We have to launch the terraform instance with t2.micro with 8gb Ebs volume

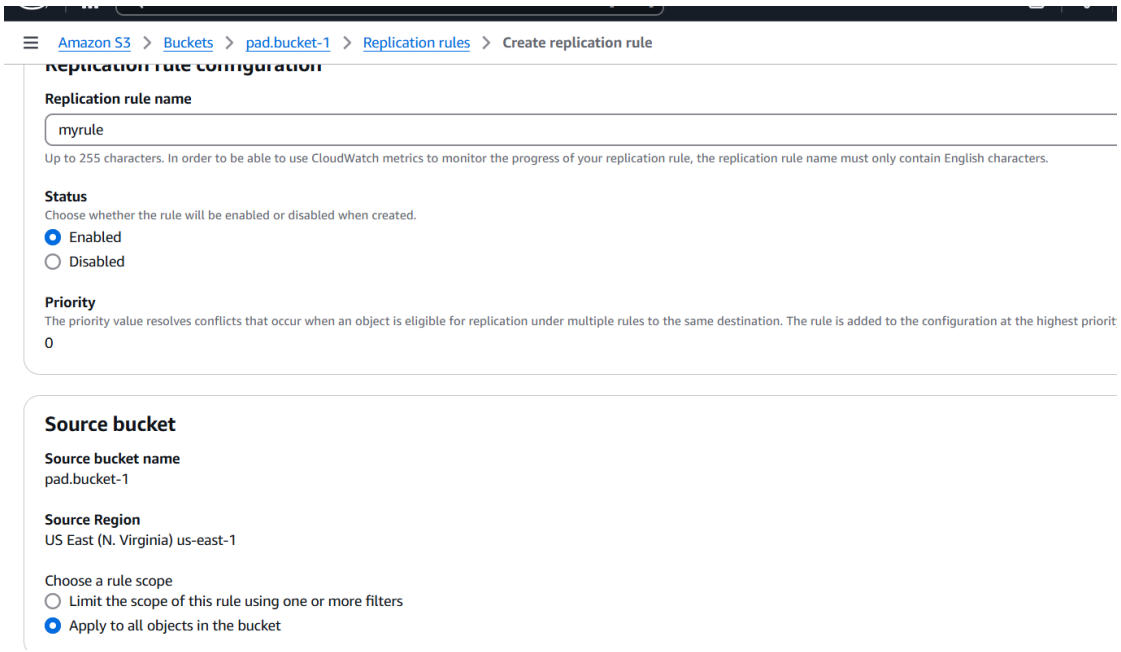
1.Created Two Buckets in S3:

Creating a bucket -> name ->Acl Disabed ->block all access -> versioning enabled -> create – bucket-name1 (pad.bucket-1)

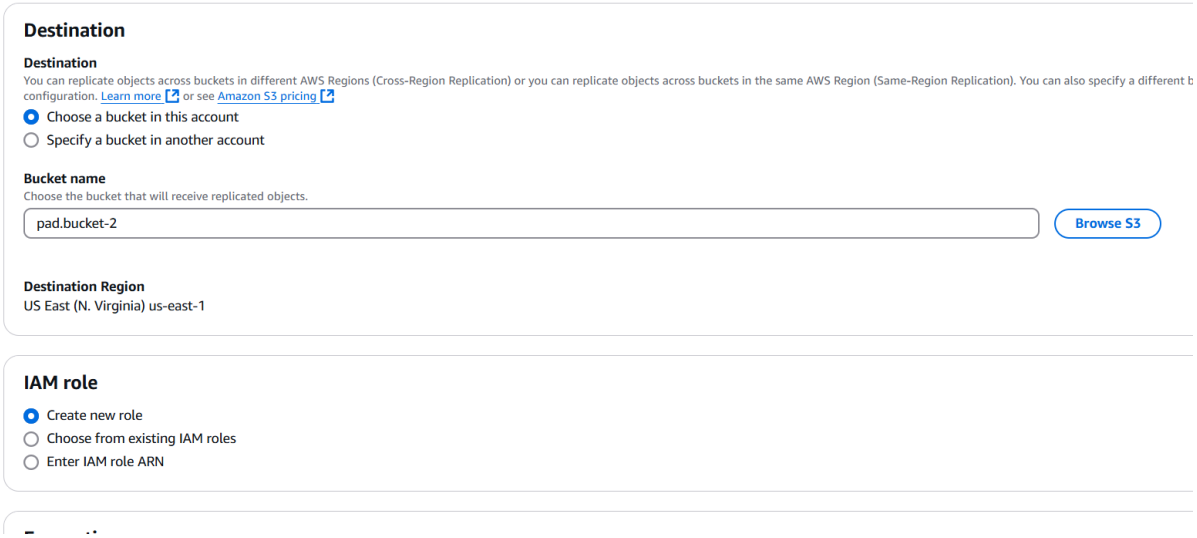
Creating a bucket -> name ->Acl Disabed ->block all access -> versioning enabled -> create – bucket-name2 (pad.bucket-2)

Now if we upload the object in bucket1 it needs to be replicate in bucket 2 so we will use **cross region replication** fot that

