Table of Contents

Table of Figures

1 Introduction

1.1 Overview of GreenTech Alliance

The GreenTech Alliance is a non-profit organization with a core mission dedicated to advancing environmental technologies and sustainability. Their primary objective is to promote the development and adoption of green technologies to mitigate the impacts of climate change and safeguard the environment.

1.2 Overview of the Problem and Goals

The specific problem at hand is the need for efficient image processing within the GreenTech Alliance. To address this, the project aims to leverage cloud redundancy and automate image processing. The overarching goal is to enhance the technological capabilities of the organization, allowing for more effective utilization of satellite image data.

1.3 Specific and General Objectives

The specific objectives include configuring two AWS storage services and implementing a serverless Lambda function for automatic image resizing and storage.

The general objectives involve optimizing image processing workflows for the GreenTech Alliance, fostering efficiency and reliability.

1.4 What is Cloud Migration?

Cloud migration involves the process of moving digital assets, such as data, applications, and services, from on-premises infrastructure to cloud-based solutions. It is a strategic move that organizations undertake to benefit from the scalability, flexibility, and cost-effectiveness offered by cloud platforms.

1.5 Benefits of Cloud Migration

Cloud migration offers several advantages for the GreenTech Alliance:

- **Utilizing Redundancy for Satellite Image Processing**: Cloud redundancy provides a reliable and scalable solution for satellite image processing. By leveraging redundant cloud services,

the GreenTech Alliance ensures that critical data processing continues seamlessly even in the event of hardware failures or other disruptions.

- **Increased Scalability and Cost-effectiveness**: Cloud migration allows for increased scalability and cost-effectiveness when compared to traditional on-premises solutions. The GreenTech Alliance can scale its image processing capabilities based on demand, ensuring optimal resource utilization, and avoiding unnecessary costs.
- **Automated Image Processing**: The cloud infrastructure enables the automation of image processing tasks, reducing manual intervention, and streamlining operations. This automation enhances efficiency and allows the GreenTech Alliance to focus on its core mission of advancing environmental technologies.

1.6 Role of the Cloud Engineer in Building the Cloud Infrastructure:

The cloud engineer plays a crucial role in designing, implementing, and maintaining the cloud infrastructure for the GreenTech Alliance. Responsibilities include configuring AWS storage services, developing serverless functions, ensuring data security, and optimizing performance. The cloud engineer acts as a key facilitator in achieving the project's objectives by leveraging cloud technologies effectively.

2 Proposed Solution

GreenTech Alliance, an environmental technology startup focusing on climate technology services, is looking to upgrade its high-resolution satellite image processing program. This upgrade involves moving the program to the AWS cloud, which offers better reliability, flexibility, and efficiency.

Our solution brings together advanced AWS technologies to optimize how GreenTech Alliance handles and processes the satellite images. By doing so, this migration aims to make the environmental analysis and climate technology initiatives more effective and impactful.

This plan focuses on keeping operations running smoothly, making the best use of resources, and improving how data is handled. The goal is to show GreenTech Alliance's commitment to using the latest cloud tools to bring meaningful changes to environmental data analysis and climate technology services.

2.1 AWS Storage Infrastructure

The implementation of an effective and scalable storage infrastructure is paramount in ensuring the seamless handling and management of the significant volume of satellite images processed by GreenTech Alliance. Our plan centers around leveraging AWS Storage Services to craft a resilient and purpose-built storage architecture that addresses the specific needs associated with high-resolution raw images and their corresponding resized versions.

2.1.1 AWS Storage Services Overview

AWS offers a diverse array of storage options, each tailored to specific use cases and performance requirements. In the context of GreenTech Alliance's satellite image processing migration, the following AWS storage services are identified as pivotal components of the proposed storage infrastructure:

1. Amazon S3 (Simple Storage Service): Amazon S3 provides scalable and highly durable object storage designed to accommodate a wide range of data types. It offers secure and seamless access to data while ensuring high levels of redundancy and reliability.

2. Amazon EBS (Elastic Block Store): Amazon EBS delivers block-level storage designed for use with Amazon EC2 instances, providing persistent and low-latency storage volumes for mission-critical applications.

3. Amazon EFS (Elastic File System): Amazon EFS offers scalable and fully managed file storage suitable for use cases requiring shared access to files in a distributed environment.

2.1.2 Configuration for Efficient Data Management

The core focus of our plan revolves around configuring dedicated storage locations within the Amazon S3 service, catering to the distinct requirements of managing and processing high-resolution satellite images. This configuration will encompass specific designations for raw images and their resized counterparts, facilitating organized and efficient data management practices. The two primary configurations are as follows:

1. Storage Location for Raw High-Resolution Images: This designated storage area within Amazon S3 will serve as the repository for the high-resolution raw images received from the

satellite. This storage location is essential for securely housing the original, unaltered image data, preserving data integrity and enabling subsequent processing stages.

2. Storage Location for Resized Images: A separate storage location within Amazon S3 will be established to accommodate the resized and optimized versions of the high-resolution images. This dedicated storage area will house the processed images, ensuring streamlined access and efficient utilization for environmental analysis and climate technology services.

2.1.3 Addressing Specific Storage Needs

The implementation of these distinct storage configurations provides a tailored approach to managing satellite image data effectively. This includes addressing the storage needs of both the original high-resolution images and the resulting resized images, enabling seamless access and retrieval for analysis and processing tasks. By carefully crafting this storage infrastructure, we aim to ensure optimal data management and accessibility for GreenTech Alliance's environmental technology initiatives.

The detailed configuration and allocation of storage resources within the AWS cloud platform will cater to the specialized needs of handling high-resolution satellite images, fostering efficient data processing, and analysis. This approach is fundamental in empowering GreenTech Alliance with a robust and scalable storage infrastructure that aligns with the core objectives of the cloud migration initiative.

2.2 Overview of Serverless (Lambda) Function

In the proposed cloud migration solution for GreenTech Alliance, the utilization of a serverless function through AWS Lambda is identified as a foundational component in automating the image processing and storage workflows. AWS Lambda, known for its event-driven and serverless computing capabilities, presents a scalable and cost-effective solution perfectly tailored to the requirements of the cloud migration initiative.

2.2.1 AWS Lambda

AWS Lambda is a cutting-edge serverless computing platform that enables the execution of code in response to a diverse range of events, thereby eliminating the need for provisioning or managing

servers. This paradigm shift in computing presents a highly scalable and cost-effective approach to executing tasks and processes in the cloud environment.

Key aspects of AWS Lambda that make it a pivotal technology choice for the cloud migration solution include:

1. Event-Driven Model: AWS Lambda's event-driven architecture allows seamless execution of code in response to various triggers or events. This aligns perfectly with the automation needs associated with image processing and storage tasks, enabling efficient and timely execution of workflows.

2. Scalability: Lambda's inherent scalability ensures that computing resources are automatically allocated based on the incoming workload, thereby eliminating the need for manual provisioning or capacity planning. This ensures that the image processing tasks can scale effortlessly in response to varying workloads.

3. Cost-Effectiveness: AWS Lambda follows a pay-as-you-go pricing model, where charges are incurred only for the compute time consumed during code execution. This cost-effective approach minimizes overhead costs, making it an attractive solution for the cloud migration initiative.

2.2.2 Automation of Image Processing and Storage Tasks

The deployment of a serverless function using AWS Lambda is particularly advantageous in facilitating the automation of image processing and storage tasks. As satellite images are uploaded to the designated storage locations within the AWS cloud platform, the Lambda function becomes the catalyst for initiating subsequent processing and storage operations.

The Lambda function, triggered by the arrival of new images, seamlessly executes the necessary code to resize and optimize the images before transmitting them to the secondary storage location. This automated workflow significantly reduces manual intervention and streamlines the entire image processing pipeline, thereby enhancing operational efficiency and enabling timely access to the processed images for environmental analysis.

2.2.3 Advantages for GreenTech Alliance's Cloud Migration Initiative

By leveraging AWS Lambda, the cloud migration initiative for GreenTech Alliance benefits from unparalleled flexibility, scalability, and cost-effectiveness in automating critical image processing

and storage workflows. The seamless execution of event-driven code and the absence of underlying server management bolsters the overall efficiency and agility of the cloud infrastructure, aligning with the core requirements of the satellite image processing program.

The adoption of AWS Lambda as the serverless function technology of choice underscores the commitment to harnessing cutting-edge cloud computing capabilities, ensuring a streamlined and efficient approach to image processing automation within the cloud environment.

2.3 Serverless Function Objective

The central goal of crafting a serverless function using AWS Lambda within the cloud migration solution for GreenTech Alliance is to automate the image processing and storage workflow, providing a seamless and efficient approach to managing the high-resolution satellite images. The serverless function acts as the cornerstone of the automation strategy, enabling a streamlined and event-driven approach to resizing, optimizing, and managing the satellite images within the AWS cloud environment.

2.3.1 Automation of Image Processing and Storage Workflow

The primary objective of the serverless function is to automate the critical image processing and storage workflow, enabling the efficient transformation and management of the high-resolution satellite images. To achieve this, the serverless function is carefully designed to trigger automatically upon the upload of an image to the first designated storage location within the AWS environment. This event-driven approach ensures that the resizing and optimization tasks are initiated seamlessly and expeditiously as new images are deposited into the storage, effectively kickstarting the automated image processing workflow.

2.3.2 Seamless Resizing and Optimization for Environmental Analysis

Upon triggering, the serverless function seamlessly executes the necessary code to resize and optimize the high-resolution satellite images, preparing them for subsequent analysis and usage. This automated workflow minimizes manual intervention and accelerates the image processing pipeline, ensuring that the optimized images are efficiently transferred to the second storage location for streamlined accessibility and utilization. By automating the resizing and optimization tasks, the serverless function facilitates a continuous and uninterrupted image processing

workflow, ultimately enhancing operational efficiency and enabling timely access to the processed images for environmental analysis and climate technology services.

2.3.3 Anticipated Benefits for Environmental Analysis

The proactive execution of the serverless function contributes significant advantages to the cloud migration initiative, particularly in the context of environmental analysis processes. By automating the resizing and optimization of satellite images, the cloud infrastructure becomes a catalyst for providing timely access to efficiently processed and optimized satellite images. This, in turn, empowers GreenTech Alliance with the essential resources needed to foster enhanced insights and analysis into environmental conditions, paving the way for advancements in environmental technology and climate analysis. With access to efficiently processed and optimized images, the organizational capabilities to derive impactful environmental insights are significantly amplified, underscoring the substantial value brought forth by the automated image processing workflow facilitated by the serverless function.

The implementation of the serverless function using AWS Lambda embodies the commitment to automation, operational efficiency, and technological advancement within the cloud migration initiative. By meticulously orchestrating the automation of image processing and storage tasks, GreenTech Alliance is poised to realize a transformative shift in the management and analysis of high-resolution satellite images, thereby driving meaningful contributions to environmental analysis processes and climate technology services.

3 Requirements

3.1 Explanation of the Two AWS Storage Services

In the context of the cloud migration initiative for GreenTech Alliance's high-resolution satellite image processing program, the implementation of two distinct AWS Storage Services plays a critical role in effectively managing and processing the vast volume of satellite images. These services are meticulously tailored to accommodate the specific storage needs associated with high-resolution raw images and their corresponding resized counterparts.

3.1.1 Storage Location One: High-Resolution Raw Photos

The first storage location within the AWS cloud infrastructure serves as the designated repository for securely housing the high-resolution raw photos received from the satellite. This storage area is meticulously designed to safeguard the original, unaltered image data, thereby preserving data integrity and enabling future processing and analytical tasks. By securely storing the high-resolution raw photos in this dedicated storage location, GreenTech Alliance ensures the availability and integrity of essential unprocessed image data for subsequent stages of analysis and processing.

3.1.2 Storage Location Two: Resized Images Repository

The second storage location in the AWS cloud environment is specifically allocated to house the resized and optimized images generated through the automated image processing workflow. This storage area serves as a pivotal repository for efficiently storing and accessing the processed satellite images, ensuring seamless retrieval and utilization for environmental analysis and climate technology services. By designating a distinct storage area for the resized images, GreenTech Alliance enables efficient and organized access to the optimized image data, underpinning an agile and streamlined approach to managing the processed images within the cloud environment.

The careful delineation and allocation of these distinct storage locations within the AWS cloud platform highlight the commitment to efficient data management and accessibility for GreenTech Alliance's environmental technology initiatives. By aligning the storage architecture to the specific needs of managing high-resolution satellite images, GreenTech Alliance ensures optimal data organization and accessibility critical for advancing environmental analysis processes and climate technology services.

This clear demarcation of storage locations within the AWS cloud platform supports the overarching objective of effectively managing and analyzing high-resolution satellite images, fostering operational efficiency, and enabling timely access to essential image data for GreenTech Alliance's environmental technology endeavors.

3.2 Description of the Desired Serverless Function Behavior

In the proposed cloud migration solution for GreenTech Alliance's high-resolution satellite image processing program, the serverless function implemented using AWS Lambda is meticulously

crafted to exhibit specific behaviors that underpin the seamless automation of image processing and storage workflows within the AWS cloud environment. The desired behavior of the serverless function encompasses two pivotal aspects, each contributing to the efficient processing and management of the high-resolution satellite images.

3.2.1 Automatic Triggering Upon Image Upload to the First Bucket

The serverless function is intricately designed to automatically trigger as soon as a new high-resolution image is uploaded to the first designated storage location within the AWS cloud environment. This event-driven behavior ensures that the image processing workflow is seamlessly initiated in response to the arrival of new image data, minimizing manual intervention and streamlining the processing pipeline. By automating the triggering process, GreenTech Alliance ensures a continuous and responsive approach to managing and processing the incoming satellite images, fostering agility and operational efficiency within the cloud infrastructure.

3.2.2. Resizing and Uploading the Optimized File to the Second Bucket

Upon automatic triggering, the serverless function executes the essential code to resize and optimize the high-resolution satellite images, preparing them for subsequent analysis and usage. Following the resizing and optimization tasks, the serverless function seamlessly proceeds to upload the processed and optimized images to the second designated storage location within the AWS cloud platform. This step ensures that the resized and optimized images are efficiently transferred to the secondary storage area, where they are readily available for further analysis, environmental insights, and climate technology services. This behavior streamlines the image processing pipeline, fostering expedited access and utilization of the processed images for impactful environmental analysis and technological initiatives.

The deliberate design of the serverless function with these specific behaviors encapsulates the commitment to automation, operational excellence, and technological advancement within the cloud migration initiative for GreenTech Alliance. By orchestrating the serverless function to exhibit automatic triggering and seamless resizing and upload behaviors, GreenTech Alliance is positioned to foster a transformative shift in the management and analysis of high-resolution satellite images, advancing significant strides in environmental analysis and climate technology services.

3.3 Importance of Resized Images for Environmental Analysis

The generation of resized images plays a pivotal role in enhancing environmental analysis processes within GreenTech Alliance's technological initiatives. The optimization and resizing of high-resolution satellite images carry profound importance, aligning with the overarching goals of streamlined data processing, efficient analysis, and the derivation of impactful insights into environmental conditions and changes.

3.3.1 Facilitation of Streamlined Data Processing

Resized images serve as the cornerstone for facilitating streamlined data processing within GreenTech Alliance's environmental technology initiatives. By optimizing the resolution and size of the satellite images, the resulting resized images are better suited for efficient handling and processing, thus enabling swift and agile data manipulation within the cloud environment. The streamlined nature of resized images ensures that computational and analytical tasks are executed with enhanced efficiency, contributing to accelerated processing and analysis workflows critical for environmental insights and climate technology services.

3.3.2 Enablement of Efficient Analysis and Insights

Optimized and resized images unlock the potential for efficient analysis and the derivation of meaningful insights into environmental conditions and changes. The reduced size and optimized format of the images provide a conducive environment for advanced analytical processes, fostering the extraction of valuable environmental data and trends. This efficient analysis, facilitated by resized images, empowers GreenTech Alliance to draw impactful insights and conclusions, thereby driving the advancement of environmental analysis capabilities and fostering meaningful contributions to climate technology initiatives.

The generation of resized images represents a pivotal enabler for GreenTech Alliance's environmental analysis processes, paving the way for enhanced operational capabilities and technological advancements in the field of environmentally conscious data analysis. Through the meticulous optimization and resizing of high-resolution satellite images, GreenTech Alliance is positioned to elevate its capacity for efficient data processing, insightful environmental analysis, and the generation of impactful insights, thereby underpinning its commitment to environmental stewardship and technological innovation.

3.4 Explanation of How CloudWatch is Required for Monitoring

CloudWatch assumes a critical role in the monitoring and oversight of the health and performance of the AWS resources deployed within the context of GreenTech Alliance's cloud migration process. This comprehensive monitoring framework is essential for ensuring operational efficiency, identifying potential issues, and proactively addressing any bottlenecks that may arise within the cloud environment.

3.4.1 Monitoring the Health and Performance

CloudWatch facilitates the continuous and detailed monitoring of AWS resources, providing real-time insights into the operational health, performance metrics, and utilization patterns of various cloud components. Through the meticulous analysis of metrics such as CPU utilization, storage capacity, network performance, and application logs, CloudWatch offers a comprehensive observability landscape, ensuring that GreenTech Alliance gains a holistic view of the health and operational performance of their cloud infrastructure.

3.4.2 Proactive Maintenance and Issue Resolution

Regular and proactive monitoring of the cloud environment using CloudWatch enables GreenTech Alliance to anticipate and address potential issues before they escalate into critical challenges. By promptly identifying anomalous behavior, resource exhaustion, or performance degradation, CloudWatch empowers GreenTech Alliance's cloud engineering team to take proactive actions, initiate remediation, and maintain operational efficiency. This proactive approach is instrumental in minimizing disruptions and ensuring seamless operations within the cloud environment.

