

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

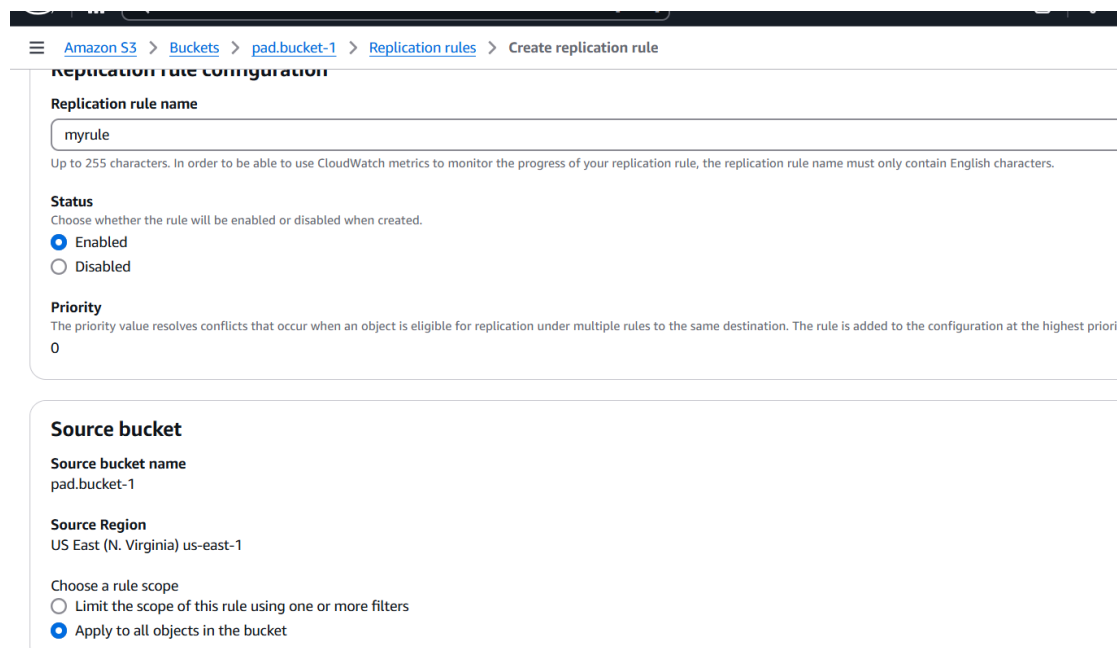
1.Created Two Buckets in S3:

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.



Amazon S3 > Buckets > pad.bucket-1 > Replication rules > Create replication rule

Replication rule configuration

Replication rule name

myrule

Up to 255 characters. In order to be able to use CloudWatch metrics to monitor the progress of your replication rule, the replication rule name must only contain English characters.

Status

Choose whether the rule will be enabled or disabled when created.

☒ Enabled

☐ Disabled

Priority

The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority.

0

Source bucket

Source bucket name

pad.bucket-1

Source Region

US East (N. Virginia) us-east-1

Choose a rule scope

☐ Limit the scope of this rule using one or more filters

☒ Apply to all objects in the bucket

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

Destination

Destination
You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket configuration. [Learn more](#) or see [Amazon S3 pricing](#)

☒ Choose a bucket in this account
☐ Specify a bucket in another account

Bucket name
Choose the bucket that will receive replicated objects.

