

## Lesson 03 Demo 03

### Using IAM Roles to Access S3 Bucket

**Objective:** To securely access Amazon S3 (Simple Storage Service) buckets from an EC2 instance using IAM (Identity and Access Management) roles

**Tools required:** AWS Lab

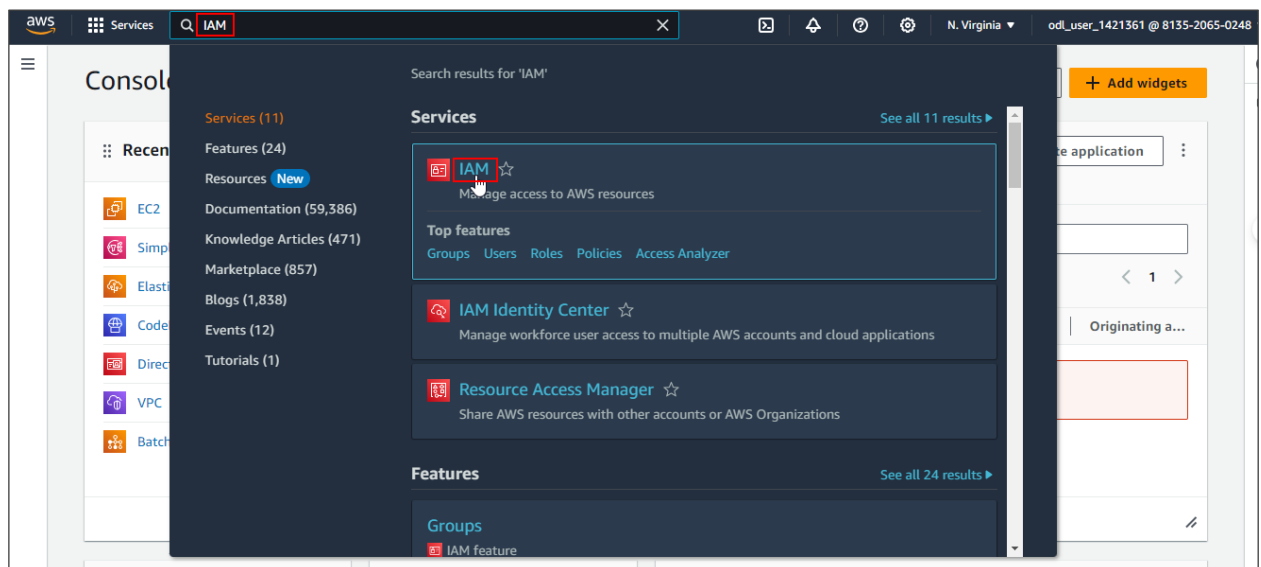
**Prerequisites:** Create an EC2 instance named S3

Steps to be followed:

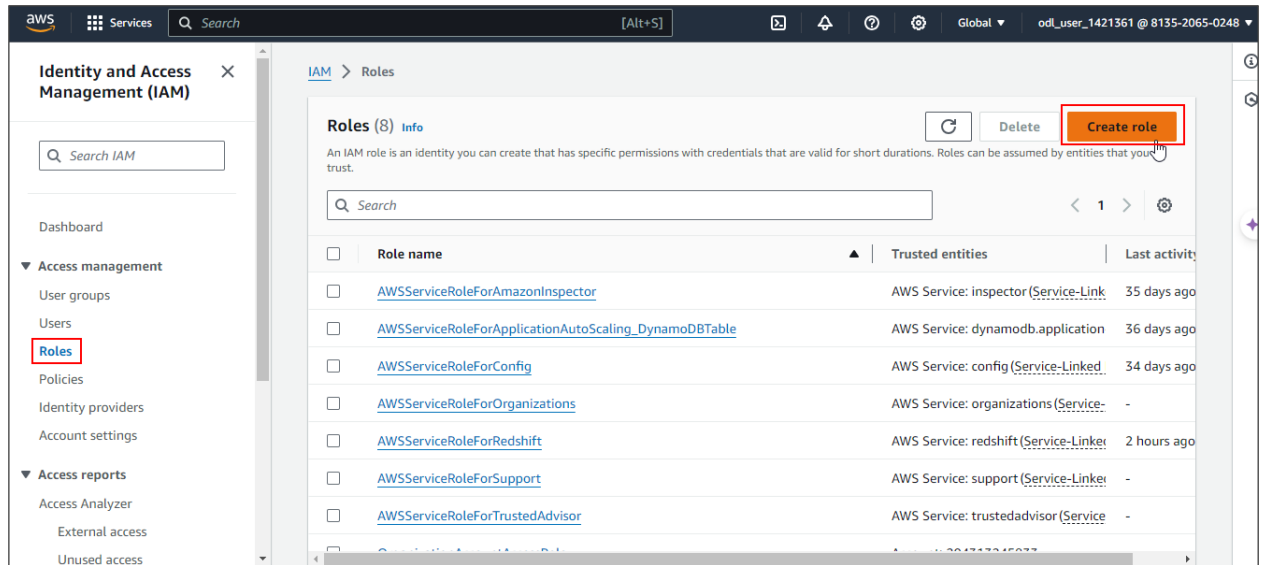
1. Create an IAM role
2. Connect IAM Profile to EC2
3. Validate access to the S3 bucket

#### Step 1: Create an IAM role

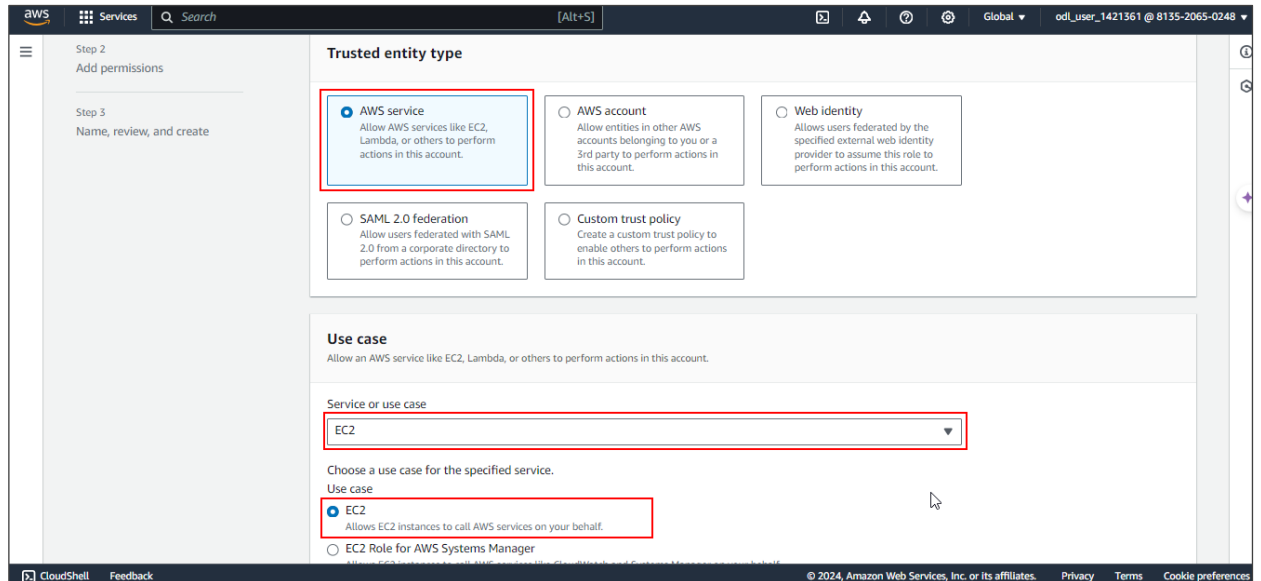
1.1 Navigate to the AWS console home dashboard, search for, and click on **IAM** as shown:

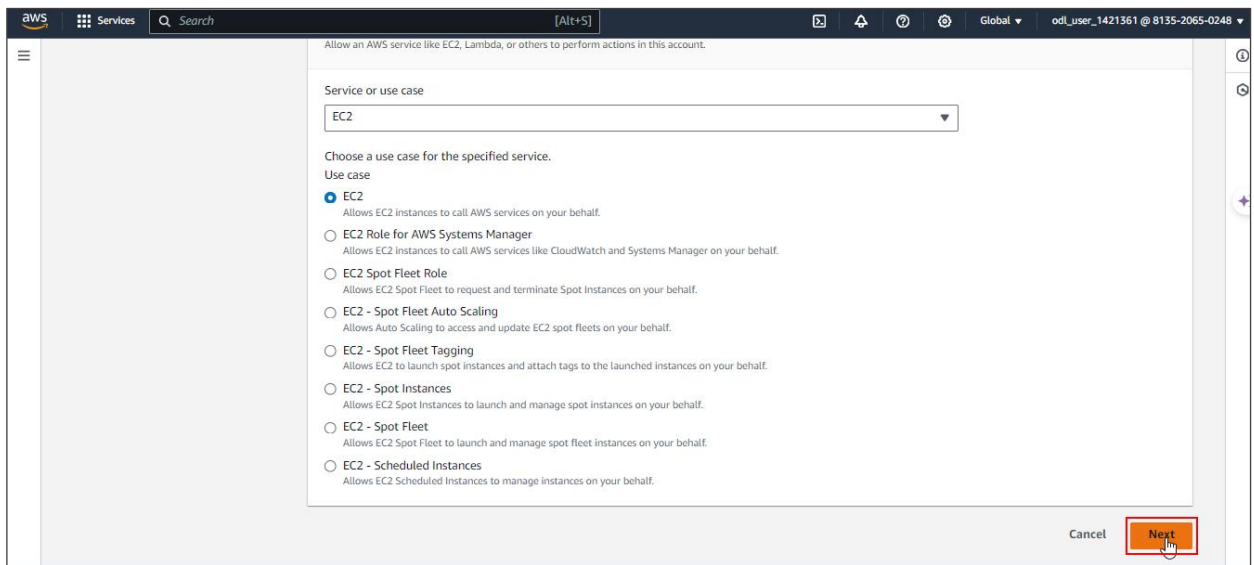


## 1.2 Navigate to **Roles** and click on the **Create role** button

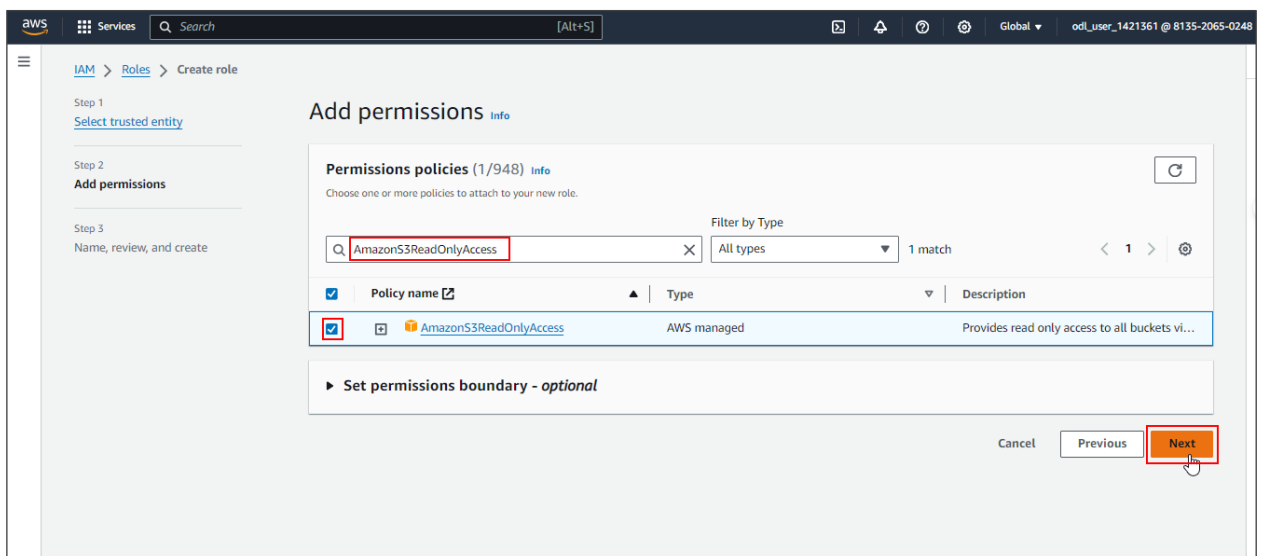


## 1.3 Select the **AWS service** option under **Trusted entity type** and the **EC2** option under **Use cases**, and click on **Next**





1.4 Search for and select **AmazonS3ReadOnlyAccess**, then click on **Next** to proceed



## 1.5 Add the role name as **S3access**, and click on **Create role**

[IAM](#) > [Roles](#) > Create role

Step 1  
[Select trusted entity](#)

Step 2  
[Add permissions](#)