3.4.3 Efficient Resource Utilization and Cost Management

By leveraging CloudWatch insights, GreenTech Alliance gains the capability to optimize resource utilization, identify underutilized capacity, and enforce cost management measures. This comprehensive visibility into resource utilization patterns and operational health is instrumental in fostering efficient resource allocation and management, thereby ensuring optimal performance and cost-effective utilization of AWS resources within the cloud migration initiative.

The incorporation of CloudWatch within the cloud migration process for GreenTech Alliance underscores the commitment to establishing a robust and meticulously monitored cloud

environment. By harnessing the capabilities of CloudWatch for comprehensive monitoring and oversight, GreenTech Alliance ensures the maintenance of operational efficiency, proactive issue resolution, and optimized resource utilization. This proactive monitoring framework is pivotal in driving the successful and seamless harnessing of cloud technology for environmental analysis and climate technology advancement, underscoring the organization's commitment to leveraging cutting-edge cloud capabilities for transformative environmental initiatives.

4 Overview of the Proposed Cloud Architecture

Below is the architecture diagram of our project:

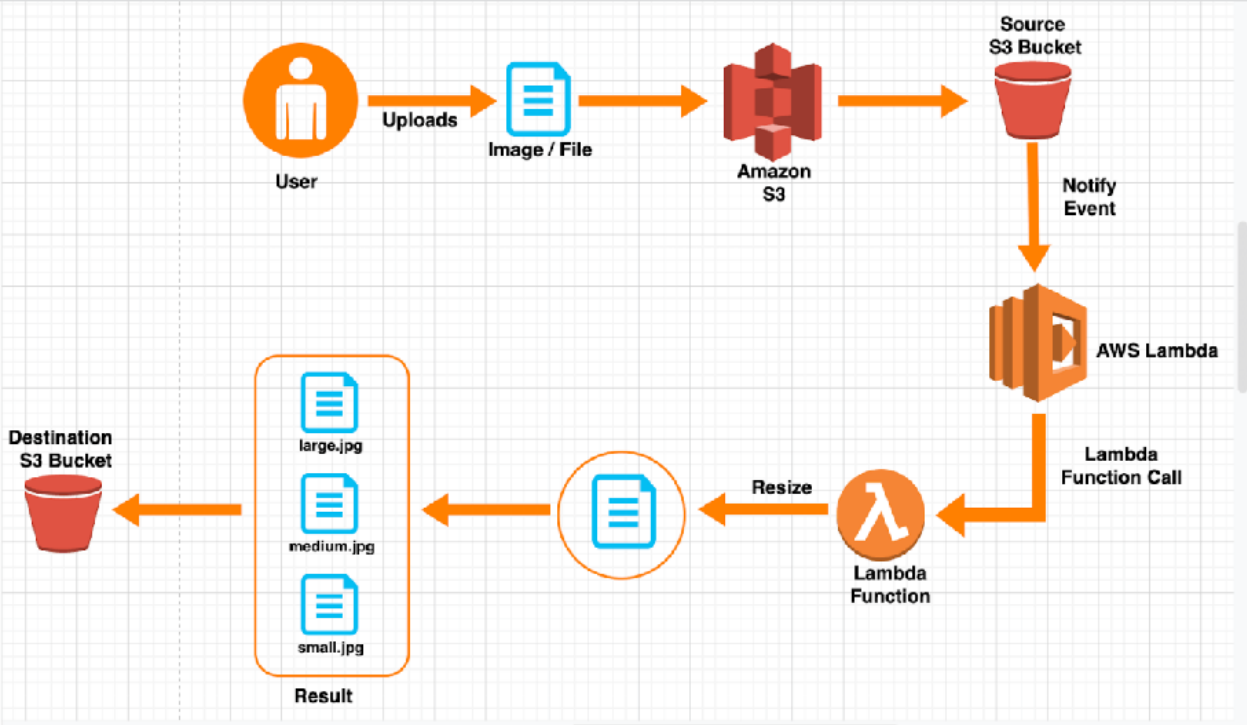

Figure 1: Architecture Diagram

4.1 Architecture Components

1. **Amazon S3**:

- **First Bucket (Raw Images):**
  - **Role:** Dedicated for storing high-resolution raw photos received from the satellite.
  - **Configuration:** Default settings or customized based on specific needs, with versioning and logging enabled.

- **Permissions:** Strictly controlled access via IAM roles to enhance security.

- **Second Bucket (Resized Images):**
  - **Role:** Dedicated for storing processed and resized images.
  - **Configuration:** Optimized settings for efficient storage utilization.
  - **Permissions:** IAM roles set up to control access, ensuring data security.

2. **AWS Lambda**: Facilitates serverless image processing and storage automation.

- **Purpose and Functionality:**
  - **Role:** Facilitates serverless image processing and storage automation.
  - I**mplementation:** Utilizes Node.js or Python with the `sharp` library for image processing.
  - **Trigger:** Configured to be automatically invoked when a new image is uploaded to the first S3 bucket.
  - **Scalability**: Scales seamlessly based on workload, ensuring efficiency during peak processing times.

4. CloudWatch

- **Role:**
  - **Monitoring:** Monitors Lambda function performance, S3 bucket activity, and potential issues.
  - **Logging:** Captures detailed logs for debugging, tracking, and performance analysis.
  - **Alarming:** Sets up alarms to notify administrators of critical events, ensuring proactive issue resolution.

4.2 Flow Overview

1. **Image Upload:**

- A high-resolution satellite image is uploaded to the "raw-satellite-images" S3 bucket.

3. **Lambda Trigger:**
   - The S3 bucket event triggers the Lambda function ("ProcessSatelliteImages").
4. **Lambda Execution:**
   - The Lambda function downloads the high-resolution image from "raw-satellite-images."
   - It processes the image (e.g., resizing) using the Sharp library.
   - The optimized image is then uploaded to the "resized-satellite-images" S3 bucket.

4. Processed Image Storage:

- The resized and optimized satellite image is stored in the "resized-satellite-images" S3 bucket.

4.3 Benefits

- **Scalability:** The serverless architecture allows automatic scaling based on the number of incoming images, ensuring efficient processing.
- **Redundancy:** AWS S3 provides data redundancy and durability, ensuring the safety of both raw and processed images.
- **Cost Efficiency:** With serverless computing, you pay for actual usage, making it cost-efficient for sporadic image processing tasks.
- **Automation:** The entire process is automated, triggered by image uploads, reducing manual intervention.
- **Environmental Impact:** GreenTech Alliance can leverage cloud services to process environmental data, contributing to sustainable practices and analysis.

This architectural setup provides a robust, scalable, and automated solution for processing and managing high-resolution satellite images in the cloud. Adjustments can be made based on specific requirements and considerations.

5 AWS Services Configuration

Below are the steps we followed to set up our two AWS S3 buckets as part of our mission to enhance GreenTech Alliance's environmental technology. We established the first bucket to store high-resolution raw photos received from the satellite and the second bucket for storing resized images, which are crucial for environmental analysis. This configuration is a critical component of our efforts to migrate the high-resolution satellite image processing program to the cloud, leveraging the redundancy advantages offered by cloud services. Additionally, we have demonstrated the creation of a Lambda function to process high-resolution satellite images upon their upload to the first bucket. The enhanced and processed images will then be seamlessly transferred to the second bucket through the utilization of this function.

5.1 Steps for Creating S3 Buckets

1. Open the AWS Management Console and Navigate to S3 service

In the AWS Management Console, go to the "Services" dropdown and then select "S3" under the "Storage" category. Alternatively, simply search for "S3" in the search bar after logging into the console.

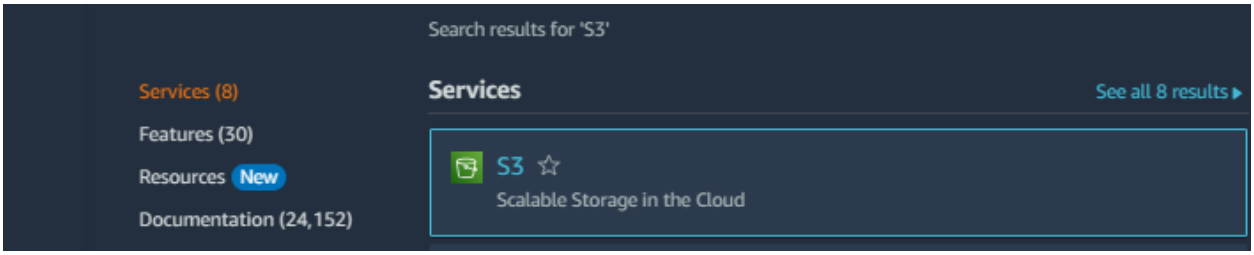

Figure 2: Searching for S3 on AWS Management Console

2. Click on "Create Bucket"

In the S3 dashboard, click the "Create Bucket" button.

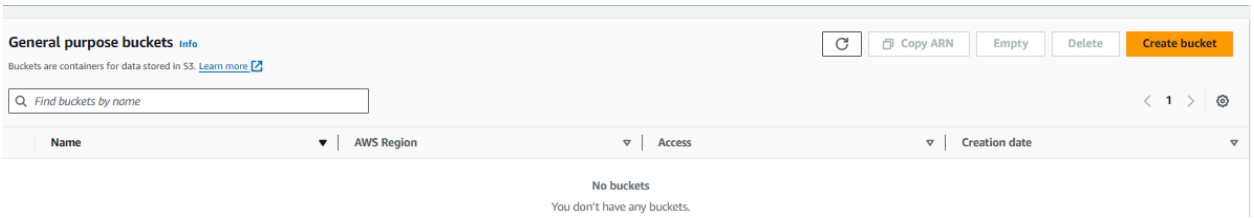

Figure 3: Navigate to S3 dashboard

3. **Configure Bucket** – Below are the various bucket configurations that are applied to both buckets.

A. **General Configuration**

- AWS Region – for the region we have used US East (N. Virginia) us-east-1.
- Bucket Name – We have named our first bucket venus-images-bucket and the second bucket venus-images-resized-bucket
- Bucket Type – There are two types of Amazon S3 buckets, general purpose buckets and directory buckets.

1. General purpose buckets are the original S3 bucket type and are recommended for most use cases and access patterns. General purpose buckets also allow objects that are stored across all storage classes, except S3 Express One Zone.
2. Directory buckets use the S3 Express One Zone storage class, which is recommended if the application is performance sensitive and benefits from single-digit millisecond PUT and GET latencies.

In this project, we have chosen the general purpose buckets, as they are recommended and allow a mix of storage classes that redundantly store objects across multiple AZs.

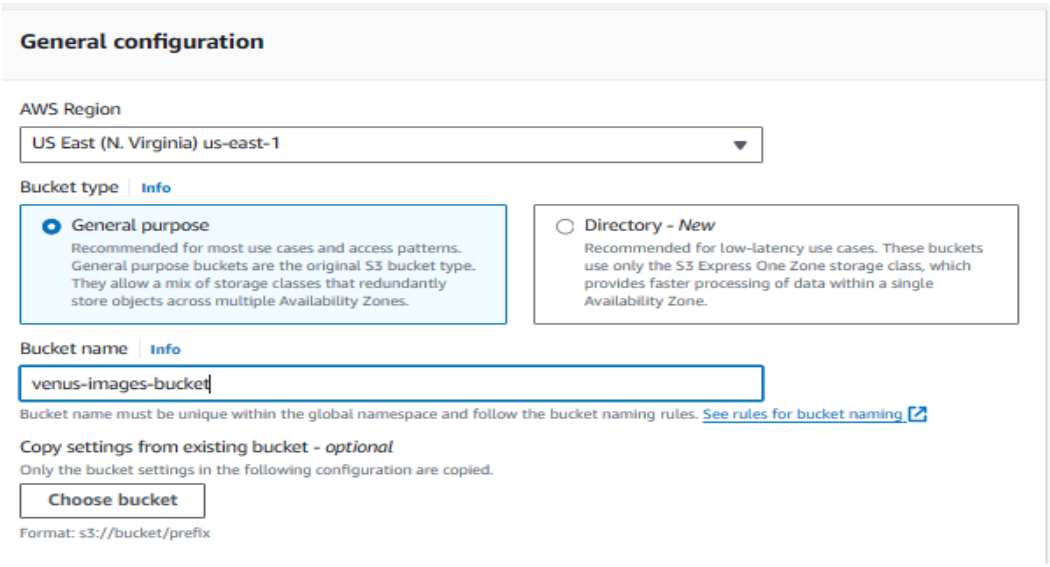

Figure 4: General configuration for bucket 1

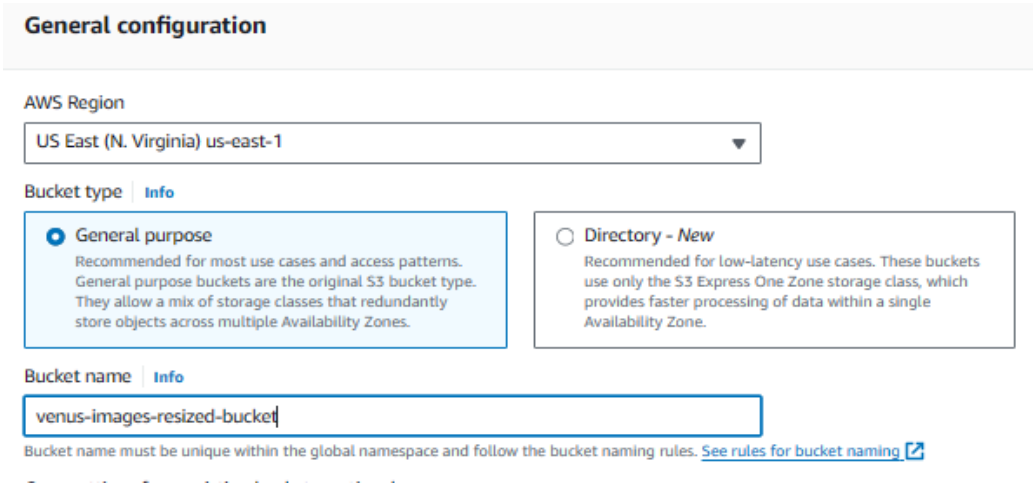

Figure 5: General configuration for bucket 2

B. **Object Ownership** – S3 Object Ownership is an Amazon S3 bucket-level setting that is used to control ownership of objects uploaded to bucket and to disable or enable access control lists (ACLs).

1. ACLs disabled – By default, Object Ownership is set to the Bucket owner enforced setting and all ACLs are disabled. When ACLs are disabled, the bucket owner owns all the objects in the bucket and manages access to data exclusively using access management policies.
2. ACLs enabled – The bucket owner owns and has full control over new objects that other accounts write to the bucket with the bucket-owner-full-control canned ACL.

In this project, we have chosen the ACL-disabled option as we do not want objects in our bucket to be owned by other AWS accounts.

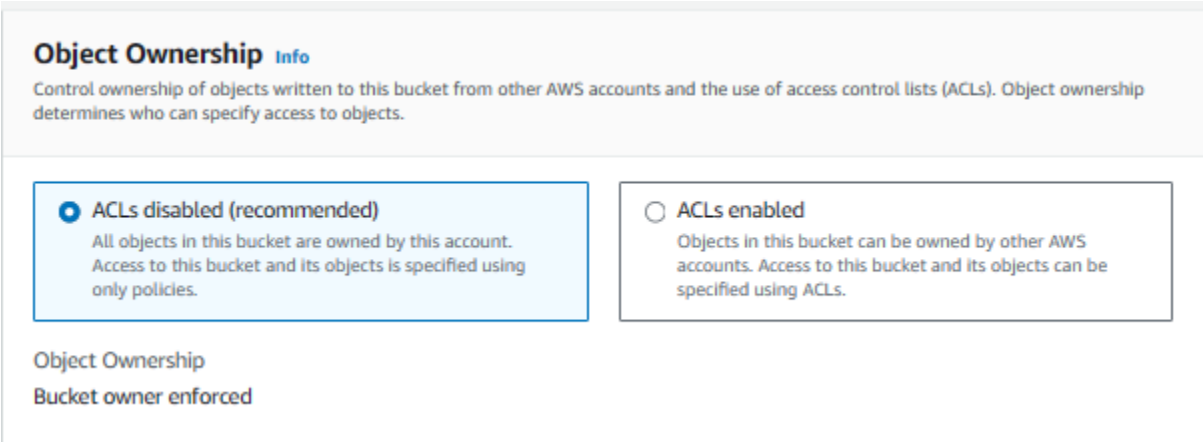

Figure 6: Object ownership

C. **Set permissions (Block or unblock all public access)** – Considering the nature of the GreenTech Alliance's environmental data processing, we have blocked public access to the AWS S3 buckets created for this purpose. This precautionary measure is essential to safeguard the confidentiality and security of the high-resolution satellite images and resized data stored within the buckets. By restricting public access, we ensure that unauthorized users and the general public cannot retrieve or view sensitive environmental data. This approach aligns with compliance requirements, addressing potential concerns related to data privacy laws and industry standards. Additionally, blocking public access helps maintain the integrity of the environmental data and mitigates privacy issues associated with exposing detailed satellite imagery to unauthorized individuals.

