

**PRACTICAL NO: - 06****A] Create Docker Image using Terraform**

## □ Checking the Docker Functionality

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TEJAS>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run      Create and run a new container from an image
exec     Execute a command in a running container
ps       List containers
build    Build an image from a Dockerfile
pull     Download an image from a registry
push     Upload an image to a registry
images   List images
login    Log in to a registry
logout   Log out from a registry
search   Search Docker Hub for images
version  Show the Docker version information
info     Display system-wide information

Management Commands:
builder  Manage builds
buildx*  Docker Buildx
compose* Docker Compose
container Manage containers
context  Manage contexts
debug*   Get a shell into any image or container
desktop* Docker Desktop commands (Alpha)
dev*     Docker Dev Environments
extension* Manages Docker extensions
```

```
C:\Users\TEJAS>docker --version
Docker version 27.1.1, build 6312585

C:\Users\TEJAS>|
```

```
Windows PowerShell
PS C:\Users\TEJAS> docker image ls
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
PS C:\Users\TEJAS> |
```

## □ Creation of docker.tf file

Create a folder named 'Terraform Scripts' in which we save our different types of scripts. Then, create a new folder named 'Docker' in the 'TerraformScripts' folder.

Then create a new docker.tf file to create a Ubuntu Linux container.

```

docker.tf > ...
1 terraform {
2   required_providers {
3     docker = {
4       source = "kreuzwerker/docker"
5       version = "2.21.0"
6     }
7   }
8 }
9
10 provider "docker" {
11   host = "npipe:////./pipe/docker_engine"
12 }
13
14 resource "docker_image" "ubuntu" {
15   name = "ubuntu:latest"
16 }
17
18 resource "docker_container" "practical6" {
19   image = docker_image.ubuntu.image_id
20   name = "practical6"
21 }
22

```

- Execute Terraform Init command to initialize the resources

```

Windows PowerShell
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
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.
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker>

```

- Execute Terraform plan to see the available resources

```

Windows PowerShell
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> 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:

# docker_container.practical6 will be created
+ resource "docker_container" "practical6" {
  + attach      = false
  + bridge      = (known after apply)
  + command     = (known after apply)
  + container_logs = (known after apply)
  + entrypoint  = (known after apply)
  + env         = (known after apply)
  + exit_code   = (known after apply)
  + gateway     = (known after apply)
  + hostname    = (known after apply)
  + id          = (known after apply)
  + image       = (known after apply)
  + init        = (known after apply)
  + ip_address  = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode    = (known after apply)
  + log_driver  = (known after apply)
  + logs        = false
  + must_run    = true
  + name        = "practical6"
  + network_data = (known after apply)
  + read_only   = false
  + remove_volumes = true
  + restart     = "no"
  + rm          = false

```

```

Windows PowerShell
+ remove_volumes = true
+ restart        = "no"
+ rm             = false
+ runtime        = (known after apply)
+ security_opts  = (known after apply)
+ shm_size       = (known after apply)
+ start          = true
+ stdin_open     = false
+ stop_signal    = (known after apply)
+ stop_timeout   = (known after apply)
+ tty            = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

# docker_image.ubuntu will be created
+ resource "docker_image" "ubuntu" {
  + id          = (known after apply)
  + image_id    = (known after apply)
  + latest      = (known after apply)
  + name        = "ubuntu:latest"
  + output      = (known after apply)
  + repo_digest = (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if
you run "terraform apply" now.
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> |

```

- Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration.

```

Windows PowerShell
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> 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:

# docker_container.practical6 will be created
+ resource "docker_container" "practical6" {
  + attach      = false
  + bridge      = (known after apply)
  + command     = (known after apply)
  + container_logs = (known after apply)
  + entrypoint  = (known after apply)
  + env         = (known after apply)
  + exit_code   = (known after apply)
  + gateway     = (known after apply)
  + hostname    = (known after apply)
  + id          = (known after apply)
  + image       = (known after apply)
  + init        = (known after apply)
  + ip_address  = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode    = (known after apply)
  + log_driver  = (known after apply)
  + logs        = false
  + must_run    = true
  + name        = "practical6"
  + network_data = (known after apply)
  + read_only   = false
  + remove_volumes = true
  + restart     = "no"
  + rm          = false

```

```

Windows PowerShell
+ start      = true
+ stdin_open = false
+ stop_signal = (known after apply)
+ stop_timeout = (known after apply)
+ tty         = false

