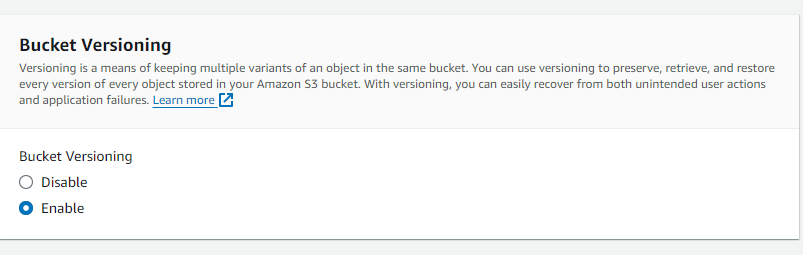
**S3 bucket versioning**

**Versioning** is a means of keeping the multiple forms of an object in the same S3 bucket. Versioning can be used to retrieve, preserve and restore every version of an object in S3 bucket.

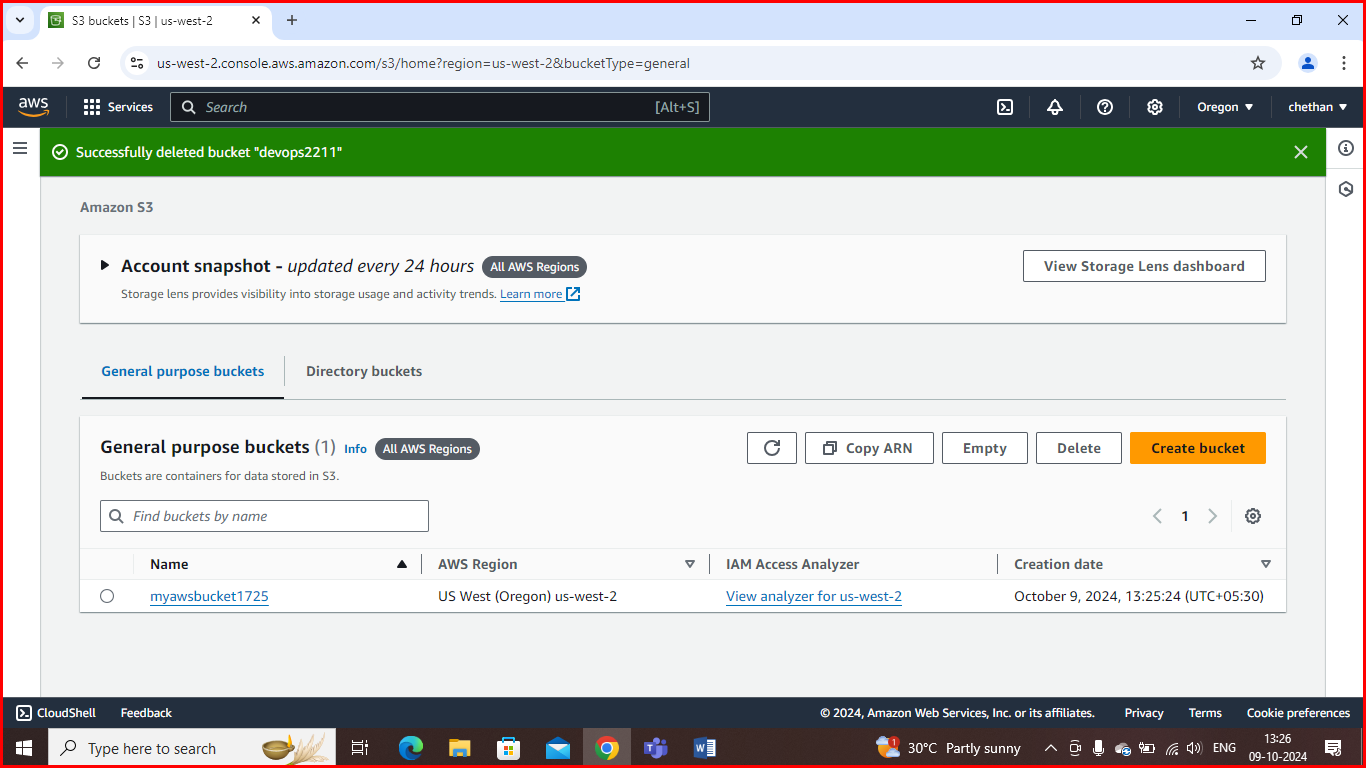
For example, bucket consists of two objects with the same key but with different version ID's such as photo.jpg (version ID is 11) and photo.jpg (version ID is 12).

Versioning-enabled buckets allow you to recover the objects from the deletion or overwrite. It serves two purposes:

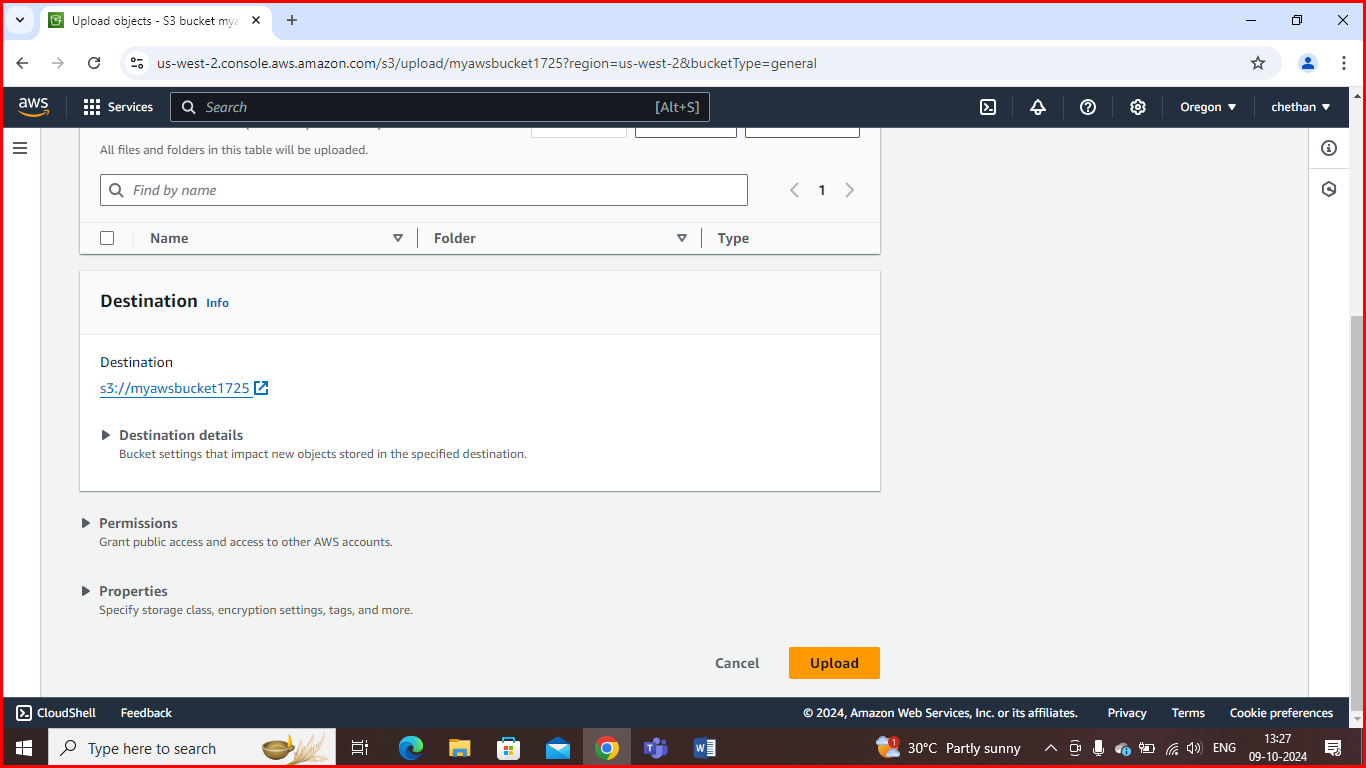
* If you delete an object, instead of deleting the object permanently, it creates a delete marker which becomes a current version of an object.
* If you overwrite an object, it creates a new version of the object and also restores the previous version of the object.



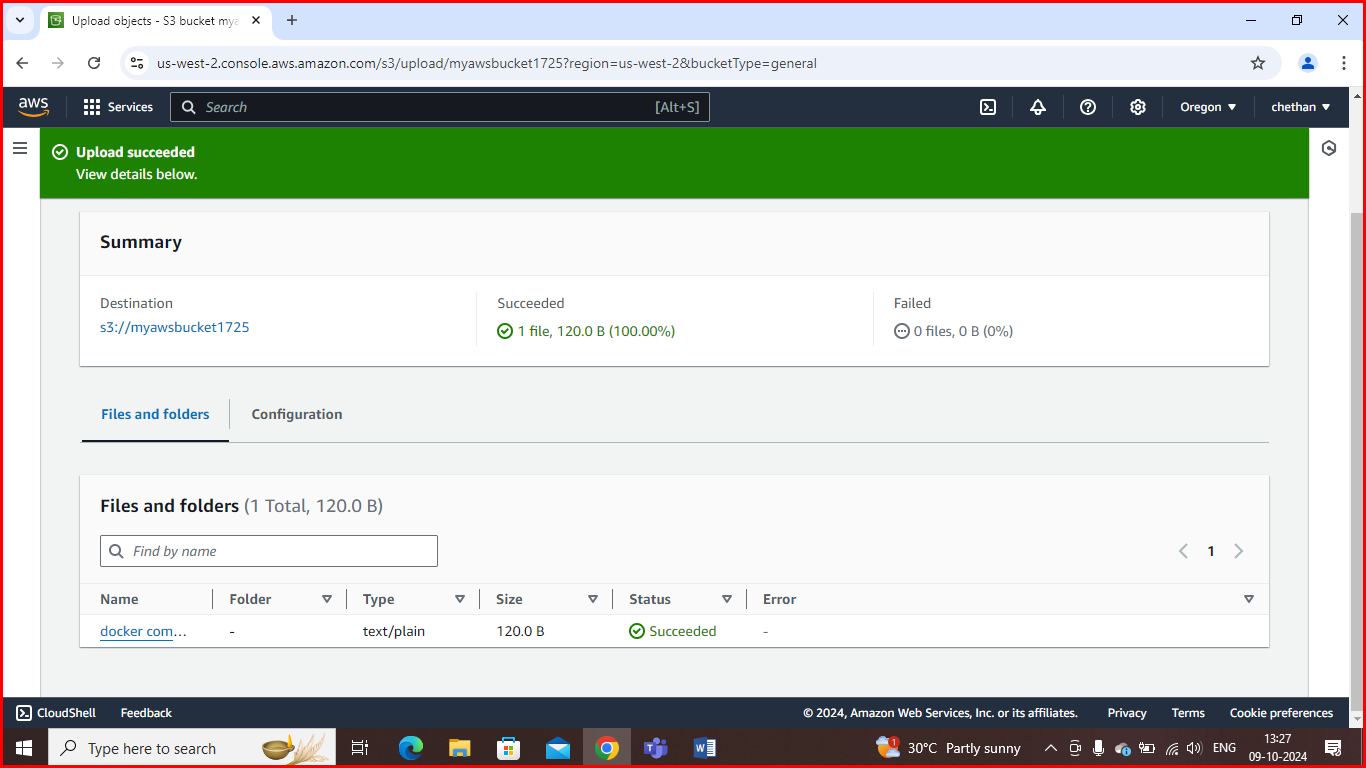
**Create bucket :** Creating bucket with the unique name myawsbucket1725

