**General tips for the Task below.**

**Task Title**

**S3**- Simple Storage Service in AWS

**Objective**

Worked on pre-requisite, as I watched the S3-AWS video posted in DevOps group in slack and task to work on S3.

**Prerequisites**

**List all requirements before starting**:  
• S3 bucket and upload some objects  
• static website  
• cross-region replication  
• Bucket policy  
• Lifecycle policies  
• AWS CLI  
• Bash script  
• 1 GB file via CLI

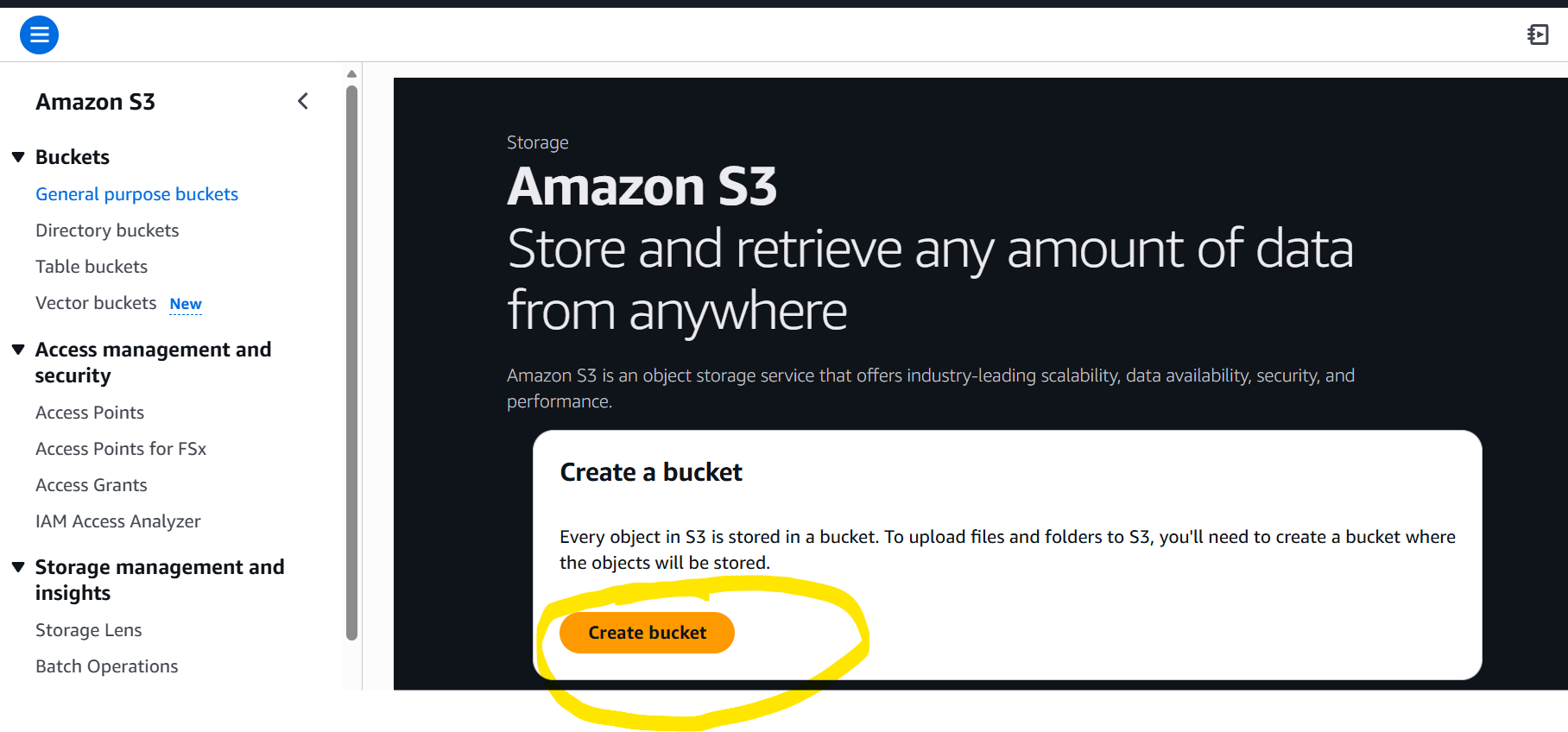
**Issues Faced** (Optional but Recommended) Briefly mention issues and how you fixed them.