+ healthcheck (known after apply)
+ labels (known after apply)
}

# docker_image.ubuntu will be created
+ resource "docker_image" "ubuntu" {
  + id          = (known after apply)
  + image_id    = (known after apply)
  + latest      = (known after apply)
  + name        = "ubuntu:latest"
  + output      = (known after apply)
  + repo_digest = (known after apply)
}

Plan: 2 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

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Creation complete after 13s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.practical6: Creating...

```

□ Docker images, Before Executing Apply step:

```
Windows PowerShell
PS C:\Users\TEJAS> docker image ls
REPOSITORY    TAG          IMAGE ID      CREATED      SIZE
PS C:\Users\TEJAS> |
```

- Docker images, After Executing Apply step:

```
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> docker images
REPOSITORY    TAG          IMAGE ID      CREATED      SIZE
ubuntu        latest       edbfe74c41f8  6 weeks ago  78.1MB
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> |
```

- Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
Windows PowerShell
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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:

# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
  - id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
  - image_id    = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - latest      = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - name        = "ubuntu:latest" -> null
  - repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> 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

docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:lat
est]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> |
```

- Docker images After Executing Destroy step

```
Destroy complete! Resources: 1 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> docker images
REPOSITORY    TAG          IMAGE ID      CREATED      SIZE
PS C:\Users\TEJAS\OneDrive\Desktop\terraform_script\docker> |
```

- B] Creating S3 Bucket using terraform.

- Write a Terraform Script in Atom for creating S3 Bucket on Amazon AWS

```
aws-s3 > main.tf > ...
1 terraform {
2   required_providers {
3     aws={
4       source="hashicorp/aws"
5       version="5.64.0"
6     }
7     random = {
8       source = "hashicorp/random"
9       version = "3.6.2"
10    }
11  }
12 }
13
14 resource "random_id" "rand_id"{
15   byte_length = 8
16 }
17
18 resource "aws_s3_bucket" "demo-bucket" {
19   bucket = "demo-bucket-${random_id.rand_id.hex}"
20 }
21
22 resource "aws_s3_object" "bucket-data" {
23   bucket = aws_s3_bucket.demo-bucket.bucket
24   source = "./myfile.txt"
25   key="newfile.txt"
26 }
```

- Create a new provider.tf file

```
aws-s3 > provider.tf > ...
1 provider "aws" {
2   access_key="ASIAXDVAUJCTAHPBYALZ"
3   secret_key="kdp24c5nflcbbuWQe49jBthzJc4RefZBns6t6J8y"
4   token="IQoJb3JpZ21uX2VjE3f////////wEaCXVzLXd1c3QtMiJIMEYCIQCSJ3i5FhAAZU/xRQHF5L2DoNBiAuOG0PAemX7fCnC2bwiHAPIRX10U0Af6Bz3Qs7oX8Cnpr62XK7LhDo3/
u4vx66jUkSACCMd////////wEQABOMNDg4ODg5NTU3MTU4IGzPDJggqUuyZ9njLwoqLAKlV6HhXOV2Vhev/w51dDaiTqa9/
lTgr1iVofAfozhGRcK5yuoAho5IXYG0thTerBx8N1n0i8fQfbqnI/dB/pEMsk02xh/5PYBrduLAHVSWuvV86LPqZzJ0wTS4iTn+l6fi6PyhM4kZtbPTPixaJZGJuvWYzrPbe6t6lHgsw/
Rsv51B91HEjyVYw7LDQsXvz+YX4ie5Dt09+XbXh+WMWAp2+GucNkQOKRg+IbkdRPF5njQHJ818h3xQYD72tIGmMwXZgBHrobK15zVkkfZMseofXimgj1h1ZQ8zGof9K0U1xd1vnrpFzH
+FhyVPV/cCEMIjE8M1euRcd1RnaD5E7PdAg3k7F8FnMAFcU4qck9uhQCJ0o0wqZyRtwY6nAGhyr0X3yVQrC1EWK3j6HbY5MOJ3C41Thme7fNTs5HykFDVSFUGZKRzo3spEsepbN/
YnNlOAadFZCEbY4ayUfSSEhMnKhyZd185xewRnp8kZTKSdxWzqyuTjoi/dL1vfq0B5jPSdJ4J0v453YIsyZmXRnWdh8ANyg2ZAyw3kC5CdipQLDEti37zP20nciYcA10y+1F+e3Cw/
vcgFR08="
5   region = "us-east-1"
6 }
7 \end{code}
```

- Execute Terraform Init command to initialize the resources

```

PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Terraform practical>
cd aws-s3
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Terraform practical\aws-s3> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Using previously-installed hashicorp/aws v5.64.0
- Using previously-installed hashicorp/random v3.6.2

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.
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Terraform practical\aws-s3>