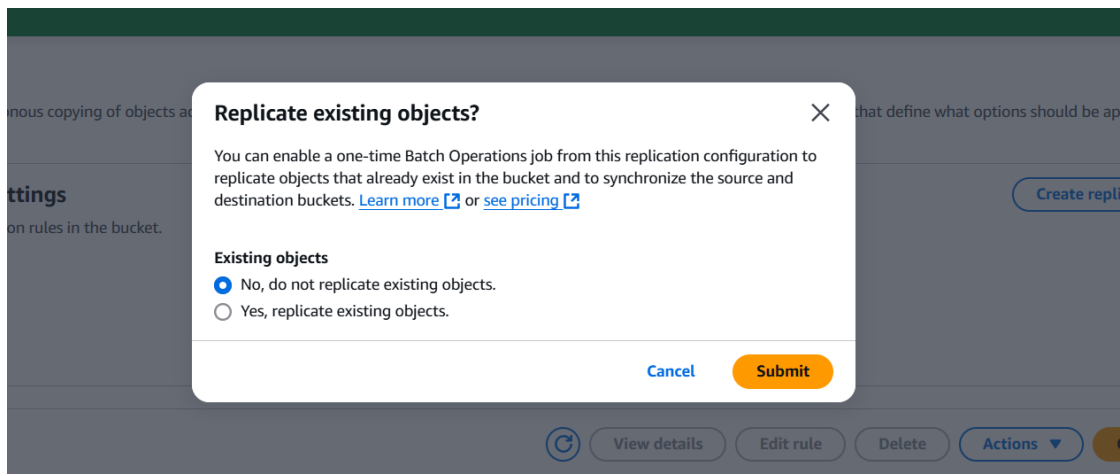
pad.bucket-2 [Browse S3](#)

Destination Region
US East (N. Virginia) us-east-1

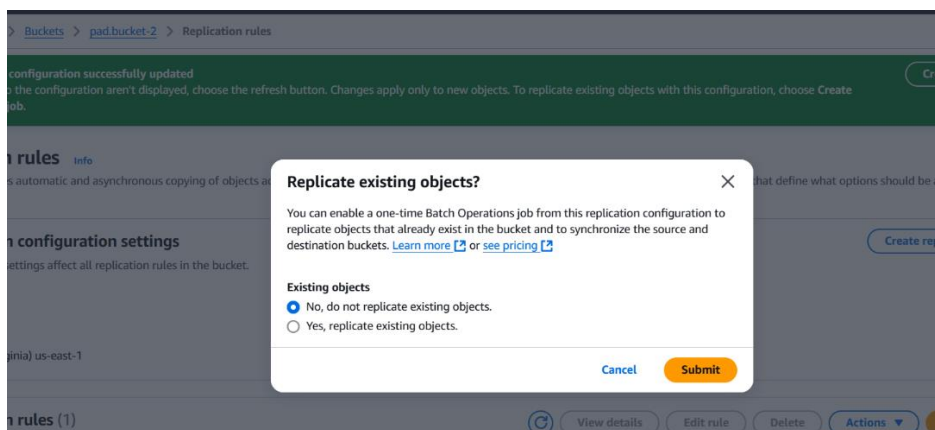
IAM role

☒ Create new role
☐ Choose from existing IAM roles
☐ Enter IAM role ARN

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

Amazon S3 > Buckets > pad.bucket-1

pad.bucket-1

ObjectsMetadata - PreviewPropertiesPermissionsMetricsManagementAccess Points

Objects (1)

Copy S3 URICopy URLDownloadOpenDeleteActionsCreate folderUpload

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](#)

Find objects by prefixShow versions

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	terraform.txt	txt	January 11, 2025, 12:10:23 (UTC+05:30)	15.0 B	Standard

File is replicated in pad.bucket-2

Amazon S3 > Buckets > pad.bucket-2

pad.bucket-2

ObjectsMetadata - PreviewPropertiesPermissionsMetricsManagementAccess Points

Objects (1)

Copy S3 URICopy URLDownloadOpenDeleteActionsCreate folderUpload

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](#)

Find objects by prefixShow versions

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	terraform.txt	txt	January 11, 2025, 12:07:52 (UTC+05:30)	268.0 B	Standard

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

Amazon S3 > Buckets > pad.bucket-2

pad.bucket-2

ObjectsMetadata - PreviewPropertiesPermissionsMetricsManagementAccess Points

Objects (2)