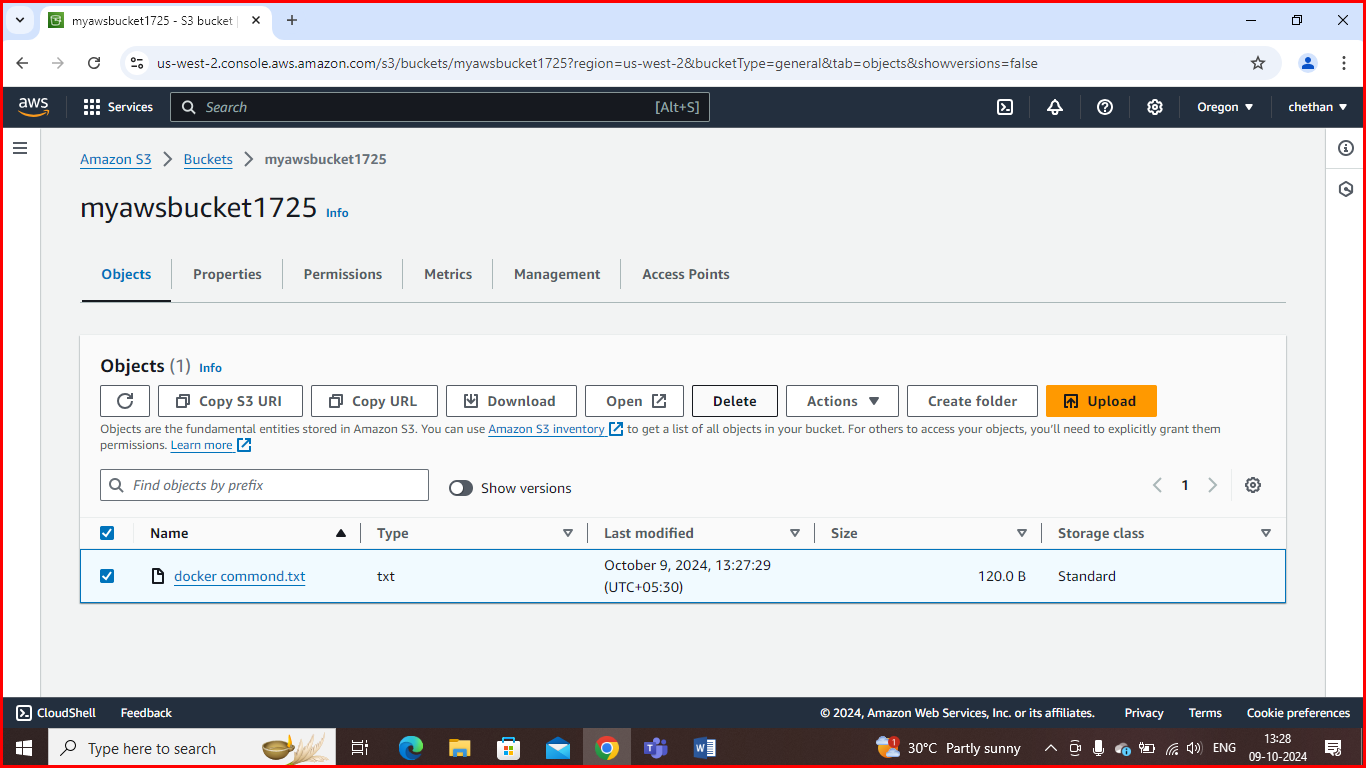


**Uploading object into bucket**

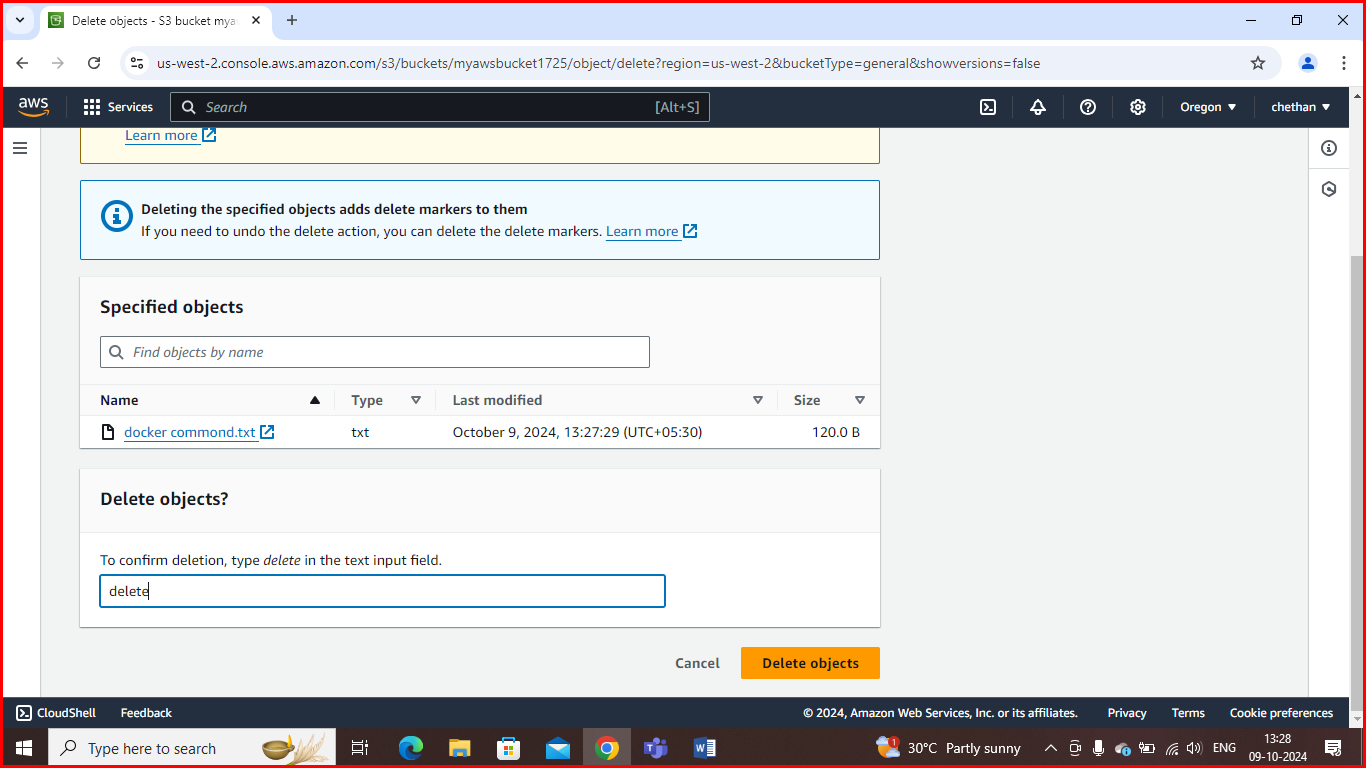


**Object has successfully uploaded into