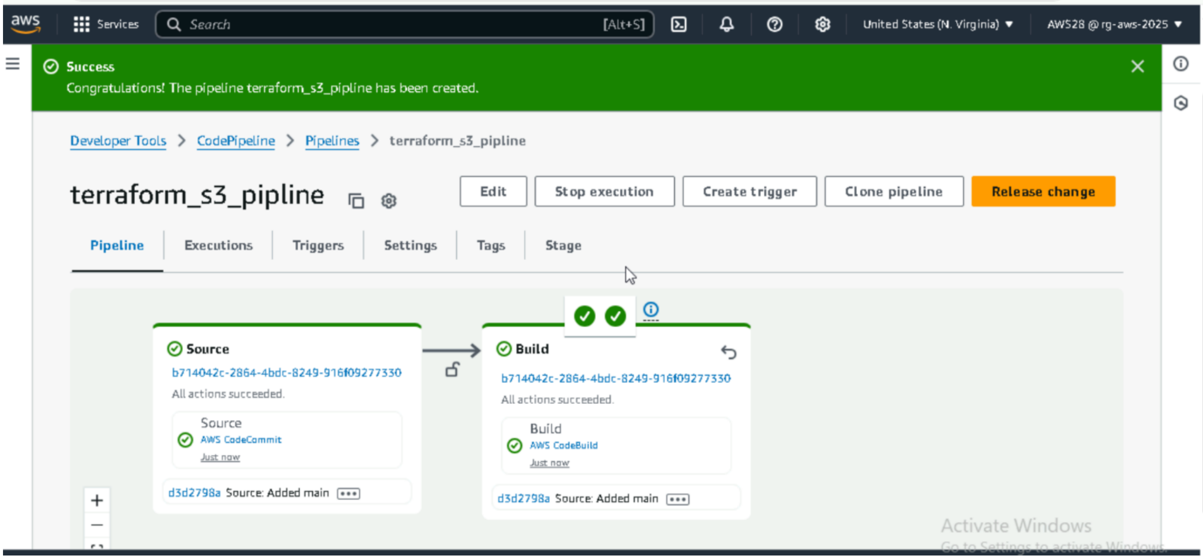
terra_build
node_codebuild

terra_build or Create project

…ct in the AWS CodeBuild console

□ Define buildspec override - *optional*
Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

## Deployed and verified



Success
Congratulations! The pipeline terraform_s3_pipline has been created.

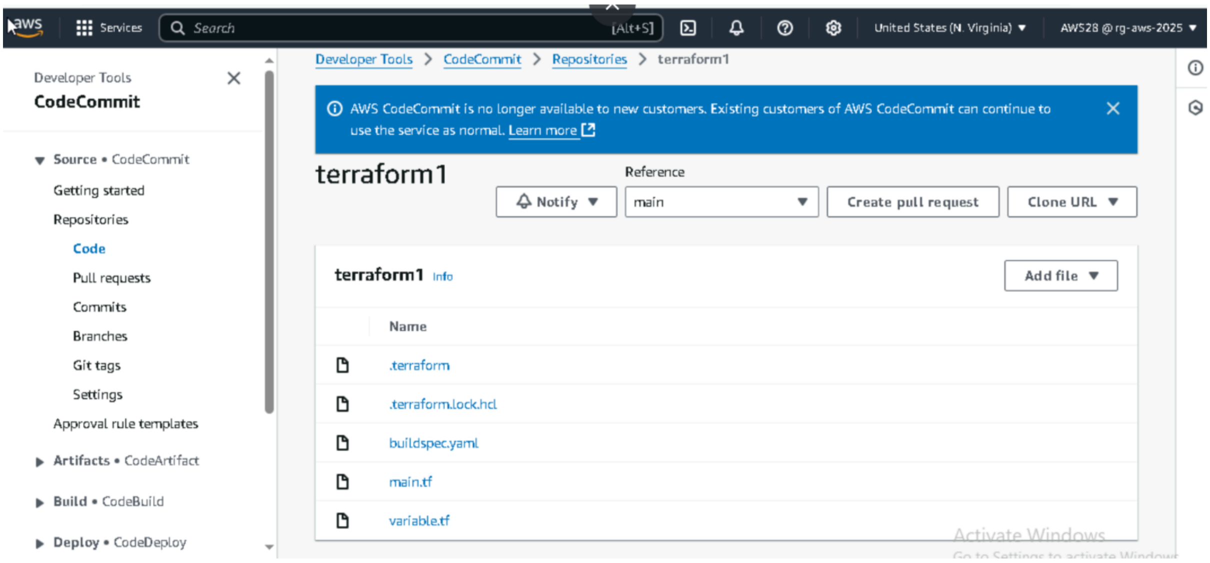
Developer Tools > CodePipeline > Pipelines > terraform_s3_pipline