**Conclusion**

Write 1–2 lines summarizing that the task was successfully completed.

1. **Create an S3 bucket and upload some objects to S3.**

* **Creating a S3 bucket** as we need to upload few objects.



* Choosing “**General purpose** bucket” for my purpose for this task.
* Named my bucket- **vicky-root-007**

A screenshot of a computer

AI-generated content may be incorrect.

* **If we have a existing bucket**, we can choose from the option and this selection is optional until and unless, we need to choose from existing one created.
* **Here, we are enabling ACLs,** because if we need to connect with other AWS services, we will be able to connect.

A screenshot of a chat

AI-generated content may be incorrect.

* **Block Public ACLs’ for this bucket**. The reason we are blocking now is, we do not want public to change any settings. But we will turn it on when we work on further tasks below.

A screenshot of a computer

AI-generated content may be incorrect.

* I have **turned on Bucket Versioning**, because that will help to identify, retrieve or restore every version of objects stored in my buckets.

A screenshot of a computer

AI-generated content may be incorrect.

* The below are **default** until and unless we need to change as per company requirements.

A screenshot of a computer

AI-generated content may be incorrect.

* **Created a bucket** with all my requirements.

A screenshot of a bucket

AI-generated content may be incorrect.

* Now, its time to upload some **objects.**
* Added **3 objects** and save.

A screenshot of a computer

AI-generated content may be incorrect.

* Now all objects are uploaded to my bucket- **vicky-root-007**

A screenshot of a computer

AI-generated content may be incorrect.

1. **Deploy a static website in the S3 bucket.**

* Selecting a bucket to perform this task- **vicky-root-007**

**A screenshot of a bucket list

AI-generated content may be incorrect.**

* Inside the bucket, select**- properties** and scroll down to **static website.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* Now **edit** the setting.

**A computer screen shot of a message

AI-generated content may be incorrect.**

* **Enable the static website hosting**, so we can work on the folders that needs to be created

**A screenshot of a computer

AI-generated content may be incorrect.**

* We have named the **index document** and **error document** as **index.html** and **error.html**
* The reason for **error document** is to work when **index document** is removed or not available.
* Then save.

**A white background with black and white clouds

AI-generated content may be incorrect.**

* Now the **static website hosting** requirements is complete.

**A screenshot of a computer

AI-generated content may be incorrect.**

* Now, we need to create an **index file and error file** in our local Gitbash terminal
* Using **vi index.html and vi error.html**, created the content to display.