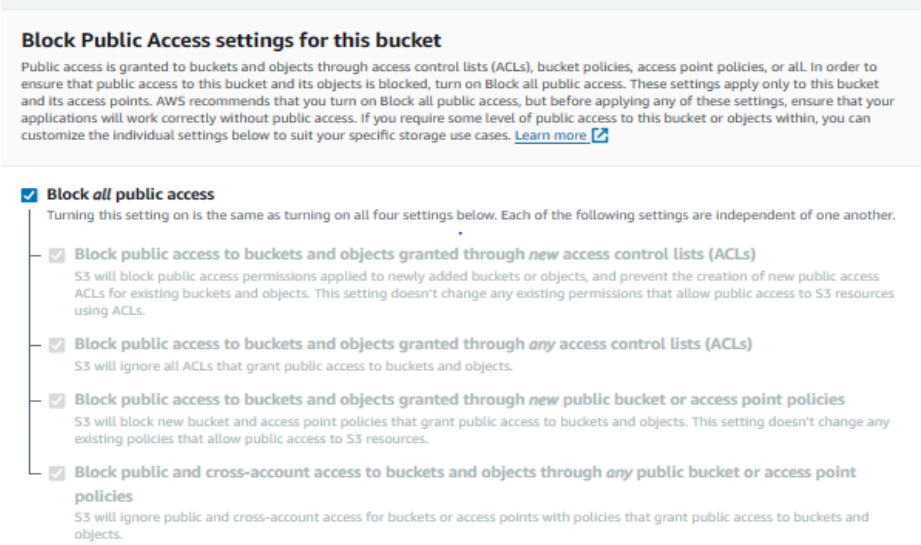

Figure 7: Block public access settings for buckets

D. **Bucket Versioning** – Versioning in Amazon S3 allows the storage of multiple versions of an object in a bucket, each assigned a unique identifier. While versioning enhances data durability and safeguards against accidental deletions, there are scenarios where it might be preferable to disable this feature. Disabling versioning can lead to reduced storage costs, simplified object management, and avoidance of unintended version accumulation. For these reasons, we have disabled versioning while creating our buckets.

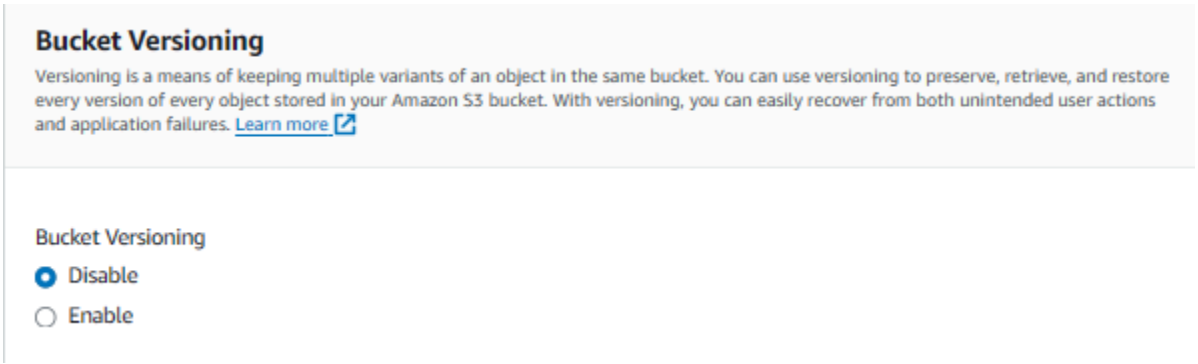

Figure 8: Bucket Versioning

E. **Default encryption**

➢ Encryption type – below the three types of encryption and the specific type used are discussed.

1. Server-side encryption with Amazon S3 managed keys (SSE-S3) – involves automatically managing encryption keys within Amazon S3 to secure data stored in the bucket. We have

implemented the Server-side encryption with Amazon S3 managed keys (SSE-S3) in this project due to its ease of use and cost-effectiveness, aligning with our priorities for a straightforward and budget-friendly data encryption solution.

2. Server-side encryption with AWS Key Management Service keys (SSE-KMS) –involves using the AWS Key Management Service to manage encryption keys, providing an additional layer of control and security for data stored in an Amazon S3 bucket.
3. Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS) – adds an extra layer of security by employing two separate layers of encryption, enhancing data protection in an Amazon S3 bucket.

➢ Bucket key – refers to a unique key generated by S3 to facilitate server-side encryption with SSE-KMS, we have left its default value as "Enable."

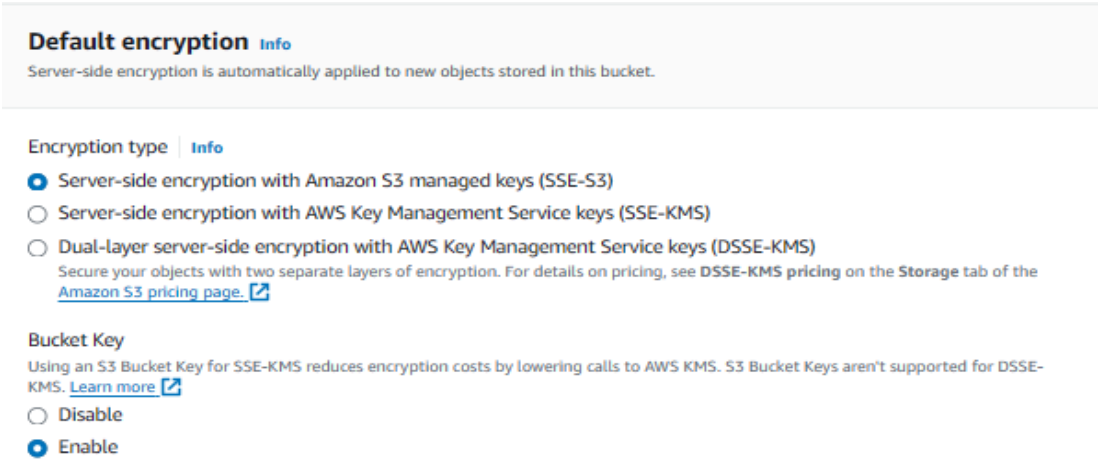

Figure 9: Default encryption

F. **Advanced Settings** – Object Lock is a feature in Amazon S3 designed to enforce retention periods on stored objects within a bucket. Given that Object Lock functions solely in versioned buckets (which we have disabled, as discussed above), we have kept it disabled.

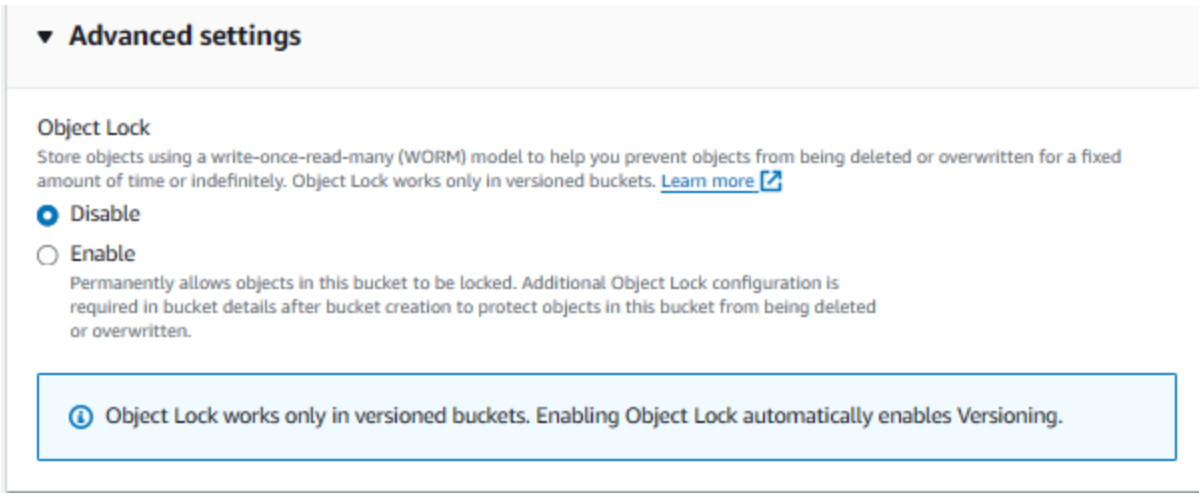

Figure 10: Advanced settings

4. **Review and create -** Review configuration settings and click the "Create bucket" button to create the S3 bucket.

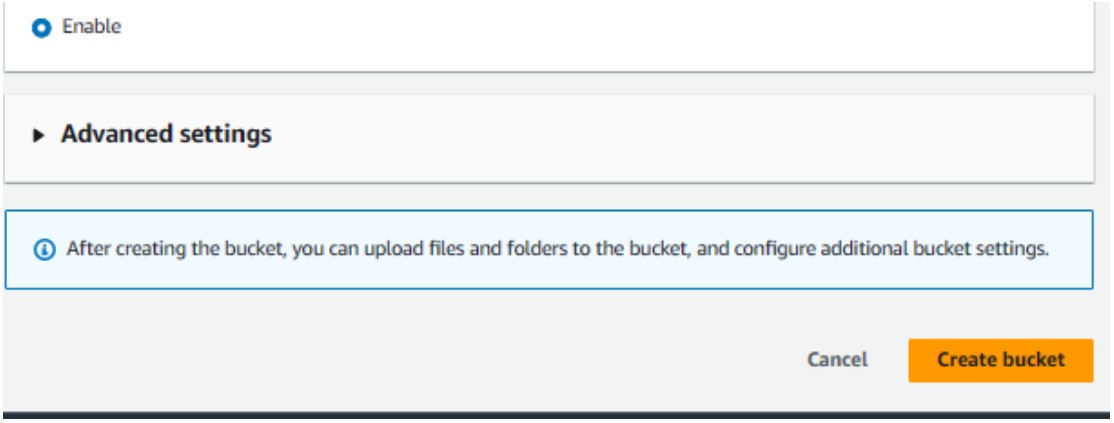

Figure 11: Create bucket

5. **Accessing the Buckets -** Once the buckets are created, it can be accessed by navigating to the S3 dashboard, selecting the newly created bucket from the list, and exploring its contents.

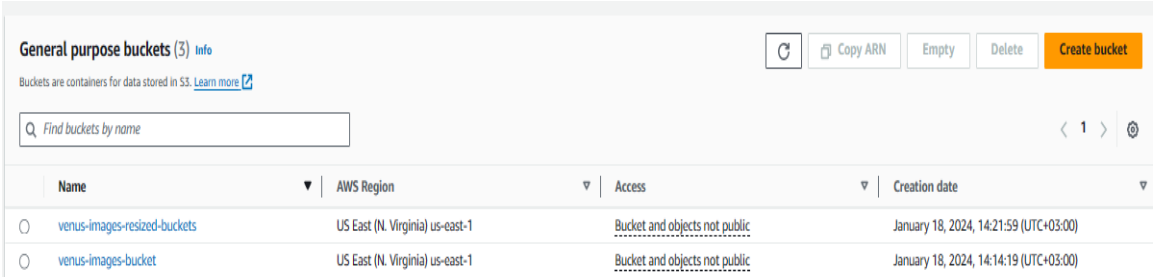

Figure 12: Buckets are successfully created

6. **Upload an image on the first bucket** – Select the "venus-images-bucket." Once inside the desired bucket, choose the "Upload" button. A file upload dialog will appear; use it to pick the image file from the local device. After selecting the image, click "Upload" to initiate the file transfer to the S3 bucket. Upon successful upload, the image will be visible in the bucket, and its properties can be configured as needed.

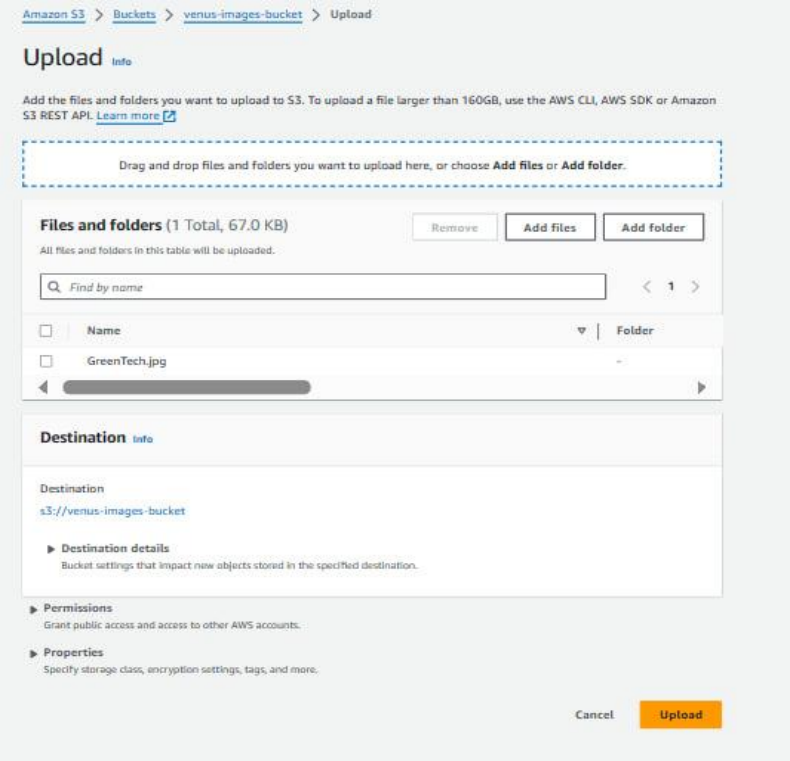

Figure 13: Uploading an image to the first bucket

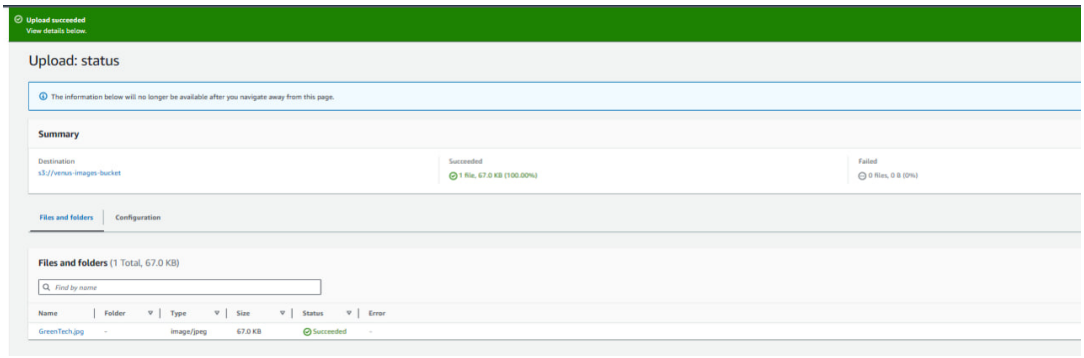

Figure 14: Successfully uploaded an image

To view the image, click on the image, then click on the "Open" tab; the image will open in a new tab.

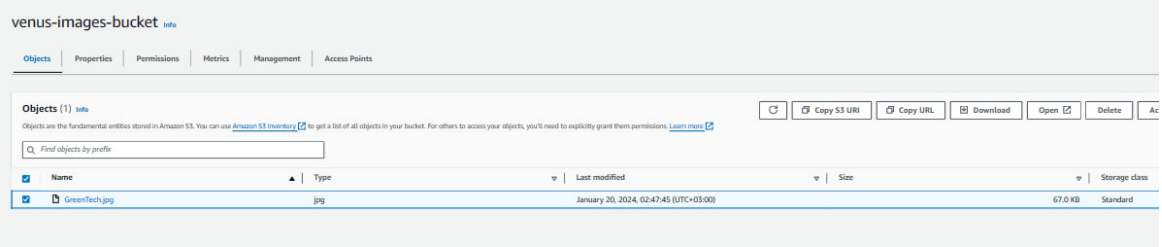

Figure 15: Click on open tab

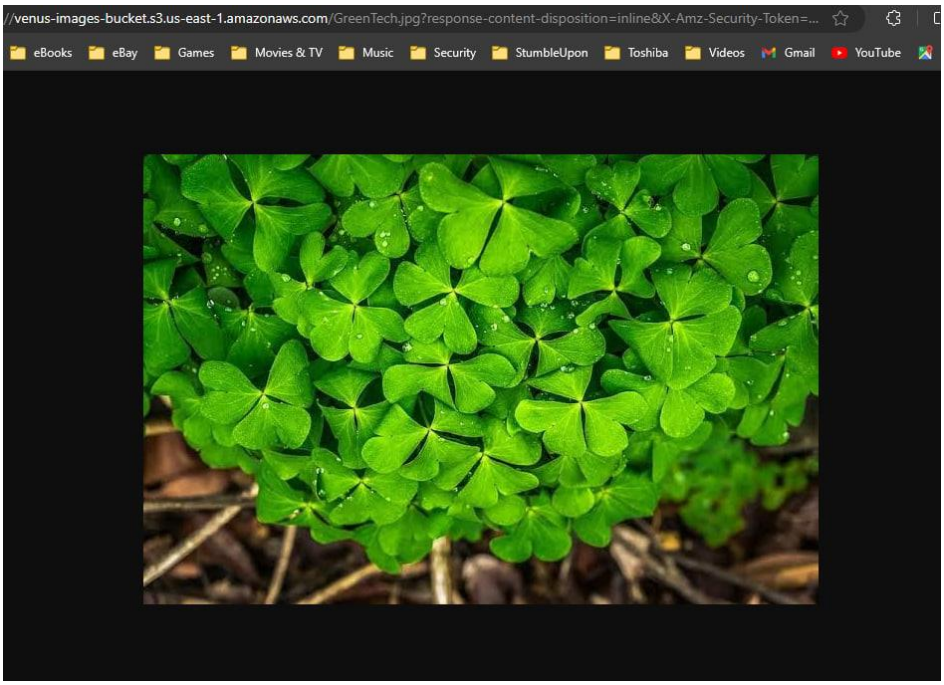

Figure 16: Viewing uploaded image

5.2 Steps for Creating AWS Lambda Function

1. **Open the AWS Management Console and Navigate to AWS Lambda -** Once signed in, navigate to the "Lambda" service. It can be found in the "Compute" section of the AWS Management Console. Alternatively, simply search for "Lambda" in the search bar after logging into the console.