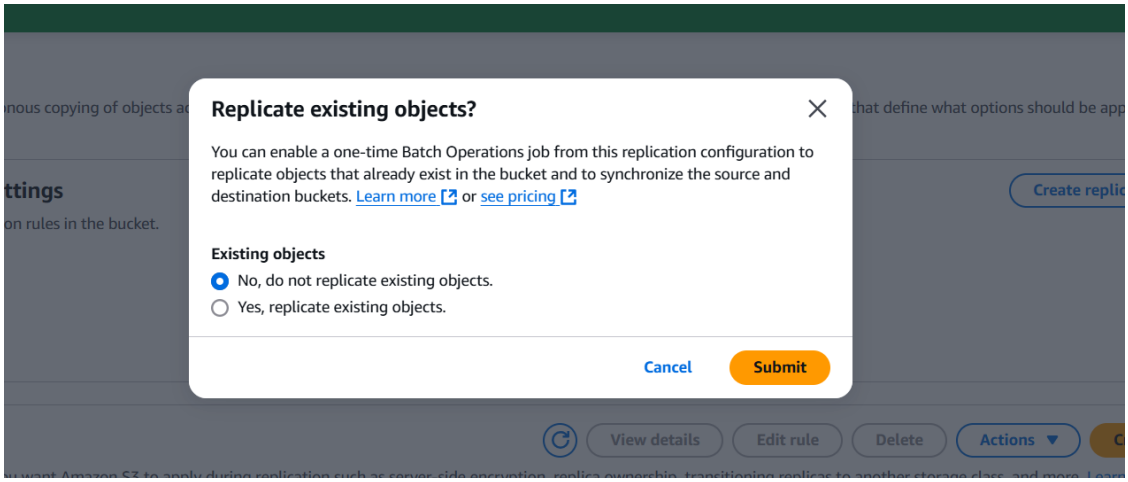
We need to go pad.bucket-1 →click on create replication rule -> enter the replicationrule name(myrule) →status -enabled → choose a role scope -apply all objects in bucket → Destination ->click on browse select bucket2 click on choose path.



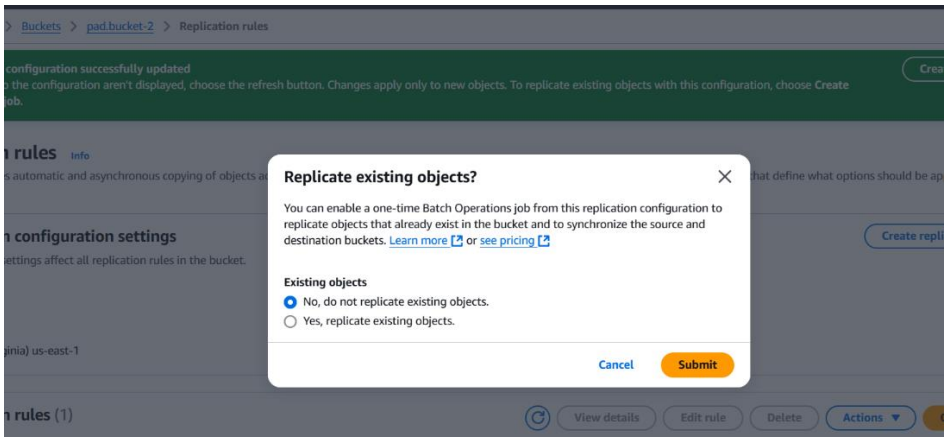
Next creating the **new role** under IAM role: It will allow the permissions to replicate the data in two buckets



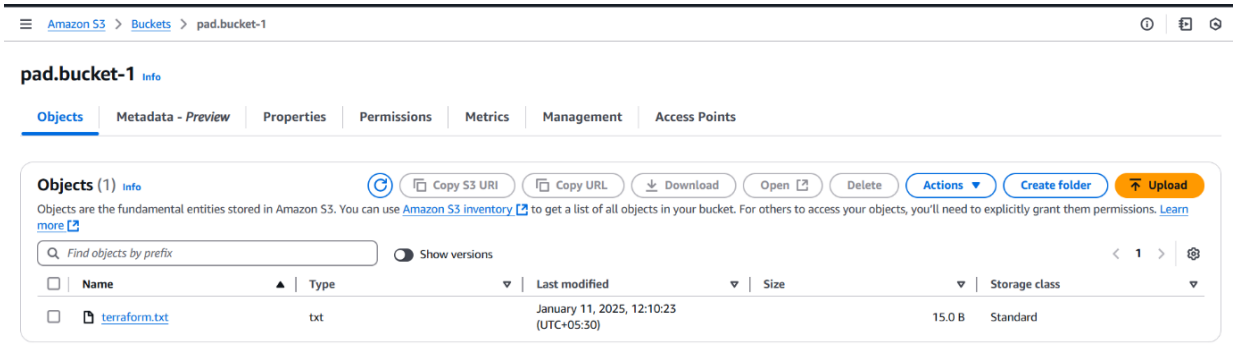
Click on save and submit.



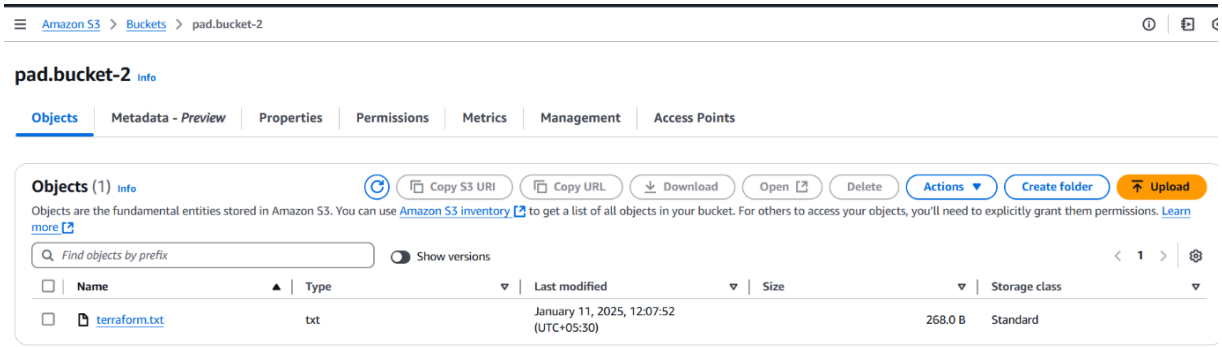
Same as create Bucket 2→ click on create replication rule -> enter the replicationrule name(myrule) →status -enabled → choose a role scope -apply all objects in bucket → Destination ->click on browse select bucket1 click on choose path → select existing IAM role →click on save and submit.