**A screenshot of a computer program

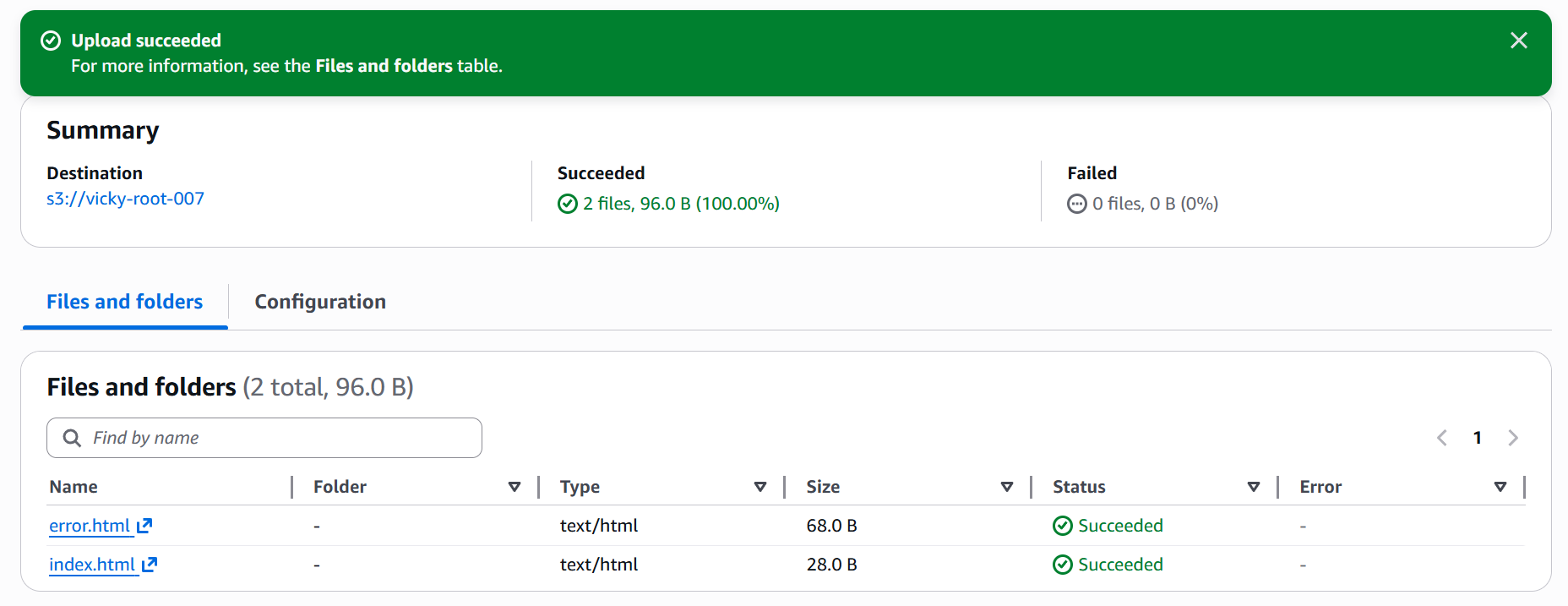
AI-generated content may be incorrect.**

* Uploaded the **index.html and error.html** files to my bucket under**- Objects.**