Copy S3 URICopy URLDownloadOpenDeleteActionsCreate folderUpload

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](#)

Find objects by prefixShow versions

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	s3.txt	txt	January 11, 2025, 12:25:02 (UTC+05:30)	11.0 B	Standard
<input type="checkbox"/>	terraform.txt	txt	January 11, 2025, 12:10:23 (UTC+05:30)	15.0 B	Standard

Data replicated in pad.bucket-1

Amazon S3 > Buckets > pad.bucket-1

pad.bucket-1 [Info](#)

[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2) [Info](#)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

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](#)

☐ Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	s3.txt	txt	January 11, 2025, 12:25:02 (UTC+05:30)	11.0 B	Standard
<input type="checkbox"/>	terraform.txt	txt	January 11, 2025, 12:10:23 (UTC+05:30)	15.0 B	Standard

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
```

☰ [Amazon S3](#) > [Buckets](#) > pad.bucket-1

pad.bucket-1 [Info](#)




[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Acc](#)

Objects (3) [Info](#)

 [Copy S3 URI](#) [Copy URL](#) [Download](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in [more](#).


☐ Show versions

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	 flm.txt	txt	January 11, 2025, 12:41:28 (UTC+05:30)
<input type="checkbox"/>	 s3.txt	txt	January 11, 2025, 12:25:02 (UTC+05:30)
<input type="checkbox"/>	 terraform.txt	txt	January 11, 2025, 12:10:23 (UTC+05:30)

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

Access control list (ACL) [Edit](#)

Grant basic read/write permissions to other AWS accounts. [Learn more](#)

 This bucket has the bucket owner enforced setting applied for Object Ownership

When [bucket owner enforced](#) is applied, use bucket policies to control access. [Learn more](#)

Edit Object Ownership [info](#)

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

⚠ Enabling ACLs turns off the bucket owner enforced setting for Object Ownership

Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket owner's permissions. ☒ I acknowledge that ACLs will be restored.

Object Ownership

☒ Bucket owner preferred

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ Object writer

The object writer remains the object owner.

Next under permissions → block access untick and save

Amazon S3 > Buckets > pad.bucket-1 > Edit Block public access (bucket settings)

Edit Block public access (bucket settings) [info](#)

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings is independent of one another.