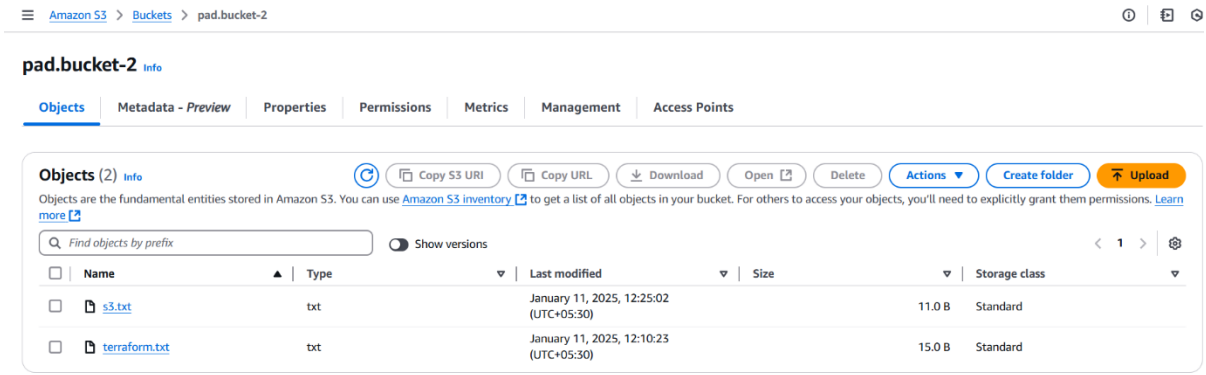
Now uploaded the file in pad.bucket-1



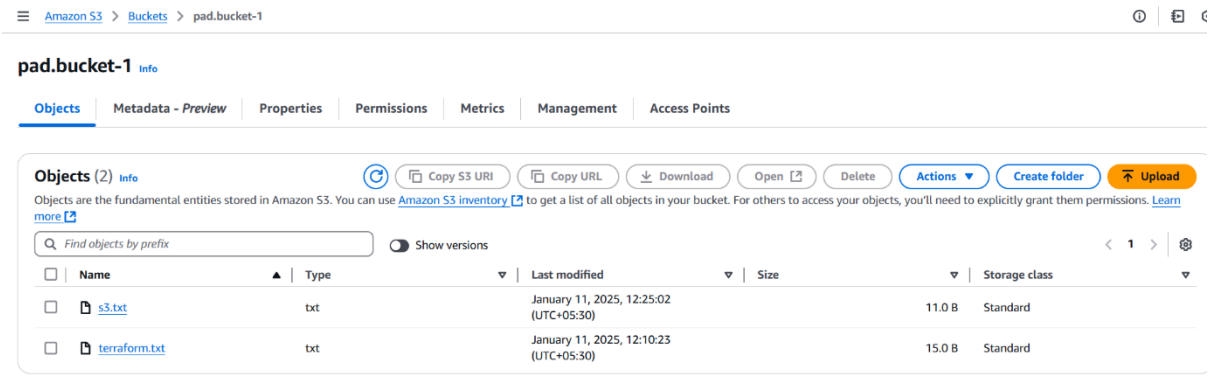
File is replicated in pad.bucket-2



Now again I uploaded the file in pad.bucket-2



Data replicated in pad.bucket-1



Now if we want see the buckets in server will use the command: aws s3 ls

Before that we need to attach the IAM user/role - s3 full permission

Here I am attached the IAM role (admin access role) its worked for me.

```
[root@ip-172-31-94-247 ~]# aws s3 ls
2025-01-11 06:29:51 pad.bucket-1
2025-01-11 06:30:32 pad.bucket-2
2024-11-30 14:20:24 siddhu.flm.k8s
2025-01-07 07:50:02 sridhar.aws.bucket
[root@ip-172-31-94-247 ~]#
```

From server if we want create the new bucket will use below command:

aws s3 mb s3://sid.bucket-3

to check the files or folders in one particular bucket will use below command:

aws s3 ls bucket_name

```
[root@ip-172-31-94-247 ~]# aws s3 mb s3://sid.bucket-3
make_bucket: sid.bucket-3
[root@ip-172-31-94-247 ~]# aws s3 ls
2025-01-11 06:29:51 pad.bucket-1
2025-01-11 06:30:32 pad.bucket-2
2025-01-11 07:01:36 sid.bucket-3
2024-11-30 14:20:24 siddhu.flm.k8s
2025-01-07 07:50:02 sridhar.aws.bucket
[root@ip-172-31-94-247 ~]# aws s3 ls pad.bucket-1
2025-01-11 06:55:02         11 s3.txt
2025-01-11 06:40:23         15 terraform.txt
[root@ip-172-31-94-247 ~]#
```

We cannot delete the bucket directly firstly we need to do empty (permanently delete) and then delete the bucket with bucket_name.

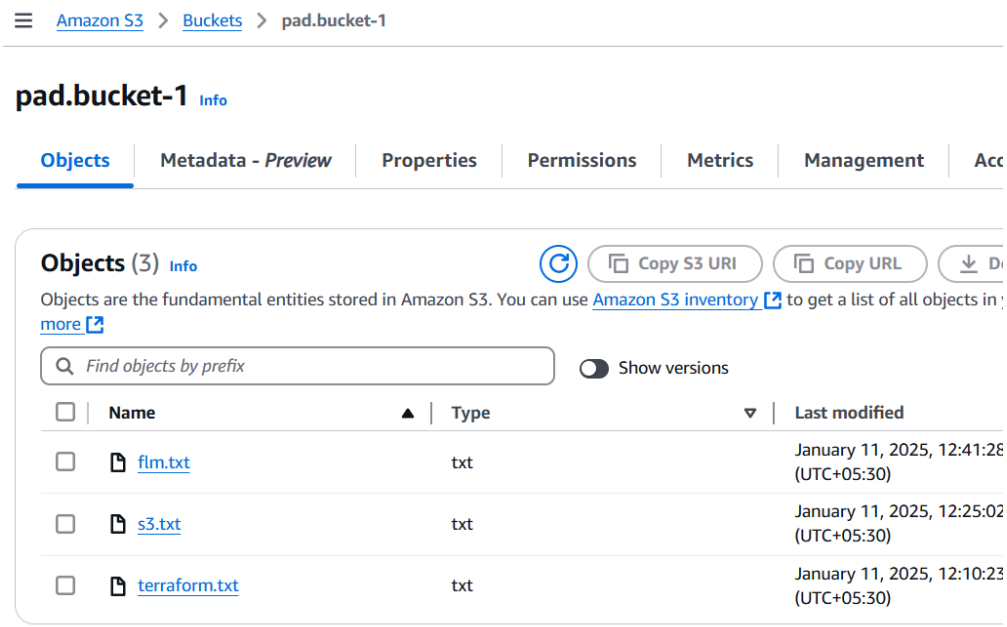
If you want to delete the bucket through CLI will use below command.