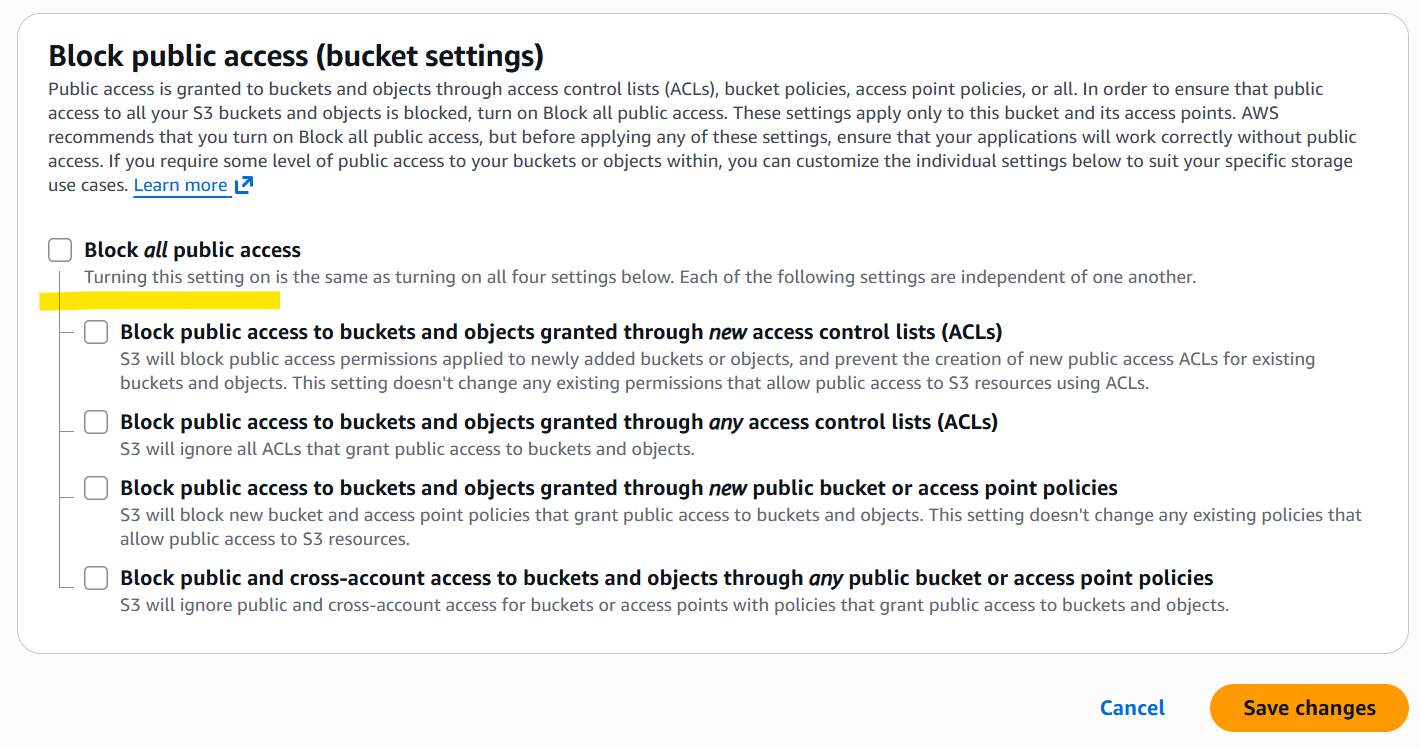
**A screenshot of a computer

AI-generated content may be incorrect.**

* **Successfully uploaded 2 objects to my bucket.**

****

* Now, its time to **Navigate to Permissions tab** in my bucket to update bucket policy.
* To **update bucket policy**, we need to **turn off the Block access**.
* The reason for **turning it off** is to **portray my files to browser**.

****

* **Under permissions tab-** uploaded a **Bucket policy** to **Public read my static website.**
* Copied the script from google and uploaded with my **bucket name- vicky-root-007.**
* **Then save.**

**A screenshot of a computer

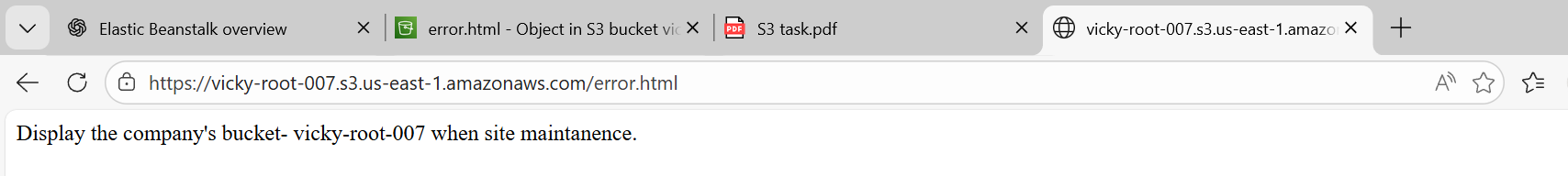
AI-generated content may be incorrect.**

* Now, navigate to **Objects tab** in bucket and click on **index.html file.**
* **Copy the URL and paste** in the **browser for static content** to display.

**A screenshot of a computer

AI-generated content may be incorrect.**

* Once the **URL is pasted in static browser**, we are able to see the content.
* Success.

****

* Now, lets remove the **index file** from **objects** and verify if the **error document** portrays.
* **Success**, we can see if the **index.html file** is removed and **error.html** page is active on the **static website.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Enable cross-region replication on S3 buckets.**

* **CRR-** Copying objects automatically from one bucket to another in different region.
* 1ST step is to check if we have **bucket** in our **default region.** I have 1 bucket in **N. Virginia** named **vicky-root-007.**
* 2nd step is to create a **new bucket** in another region. I have created a new bucket in **Stockholm region**.
* The below is my **N. Virginia bucket.**

A screenshot of a bucket

AI-generated content may be incorrect.

