

Creating a s3 bucket and uploading a file into the bucket

Step 1: Launch an instance and connect to the server.

- After connecting to the server install aws cli and terraform

```
apt install unzip -y

Installing aws cli

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install

Installing terraform

wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
```

- Then pass aws credentials to the terraform using command
“aws configure” -> “<Give access key>” -> “<Secret access key>” -> “<region>”

“cd .aws” -> “vi credentials” -> “<change name>”

Step 2: Make a directory by naming terraform and change to terraform directory.

- Create a “terraform_settings_block.tf”

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.60.0"
    }
  }
}
```

- Create a “provider.tf”

```
provider "aws" {
  region = "eu-north-1"
  profile = "venky"
}
```

- Create a text file in terraform directory “venky.txt”
- Create a “resource.tf” for creating s3 bucket and uploading file into it

```
resource "aws_s3_bucket" "venky-31" {
  bucket = "asus-venky-5"

  tags = {
```

```

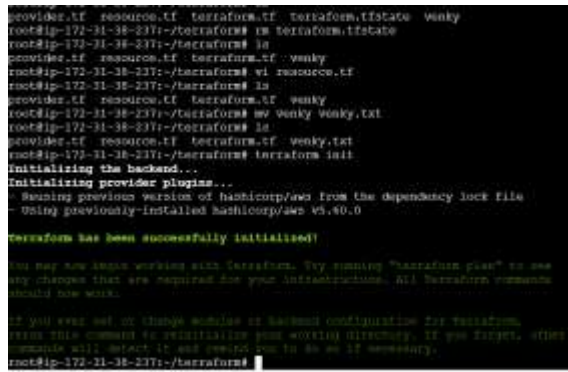
    Name      = "venky-31"
    Environment = "Dev"
  }
}

# Uploading a file into s3_bucket
resource "aws_s3_object" "object" {
  bucket = aws_s3_bucket.venky-31.id
  key     = "venky.txt"
  source  = "/root/terraform/venky.txt"
}

```

Step 3: Now we will use four commands to launch our resource.

- “terraform init” -> Initialize the configuration files



```

provider.tf resource.tf terraform.tf terraform.tfstate wenky
root@ip-172-31-38-237:~/terraform# terraform init
root@ip-172-31-38-237:~/terraform# ls
provider.tf resource.tf terraform.tf wenky
root@ip-172-31-38-237:~/terraform# vi resource.tf
root@ip-172-31-38-237:~/terraform# ls
provider.tf resource.tf terraform.tf wenky wenky.txt
root@ip-172-31-38-237:~/terraform# mv wenky wenky.txt
root@ip-172-31-38-237:~/terraform# ls
provider.tf resource.tf terraform.tf wenky.txt
root@ip-172-31-38-237:~/terraform# terraform init
Initializing the backend...
Initializing provider plugins...
- Searching previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.40.0

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,
press this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-38-237:~/terraform#

```

- “terraform validate” -> Validates our configuration files



```

root@ip-172-31-38-237:~/terraform# terraform validate
Success! The configuration is valid.
root@ip-172-31-38-237:~/terraform#

```

- “terraform apply –auto-approve” -> Apply the actions to achieve the desired state and it will also plan automatically how many resources to be add.

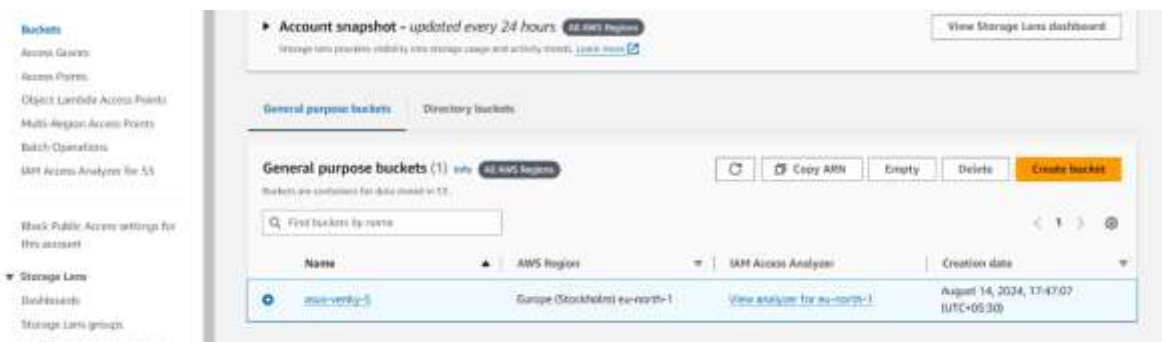
```

- bucket_key_enabled = (known after apply)
- checksum_crc32 = (known after apply)
- checksum_crc32c = (known after apply)
- checksum_sha1 = (known after apply)
- checksum_sha256 = (known after apply)
- content_type = (known after apply)
- etag = (known after apply)
- force_destroy = false
- id = (known after apply)
- key = "venky.txt"
- kms_key_id = (known after apply)
- server_side_encryption = (known after apply)
- source = "/root/.terraform/venky.txt"
- storage_class = (known after apply)
- tags_all = (known after apply)
- version_id = (known after apply)
}

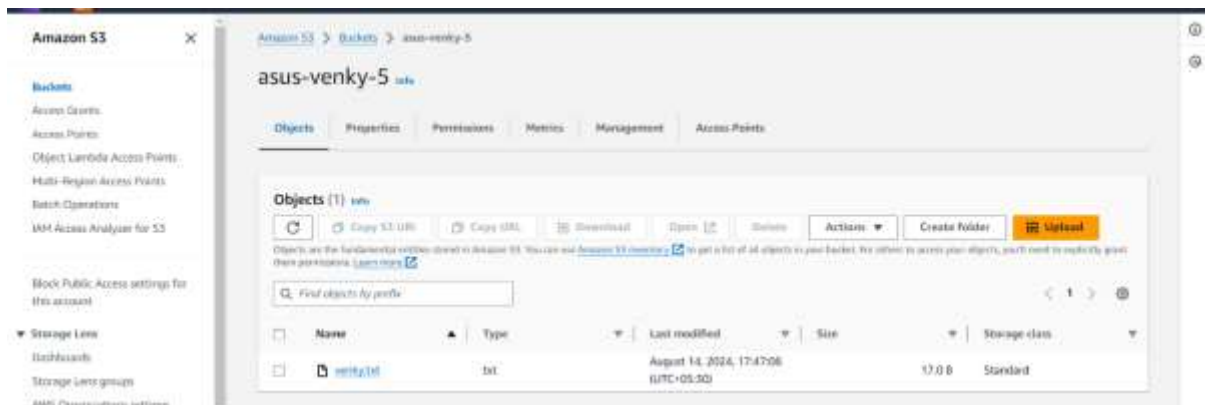
Plan: 2 to add, 0 to change, 0 to destroy.
aws_s3_bucket.venky-31: Creating...
aws_s3_bucket.venky-31: Creation complete after 2s [id=asus-venky-5]
aws_s3_object.object: Creating...
aws_s3_object.object: Creation complete after 0s [id=venky.txt]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@iz-172-31-38-237:~/terraform

```

Step 4: Now check in the console whether the s3 bucket is created or not and also check for the file, whether it is uploaded or not.



We can see that the bucket is created with the name of “asus-venky-5”



We can see that the file has also been uploaded.

Conclusion: We concluded that by using terraform we can create resources.