

Creating S3 Bucket using terraform

Rushabh Jain
D15B 25

Step 1: Write a Terraform Script in Atom for creating S3 Bucket on Amazon AWS

```
resource "aws_s3_bucket" "rushabh" {  
  bucket = "rush"  
  
  tags = {  
    Name      = "My Bucket"  
    Environment = "Dev"  
  }  
}
```

Create a new provider.tf file and write the following contents into it.

```
provider "aws" {  
  access_key = "AKIAIYIY3PKM7VRJB3TFP"  
  secret_key = "ubbfj7ZgxAiL87aSYCGUCSH2Cy8SE53ZHr-fMqNfZ"  
  region = "ap-south-1"  
}
```

Save both the files in same directory Terraform_Scripts/S3

Step 2: Open Command Prompt and go to Terraform_Script\S3 directory where our .tf files are stored

```
Command Prompt

C:\>cd terraform_scripts

C:\Terraform_Scripts>cd s3

C:\Terraform_Scripts\S3>dir
Volume in drive C has no label.
Volume Serial Number is 2E74-E8C2

Directory of C:\Terraform_Scripts\S3

08/11/2022  09:01 AM    <DIR>          .
08/11/2022  09:01 AM    <DIR>          ..
08/11/2022  09:05 AM                135 provider.tf
08/11/2022  09:05 AM                151 s3.tf
                2 File(s)                286 bytes
                2 Dir(s)  133,766,430,720 bytes free

C:\Terraform_Scripts\S3>
```

Step 3: Execute Terraform Init command to initialize the resources

```
C:\terraform_1.9.3_windows_amd64\script>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.63.0...
- Installed hashicorp/aws v5.63.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Step 4: Execute Terraform plan to see the available resources

```

C:\terraform_1.9.3_windows_amd64\script>terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_s3_bucket.Rush will be created
+ resource "aws_s3_bucket" "Rush" {
  + acceleration_status      = (known after apply)
  + acl                     = (known after apply)
  + arn                    = (known after apply)
  + bucket                 = "RushabhValoDegenerate88"
  + bucket_domain_name     = (known after apply)
  + bucket_prefix          = (known after apply)
  + bucket_regional_domain_name = (known after apply)
  + force_destroy          = false
  + hosted_zone_id         = (known after apply)
  + id                    = (known after apply)
  + object_lock_enabled     = (known after apply)
  + policy                 = (known after apply)
  + region                 = (known after apply)
  + request_payer           = (known after apply)
  + tags                   = {
    + "Environment" = "Dev"
    + "Name"        = "My Bucket"
  }
  + tags_all              = {
    + "Environment" = "Dev"
    + "Name"        = "My Bucket"
  }
  + website_domain         = (known after apply)
  + website_endpoint       = (known after apply)

  + cors_rule (known after apply)

  + grant (known after apply)

  + lifecycle_rule (known after apply)

  + logging (known after apply)

  + object_lock_configuration (known after apply)

  + replication_configuration (known after apply)

  + server_side_encryption_configuration (known after apply)

  + versioning (known after apply)

  + website (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

```

Step 5: Execute Terraform apply to apply the configuration, which will automatically create an S3 bucket based on our configuration.