* The below is my **Stockholm region bucket**.
* Now, we can see **2 buckets** are in **2 different regions**.

A screenshot of a computer

AI-generated content may be incorrect.

* Now, we need to navigate to default region- **N. Virginia** and update the **bucket versioning to enabled**.

A screenshot of a computer

AI-generated content may be incorrect.

* The below is my default region- **N. Virginia, bucket versioning enabled.**

A screenshot of a computer

AI-generated content may be incorrect.

* The below is my **Stockholm region, Bucket versioning enabled.**

A screenshot of a computer

AI-generated content may be incorrect.

* **Set-up CRR rule**
* Navigate to **default region** and go to **Management tab** in bucket.
* Create replication rule.
* Named replication rule name- **my-replication rule**
* Status- **Enabled**

A screenshot of a computer

AI-generated content may be incorrect.

* **Rule scope**- Apply to all objects in the bucket.

A white background with black and white clouds

AI-generated content may be incorrect.

* **Destination**
* Account is the same, so same **account number.**
* **My other region bucket name entered** and destination region got automatically populated.

A screenshot of a computer

AI-generated content may be incorrect.

* We need to create a new I AM role.
* Now save with all default.

A screenshot of a computer

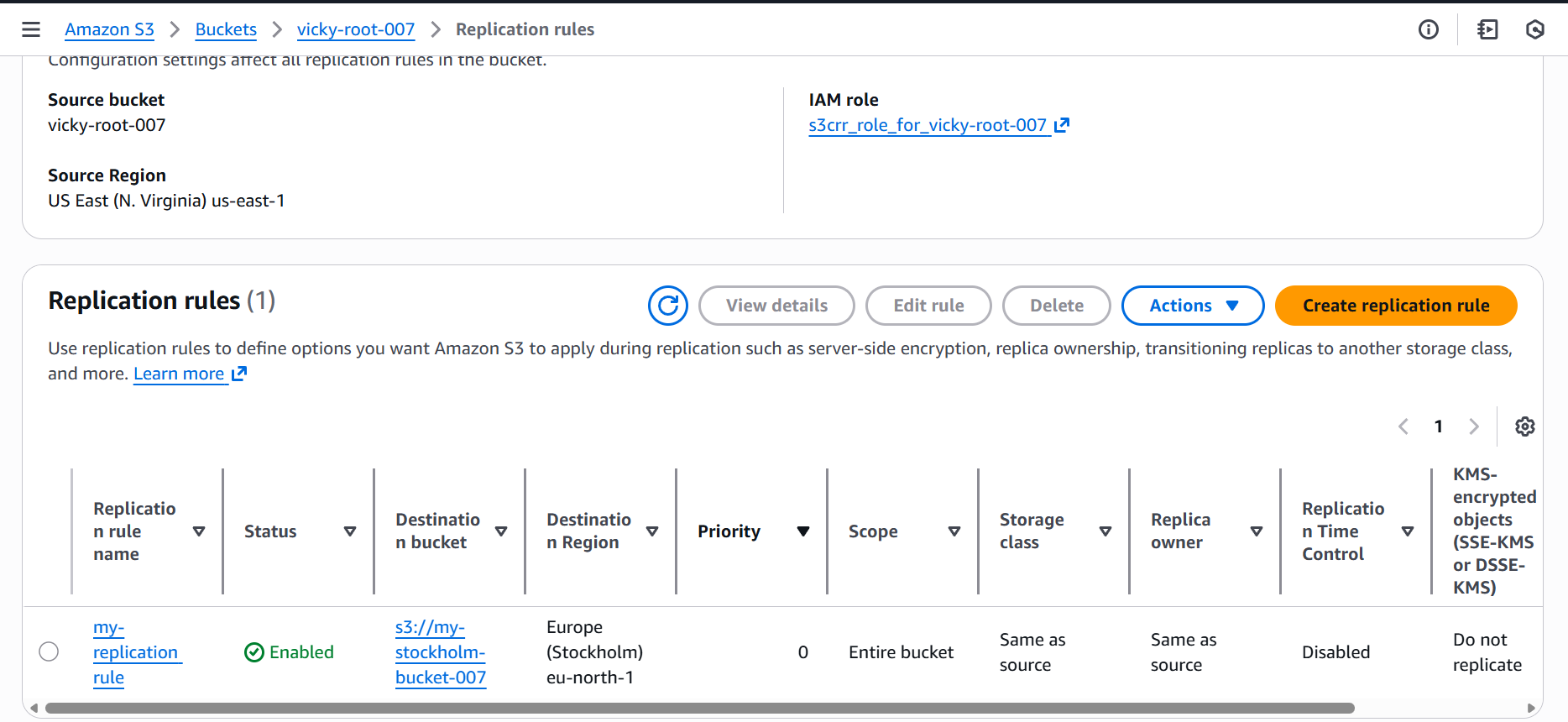
AI-generated content may be incorrect.