the bucket**

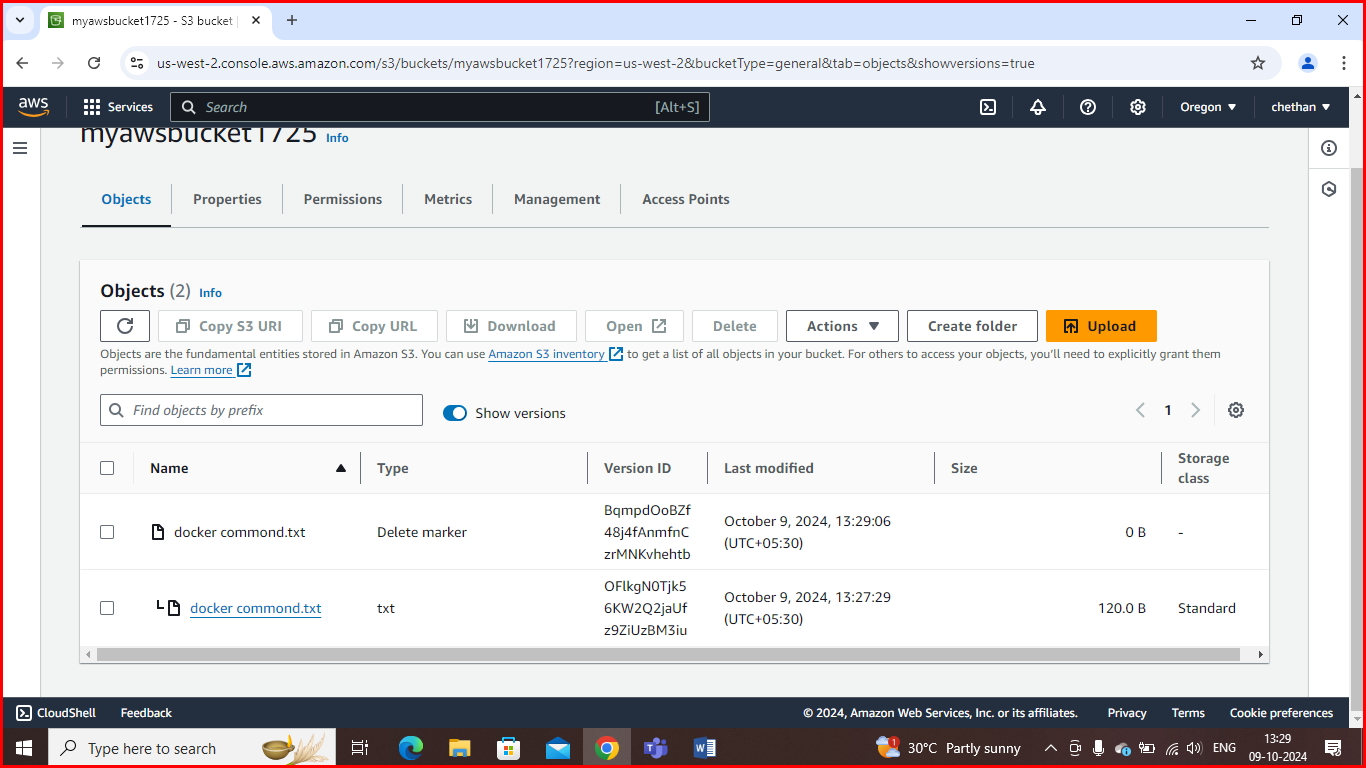




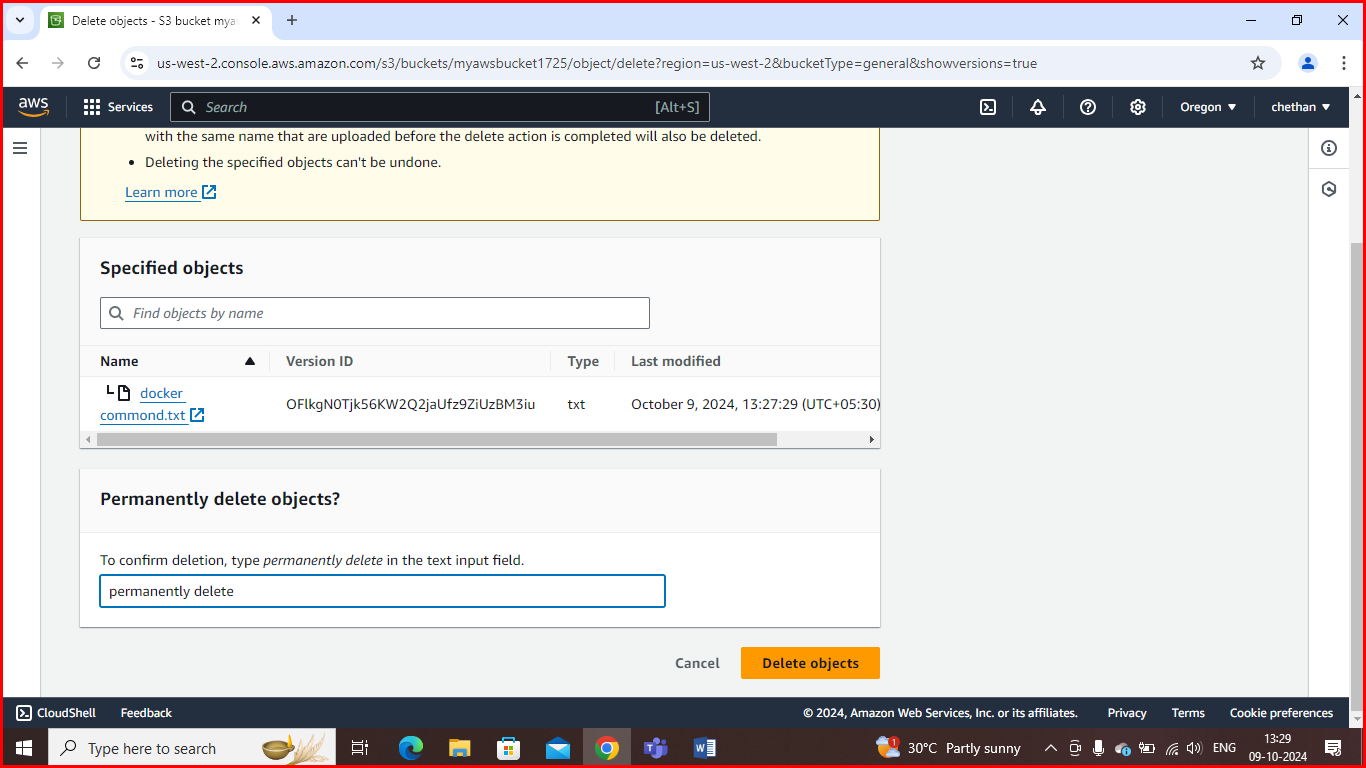
**Deleting object in bucket when versioning is enable**