aws s3 rm s3://sid.bucket-3 –recursive (for removing files in bucket)

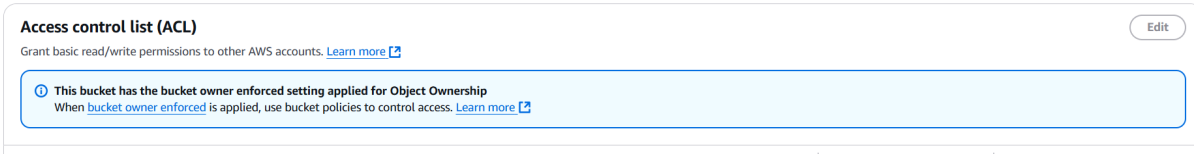
aws s3 rb s3://sid.bucket-3 (to delete the bucket)

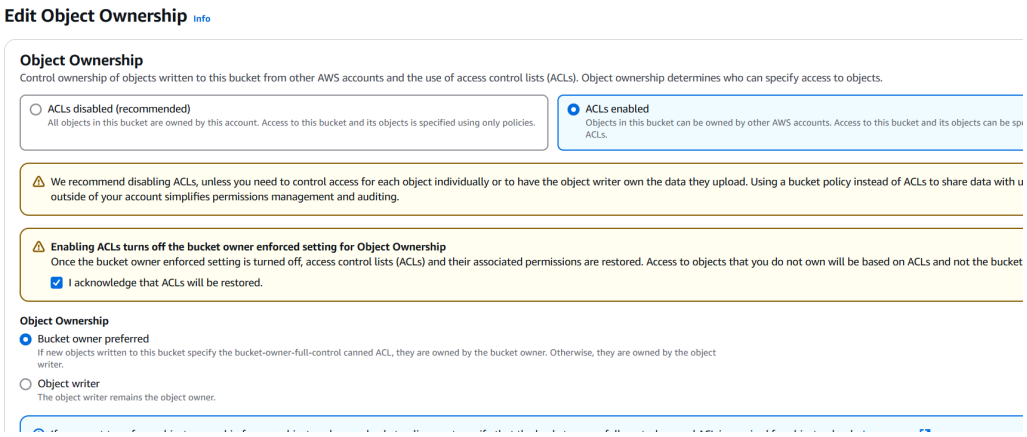
Next I created the file in server(flm.txt) that I want send to bucket nothing to copy so will use below command

aws s3 cp flm.txt s3://pad.bucket-1

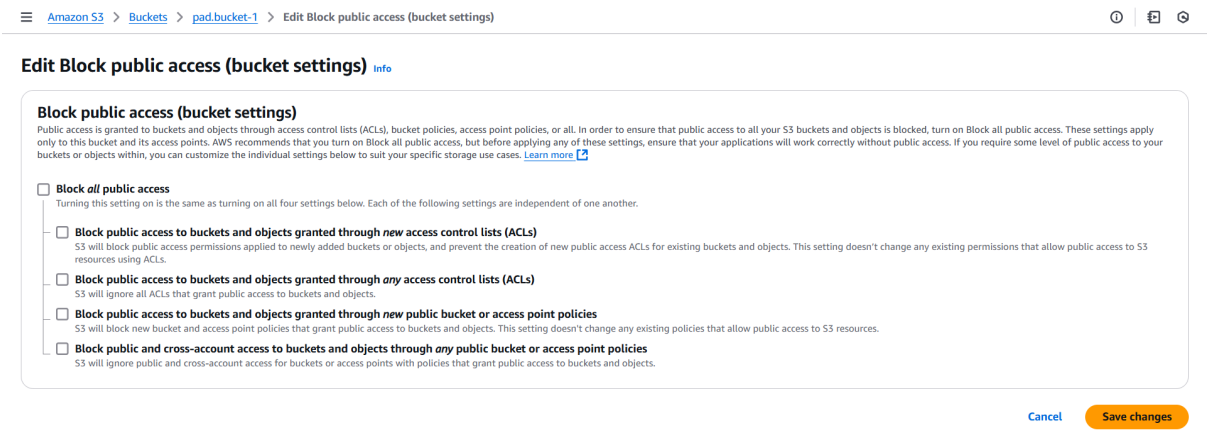


If you want to allow others to see objects we have to go permissions →Access control list(ACL)→ACLs enabled → click on Save changes