Step 3  
**Name, review, and create**

### Name, review, and create

Role details

**Role name**  
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and "+=, @>\_" characters.

**Description**  
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+=, @>\_/{[]}\$%'^\*~`-~`

**Step 1: Select trusted entities** Edit

**Trust policy**

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [

```

aws Services Search [Alt+S] Global odl\_user\_1421361 @ 8135-2065-0248

14 }  
15 }  
16 }

**Step 2: Add permissions** Edit

**Permissions policy summary**

Policy name	Type	Attached as
<a href="#">AmazonS3ReadOnlyAccess</a>	AWS managed	Permissions policy

**Step 3: Add tags**

**Add tags - optional** [Info](#)  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

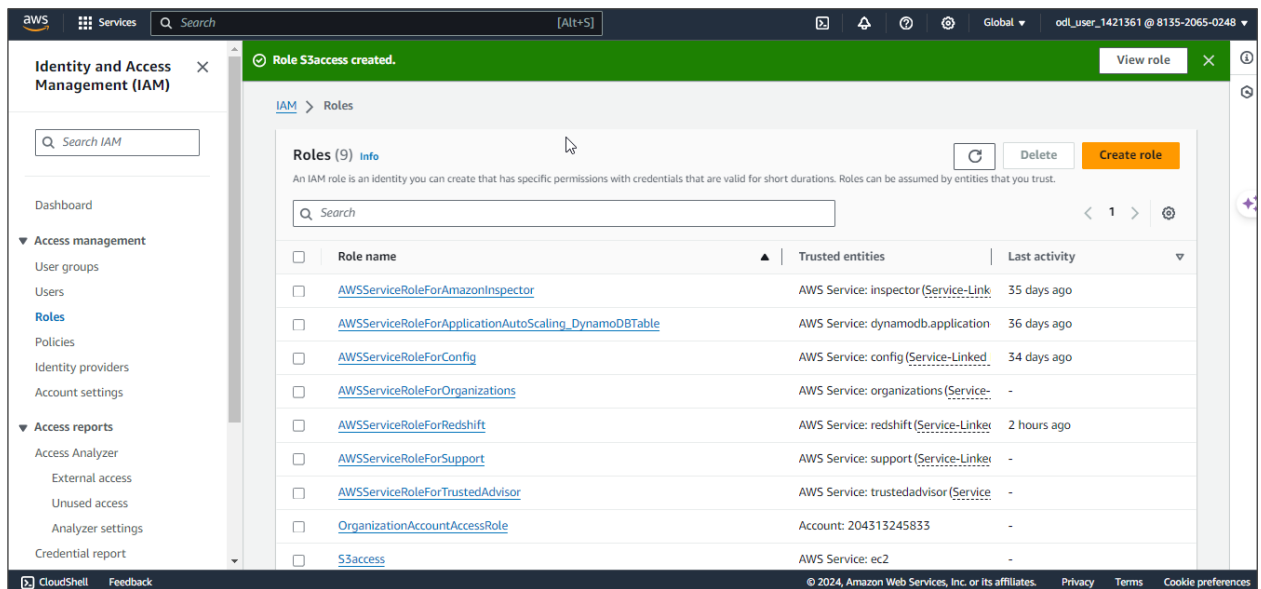
No tags associated with the resource.

Add new tag  
You can add up to 50 more tags.

Cancel Previous Create role

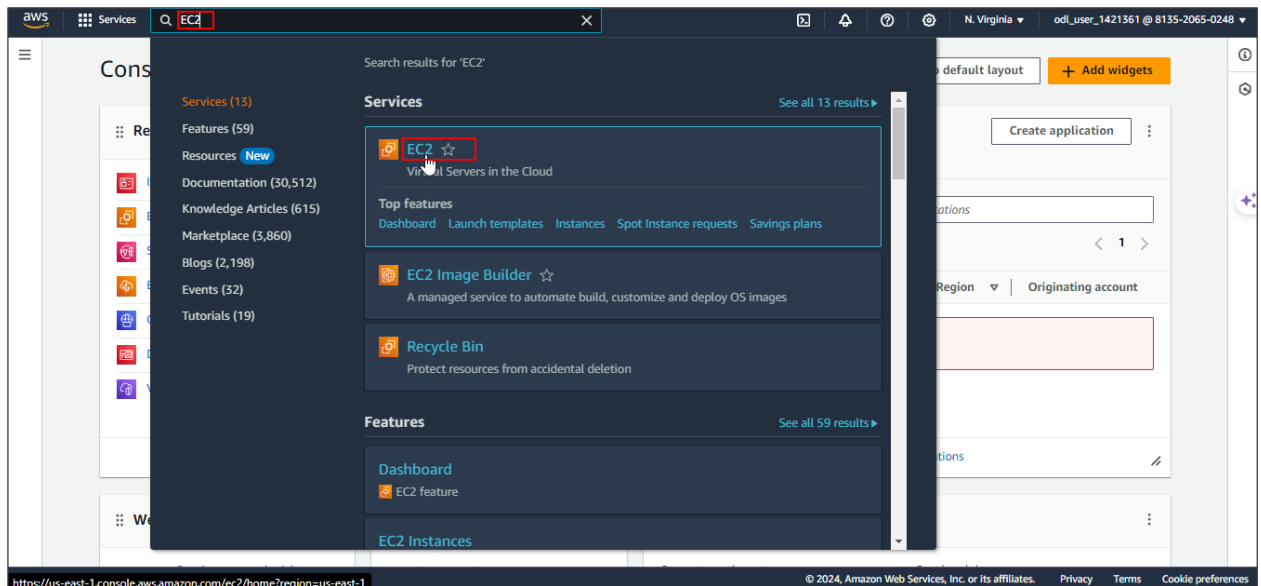
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The IAM role is successfully created as shown:

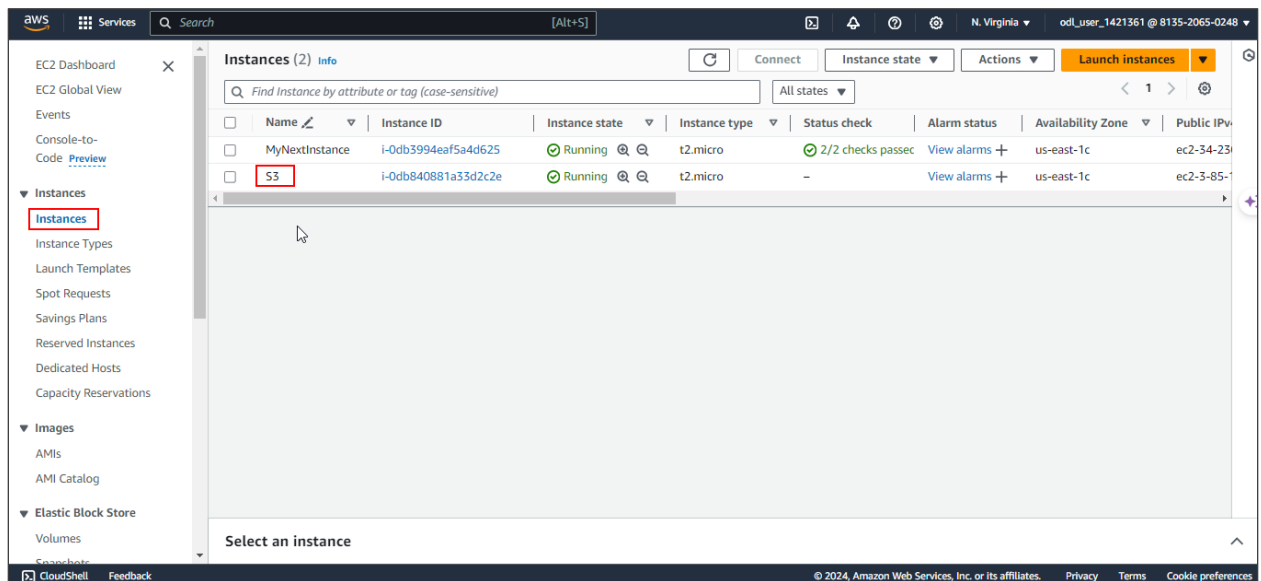


## Step 2: Connect IAM Profile to EC2

2.1 Navigate to the AWS console home dashboard, search for, and click on **EC2** as shown:

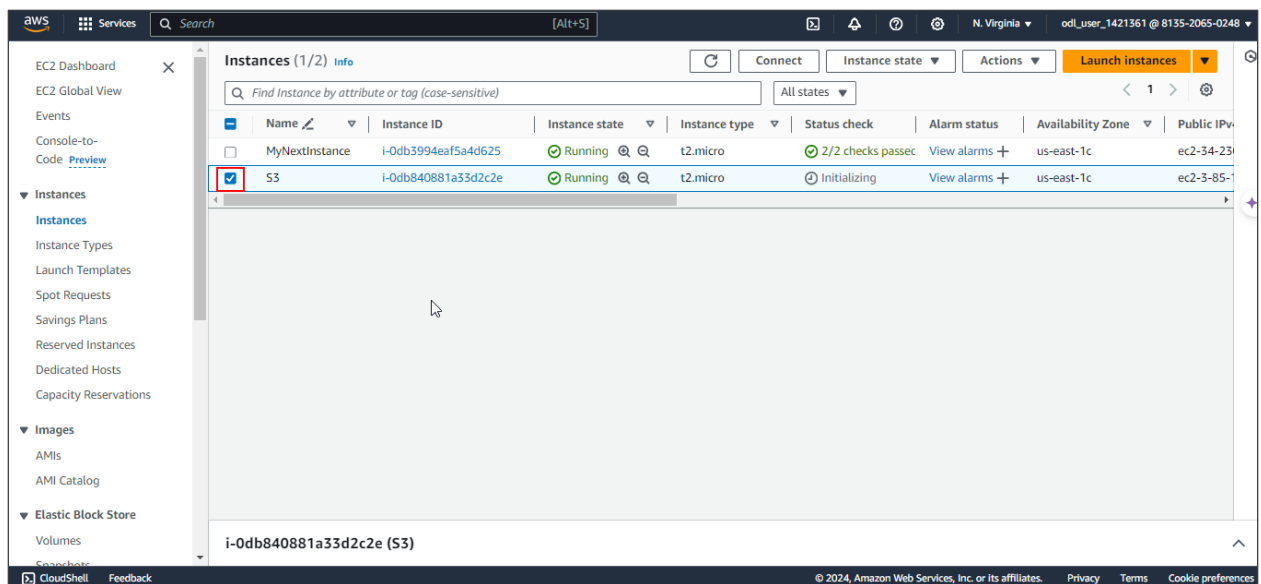


## 2.2 Click on **Instances** and launch a new instance named **S3**

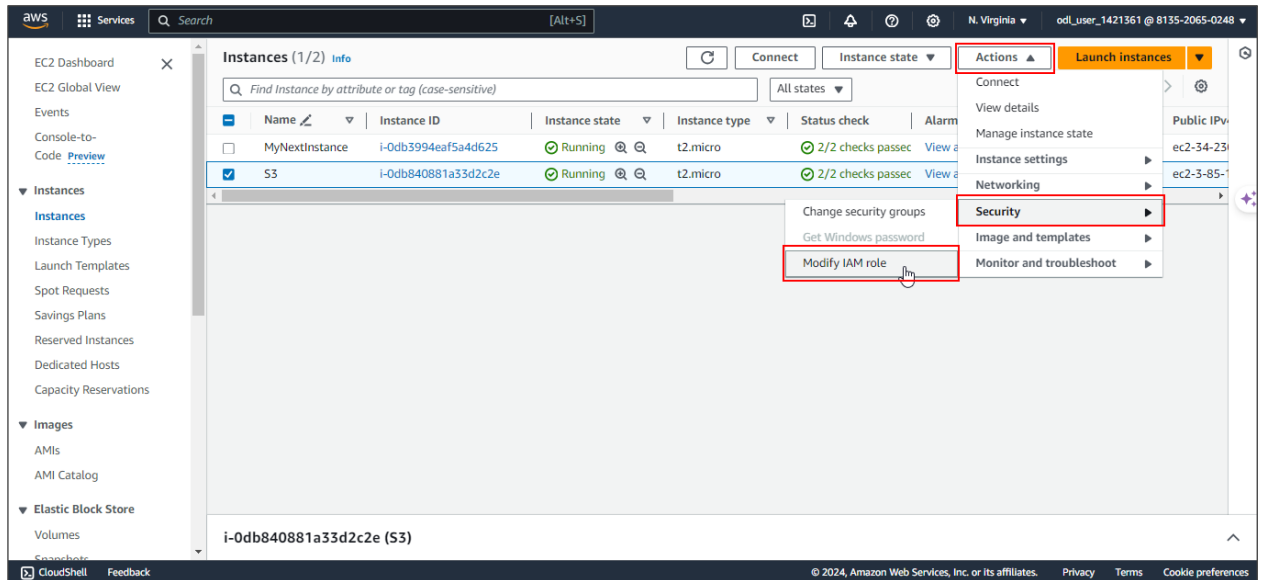