```

## □ Execute Terraform plan to see the available resources

```

PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Terraform practical\aws-s3> terraform plan
random_id.rand_id: Refreshing state... [id=mM9L9tLf-c]
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Refreshing state... [id=newfile.txt]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

# aws_s3_bucket.demo-bucket has been deleted
- resource "aws_s3_bucket" "demo-bucket" {
  - bucket              = "demo-bucket-98c27d2fdb4b7fe7" -> null
    id                  = "demo-bucket-98c27d2fdb4b7fe7"
  # (12 unchanged attributes hidden)

  # (3 unchanged blocks hidden)
}

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to
undo or respond to these changes.

```

---

```

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.demo-bucket will be created
+ resource "aws_s3_bucket" "demo-bucket" {
  + acceleration_status = (known after apply)
  + acl                  = (known after apply)
  + arn                  = (known after apply)
  + bucket                = "demo-bucket-98c27d2fdb4b7fe7"
  + bucket_domain_name   = (known after apply)
  + bucket_prefix         = (known after apply)
  + bucket_regional_domain_name = (known after apply)
  + force_destroy         = false

```

```

+ replication_configuration (known after apply)

+ server_side_encryption_configuration (known after apply)

+ versioning (known after apply)

+ website (known after apply)
}

# aws_s3_object.bucket-data will be created
+ resource "aws_s3_object" "bucket-data" {
+   acl                = (known after apply)
+   arn                = (known after apply)
+   bucket             = "demo-bucket-98c27d2fdb4b7fe7"
+   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                = "newfile.txt"
+   kms_key_id         = (known after apply)
+   server_side_encryption = (known after apply)
+   source              = "./myfile.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.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Terraform practical\aws-s3>

```

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

```

PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Terraform practical\aws-s3> terraform apply
random_id.rand_id: Refreshing state... [id=mM9L9tLf-c]
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Refreshing state... [id=newfile.txt]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

# aws_s3_bucket.demo-bucket has been deleted
- resource "aws_s3_bucket" "demo-bucket" {
-   bucket             = "demo-bucket-98c27d2fdb4b7fe7" -> null
-   id                 = "demo-bucket-98c27d2fdb4b7fe7"
-   # (12 unchanged attributes hidden)
-
-   # (3 unchanged blocks hidden)
- }

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to undo or respond to these changes.

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.demo-bucket will be created
+ resource "aws_s3_bucket" "demo-bucket" {
+   acceleration_status = (known after apply)
+   acl                 = (known after apply)
+   arn                 = (known after apply)
+   bucket              = "demo-bucket-98c27d2fdb4b7fe7"
+   bucket_domain_name  = (known after apply)
+   bucket_prefix       = (known after apply)
+   bucket_regional_domain_name = (known after apply)
+   force_destroy       = false

```



```
# aws_s3_object.bucket-data will be created
+ resource "aws_s3_object" "bucket-data" {
+   acl                = (known after apply)
+   arn                = (known after apply)
+   bucket             = "demo-bucket-98c27d2fdb4b7fe7"
+   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                = "newfile.txt"
+   kms_key_id         = (known after apply)
+   server_side_encryption = (known after apply)
+   source              = "./myfile.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.

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.demo-bucket: Creating...
aws_s3_bucket.demo-bucket: Creation complete after 7s [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Creating...
aws_s3_object.bucket-data: Creation complete after 2s [id=newfile.txt]

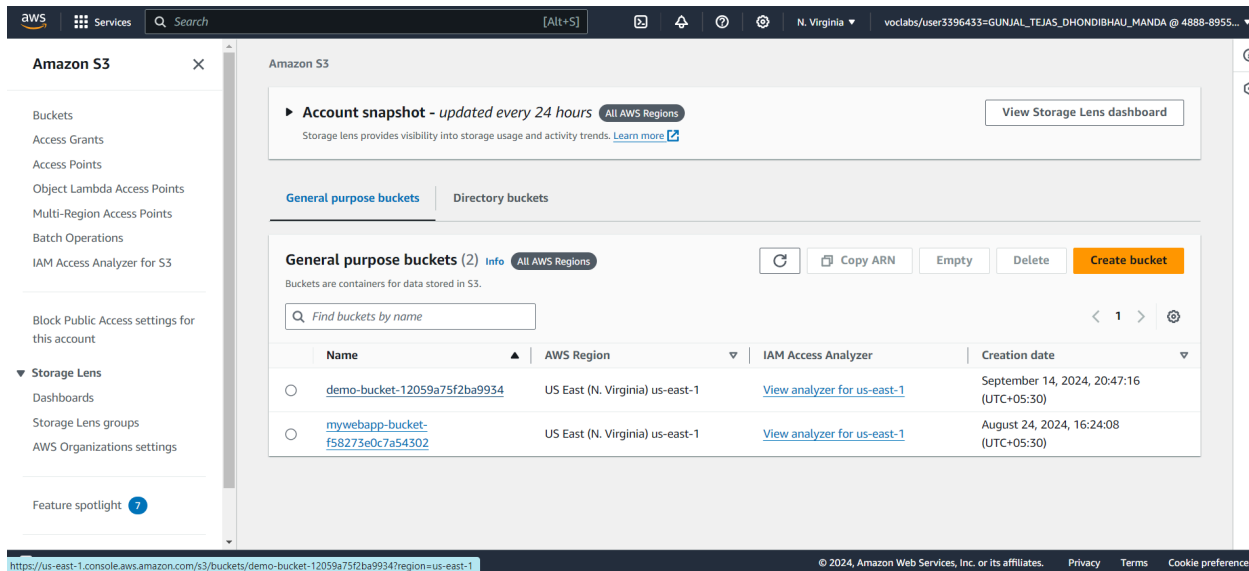
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3>
```