```

C:\terraform_1.9.3_windows_amd64\script>terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_s3_bucket.Rush will be created
+ resource "aws_s3_bucket" "Rush" {
+   acceleration_status      = (known after apply)
+   acl                     = (known after apply)
+   arn                     = (known after apply)
+   bucket                  = "rushabhvalodegenerate"
+   bucket_domain_name      = (known after apply)
+   bucket_prefix           = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   force_destroy           = false
+   hosted_zone_id          = (known after apply)
+   id                     = (known after apply)
+   object_lock_enabled      = (known after apply)
+   policy                  = (known after apply)
+   region                 = (known after apply)
+   request_payer           = (known after apply)
+   tags                   = {
+     "Environment" = "Dev"
+     "Name"        = "My Bucket"
+   }
+   tags_all              = {
+     "Environment" = "Dev"
+     "Name"        = "My Bucket"
+   }
+   website_domain        = (known after apply)
+   website_endpoint      = (known after apply)
+   cors_rule             (known after apply)
+   grant                 (known after apply)
+   lifecycle_rule        (known after apply)
+   logging               (known after apply)
+   object_lock_configuration (known after apply)
+   replication_configuration (known after apply)
+   server_side_encryption_configuration (known after apply)
+   versioning            (known after apply)
+   website               (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_s3_bucket.Rush: Creating...
aws_s3_bucket.Rush: Creation complete after 7s [id=rushabhvalodegenerate]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

```

AWS S3bucket dashboard, Before Executing Apply command:

The screenshot shows the AWS S3 Buckets dashboard. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and user information. Below this is a blue banner with a tutorial link. The main content area is titled 'Amazon S3 > Buckets'. It features an 'Account snapshot' section with a 'View Storage Lens dashboard' button. Below that, the 'Buckets (2)' section is displayed, including a 'Copy ARN' button, 'Empty', 'Delete', and 'Create bucket' buttons. A search bar for finding buckets by name is present. A table lists the existing buckets:

	Name	AWS Region	Access	Creation date
<input type="radio"/>	codepipeline-ap-south-1-738902759473	Asia Pacific (Mumbai) ap-south-1	Objects can be public	July 20, 2022, 13:26:08 (UTC+05:30)
<input type="radio"/>	elasticbeanstalk-ap-south-1-568603923263	Asia Pacific (Mumbai) ap-south-1	Objects can be public	July 7, 2022, 08:05:29 (UTC+05:30)

AWS S3 Bucket dashboard, After Executing Apply step:

The screenshot shows the AWS S3 Buckets dashboard. At the top, there's a breadcrumb 'Amazon S3 > Buckets' and a 'View Storage Lens dashboard' button. Below this is an 'Account snapshot' section with a 'Learn more' link. The main section is titled 'Buckets (2)' and includes buttons for 'Refresh', 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A search bar is present with the placeholder 'Find buckets by name'. Below the search bar is a table with the following data:

	Name	AWS Region	Access	Creation date
<input type="radio"/>	elasticbeanstalk-ap-south-1-568603923263	Asia Pacific (Mumbai) ap-south-1	Objects can be public	July 7, 2022, 08:05:29 (UTC+05:30)
<input type="radio"/>	my-bj-terraform-test-bucket	Asia Pacific (Mumbai) ap-south-1	Public	August 11, 2022, 09:33:32 (UTC+05:30)

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete an EC2 instance

```
C:\terraform_1.9.3_windows_amd64\script>terraform destroy
aws_s3_bucket.Rush: Refreshing state... [id=rushabhvalodegenerate]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_s3_bucket.Rush will be destroyed
- resource "aws_s3_bucket" "Rush" {
  - arn = "arn:aws:s3:::rushabhvalodegenerate" -> null
  - bucket = "rushabhvalodegenerate" -> null
  - bucket_domain_name = "rushabhvalodegenerate.s3.amazonaws.com" -> null
  - bucket_regional_domain_name = "rushabhvalodegenerate.s3.us-east-1.amazonaws.com" -> null
  - force_destroy = false -> null
  - hosted_zone_id = "Z3AQ8STGFYJSTF" -> null
  - id = "rushabhvalodegenerate" -> null
  - object_lock_enabled = false -> null
  - region = "us-east-1" -> null
  - request_payer = "BucketOwner" -> null
  - tags = {
    - "Environment" = "Dev"
    - "Name" = "My Bucket"
  } -> null
  - tags_all = {
    - "Environment" = "Dev"
    - "Name" = "My Bucket"
  } -> null
  # (3 unchanged attributes hidden)

  - grant {
    - id = "5cf69121f6024296a31f10380ffb5fedeb079c92a2bbe87098a3e6d4389f5ef1" -> null
    - permissions = [
      - "FULL_CONTROL",
    ] -> null
    - type = "CanonicalUser" -> null
    # (1 unchanged attribute hidden)
  }

  - server_side_encryption_configuration {
    - rule {
      - bucket_key_enabled = false -> null

      - apply_server_side_encryption_by_default {
        - sse_algorithm = "AES256" -> null
        # (1 unchanged attribute hidden)
      }
    }
  }

  - versioning {
    - enabled = false -> null
    - mfa_delete = false -> null
  }
}

Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_s3_bucket.Rush: Destroying... [id=rushabhvalodegenerate]
aws_s3_bucket.Rush: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
```

AWS EC2 dashboard, After Executing Destroy step:

Amazon S3 > Buckets

Account snapshot [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

Buckets (1) [Info](#) [Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3. [Learn more](#)

< 1 > [Settings](#)

	Name	AWS Region	Access	Creation date
<input type="radio"/>	elasticbeanstalk-ap-south-1-568603923263	Asia Pacific (Mumbai) ap-south-1	Objects can be public	July 7, 2022, 08:05:29 (UTC+05:30)