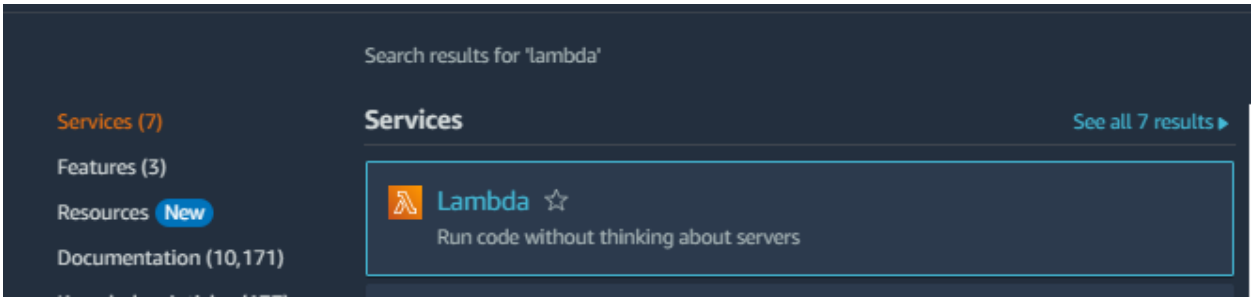

Figure 17: Searching for Lambda on AWS Management Console

2. **Click on "Create function"** – In the Lambda dashboard, click the "Create function" button.

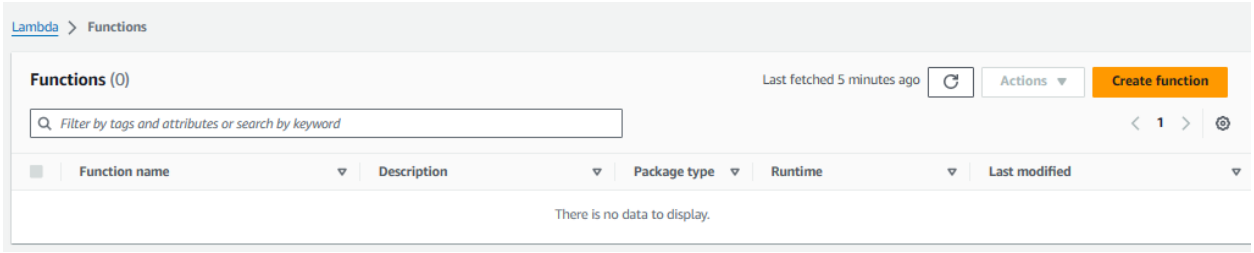

Figure 18: Click on create Lambda function

3. **Choose Author from Scratch** – the "Author from scratch" in AWS Lambda means creating a new function without relying on existing templates. It provides the flexibility to define all aspects of the function, including its name, runtime, and execution role, allowing for a customized setup tailored to specific needs.

Figure 19: Author from scratch

4. **Configure function** – Below are the various configurations that are applied on our lambda function.

- Provide a name for Lambda function – We have named our lambda function “venus-image-resizer-lambda”
- Choose the runtime environment for lambda function – In In this project, Node.js 20.x has been chosen as the runtime environment. Node.js was selected due to its quick startup time, a rich ecosystem of npm packages, ability to enable rapid scaling, provide low-latency responses,

and facilitate easy integration of third-party modules. This choice enhances the development speed and versatility of Lambda functions.

- Set up the execution role (IAM role) for lambda function - This role defines the permissions the function has to access other AWS services (S3 bucket). To create a custom role, the following steps were followed.

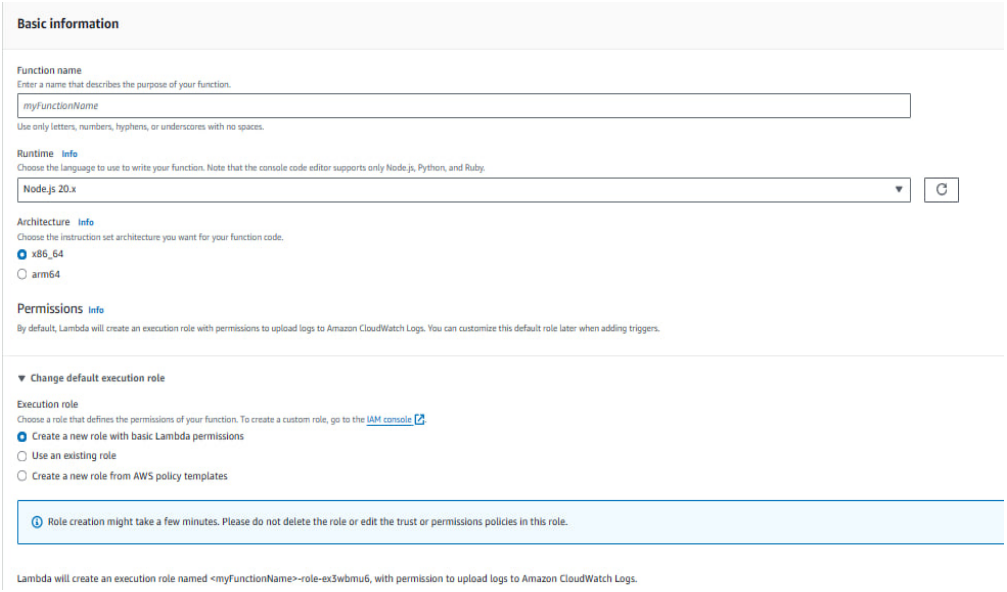

Figure 20: Basic information

Click on the IAM console, and the subsequent page appears. For the Trusted entity, AWS service was selected, and Lambda was chosen under the use case to permit it to perform actions in the account.

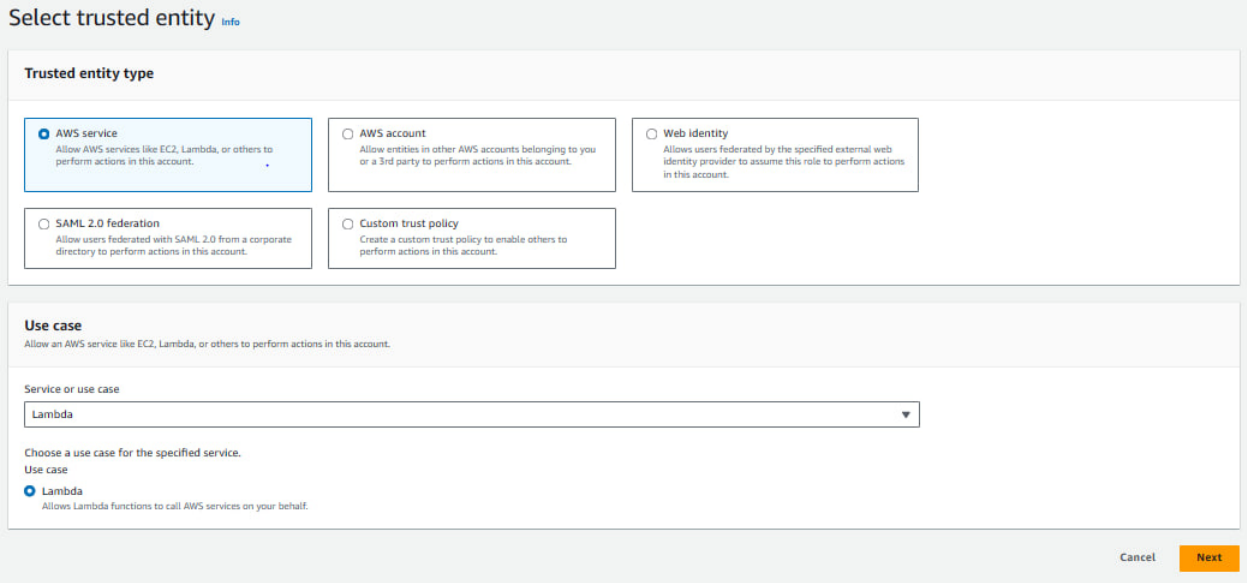

Figure 21: Select trusted entity and use case

Search for a policy in the search bar to attach it to the role. A policy defines a set of specific permissions. After opening policies in a new tab, the page below appears.

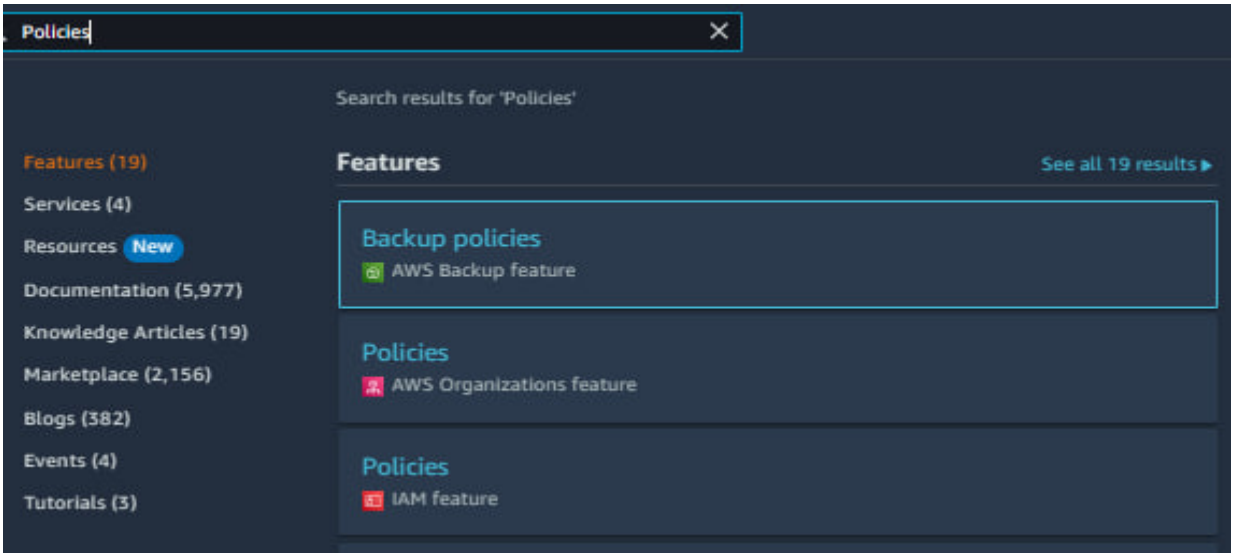

Figure 22: Search for policies on search bar

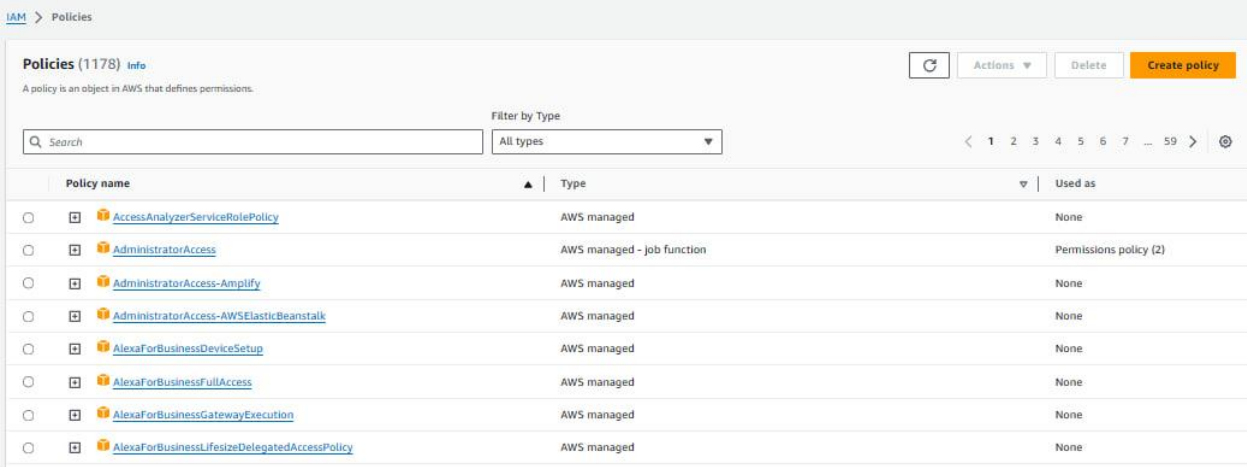

Figure 23: Policies

Then click on "Create Policy." Afterward, click on the JSON tab and paste the custom policy. This custom policy grants the Lambda access to send logs to CloudWatch, read access to the first bucket (the bucket to store raw images), and write access to the second bucket (the bucket to store resized images). The "/*" added at the end of both buckets refers to every object within the bucket.

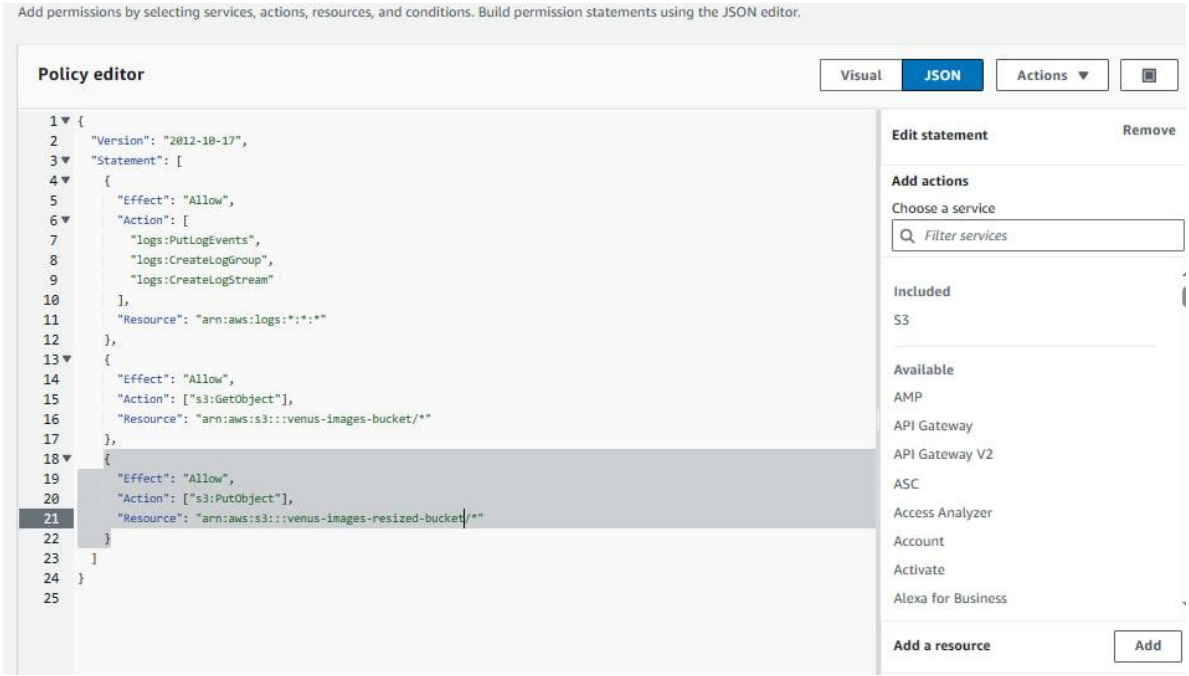

Figure 24: Policy editor

Click on "Next" and give the policy a name as shown below.

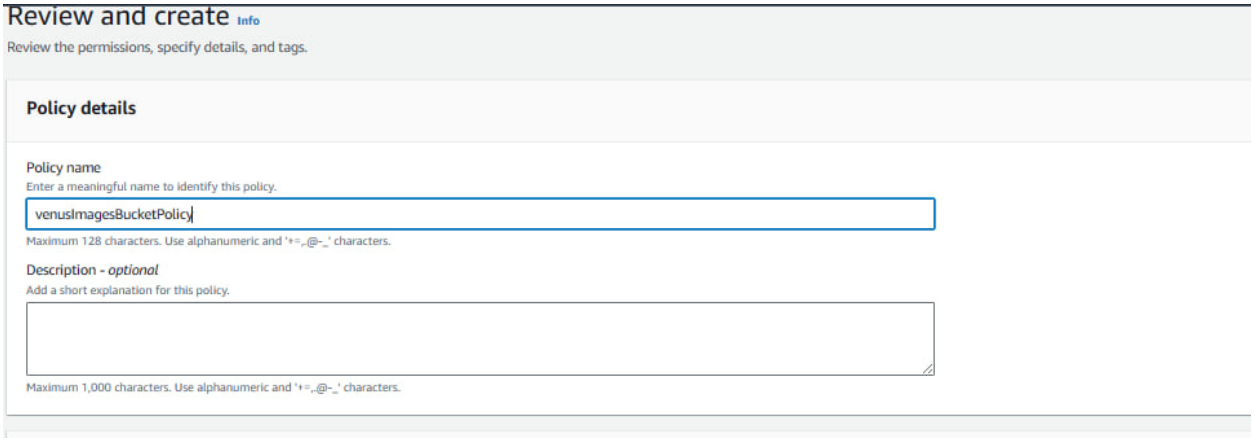

Figure 25: Policy name

In the policy editor we have allowed our lambda to have a write access to CloudWatch and read and write access to our S3 buckets. This can be seen in the image below.