□ AWS S3bucket dashboard, Before Executing Apply command:

The screenshot displays the AWS S3 console interface. On the left, a navigation menu lists various S3 features. The main content area shows the 'General purpose buckets' tab with a table of existing buckets. One bucket is listed with the name 'mywebapp-bucket-f58273e0c7a54302' in the 'US East (N. Virginia) us-east-1' region. The console also features an 'Account snapshot' section at the top and a 'Create bucket' button.

Name	AWS Region	IAM Access Analyzer	Creation date
mywebapp-bucket-f58273e0c7a54302	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 24, 2024, 16:24:08 (UTC+05:30)

□ AWS S3 Bucket dashboard, After Executing Apply step:



□ Execute Terraform destroy to delete the configuration, which will automatically delete the bucket

```
PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3> terraform destroy
random_id.rand_id: Refreshing state... [id=mJ9L9tLf-c]
aws_s3_bucket.demo-bucket: Refreshing state... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_object.bucket-data: Refreshing state... [id=newfile.txt]

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.demo-bucket will be destroyed
- resource "aws_s3_bucket" "demo-bucket" {
  - arn = "arn:aws:s3:::demo-bucket-98c27d2fdb4b7fe7" -> null
  - bucket = "demo-bucket-98c27d2fdb4b7fe7" -> null
  - bucket_domain_name = "demo-bucket-98c27d2fdb4b7fe7.s3.amazonaws.com" -> null
  - bucket_regional_domain_name = "demo-bucket-98c27d2fdb4b7fe7.s3.us-east-1.amazonaws.com" -> null
  - force_destroy = false -> null
  - hosted_zone_id = "Z3AQBSTGFYJSTF" -> null
  - id = "demo-bucket-98c27d2fdb4b7fe7" -> null
  - object_lock_enabled = false -> null
  - region = "us-east-1" -> null
  - request_payer = "BucketOwner" -> null
  - tags = {} -> null
  - tags_all = {} -> null
  # (3 unchanged attributes hidden)

- grant {
  - id = "89f1916699ac713a2d0a157aa4176d8ce16adda216e3bafb6e7f0811cb559dca" -> 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
```

**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_object.bucket-data: Destroying... [id=newfile.txt]
aws_s3_object.bucket-data: Destruction complete after 1s
aws_s3_bucket.demo-bucket: Destroying... [id=demo-bucket-98c27d2fdb4b7fe7]
aws_s3_bucket.demo-bucket: Destruction complete after 1s
random_id.rand_id: Destroying... [id=mMJ9L9tLf-c]
random_id.rand_id: Destruction complete after 0s
```

**Destroy complete! Resources: 3 destroyed.**

PS C:\Users\TEJAS\OneDrive\Desktop\SEM 5 VESIT\Teraform practical\aws-s3> █

□ AWS S3 Bucket dashboard, After Executing Destroy step:

The screenshot displays the AWS S3 console interface. On the left, a sidebar contains navigation links for various S3 services. The main panel shows the 'General purpose buckets' section, which includes a search bar and a table of existing buckets. The table has the following data:

Name	AWS Region	IAM Access Analyzer	Creation date
mywebapp-bucket-f58273e0c7a54302	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 24, 2024, 16:24:08 (UTC+05:30)