terraform_s3_pipline

Source
b714042c-2864-4bdc-8249-916f09277330
All actions succeeded.
Source
AWS CodeCommit
Just now
d3d2798a Source: Added main

Build
b714042c-2864-4bdc-8249-916f09277330
All actions succeeded.
Build
AWS CodeBuild
Just now
d3d2798a Source: Added main

Store terraform files in aws code commit :



AWS CodeCommit is no longer available to new customers. Existing customers of AWS CodeCommit can continue to use the service as normal. Learn more

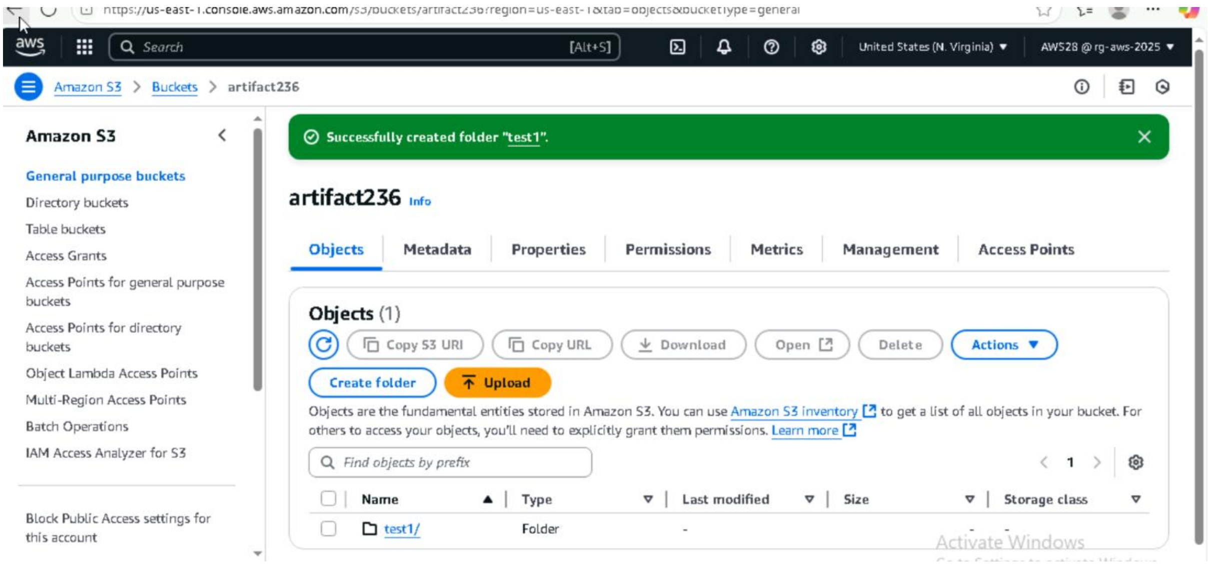
terraform1

Reference: main

terraform1 Info

| Name |
|---|
| .terraform |
| .terraform.lock.hcl |
| buildspec.yaml |
| main.tf |
| variable.tf |

# S3 bucket


https://us-east-1.console.aws.amazon.com/s3/buckets/artifact236?region=us-east-1&tab=objects&bucketType=general
Search [Alt+S]
United States (N. Virginia)
AW528 @ rg-aws-2025
Amazon S3 > Buckets > artifact236
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points for general purpose buckets
Access Points for directory buckets
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Activate Windows

Successfully created folder "test1".

artifact236 Info

Objects (1)

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

| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|
| test1/ | Folder | - | - | - |