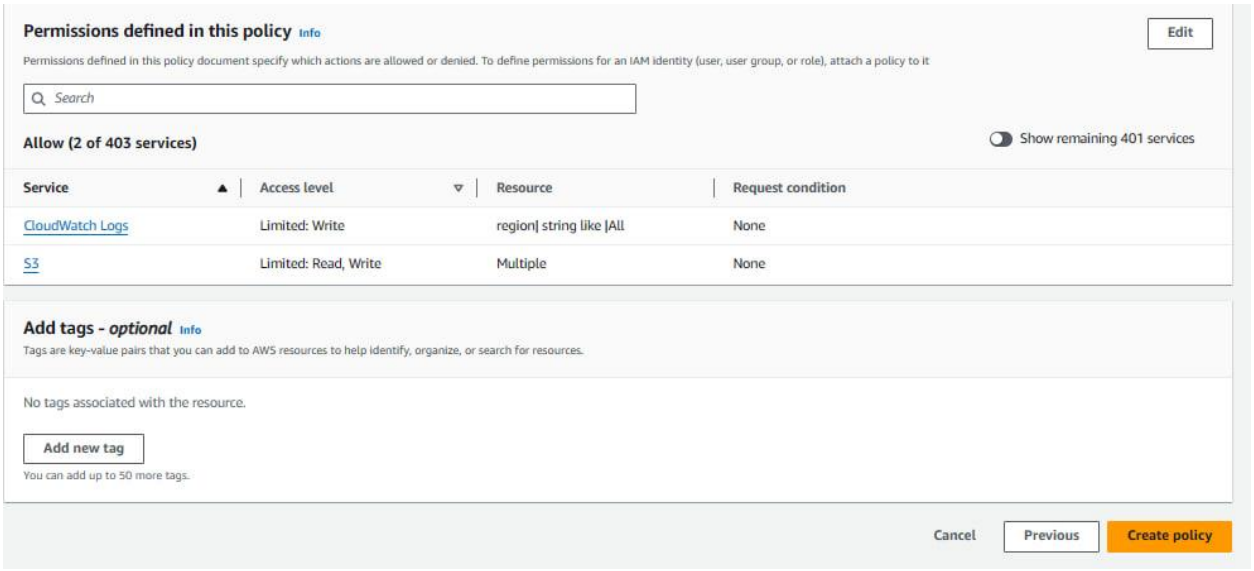

Figure 26: Permissions defined in the policy

Click on “Create Policy” after carefully reviewing the configurations.

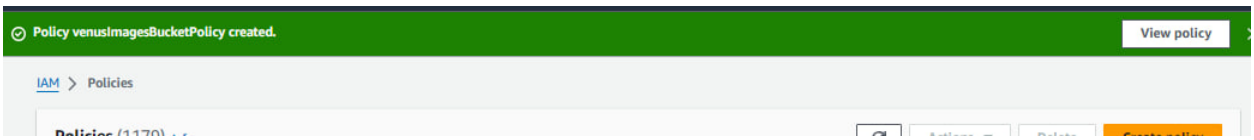

Figure 27: Successfully created policy

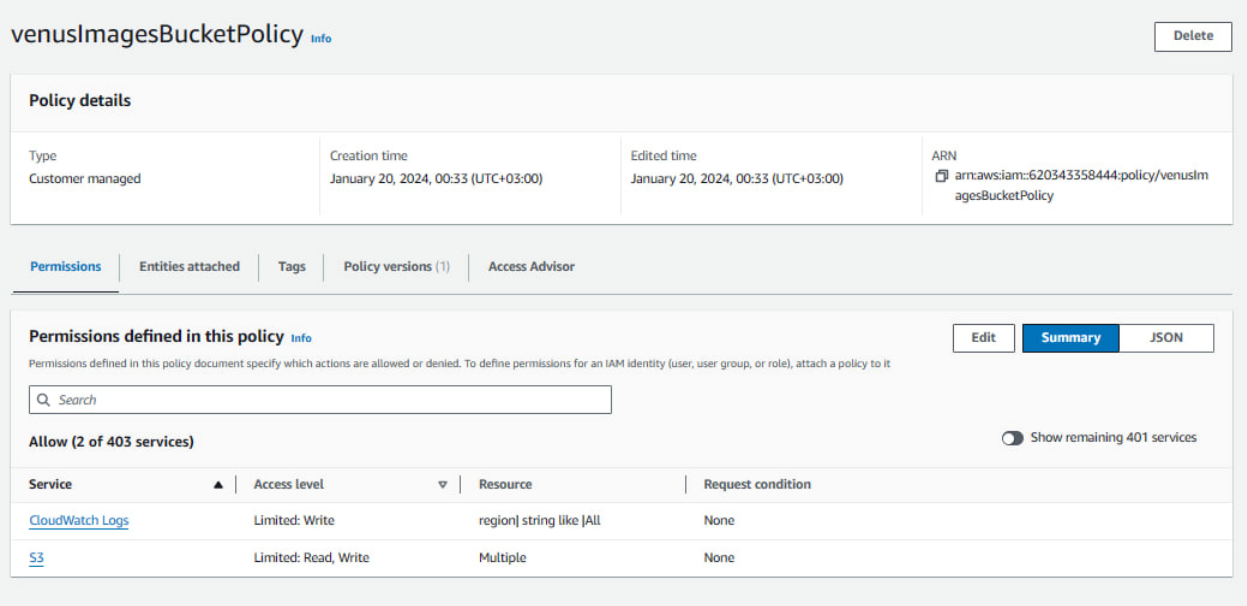

Figure 28: Details of created policy

Now go back to the role page, refresh, and then select the created policy from the list.

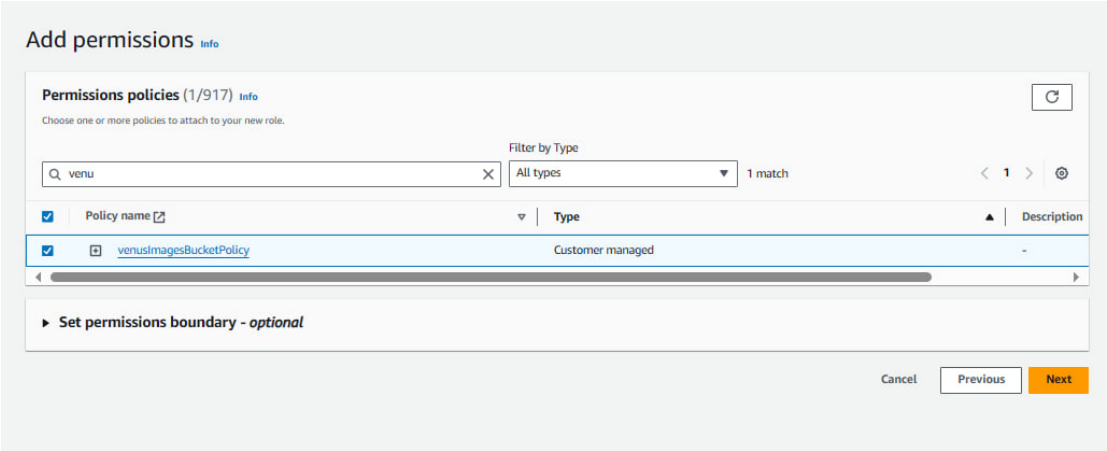

Figure 29: Selecting the created policy

After clicking on "Next," the Name, Review, and Create page will appear. Properly fill out the role details, then click on "Create role."

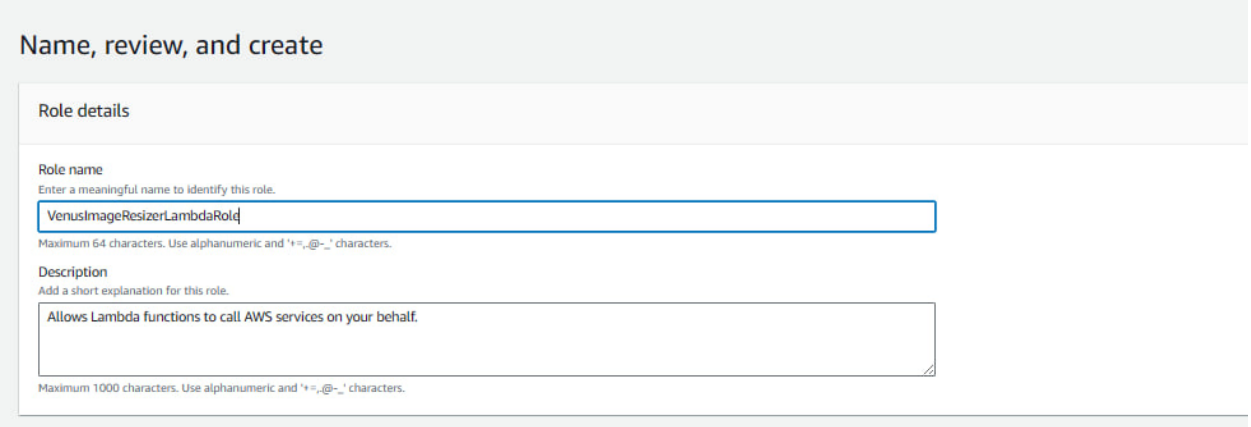

Figure 30: Role name

The below JSON snippet observed on the Trusted entity section, outlines a role trust policy, granting the AWS Lambda service (`lambda.amazonaws.com`) permission to assume the associated IAM role. The policy allows AWS Lambda to perform the `sts:AssumeRole` action, enabling it to assume this specific role. This configuration is commonly utilized when delegating specific permissions to AWS Lambda functions for interaction with other AWS services or resources.

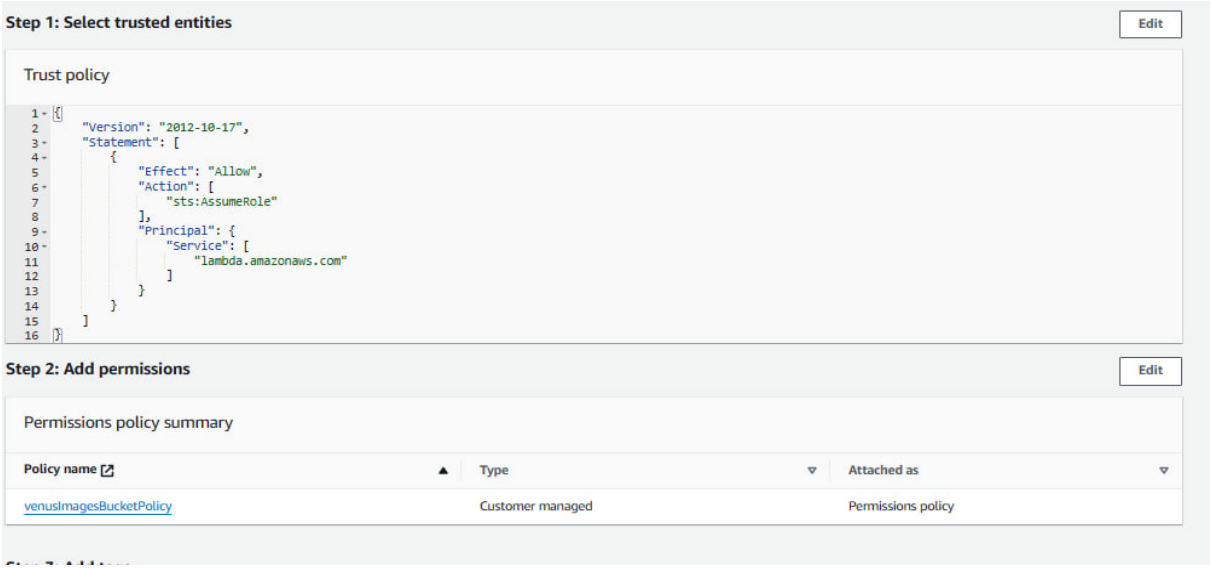

Figure 31: Select trusted entities and add permissions

Click on “Create Role” after carefully reviewing every detail.

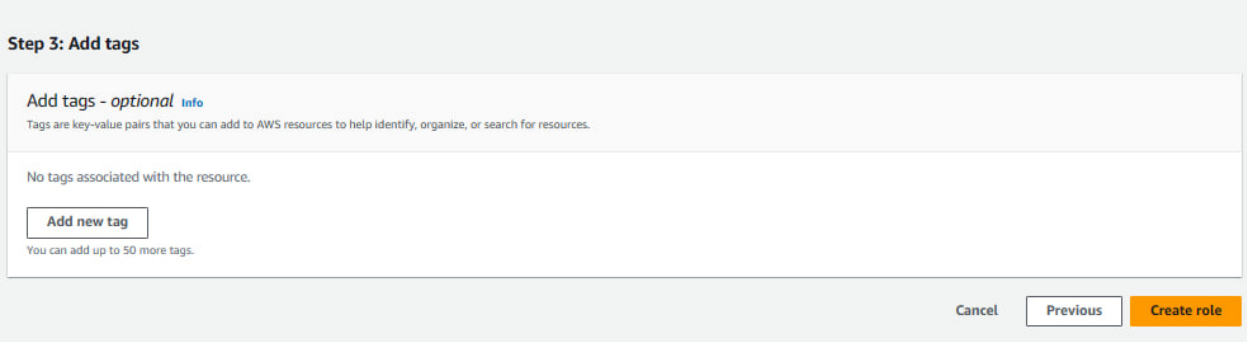

Figure 32: Create role

As shown below, the previously created policy has been successfully attached to the role.

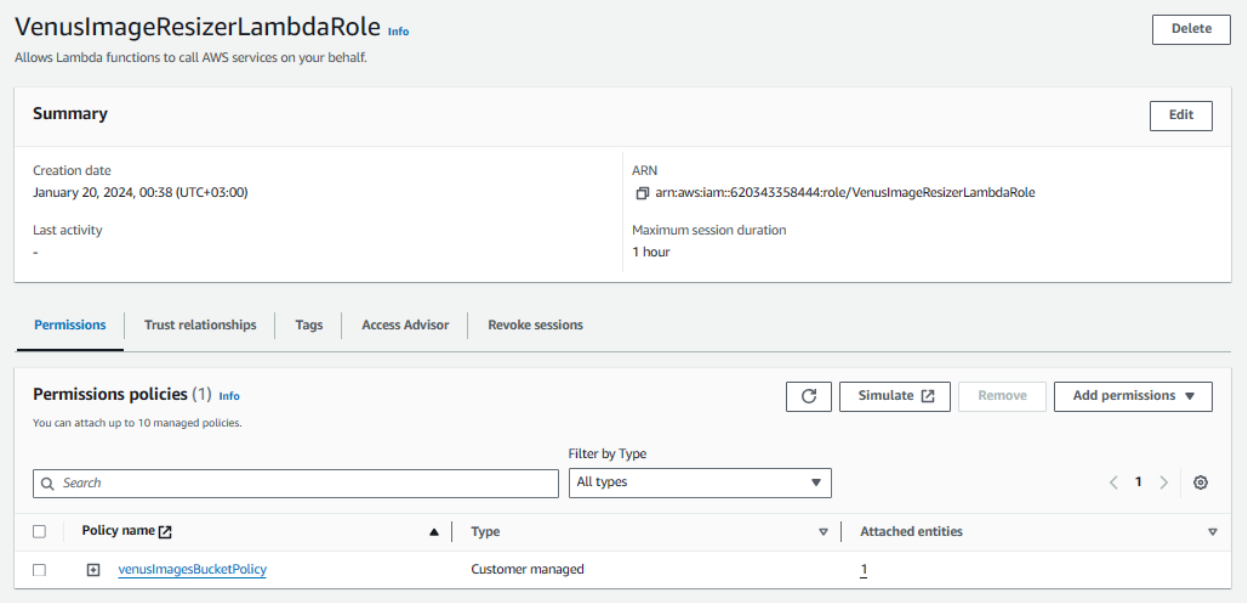

Figure 33: Details of the created role

Now that the role has been created, go back to the Lambda page, refresh, and attach that role to the Lambda. Choose "Existing role" and select the role from the dropdown menu.