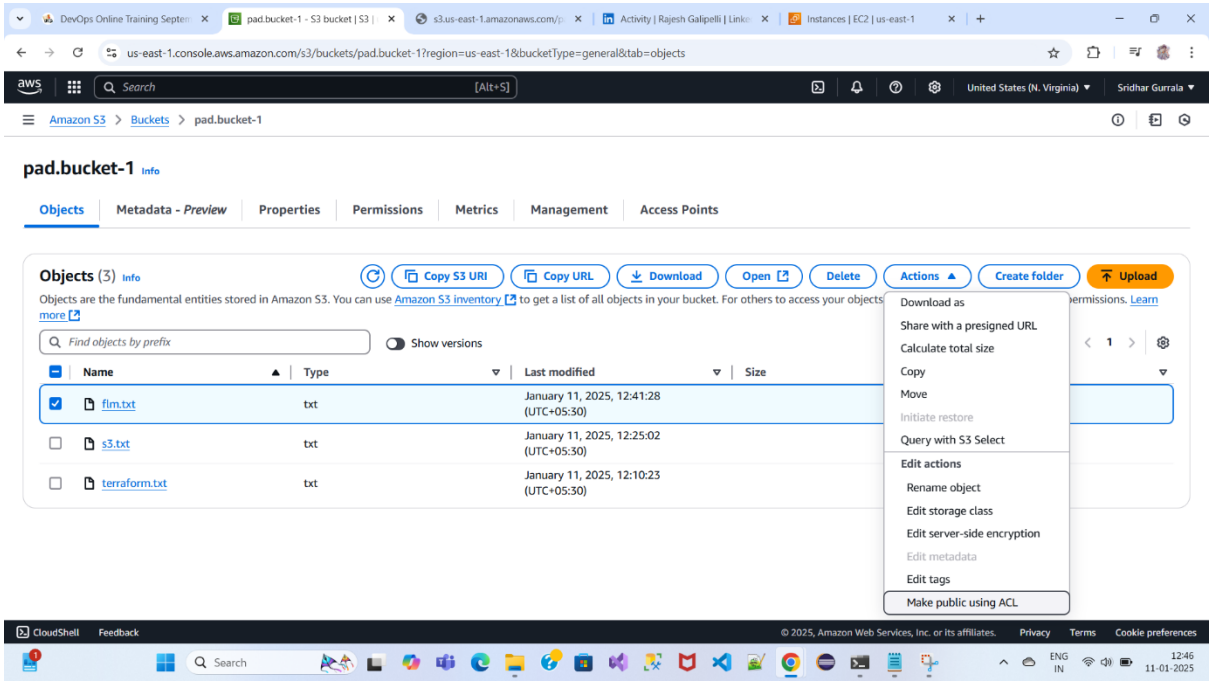




Next under permissions → block access untick and save



After that Actions →make public using ACL so everyone will access the file



2.By using S3 we can host websites:

We need to create bucket → ACL enabled → uncheck block all access → versioning enabled →click on create bucket(pad.website)

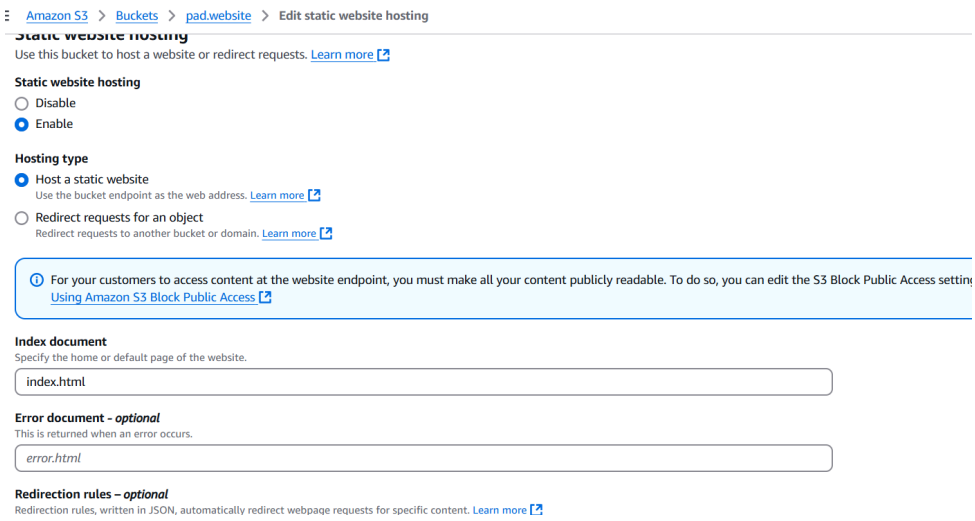
Now we have to upload the html file in bucket so we need to create the html file in server

vim index.html

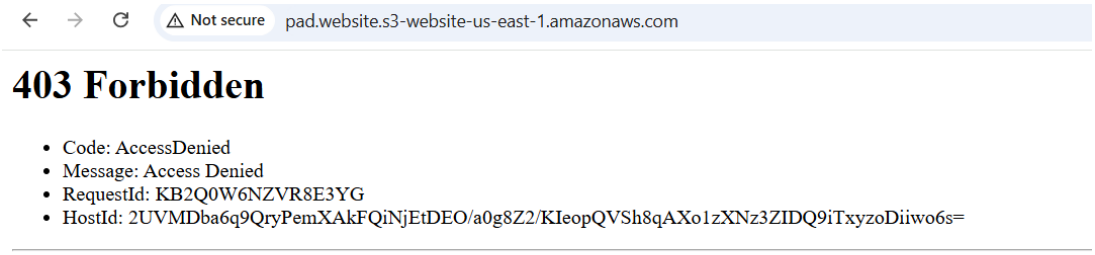
now copy the file to the pad.website

aws s3 cp index.html s3://pad.website

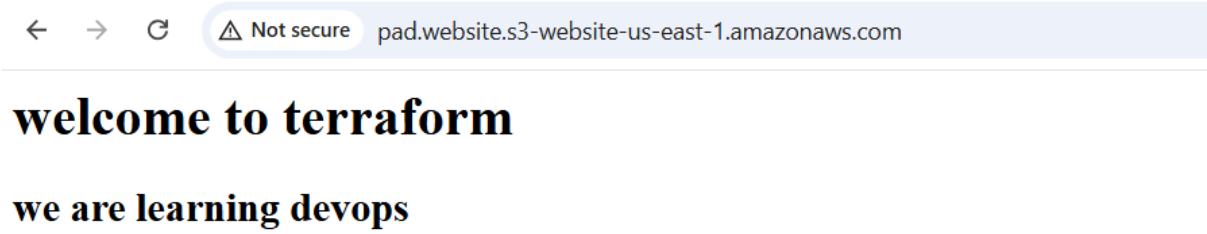
now if we want enable the website →go to index.html → properties →static website hosting → under index document – **index.html** →click on save changes



After that One URL will be generate if we access should be denied because we have to make public access



Go to bucket(pad.website)->index.html -> actions →make public using ACL so everyone will access