* Now, we will get the question populating as “Replace existing objects?”. Select submit.

A screenshot of a computer

AI-generated content may be incorrect.

* Now, we can see the **replication rule** is set.



* Now, everything is set. Let’s verify and test.
* I have created an object in my default region- **N. Virginia** and shall verify in **Stockholm bucket**.

A screenshot of a computer

AI-generated content may be incorrect.

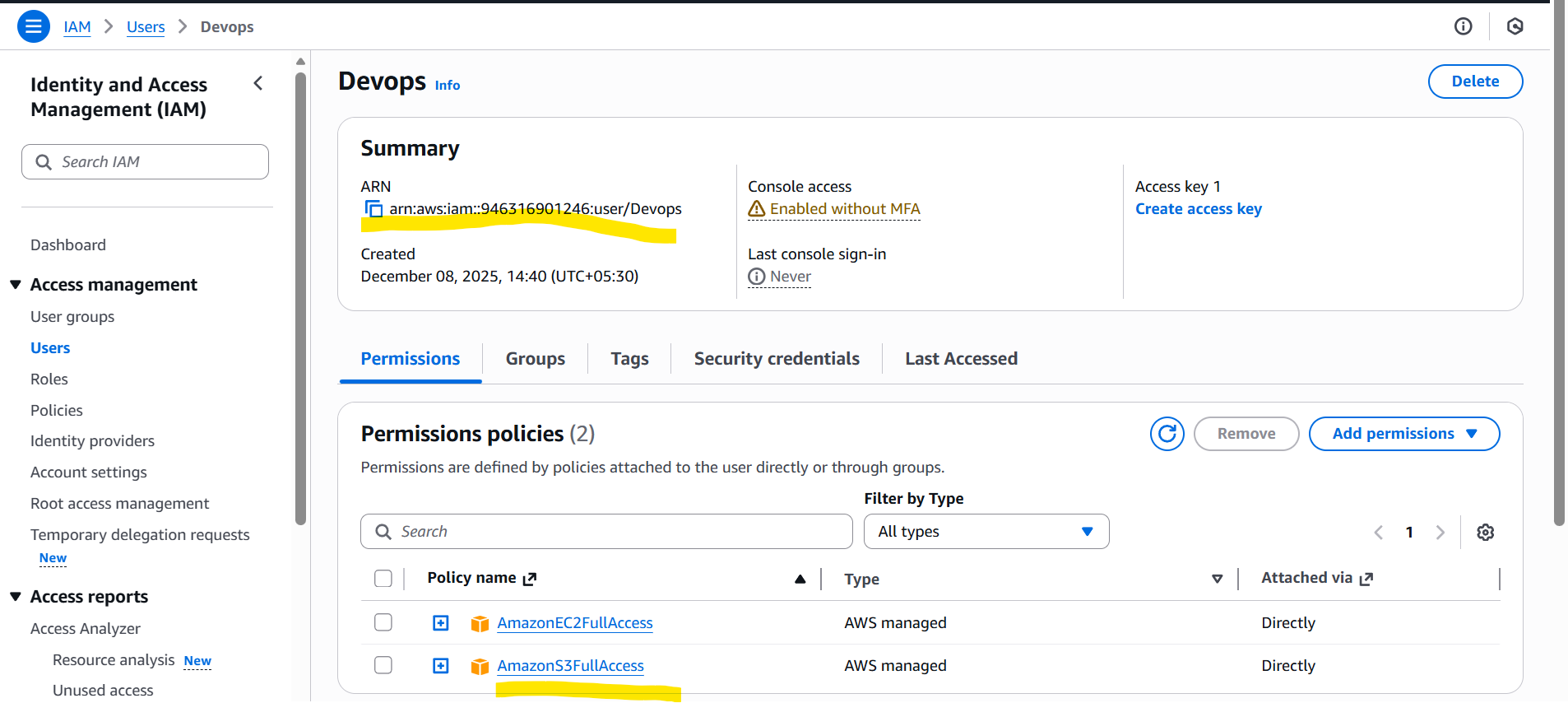
* **Success**. I can verify the **Stockholm region** bucket and can see the object present. Please see the below screenshot.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Configure a bucket policy so only the Admin user can see the objects of the S3 bucket.**