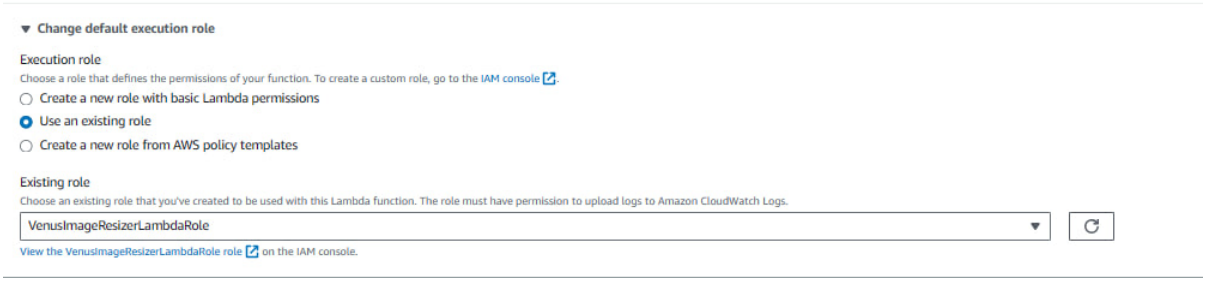

Figure 34: Change default execution role

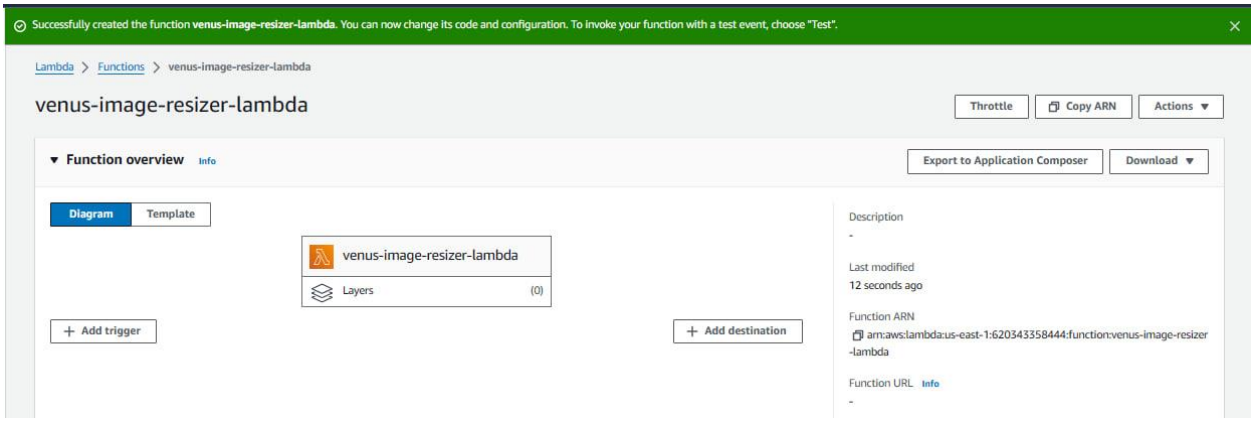

Figure 35: Lambda successfully created

5. Write function code – In the "Function code" section, write or upload the code for the Lambda function. Specify the handler function and set environment variables as needed.

Clear the code that was available by default in the lambda editor, click on the upload form button, and select the .zip file option from the dropdown.

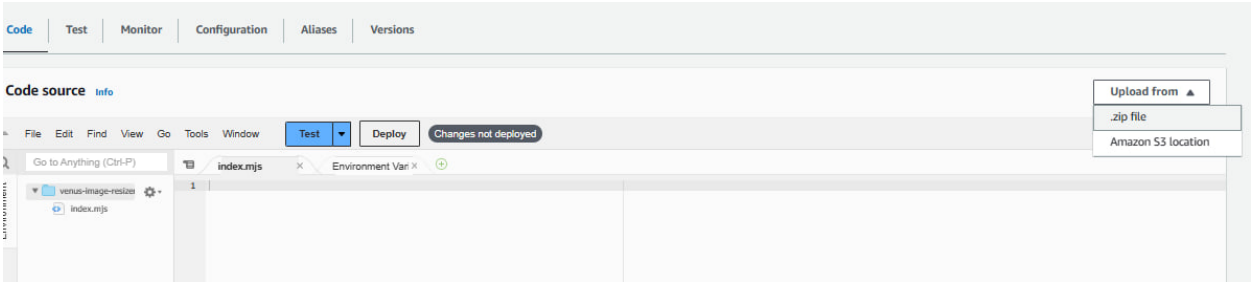

Figure 36: Upload form

After choosing the .zip file option, upload a file on this page and then click on the save button.

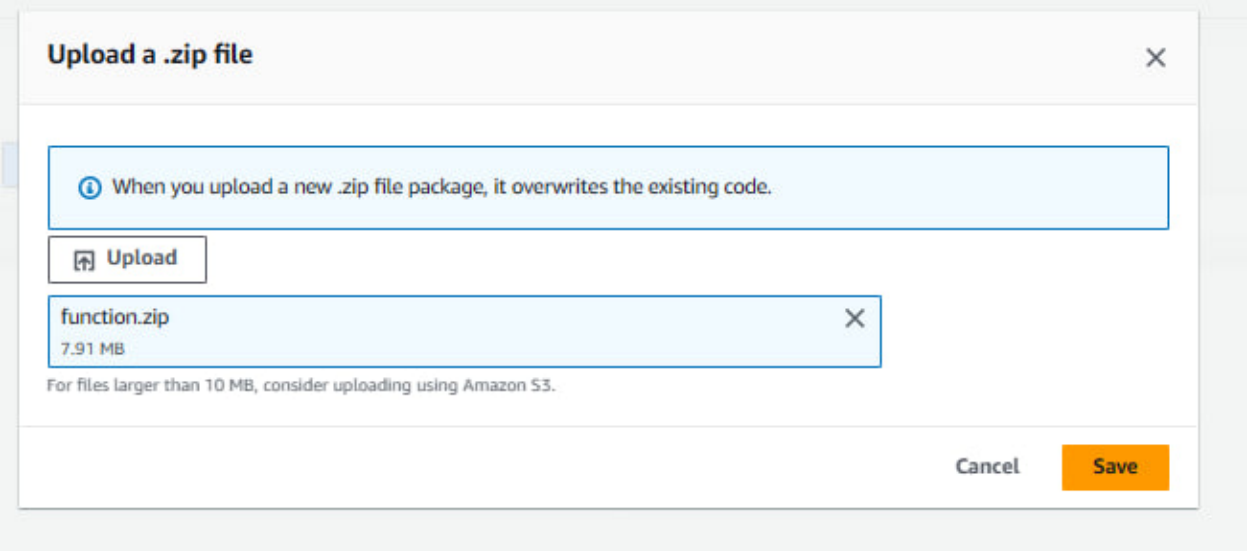

Figure 37: Upload a .zip file

The code file has been successfully uploaded

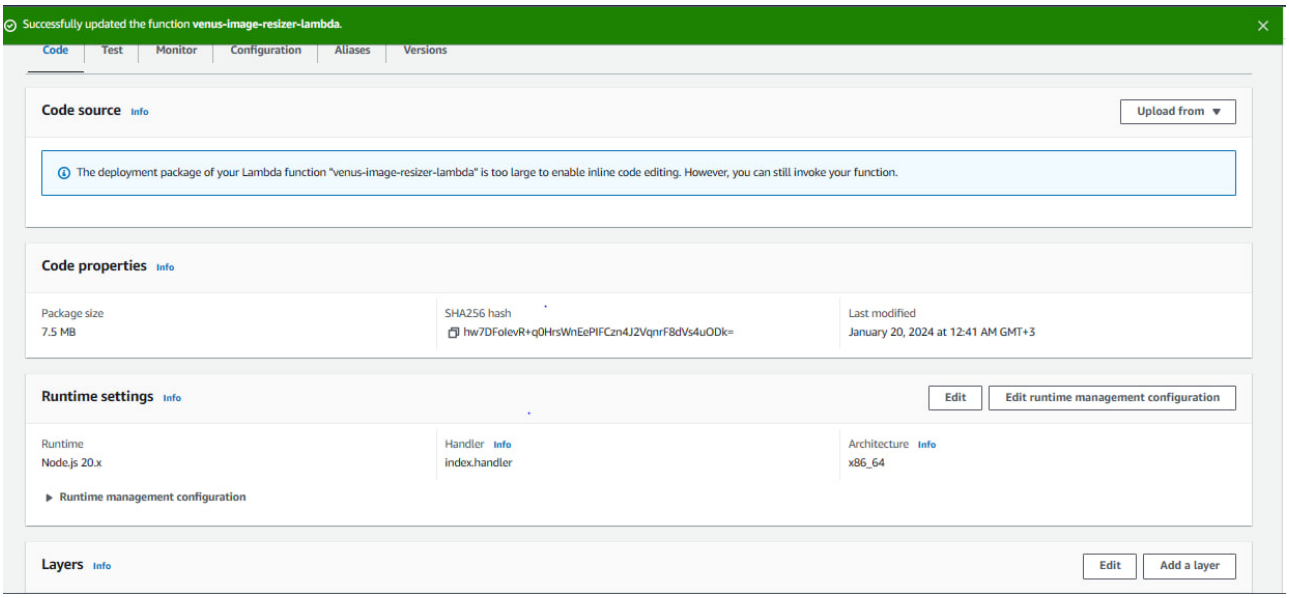

Figure 38: Successfully updated the function

Once the code is successfully uploaded, an environment variable needs to be added. An environment variable is a key-value pair that is external to the code, allowing developers to configure aspects of the function's behavior without modifying the code itself. In our code, we have defined "DEST_BUCKET," which is a variable designated for the destination bucket. This variable needs to be added through the console. To do so, navigate through configuration, then go

to environment variables, click Edit, and then click on "Add environment variable." For the Key, add the variable "DEST_BUCKET," and for the Value, add the second bucket that was created before, namely, "venus-images-resized-bucket."

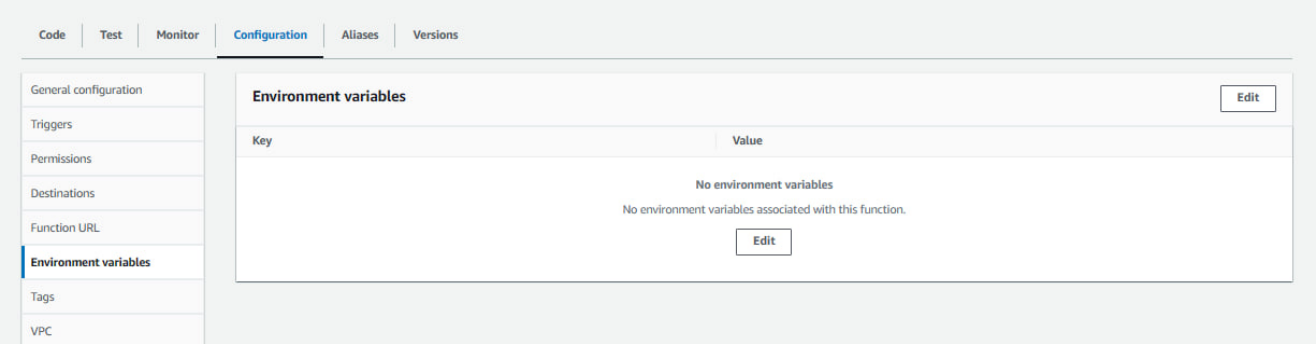

Figure 39: Navigate to configuration and environment variables

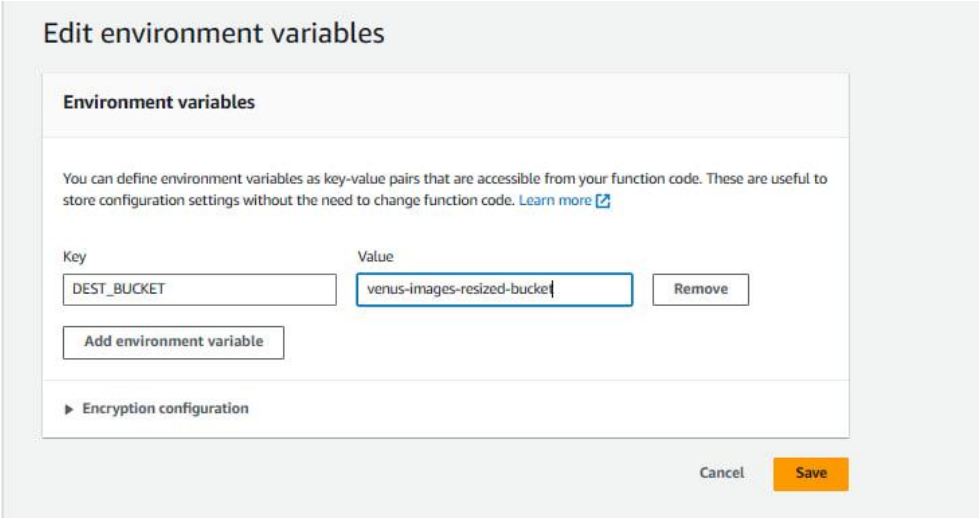

Figure 40: Edit environment variables

To manually test our Lambda function, we can navigate to the Test tab, and under the template, search for "s3-put." "S3 Put" in the context of AWS Lambda typically refers to an event trigger associated with an Amazon S3 bucket. Under the Event JSON, edit the name section of the S3 bucket and replace it with the first bucket name, "venus-images-bucket," and also replace the ARN value similarly with the name of the first bucket. ARN stands for Amazon Resource Name, and it is a unique identifier assigned to AWS resources. Under the object section, replace the key with the name of the image that has been uploaded in the first bucket before, which is "GreenTech.jpg."

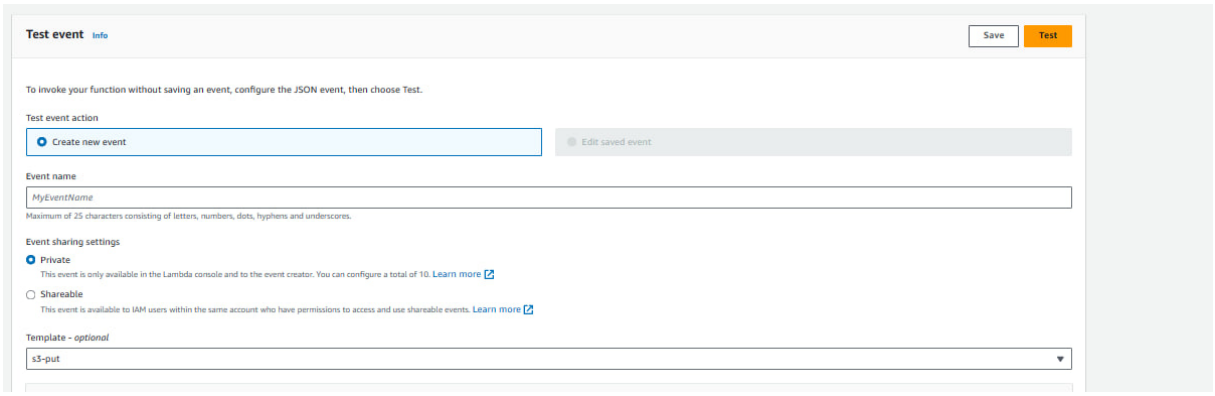

Figure 41: Choose s3-put under template

Figure 42: Modify event JSON

Now that this has been set, click on the Test button. As observed in the image below, a success message has been displayed. Now, go back to the second bucket and verify that the image has been resized.

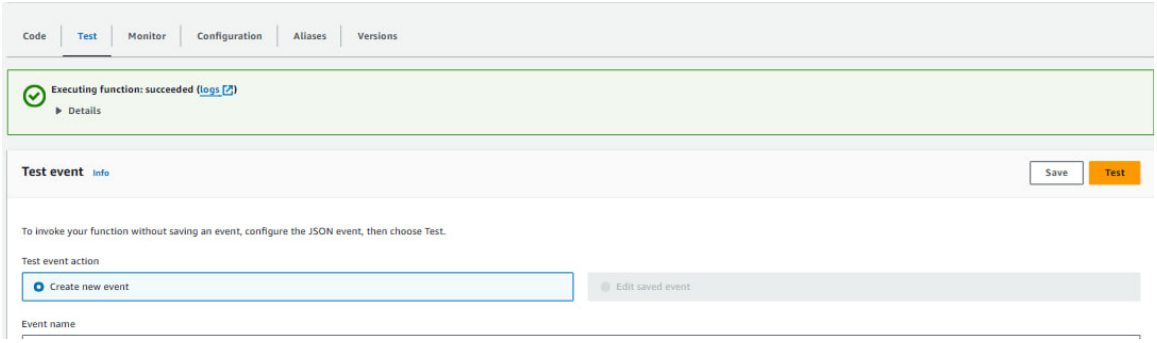

Figure 43: Executing function succeeded

Figure 44: Success message

Return to the second bucket and check if the image has been resized (we have set both the width and the height to be 200 px and saved to the second bucket.

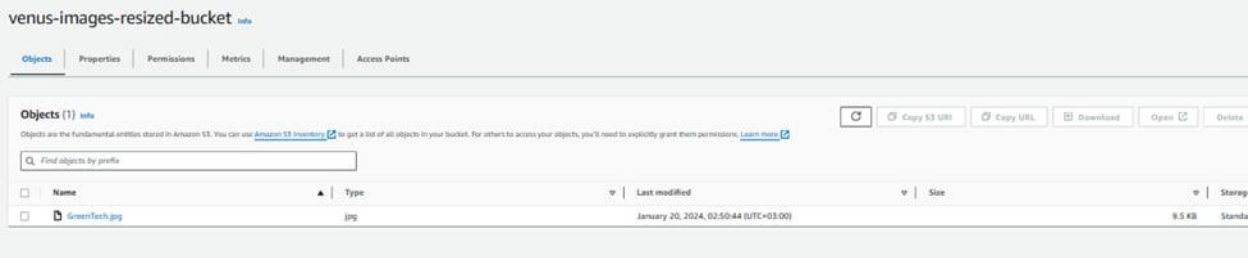

Figure 45: Image stored on second bucket

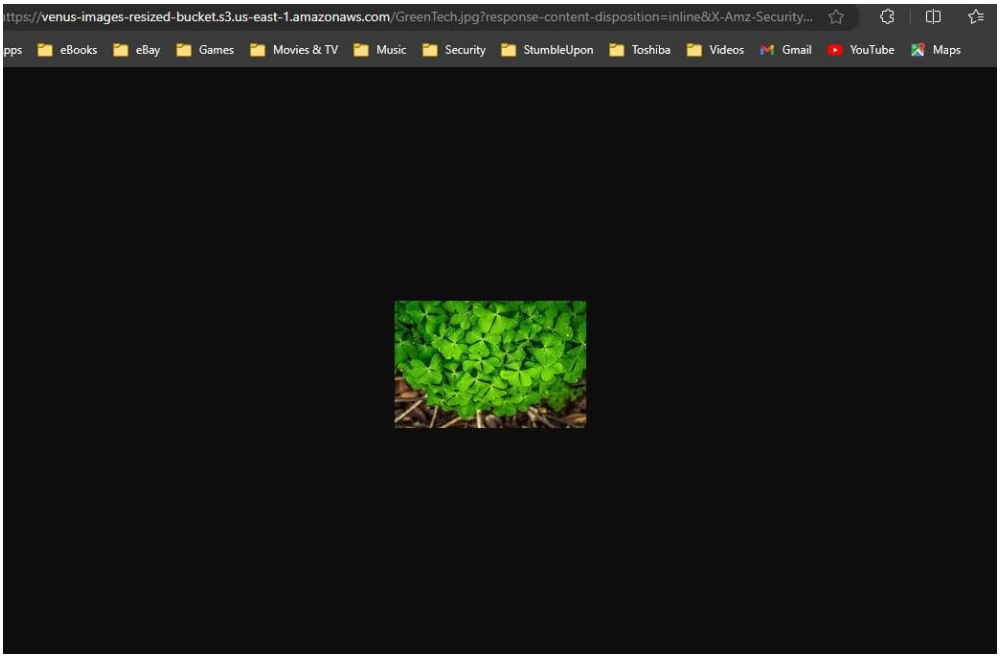

Figure 46: Image successfully resized

Right now, the Lambda won't run unless it's directly invoked, so we need to set up a trigger on the Lambda function.