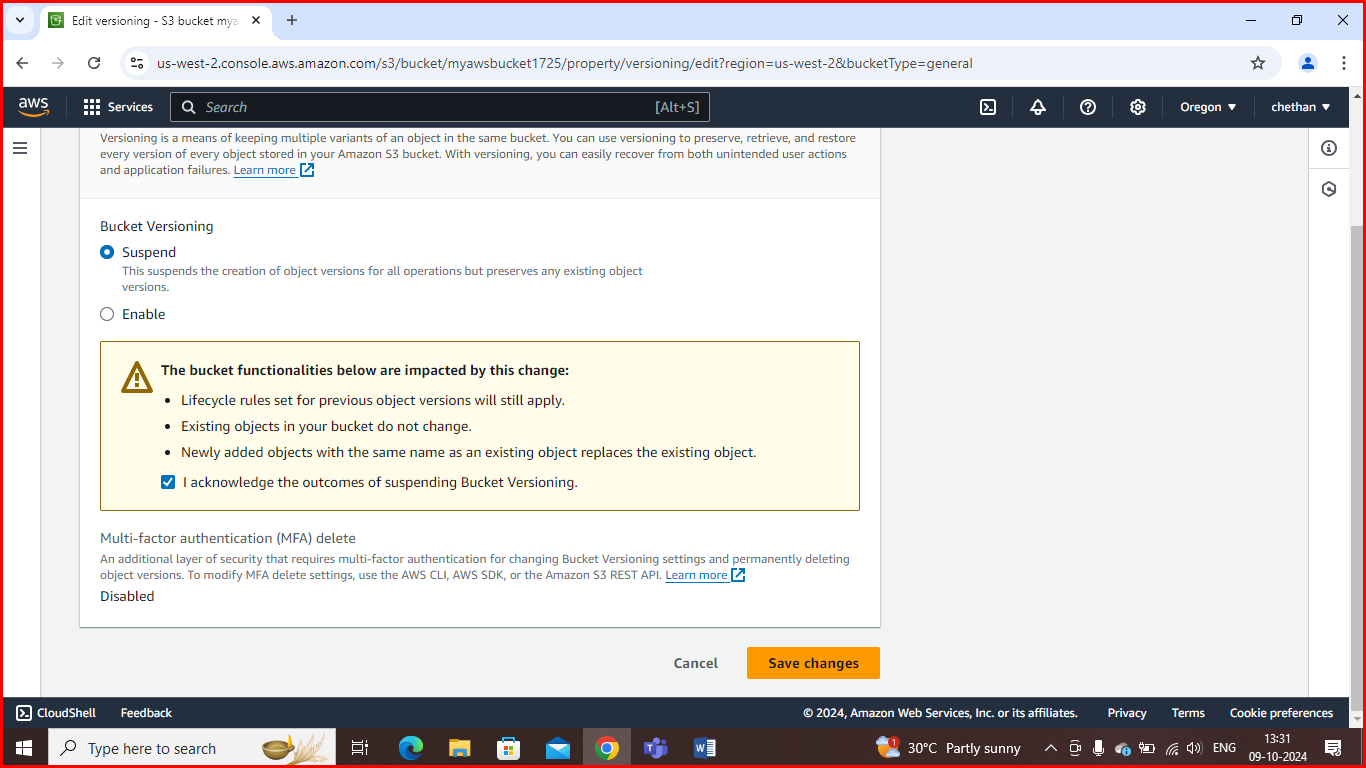
After deleting object, instead of deleting the object permanently, it creates a delete marker which becomes a current version of an object