☐ Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Cancel

Save changes

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

The screenshot shows the AWS Management Console interface for the 'pad.bucket-1' bucket. The 'Block public access' settings are visible, and the 'Actions' menu is open, showing the option 'Make public using ACL'. The console also displays a list of objects in the bucket, including 'flm.txt', 's3.txt', and 'terraform.txt'.

Objects (3) [info](#)

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 must grant permissions. [Learn more](#)

Find objects by prefix ☒ Show versions

Name	Type	Last modified	Size
<input checked="" type="checkbox"/> flm.txt	txt	January 11, 2025, 12:41:28 (UTC+05:30)	
<input type="checkbox"/> s3.txt	txt	January 11, 2025, 12:25:02 (UTC+05:30)	
<input type="checkbox"/> terraform.txt	txt	January 11, 2025, 12:10:23 (UTC+05:30)	

Actions menu options:

- Download as
- Share with a presigned URL
- Calculate total size
- Copy
- Move
- Initiate restore
- Query with S3 Select
- Edit actions
- Rename object
- Edit storage class
- Edit server-side encryption
- Edit metadata
- Edit tags
- Make public using ACL

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

Amazon S3 > Buckets > pad.website > Edit static website hosting

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
☐ Disable
☒ Enable

Hosting type
☒ Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
☐ Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access setting. [Using Amazon S3 Block Public Access](#)

Index document
Specify the home or default page of the website.

Error document - optional
This is returned when an error occurs.

Redirection rules - optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

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

← → ↻ ⚠ Not secure pad.website.s3-website-us-east-1.amazonaws.com

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: KB2Q0W6NZVR8E3YG
- HostId: 2UVMDba6q9QryPemXAkFQiNjEtDEO/a0g8Z2/KIeopQVSh8qAXo1zXNz3ZIDQ9iTxyzoDiiwo6s=

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

← → ↻ ⚠ Not secure pad.website.s3-website-us-east-1.amazonaws.com

welcome to terraform

we are learning devops

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

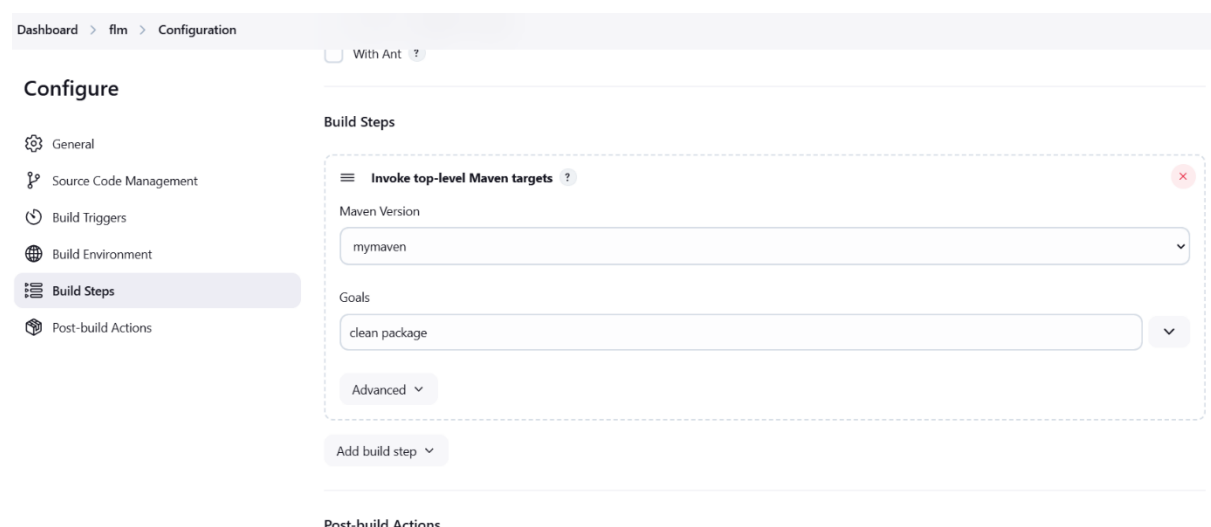
Integrate S3 to jenkins Manage jenkins → system → S3 profiles → click on Add → profile name (devops) → attached IAM user → give the access and secret key → click on test connection and save

[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.



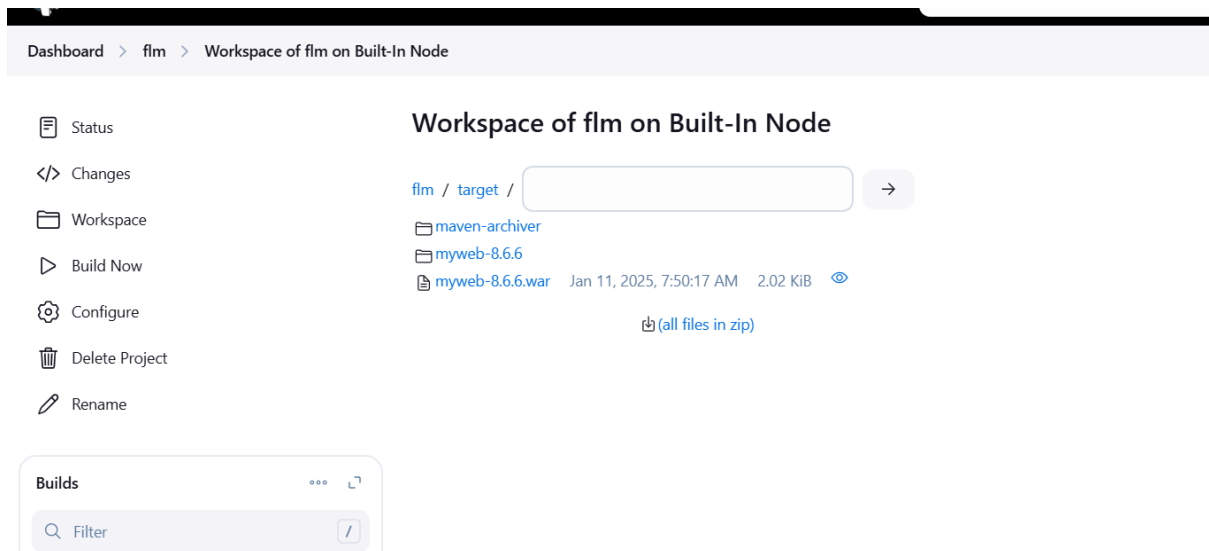
Now build the flm job