3.Next we want store jenkins war files in S3 Bucket:

We need to install jenkins in our server : vim jenkins.sh

Jenkins script:

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

yum install java-17-amazon-corretto -y

yum install jenkins -y

systemctl start jenkins

systemctl status jenkins

Execute jenkins→ sh jenkins.sh

cat /var/lib/jenkins/secrets/initialAdminPassword will get the password

yum install git -y

yum install java-1.8.0-openjdk maven -y (for maven we have another option to install in jenkins under TOOLs → under add maven give the name mymaven and click on save

create the job with name of Flm and select freelifestyle and click on create

Manage jenkins→ plugins → S3 publisher(artifact uploaders) install

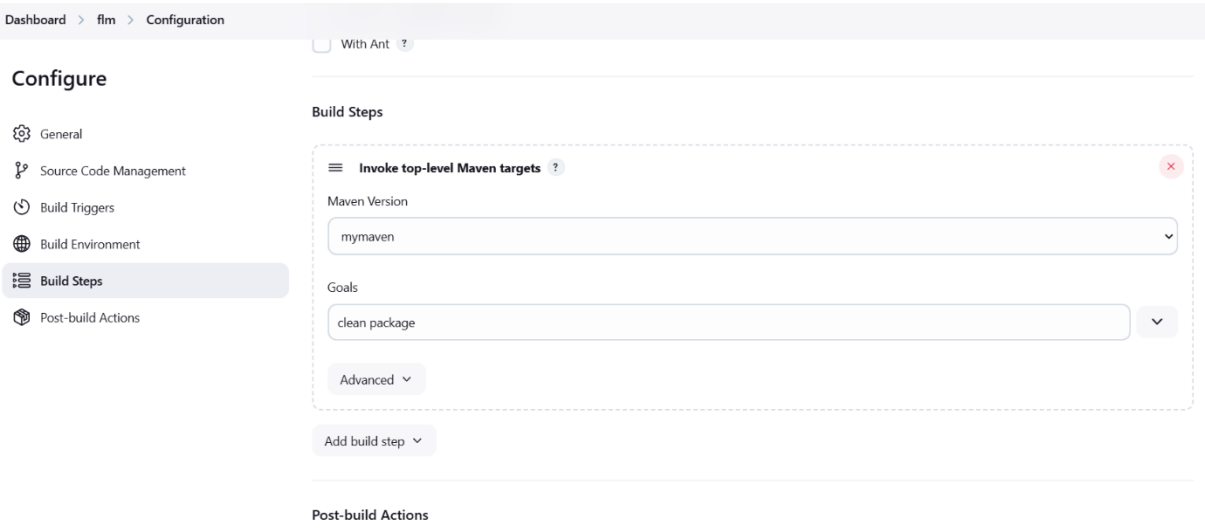
Integrate S3 to jenkins Manage jenkins -> system → S3 profiles → click on Add → profile name (devops) → attached IAM user→ give the access and secret key → click on test connection and save

[To Create the IAM user → create user → user name (any name) → attach policies directly → permission policies- Amaons3fullaccess → click on next →click on create user

Now go to user under security credentials →click on create access key →choose other option → click on next -> create access key]

Now we have to go the flm job → source code management → under git we have given one repository https://github.com/Sridhar2628/one.git and save.

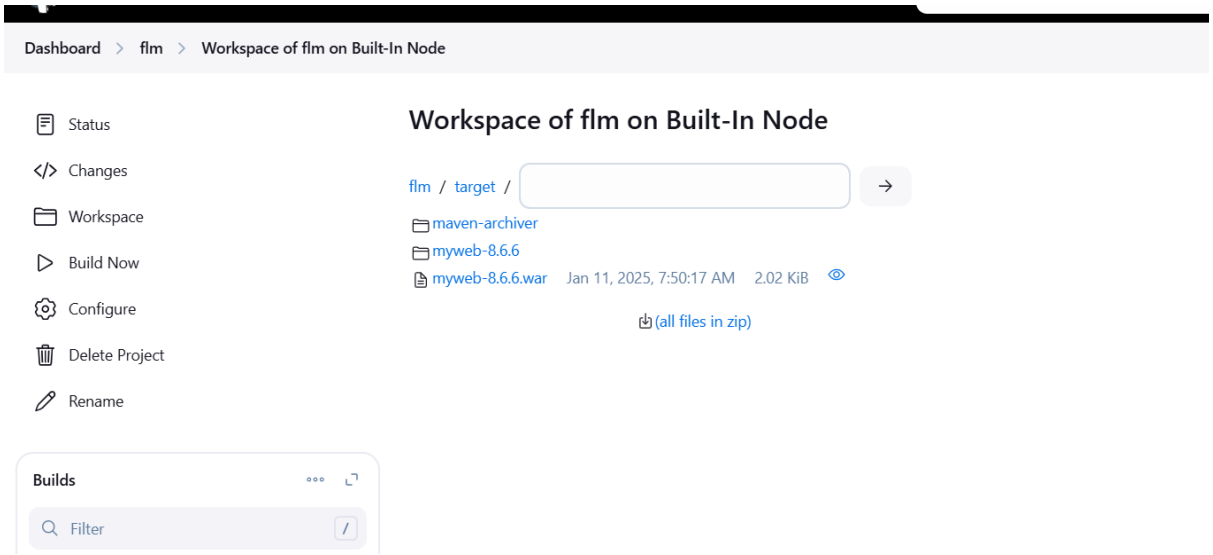
Next under build steps In Maven version → choose mymaven -> under goals – clean package and click save.