**Deleting object permanently**

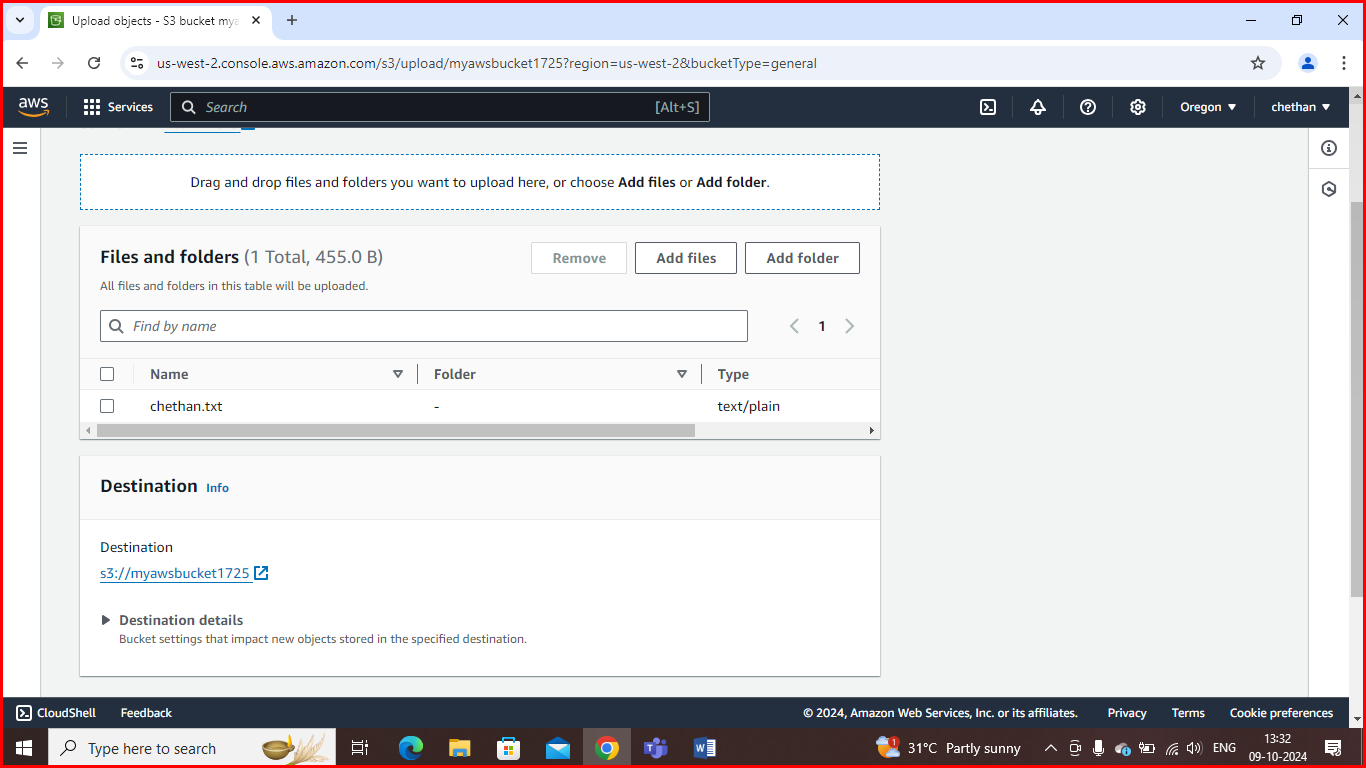


**Suspending bucket versioning and saving the changes**

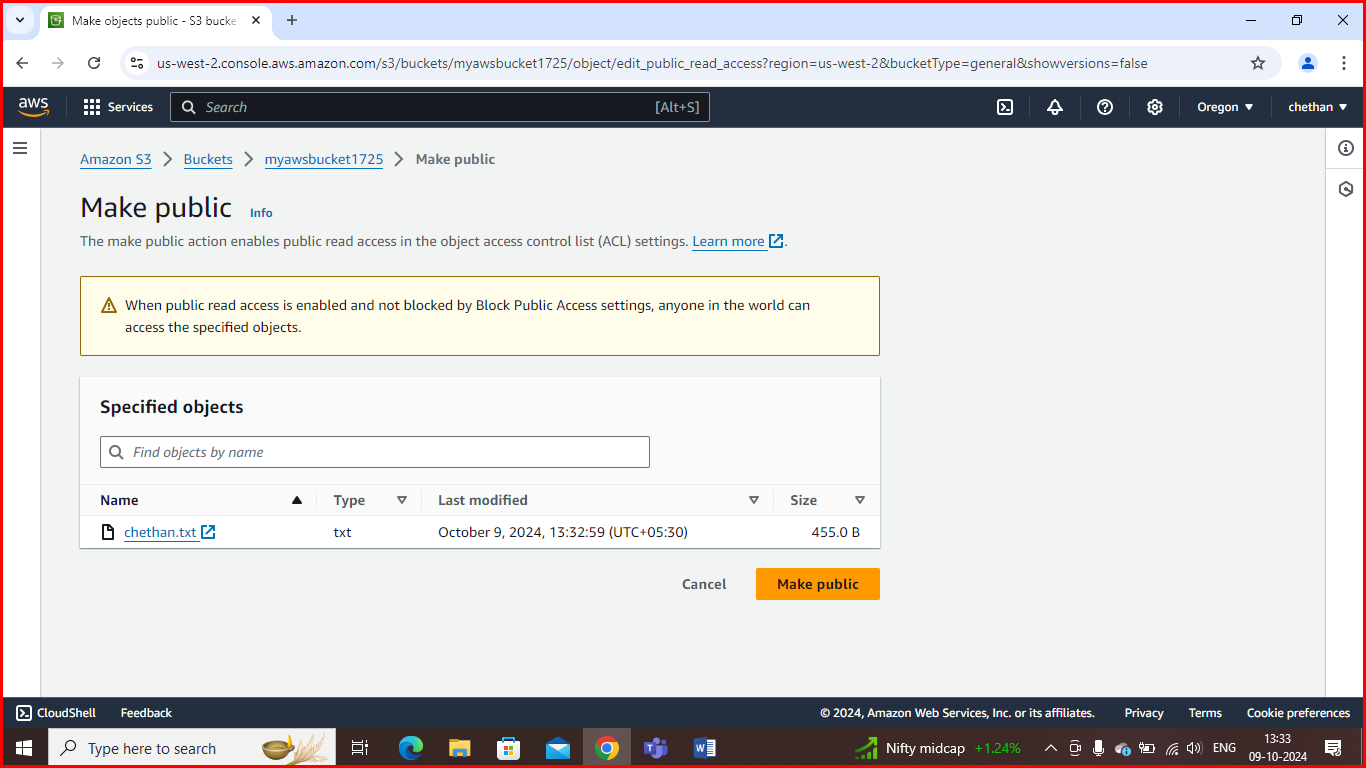




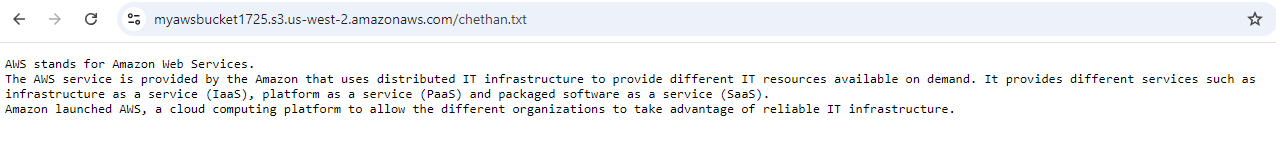
**Uploading objects into the bucket after suspending bucket version**