**Note:** Refer to Demo 01 of Lesson 03 for creating instances

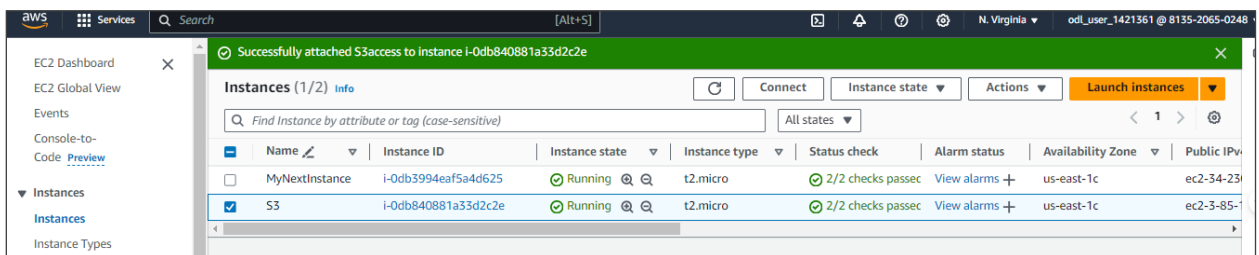
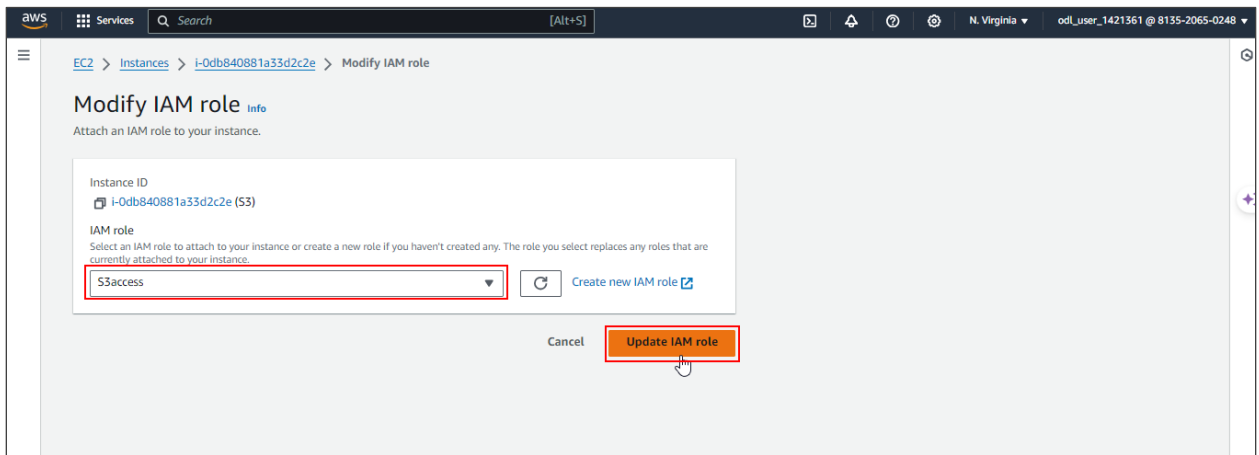
## 2.3 Select the **S3** instance



## 2.4 Under the **Actions** section, select the **Security** option and click on **Modify IAM role**

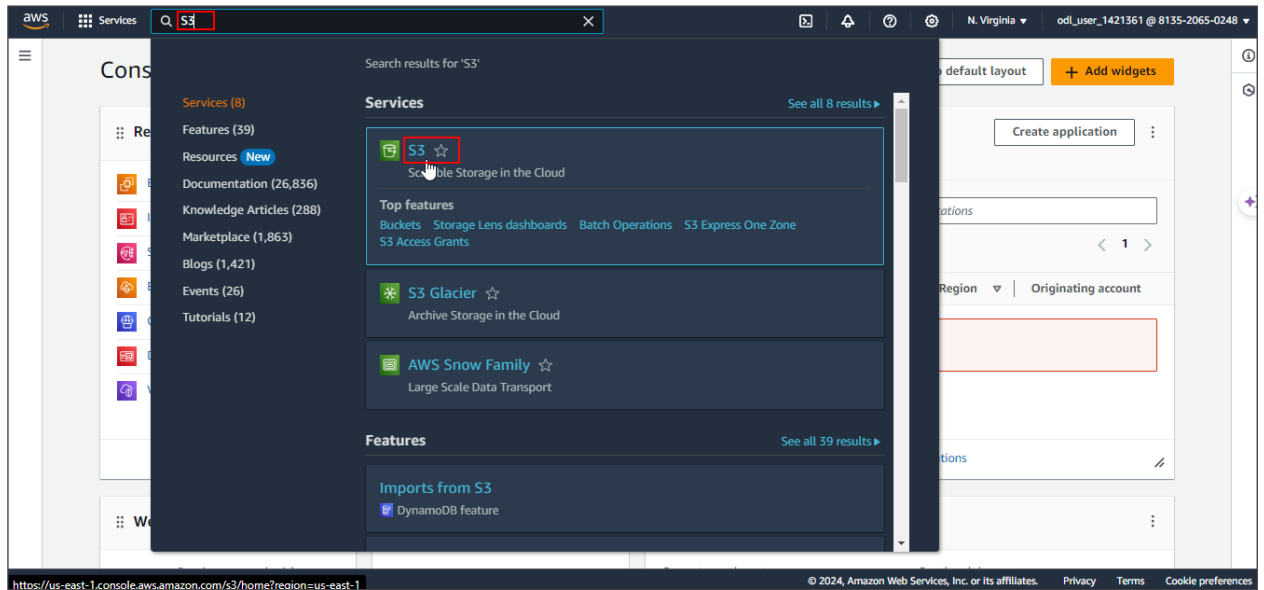


## 2.5 Select the previously created role **S3access** in the **IAM role** section and click on **Update IAM role**

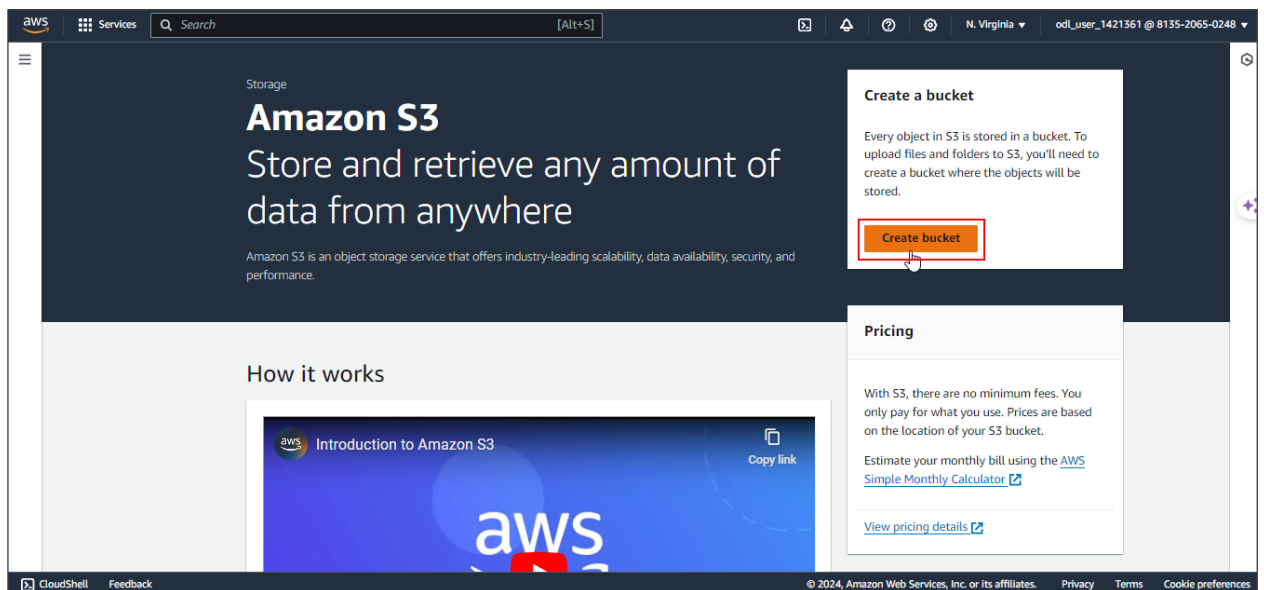


## Step 3: Validate access to the S3 bucket

3.1 Navigate to the AWS console home dashboard, search for, and click on **S3** as shown:



3.2 Click on **Create bucket** as shown:





### 3.3 Add a unique bucket name and click on **Create bucket** as shown:

**General configuration**

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

Bucket type [Info](#)

- ☒ **General purpose**  
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
- ☐ **Directory - New**  
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

s3bucketec211

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

**Object Ownership** [Info](#)

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)**
- ☐ ACLs enabled

**Default encryption** [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

- ☒ **Server-side encryption with Amazon S3 managed keys (SSE-S3)**
- ☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

Bucket Key  
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- ☐ Disable
- ☒ **Enable**

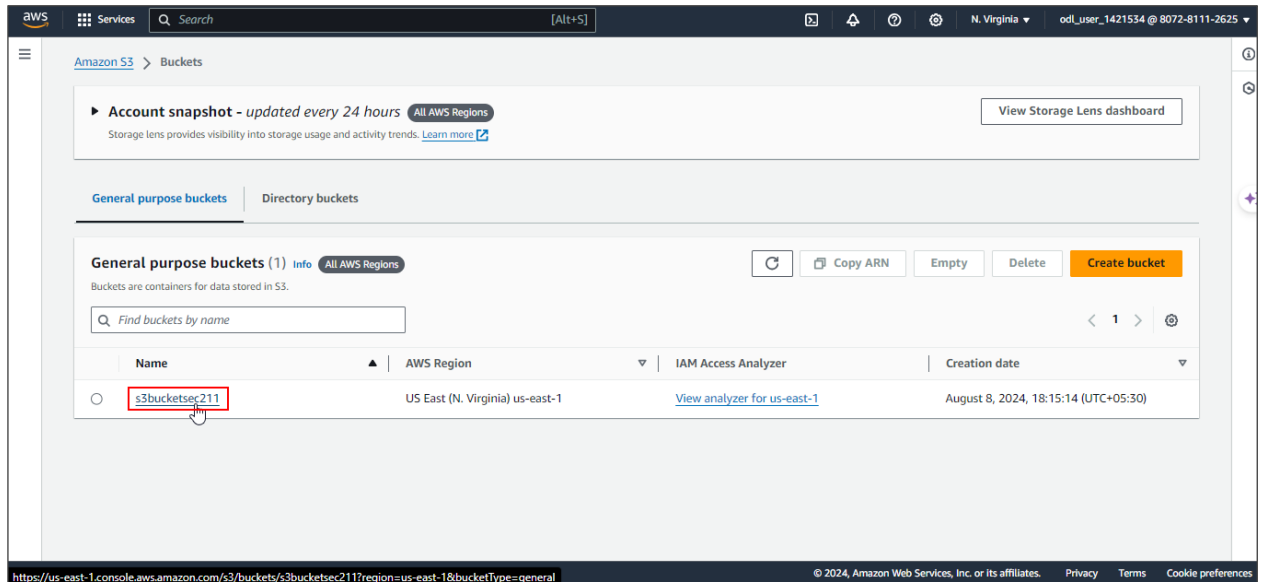
**Advanced settings**

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel [Create bucket](#)

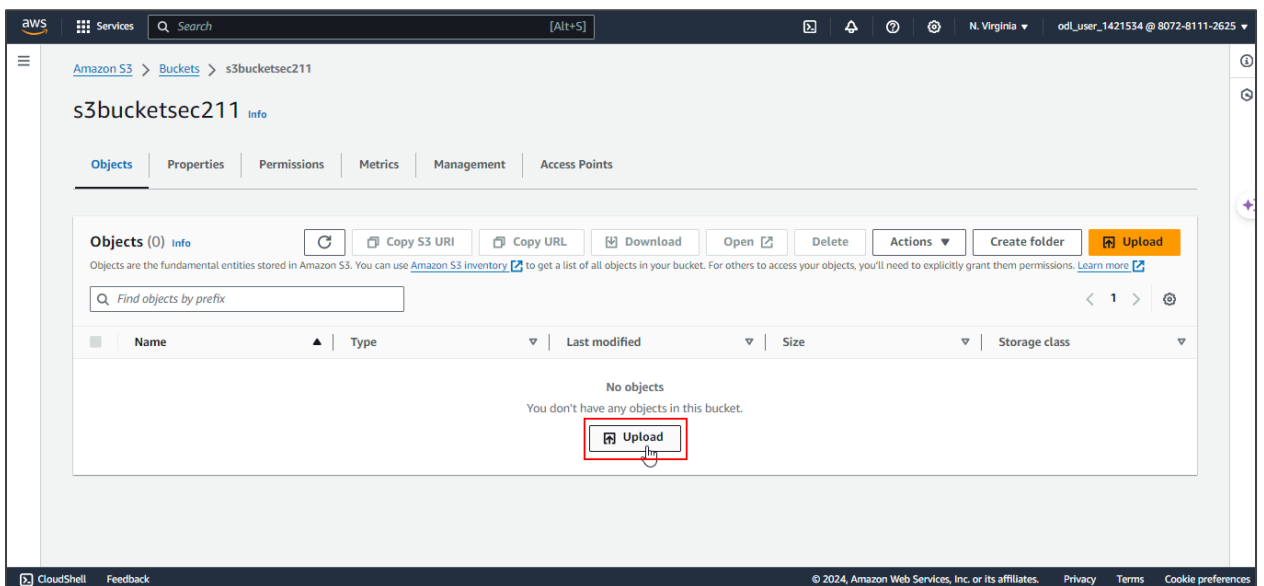
**Note:** Keep all the other options as default

### 3.4 Click on the **s3bucketsec2** bucket to verify

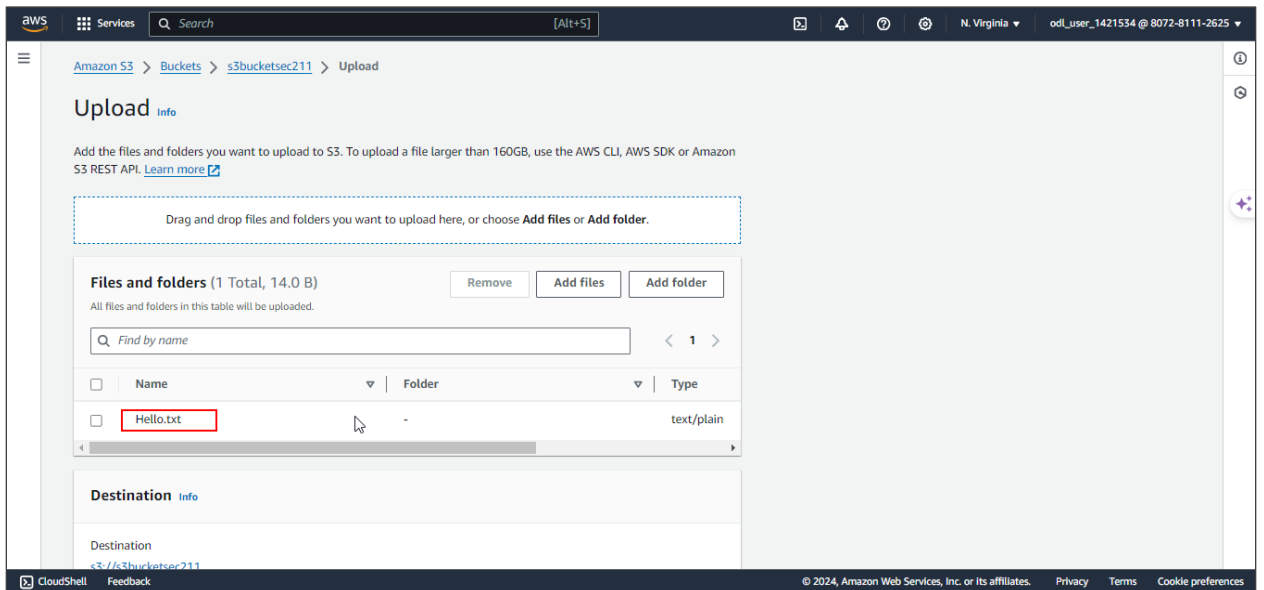
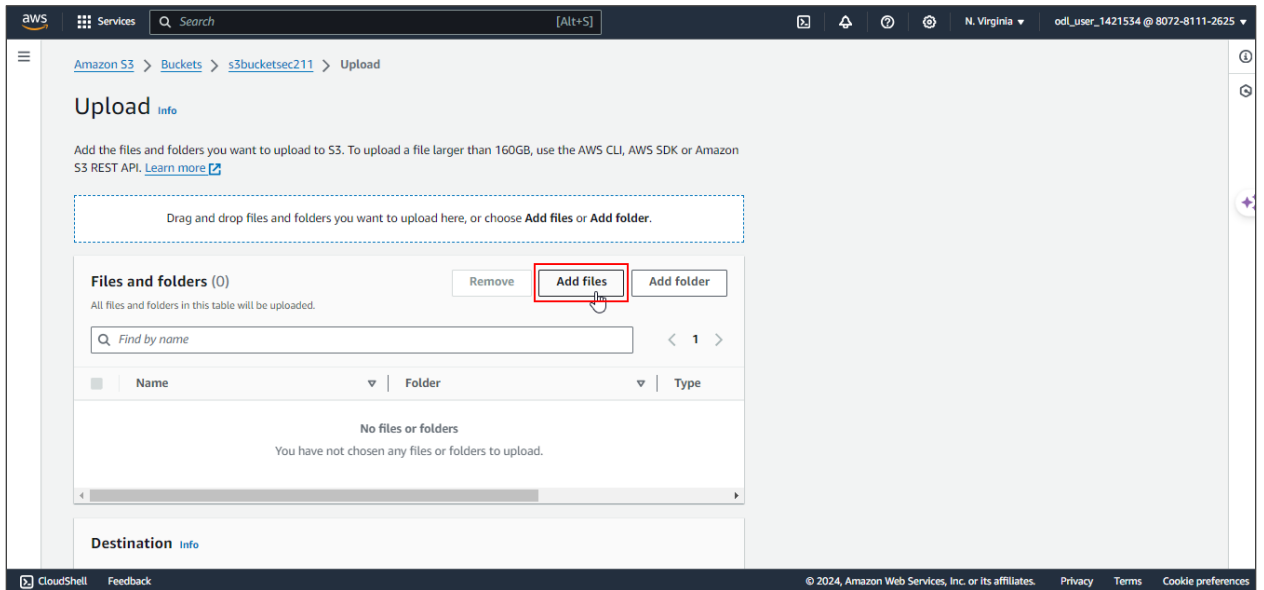


**Note:** Upload a **.txt** file to the S3 bucket

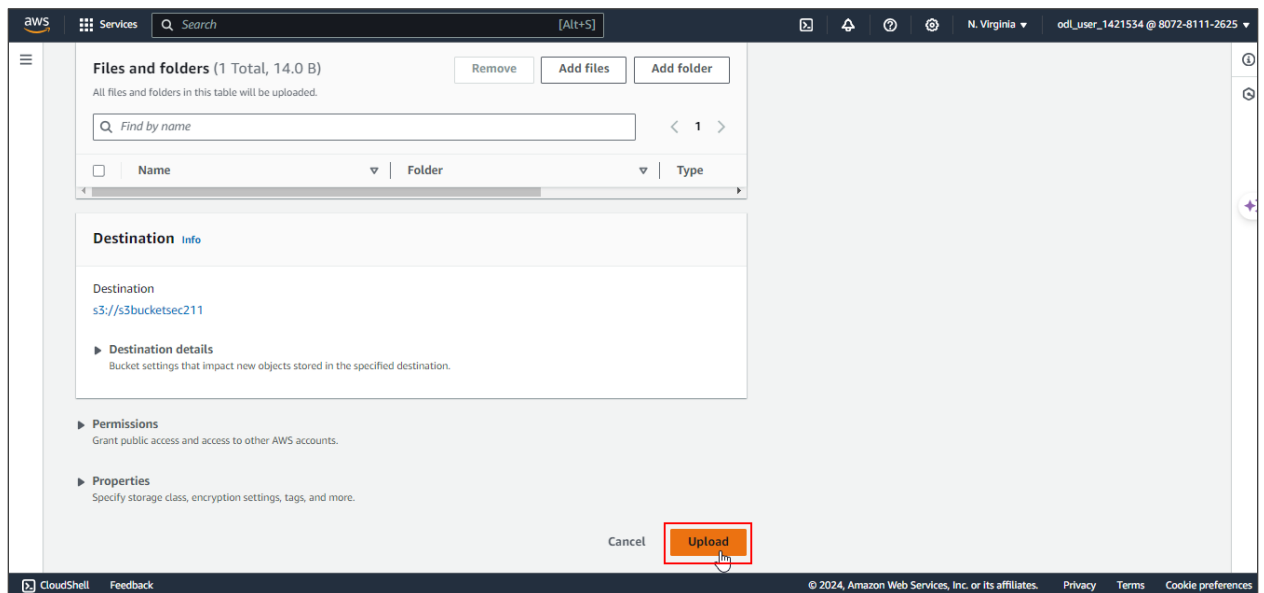