6. **Configure Triggers** – Event sources or triggers can be configured for the Lambda function, and these may involve services such as S3, API Gateway, or CloudWatch Events. In the context of this project, S3 has been selected for the services section, and a source bucket has been chosen from the dropdown. This selected bucket is the initial repository for storing raw satellite images.

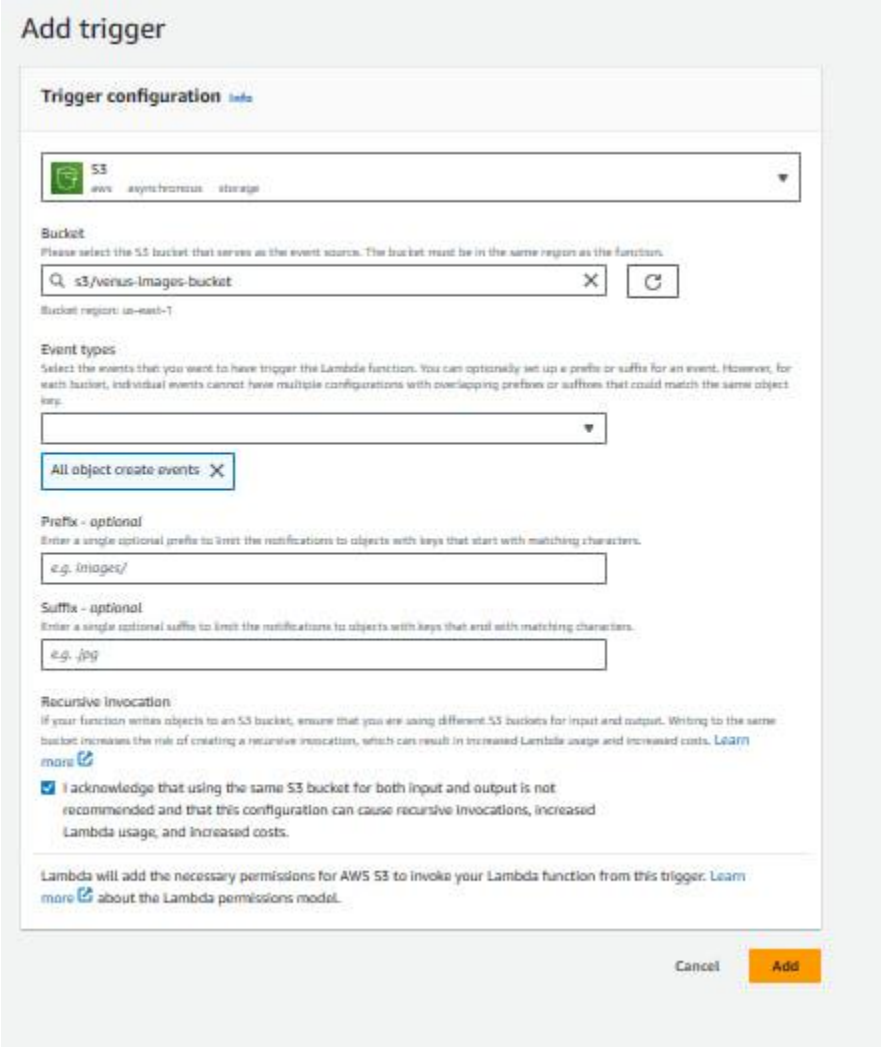

Figure 47: Add trigger to lambda function

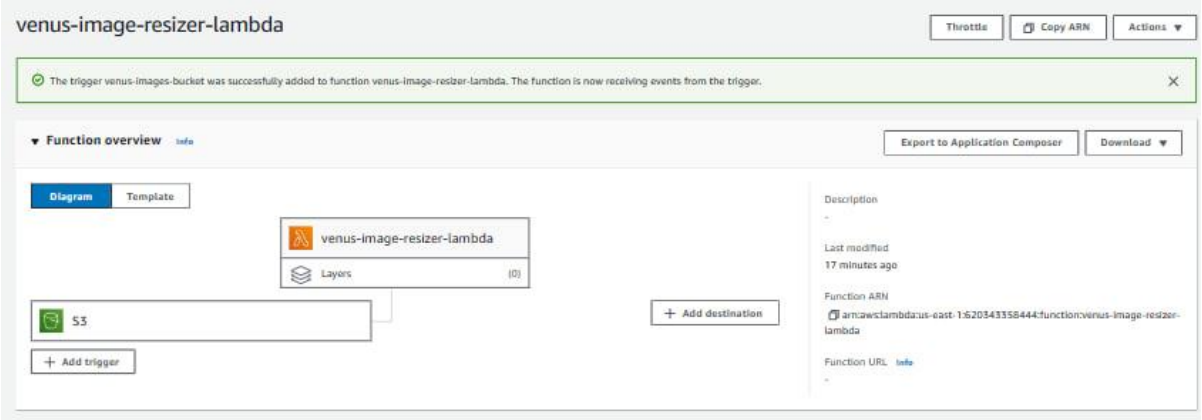

Figure 48: Successfully added trigger

To check if the trigger is functioning properly, upload a second image (GreenTech2.jpg) to the first bucket and observe if the image has been resized and has been uploaded to the second bucket. After entering the designated bucket, select the "Upload" button. A file upload dialog will be displayed; utilize it to select the image file from your local device. Upon selecting the image, click "Upload" to commence the file transfer to the S3 bucket.

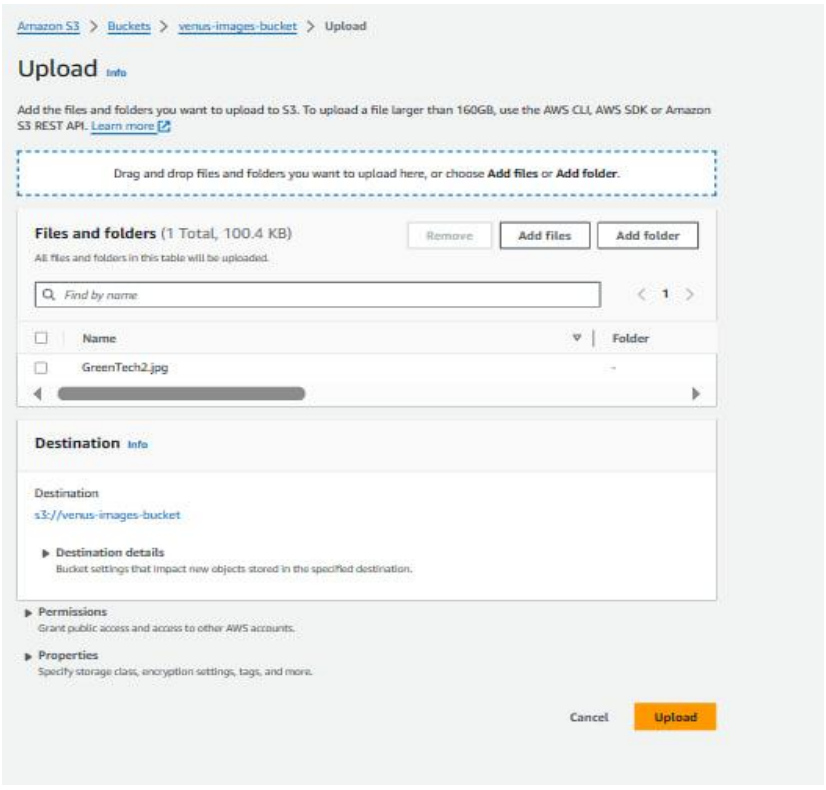

Figure 49: Upload a second image to the first bucket

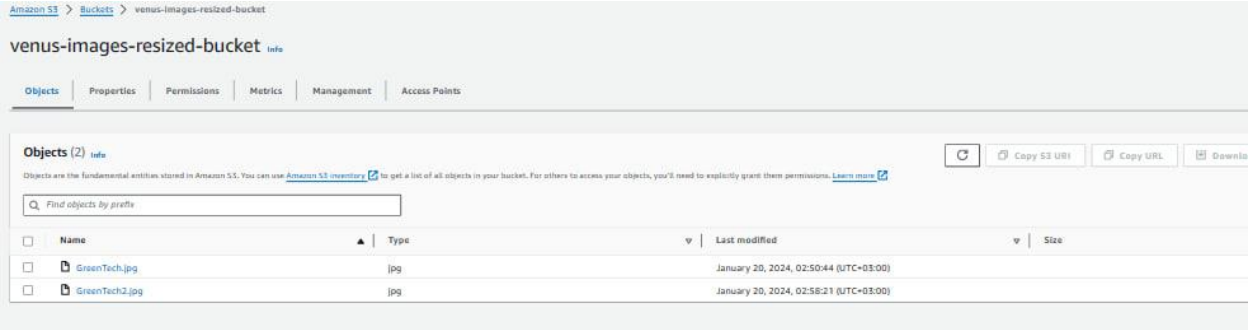

Figure 50: Images uploaded in the first bucket

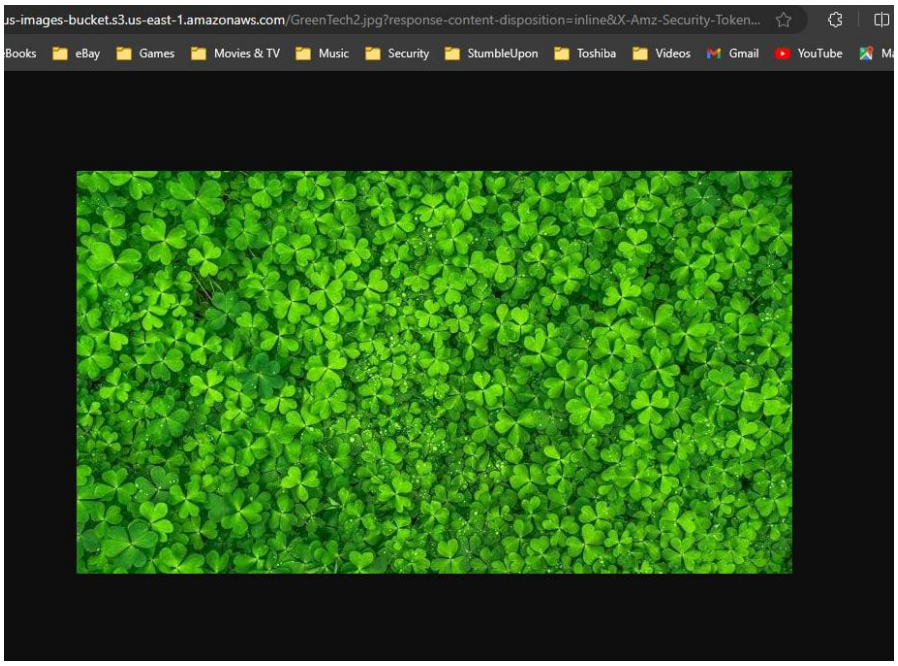

Figure 51: Second image uploaded on first bucket

Now, go back and check whether the Lambda has been triggered, and a Lambda function call has occurred.

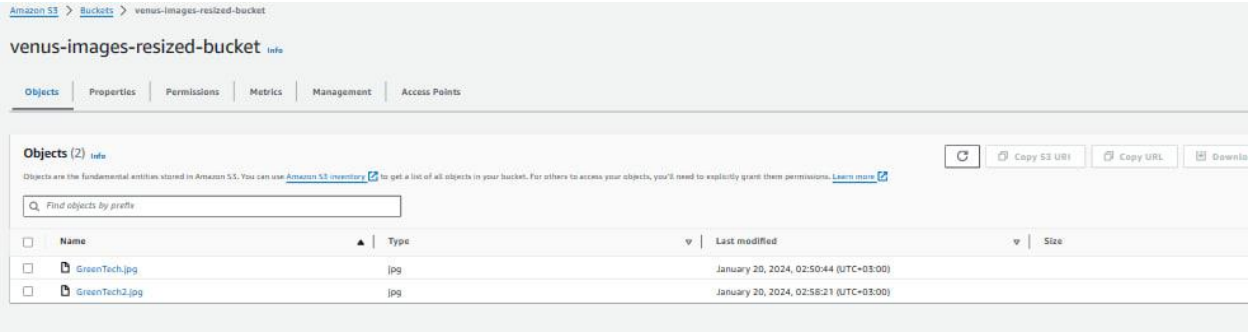

Figure 52: Resized image is stored in second bucket

To view the image, click on the image, then click on the "Open" tab; the image will open in a new tab.

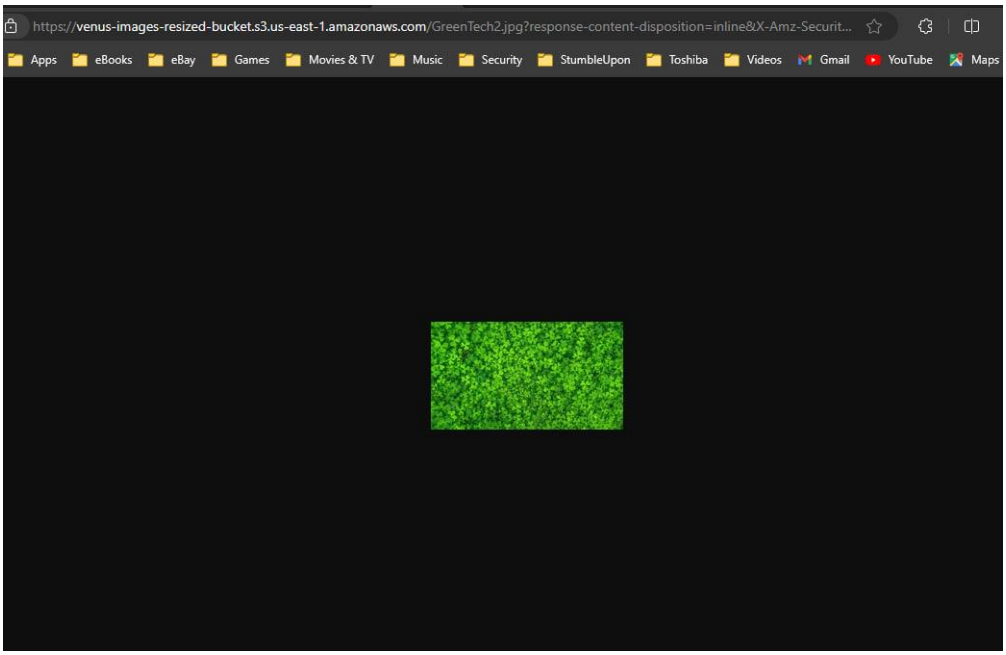

Figure 53: Second image is successfully resized

6 Monitoring Aspect of the project

When working on the project described, there are several aspects you can monitor using AWS CloudWatch to ensure the smooth operation and performance of your serverless function (lambda function) and storage services (S3 bucket). Here are some key monitoring considerations:

Lambda Function Metrics:

- Invocation count: Monitor the number of times your Lambda function is invoked to track its usage and workload.
- Duration: Monitor the execution time of your Lambda function to identify any performance issues or improvements.
- Errors: Track the number of errors or exceptions encountered during the execution of your Lambda function.

S3 Bucket Metrics:

- Bucket size: Monitor the size of your S3 buckets to track storage utilization and plan for capacity management.
- Put and Get requests: Monitor the number of requests made to upload images to the first bucket and retrieve resized images from the second bucket.
- Data transfer: Track the amount of data transferred in and out of the buckets to monitor network usage and costs.

CloudWatch Events:

- Monitor the execution of the event triggers that invoke your Lambda function when images are uploaded to the first bucket.
- Track the success rate and any failures in triggering the Lambda function to ensure the automation is working as expected.

Lambda Function Logs:

- Enable AWS CloudWatch Logs for your Lambda function and monitor the logs for any errors, warnings, or debugging information.
- Configure log streaming and retention to ensure critical logs are captured and retained for analysis.

By monitoring these aspects using AWS CloudWatch, you can proactively identify and address any performance issues, errors, or capacity constraints in your serverless function and storage services. This will help ensure the reliable and efficient processing and storage of high-resolution images for environmental analysis.

To monitor an Amazon S3 bucket using Amazon CloudWatch, here are the steps to follow:

Sign in to the AWS Management Console and open the CloudWatch console.