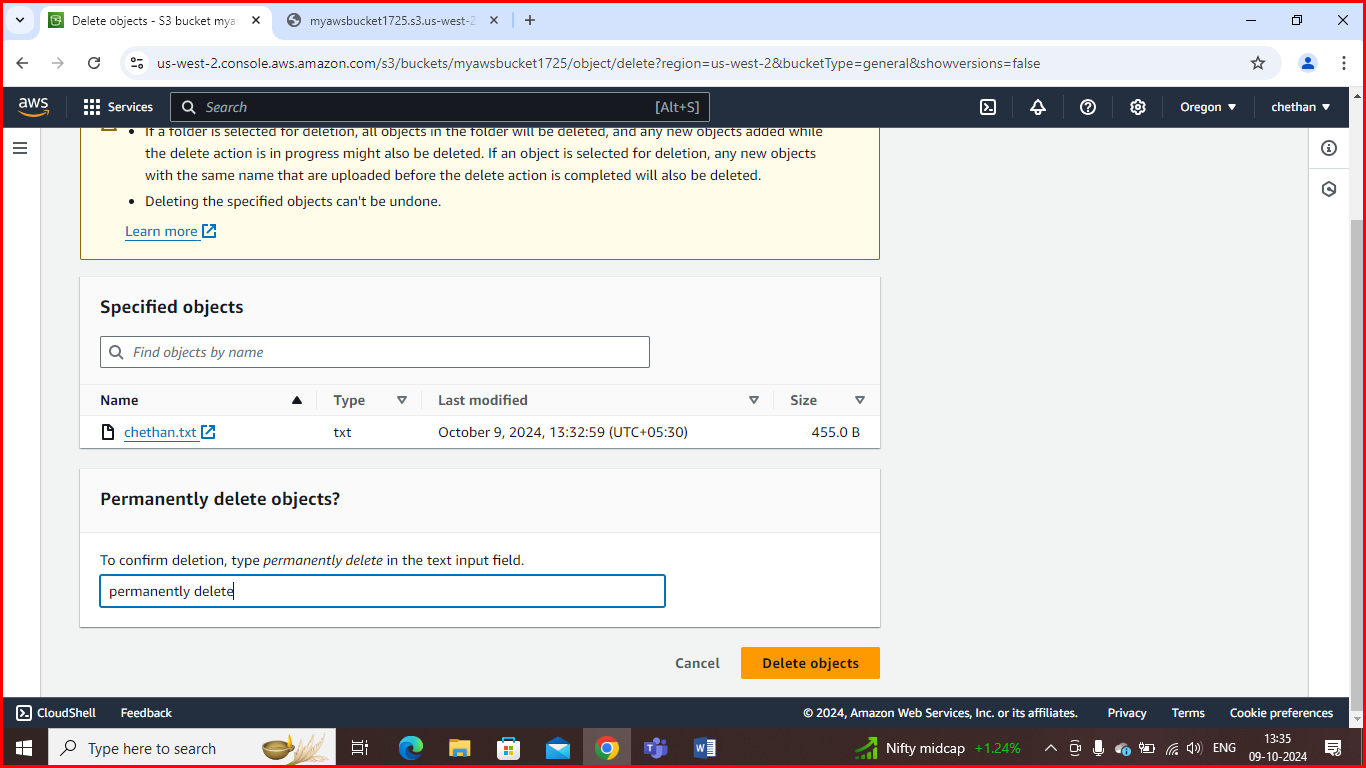
**Make object public**



**Copy object URL and paste in browser**



When bucket versioning is disabled, deleting a file removes it permanently



**S3 versioning using Terraform:**

Enabling versioning on an Amazon S3 bucket allows you to keep multiple versions of an object in the same bucket. This is useful for recovering previous versions of files or for maintaining a history of changes.

**Code:**

provider "aws" {

  region = "us-west-2"

}

resource "aws\_s3\_bucket" "my\_bucket" {

  bucket = "devops2211"

  versioning {

    enabled = true

  }

  acl = "private"

  tags = {

    Name        = "My bucket"

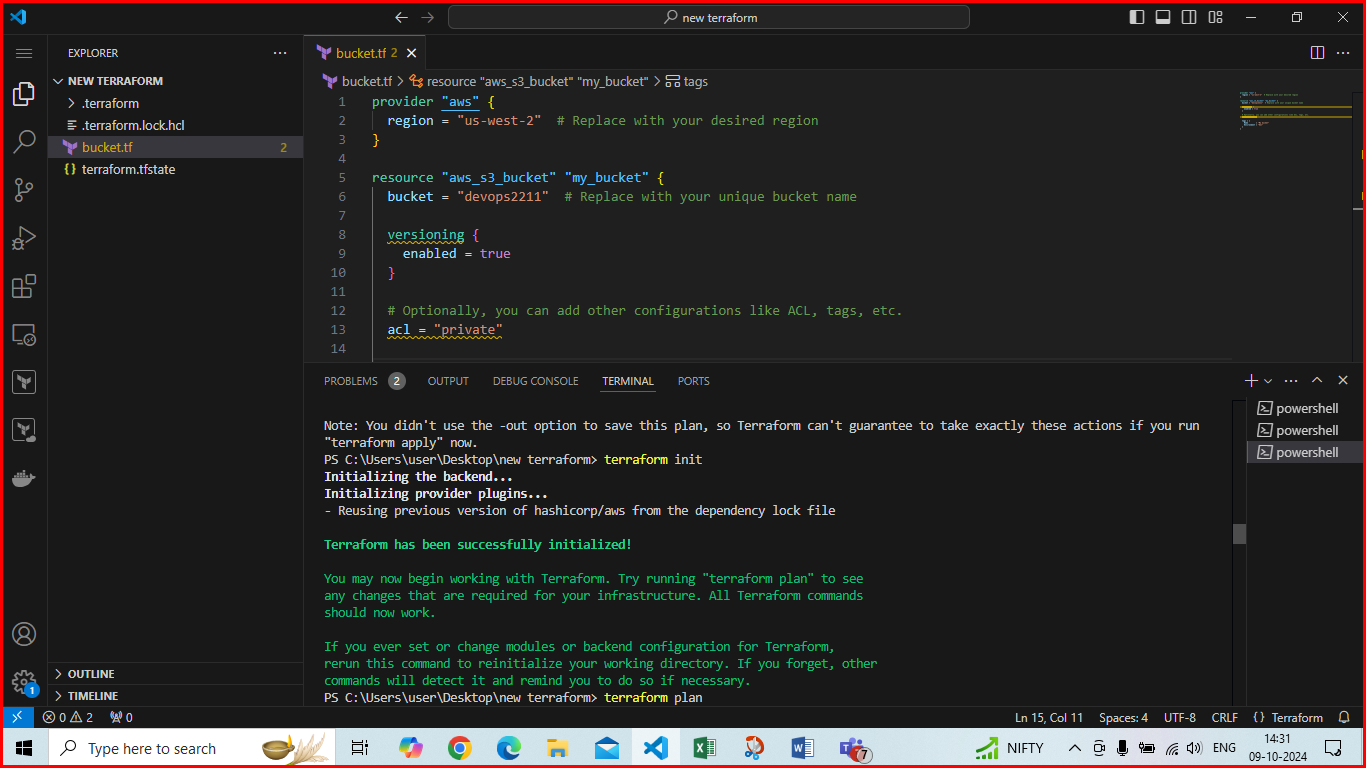
    Environment = "Dev"