* **Steps to follow for this task.**
* This bucket will only be accessed by **Admin** and a **IAM user**.
* Rest everyone except **Admin and IAM user,** the access will be denied.
* **Admin user – IAM user ARN specify required.**
* We have an existing user named **“Devops”** and we are going to make this user to have **S3 full access** and have **access to bucket policy**.

****

* **Navigate** to **S3** and click **bucket 🡪permissions** and **Bucket policy** and let’s edit the policy through **json format.**
* Now, we have made **Devops** user the only user to **access the bucket** as per below screenshot by adding a script under **bucket policy.**
* Even **root user** will not have permission to **access the bucket.**
* The below policy will allow only **Devops user** to access the **vicky-root-007** bucket in S3, **even root user** will not have access to the bucket.

**A screenshot of a computer

AI-generated content may be incorrect.** **A computer screen with text

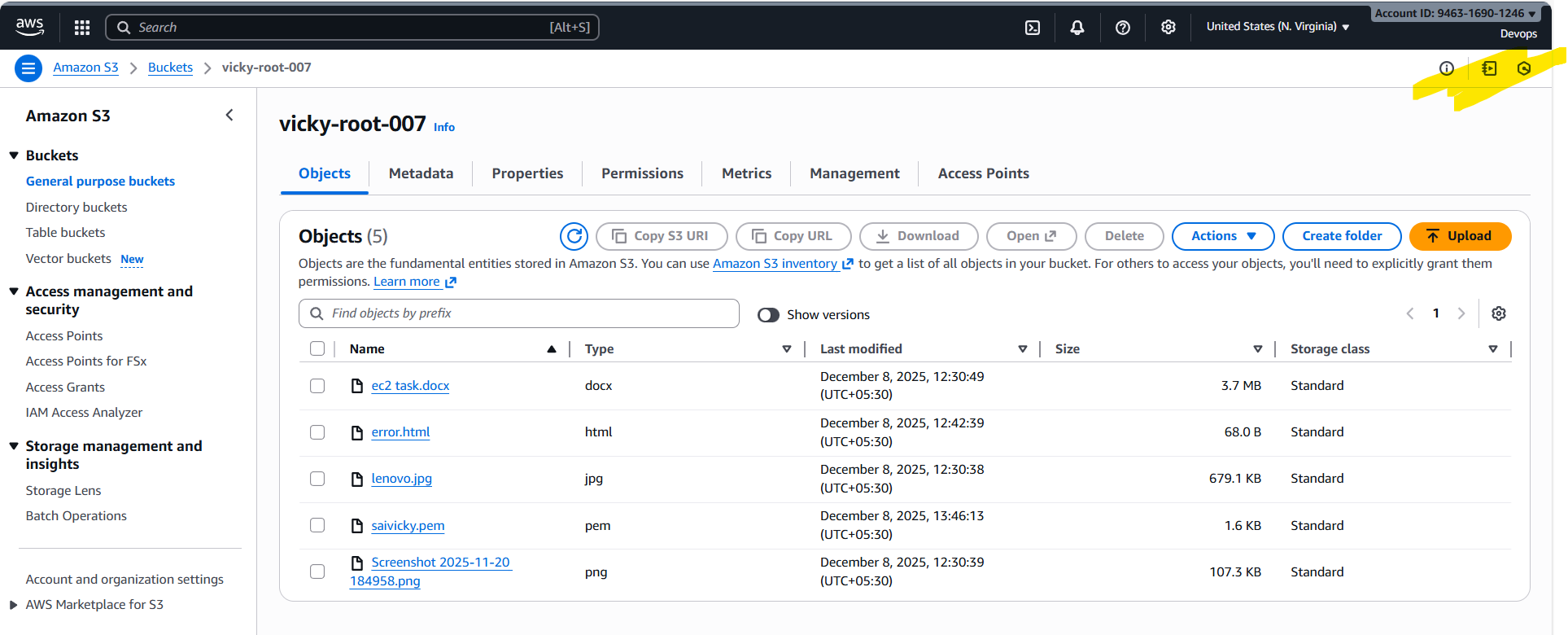
AI-generated content may be incorrect.**

* In the below screenshot, I can confirm that, even myself as a **root,** got no access to updating **s3 bucket.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* Success, in the below screenshot, we can see the user **“Devops”** has the **full access to s3 bucket.**

****

* Success, we tried to change the **IAM user and** verified that, the new user cannot access any **buckets/objects** in **s3.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Set up lifecycle policies to automatically transition or delete objects based on specific criteria.**

To set up S3 Lifecycle Policies so objects automatically *transition* to cheaper storage tiers or get *deleted* after a specified time, we configure Lifecycle Rules inside our S3 bucket.