Now build the flm job

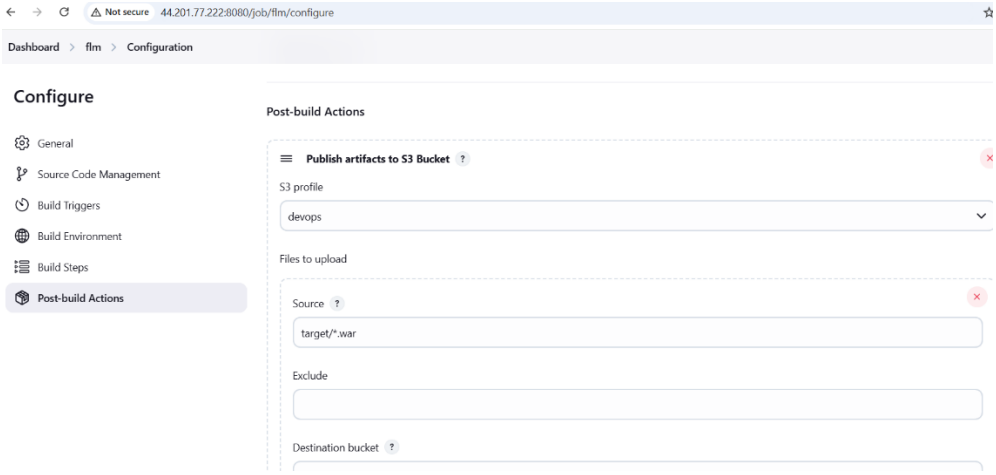
```
[root@ip-172-31-94-247 ~]# cd /var/lib/jenkins/workspace/flm/
[root@ip-172-31-94-247 flm]# ll
total 4
-rw-r--r-- 1 jenkins jenkins 1282 Jan 11 07:50 pom.xml
drwxr-xr-x 3 jenkins jenkins   18 Jan 11 07:50 src
drwxr-xr-x 4 jenkins jenkins   70 Jan 11 07:50 target
[root@ip-172-31-94-247 flm]#
[root@ip-172-31-94-247 flm]# ll target/
total 4
drwxr-xr-x 2 jenkins jenkins   28 Jan 11 07:50 maven-archiver
drwxr-xr-x 4 jenkins jenkins   54 Jan 11 07:50 myweb-8.6.6
-rw-r--r-- 1 jenkins jenkins 2064 Jan 11 07:50 myweb-8.6.6.war
[root@ip-172-31-94-247 flm]#
```

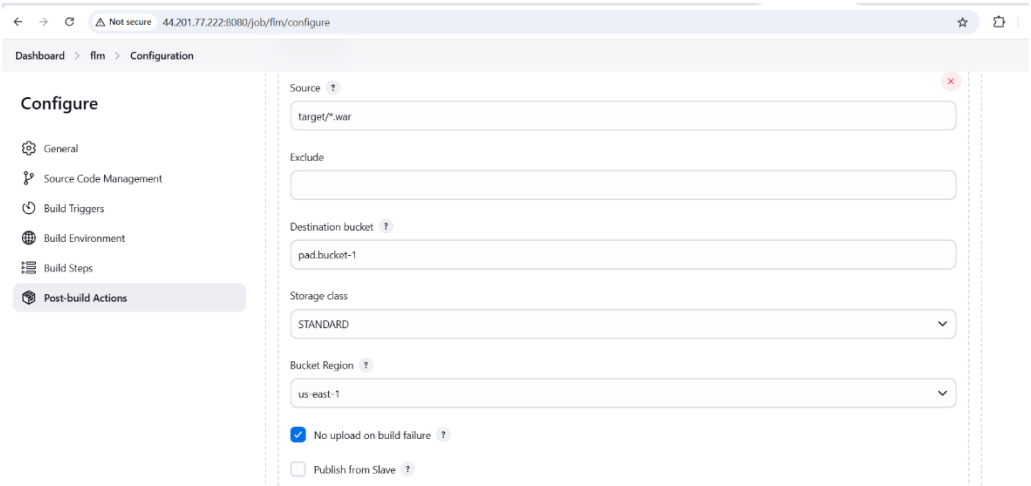
Now below war file we need to store in S3



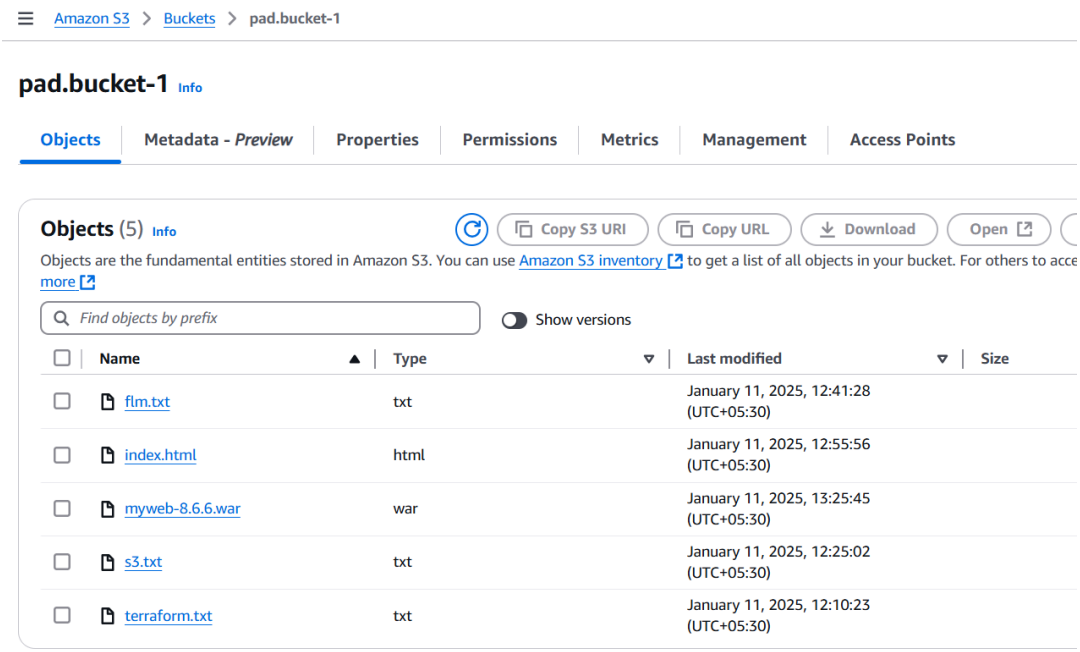
If we want store the above war file in S3 we have to make configurations in flm job

Under configuartions → post build actions → s3 profile – devops(initial we have given same name) →source- target/*.war → destination bucket - pad.bucket-1 →Bucket Region – us-east-1 → select no upload on build failure.