  }

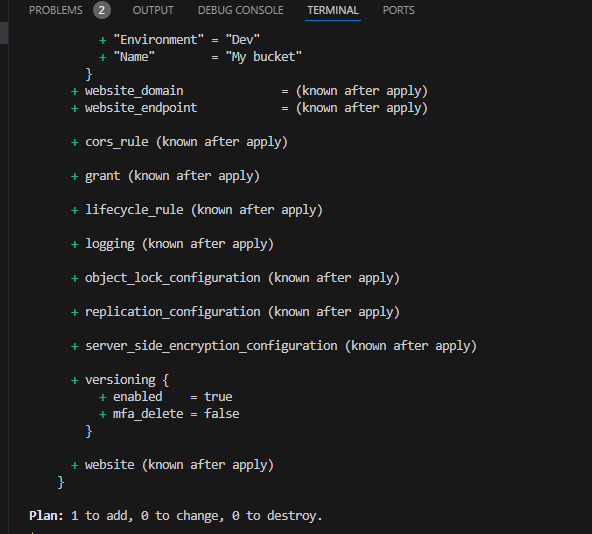
}

**Aws configure:** To configure AWS with Terraform, you typically need to set up your AWS credentials and specify the desired region.

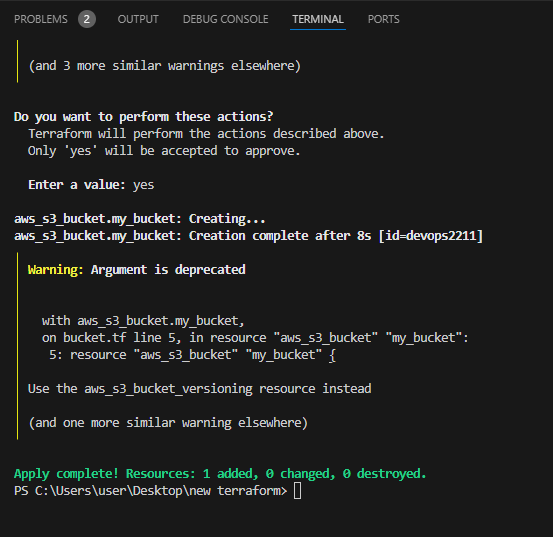
**Terraform init:** The terraform init command is the first command you run in a new or existing Terraform configuration directory. It initializes the directory containing Terraform configuration files and prepares it for other Terraform commands.



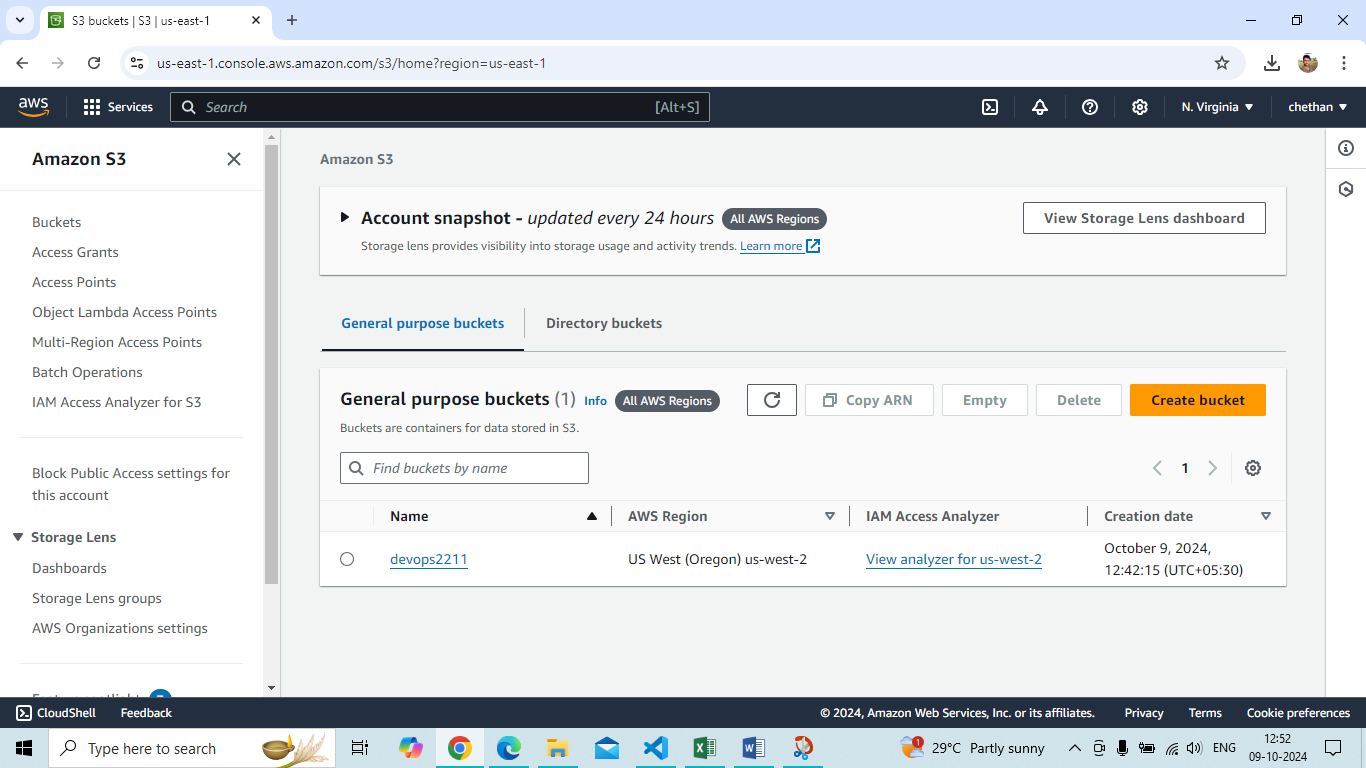
**Terraform plan:** The terraform plan command is used to create an execution plan for your Terraform configuration. It helps you understand what changes Terraform will make to your infrastructure before actually applying those changes



**Terraform apply:** The terraform apply command is used to execute the changes specified in your Terraform configuration and plan. It creates, updates, or destroys infrastructure resources based on your defined configuration



Bucket has created: devops2211 bucket has created.



**S3 versionig enabled:** Amazon S3 (Simple Storage Service) versioning is a feature that allows you to keep multiple versions of an object in the same S3 bucket. When versioning is enabled, every time you upload a file with the same name (key), S3 retains the previous versions of that file, allowing you to access, restore, or delete them as needed.