Lifecycle policies are useful for cost optimization, archival, and automated cleanup.

**🔧 1. Configure Lifecycle Rules via AWS Console**

**Steps:**

1. Open S3 Console
2. Select your Bucket
3. Go to Management Tab
4. Click Create lifecycle rule
5. Specify:
   * Rule name
   * Filters (prefix/folder, tags, or entire bucket)
6. Choose Actions
   * Transition storage class (Standard → IA → Glacier)
   * Set object expiration (auto-delete)
   * Delete previous versions (if versioning on)
   * Expire incomplete uploads

**Example Lifecycle Settings**

| **Goal** | **Action** |
| --- | --- |
| Move to S3 Standard-IA after 30 days | Transition rule |
| Move to Glacier after 90 days | Archive rule |
| Permanently delete after 365 days | Expiration rule |
| **This is the most common lifecycle rule used in projects.** |  |

* **Steps to navigate**
  + S3
  + Buckets
  + Existing bucket
  + Management
  + Lifecycle configuration

**A screenshot of a computer

AI-generated content may be incorrect.**

* Created a new rule name as **Delete-objects-olderones**
* Rule scope- **Applying to all objects**
* **Acknowledged.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **Actions-** Selected **Expire current versions of objects after 30 days.**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **Created rule** as per below screenshot.

**A screenshot of a computer

AI-generated content may be incorrect.**

* **We have created a lifecycle rule where-** 
  + All the objects are deleted from the bucket after 30 days.
  + Success.

1. **Push some objects to S3 using the AWS CLI.**
2. **Write a Bash script to create an S3 bucket.**
3. **Upload a 1 GB file to S3 using the CLI.**