### 3.5 Click on **Upload**



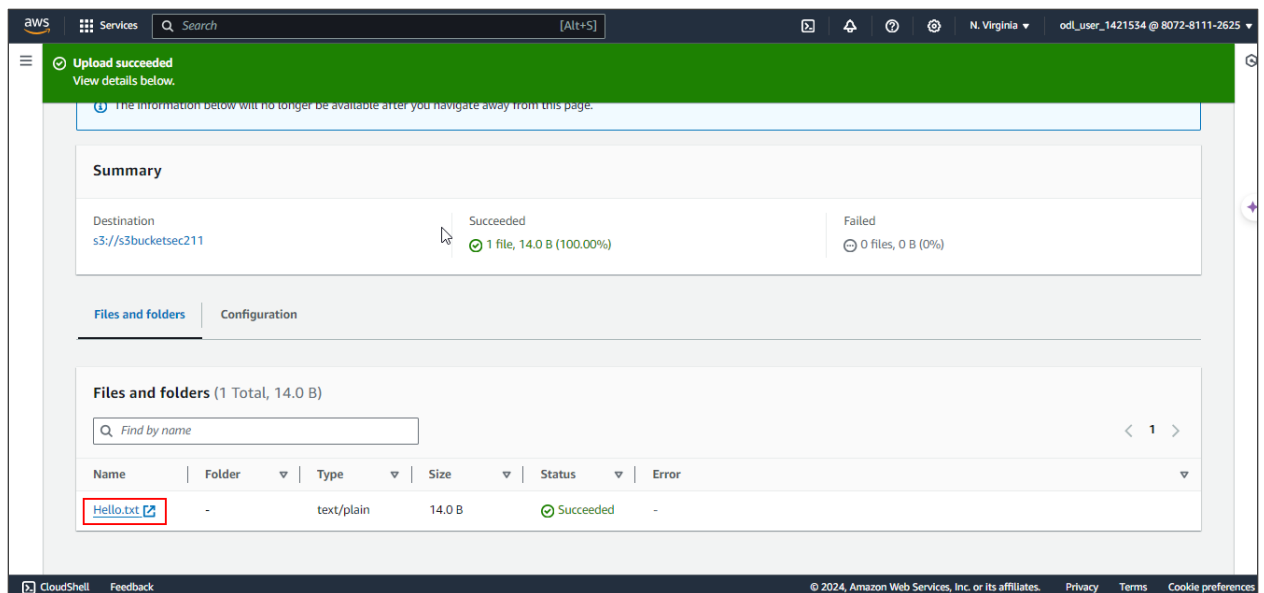
### 3.6 Click on **Add files** as shown and select the desired **.txt** file from your operating system



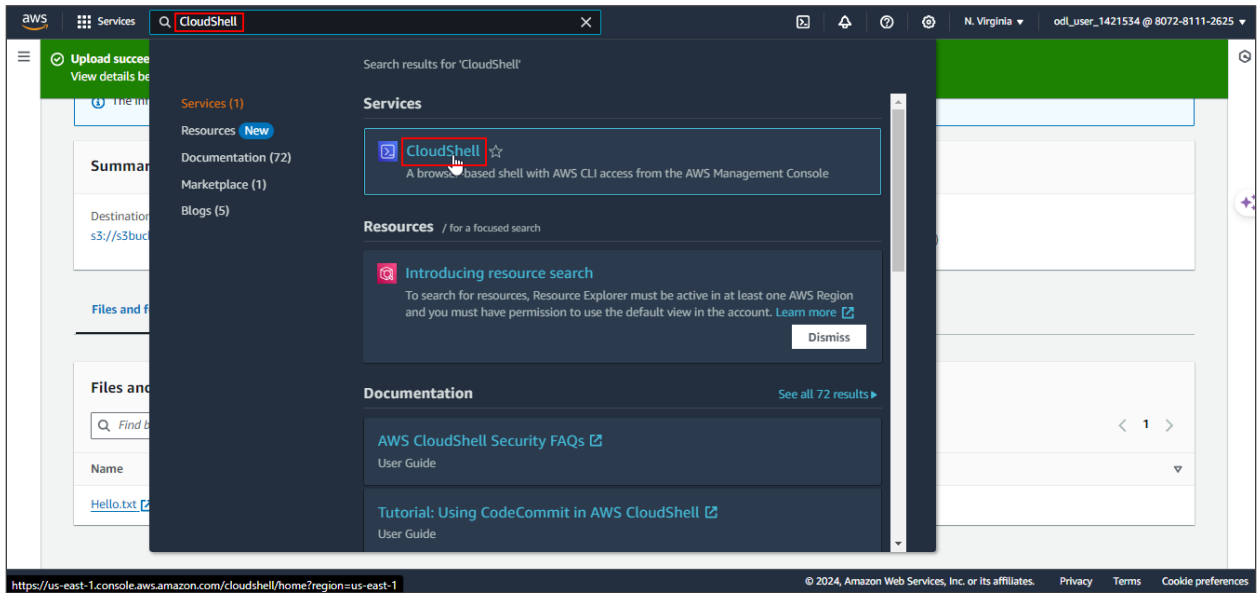
### 3.7 Click on the **Upload** button as shown:



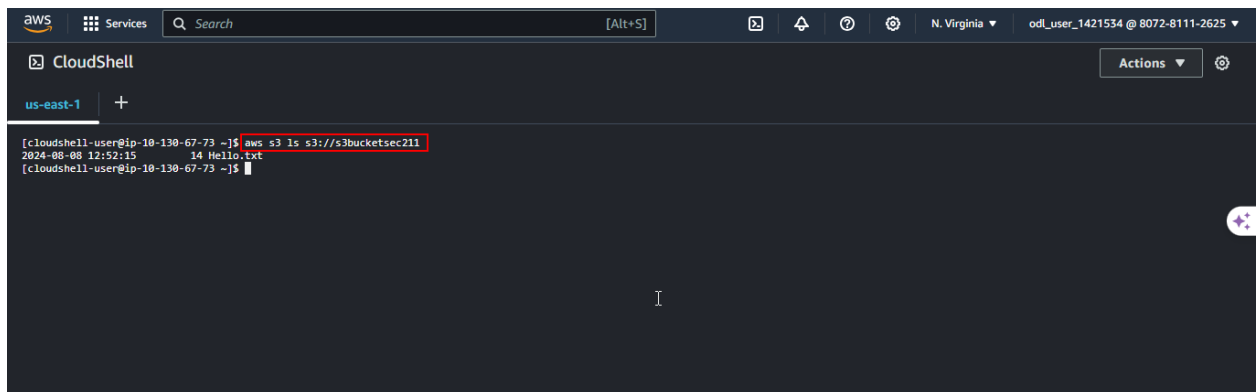
The file is uploaded successfully, as shown:



3.8 Search for and click on **CloudShell** as shown:



3.9 Add the following command in the terminal:  
**aws s3 ls s3://<YOUR\_BUCKET\_NAME>**



**Note:** Replace <YOUR\_BUCKET\_NAME> with your bucket name

By following these steps, you have successfully secured access to Amazon S3 buckets from an EC2 instance using IAM roles.