```

bash: cd: /var/lib/jenkins/workspace/flm/: No such file or directory
[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 configurations → 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.

← → ↻ Not secure 44.201.77.222:8080/job/flm/configure ☆

Dashboard > flm > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Post-build Actions

≡ Publish artifacts to S3 Bucket ?

✕

S3 profile

devops

Files to upload

Source ?

✕

target/*.war

Exclude

Destination bucket ?

pad.bucket-1

Storage class

STANDARD

Bucket Region ?

us-east-1

☒ No upload on build failure ?

☐ Publish from Slave ?

← → ↻ Not secure 44.201.77.222:8080/job/flm/configure ☆

Dashboard > flm > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Source ?

✕

target/*.war

Exclude

Destination bucket ?

pad.bucket-1

Storage class

STANDARD

Bucket Region ?

us-east-1

☒ No upload on build failure ?

☐ Publish from Slave ?

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

≡ [Amazon S3](#) > [Buckets](#) > pad.bucket-1

pad.bucket-1 [Info](#)

[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (5) [Info](#)

[Copy S3 URI](#)[Copy URL](#)[Download](#)[Open](#)


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 more

☐ Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	flm.txt	txt	January 11, 2025, 12:41:28 (UTC+05:30)	
<input type="checkbox"/>	index.html	html	January 11, 2025, 12:55:56 (UTC+05:30)	
<input type="checkbox"/>	myweb-8.6.6.war	war	January 11, 2025, 13:25:45 (UTC+05:30)	
<input type="checkbox"/>	s3.txt	txt	January 11, 2025, 12:25:02 (UTC+05:30)	
<input type="checkbox"/>	terraform.txt	txt	January 11, 2025, 12:10:23 (UTC+05:30)	






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

Amazon S3 > Buckets > pad.bucket-1

Objects (10) [Info](#)  [Copy S3 URI](#) [Copy URL](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of objects.

☒ Show versions

<input type="checkbox"/>	Name	Type	Version ID
			Og
<input type="checkbox"/>	 flm.txt	txt	skLiNPFuMfiXFn.ppMW6zA3KT77apc
<input type="checkbox"/>	 index.html	html	6rTEa5UMExHcDyGQBDEnERHKjMw7f
<input type="checkbox"/>	 myweb-8.6.6.war	war	76b38gsDY\U6IHq.PC2.rhr9nOaAcJn
<input type="checkbox"/>	 myweb-8.6.6.war	war	uf_M4SgkHvkn5F_bKxW:gOpYZhEcpvP
<input type="checkbox"/>	 s3.txt	txt	sjGgQpLMj6PIMJ.EZydC1Q29JBo4vPI

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)

jack.terra.bucket [Info](#)

[Objects](#) | [Metadata - Preview](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (0) [Info](#)

[Copy S3 URI](#)[Copy URL](#)[Download](#)[Open](#)

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 more

☐ Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size
--------------------------	------	------	---------------	------

No objects

You don't have any objects in this bucket.

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 the bucket.

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

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