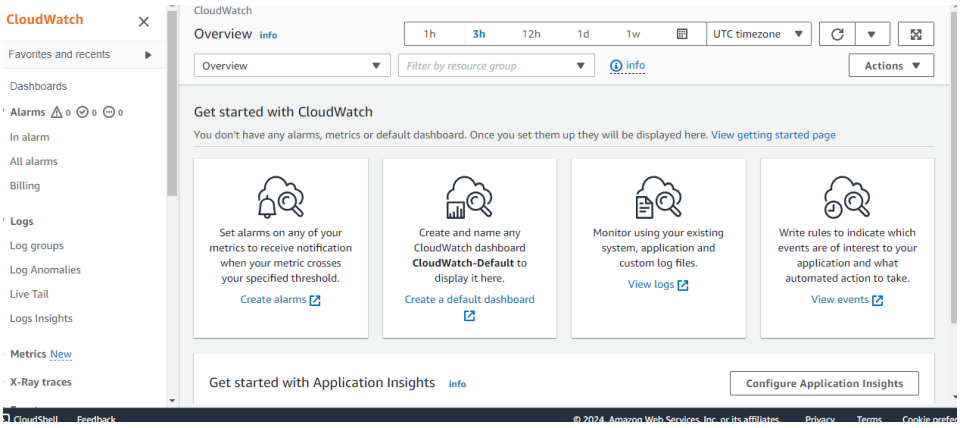

Figure 54: CloudWatch console

In the navigation pane, click on "Rules" under "Events" to create a new rule.

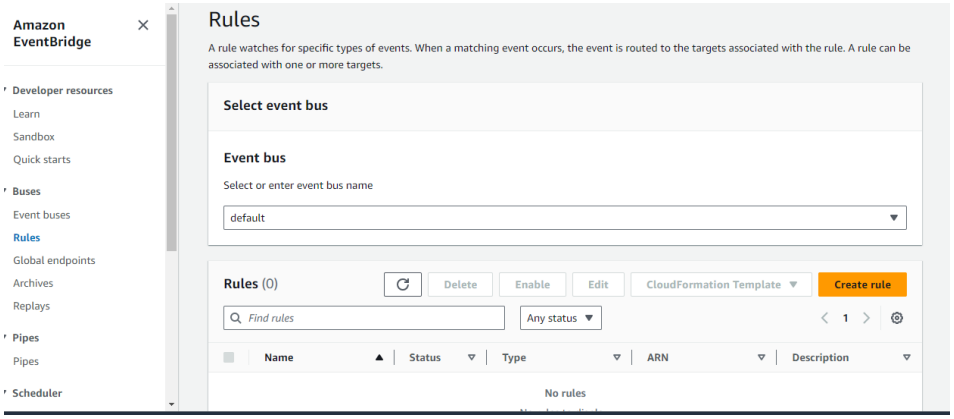

Figure 55: Create rule

Click on "Create rule."

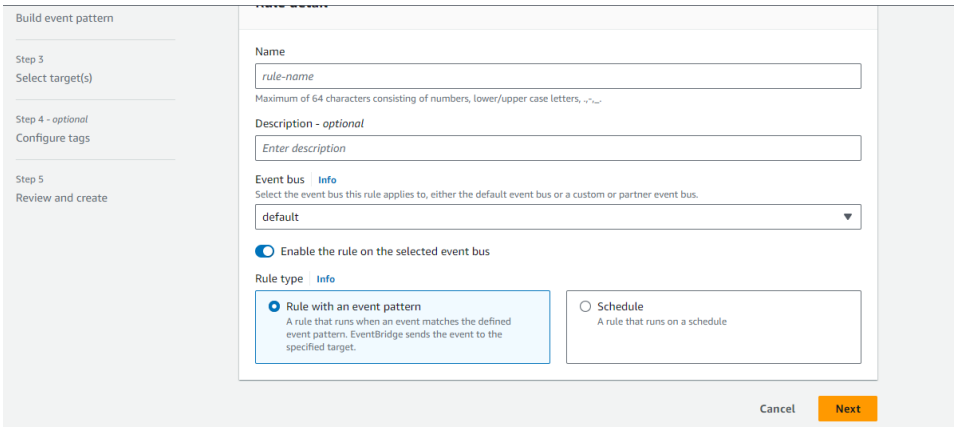

Figure 56: Rule name

Under "Event Source," select "Event Pattern."

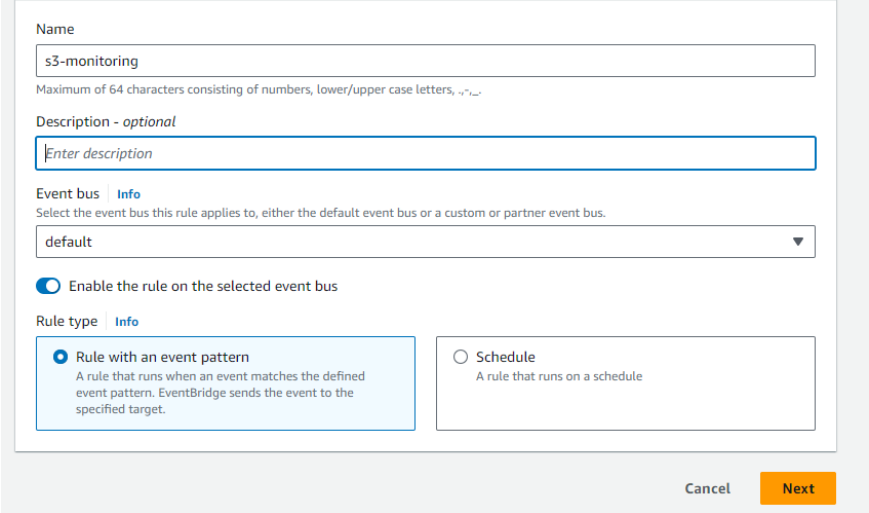

Figure 57: Event pattern

In the "Event Pattern Preview" section, click on "Edit." And then in the "Event pattern" editor, specify the S3 bucket as the event source. For example, you can use the following JSON pattern:

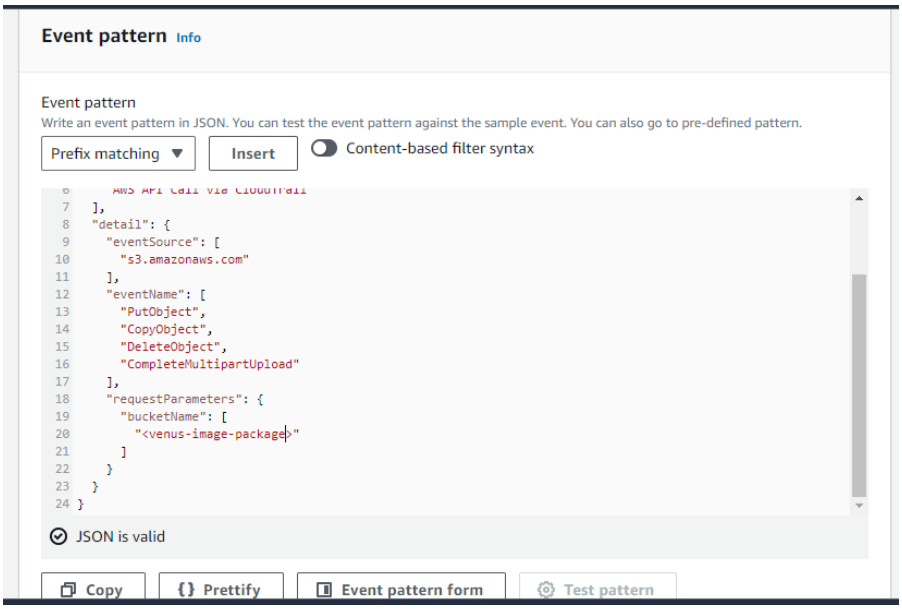

Figure 58: Event pattern editor

Then click next

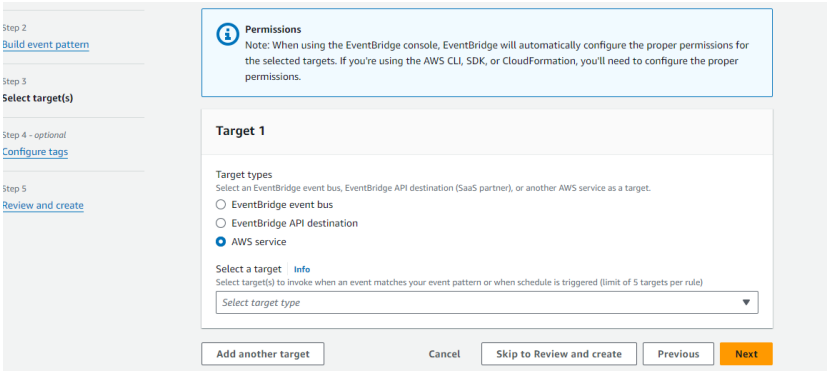

Figure 59: Click next

Under "Targets," click on "Add target," and select "CloudWatch Logs". And configure the log settings, such as log group and log stream names.

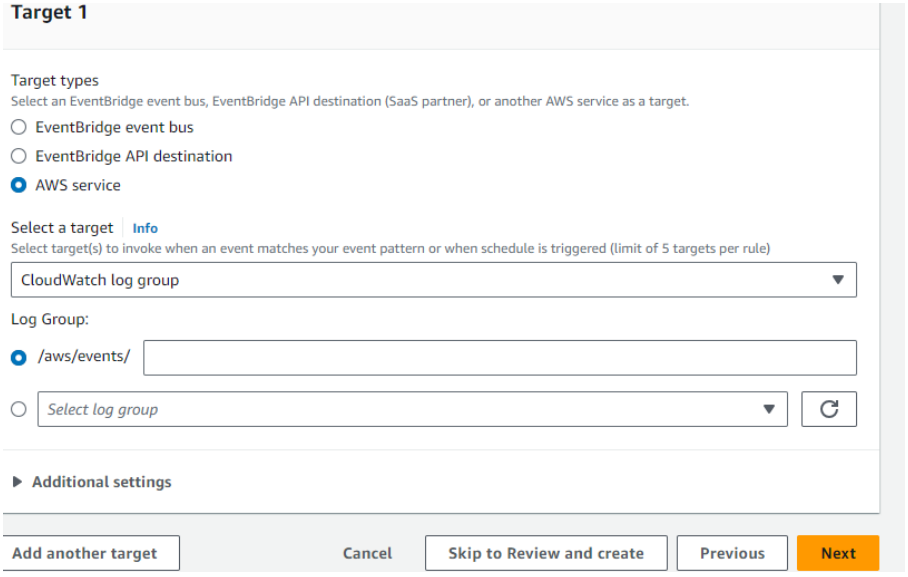

Figure 60: Add target

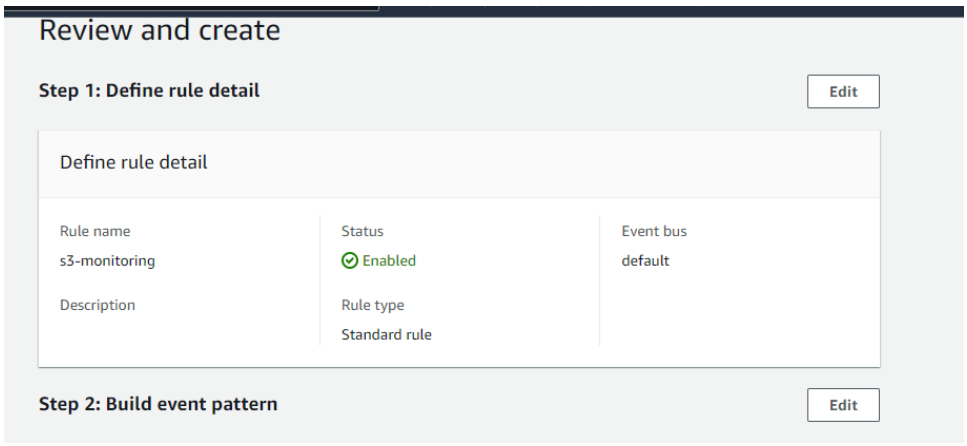

Figure 61: Review and create: define rule detail

Figure 62: Review and create: build event pattern

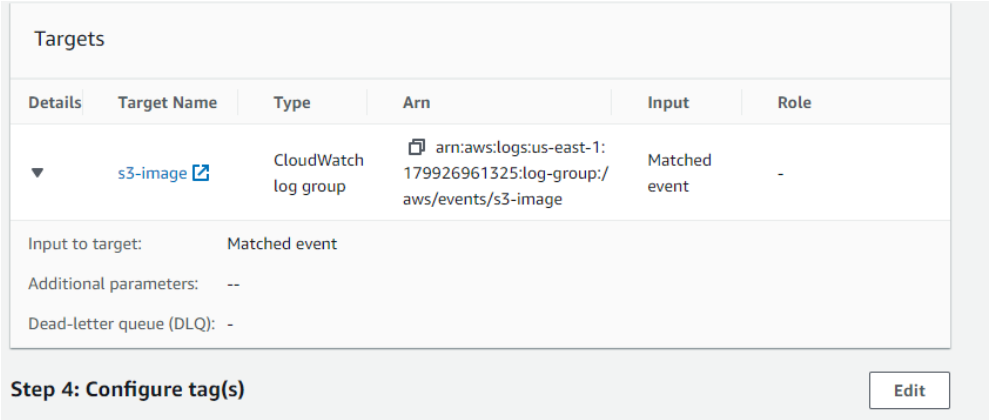

Figure 63: Targets

Finally create rule

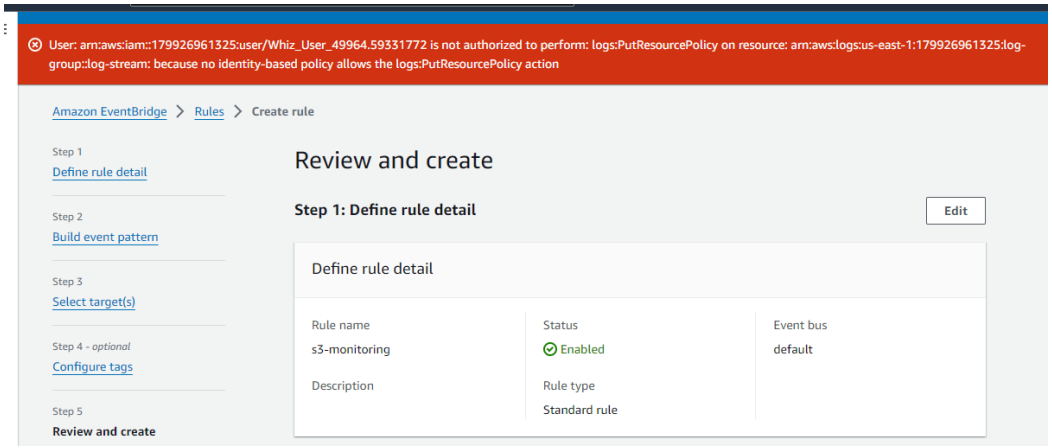

Figure 64: Review and create: Error

Once the CloudWatch rule is created, it will capture events related to the specified S3 bucket, such as object uploads, copies, deletions, and multipart uploads. These events will be sent to the specified CloudWatch Logs log group and log stream for further analysis and monitoring. However it says, "User: arn:aws:iam::179926961325:user/Whiz_User_49964.59331772 is not authorized to perform: logs:PutResourcePolicy on resource: arn:aws:logs:us-east-1:179926961325:log-group::log-stream: because no identity-based policy allows the logs:PutResourcePolicy action" due the limited access of sandbox.

We can set up CloudWatch Alarms on the log data to trigger notifications or automated actions based on specific conditions, such as a sudden increase in S3 object deletions or failed uploads. But to do all this thing we need to have the necessary permissions to create CloudWatch rules and access CloudTrail logs to monitor S3 bucket events using CloudWatch.

7 Conclusion

7.1 Summary of Achievement

The successful completion of the cloud migration process signifies a significant achievement for the GreenTech Alliance. The organization can now rely on a robust cloud infrastructure that ensures reliable, scalable, and automated satellite image processing.

7.2 Importance of Cloud Infrastructure for Environmental Technology Startups

The adoption of cloud infrastructure is particularly crucial for environmental technology startups like the GreenTech Alliance. It not only enhances operational efficiency but also provides a foundation for innovation and growth. Cloud technologies empower organizations to leverage advanced computing capabilities without the need for substantial upfront investments.

7.3 Challenges Encountered

We were unable to incorporate CloudWatch into our project due to our limited access in the Sandbox. However, we have provided a brief explanation of how to set up CloudWatch to monitor the performance and overall health of both our Lambda function and two S3 buckets. The error message we encountered indicates that the AWS Identity and Access Management (IAM) user lacks the necessary permissions to perform the "logs:PutResourcePolicy" action on the specified CloudWatch Logs resource. Specifically, there is no identity-based policy attached to the user allowing the execution of the "logs:PutResourcePolicy" action on the identified log group and log stream within the specified AWS region (us-east-1). To resolve this issue, we need to review and update the IAM policies associated with the mentioned IAM user to include the required permissions for managing CloudWatch Logs resources, ensuring that the necessary actions are explicitly allowed in the policies associated with the user's role or group.

7.4 Future Recommendations and Potential Improvements

To further enhance the GreenTech Alliance's cloud infrastructure, future recommendations may include continuous monitoring and optimization of resources, exploring additional cloud services for expanded functionalities, and staying abreast of emerging technologies. Additionally, the organization can consider implementing training programs to empower its team with the necessary

skills for effective utilization of the cloud environment. These measures will contribute to sustained growth and success in leveraging cloud technologies for environmental sustainability.