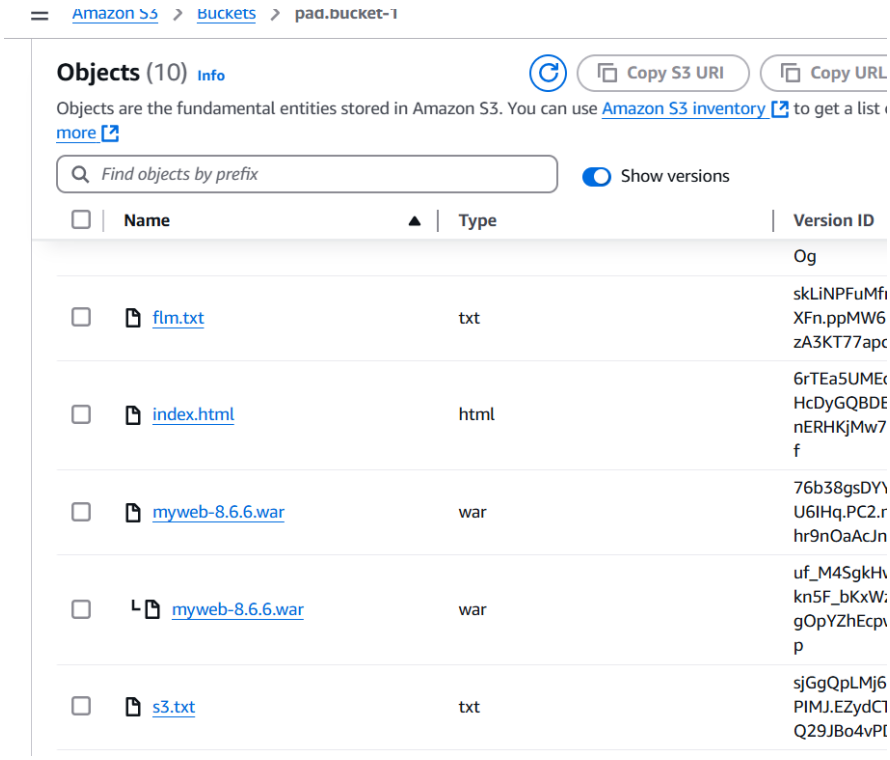




Now build the flm job the war file stored in pad.bucket-1



Now if we build the job again war multiple file will be store.



4.Store the terraform state files in s3 bucket:

Before that we need to install terraform

sudo yum install -y yum-utils

sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

sudo yum -y install terraform

To Check the version: **terraform version**

Creating the vim s3.tf

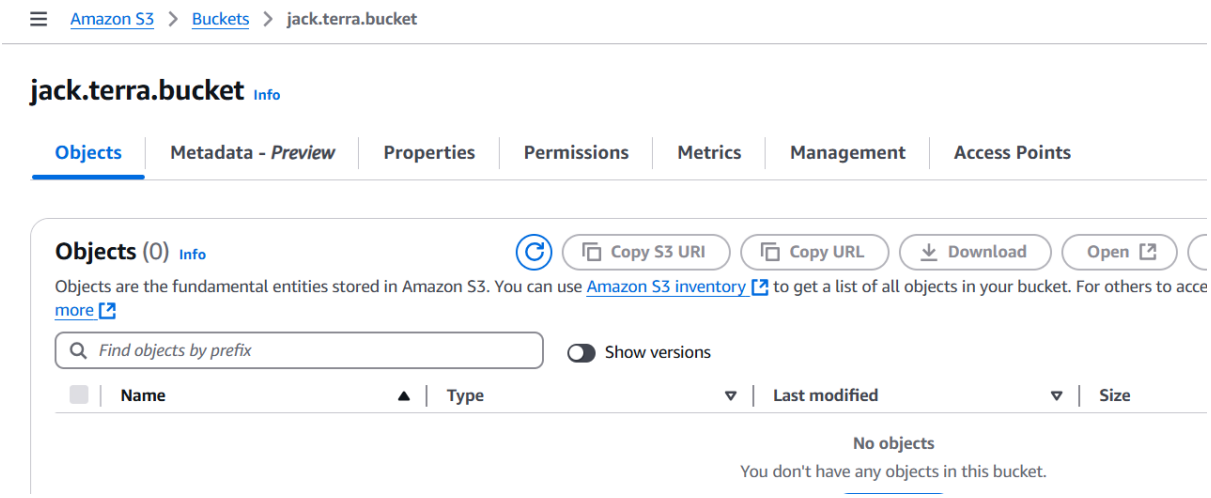
```
provider "aws" {
region = "us-east-1"
}

resource "aws_s3_bucket" "one" {
bucket = "jack.terra.bucket"
}

resource "aws_s3_bucket" "one" {
bucket = "jack.terra.bucket"
}

resource "aws_s3_bucket_versioning" "two" {
bucket = aws_s3_bucket.one.id
versioning_configuration {
status = "Enabled"
}
}
```

terraform init

terraform plan

terraform apply –auto-approve

It will create the bucket and also we can see versioning enabled(show version option)



Create the file vim backend.tf

```
terraform {
backend "s3" {
bucket = "jack.terra.bucket"
key = "imp-folder/terraform.tfsate"
region = "us-east-1"
}
}
```

Note: key = "imp-folder/terraform.tfsate" this folder will be create and under the folder terraform.tfstate will be store.

terraform init

it will ask **Do you want to copy existing state to the new backend? Yes**

```
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes


Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.83.1

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.
[root@ip-172-31-94-247 ~]#
```

Now if we want to delete the bucket first we need to remove files and folders in bucket and then delete then bucket.

```
aws s3 rm s3:// jack.terra.bucket –recursive
```

```
terraform destroy –auto-approve
```