**AWS Capstone Project**

**Problem Statement**

An organization requires a solution for hosting a static web page that ensures firewall protection, failover capabilities, geographical restrictions, and low latency access. The current infrastructure lacks the necessary security measures, redundancy, and global accessibility, leading to potential security breaches, downtime, and slow website performance. To address these concerns, the organization needs a comprehensive hosting solution that can provide robust firewall protection, automatic failover mechanisms, the ability to enforce geographical restrictions for access, and ensure low latency access to the web page for users worldwide.

**Goal:**

**To provide a hosting solution to ensure secure, reliable and efficient global access to their web page**

**Resources to be used to provide a solution:**

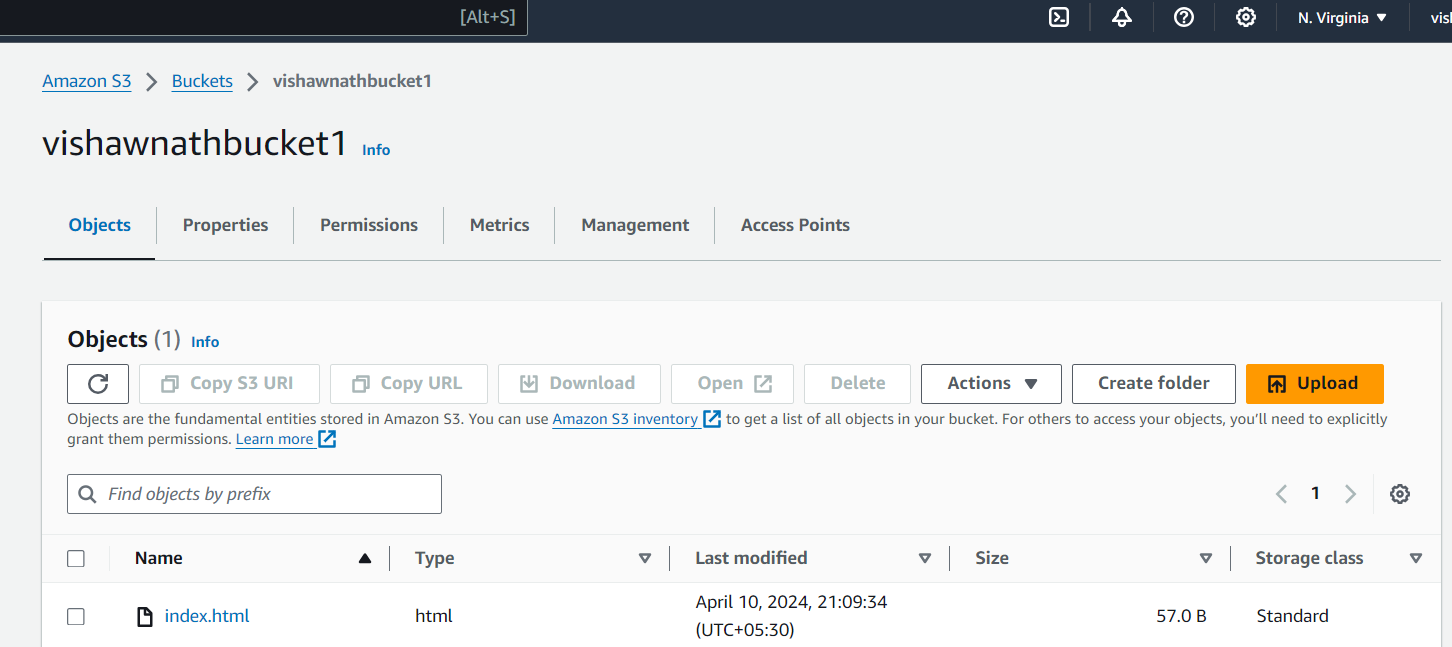
* **Amazon S3** (To host the website)
* **Amazon CloudFront Distribution** (to provide automatic failover mechanism, to enforce geographical restrictions for access, and ensure low latency access to the web page for users worldwide)
* **Amazon WAF** (firewall protection)

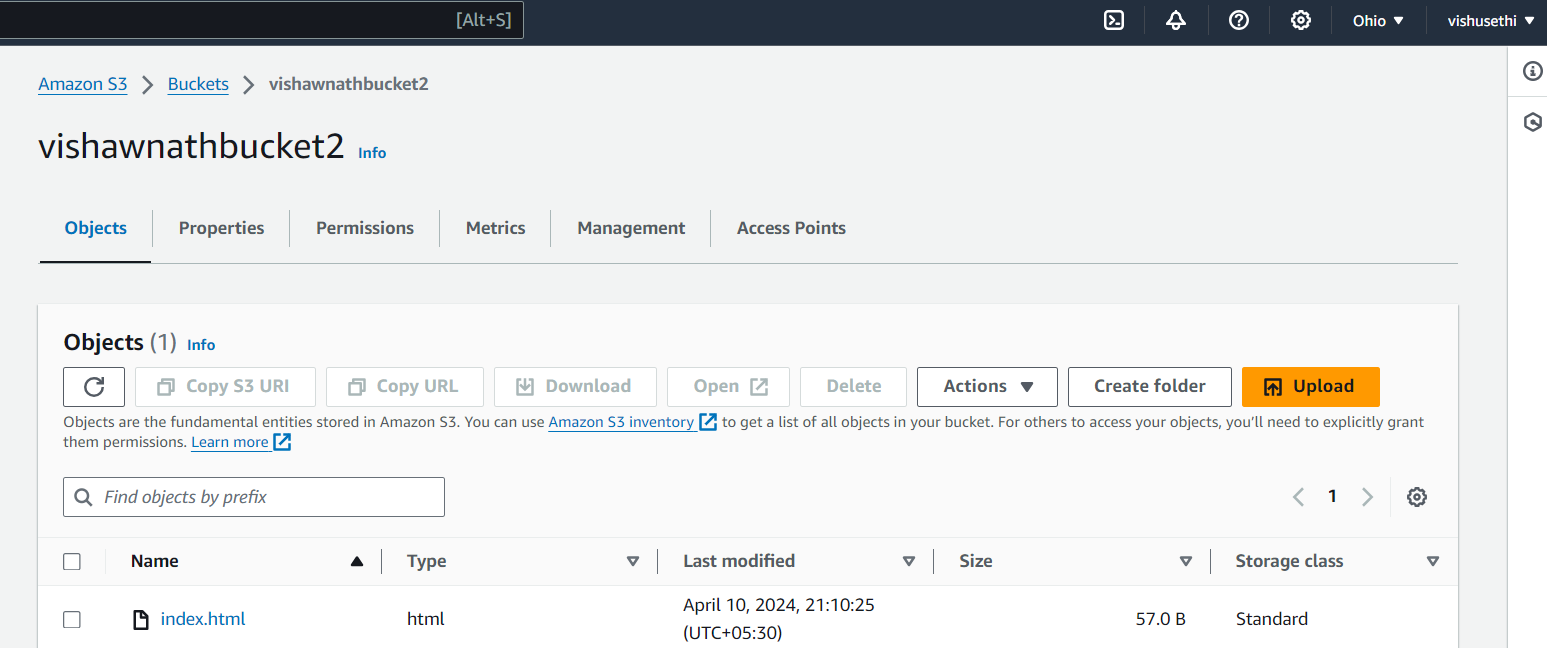
**Task-1 Created two S3 buckets**

**Objective: Set up primary and secondary buckets to host the static website**

**Steps Followed:**

* Primary bucket- Vishawnathbucket1
* Secondary bucket- Vishawnathbucket2
* Created Index.html file and uploaded to primary and secondary buckets.



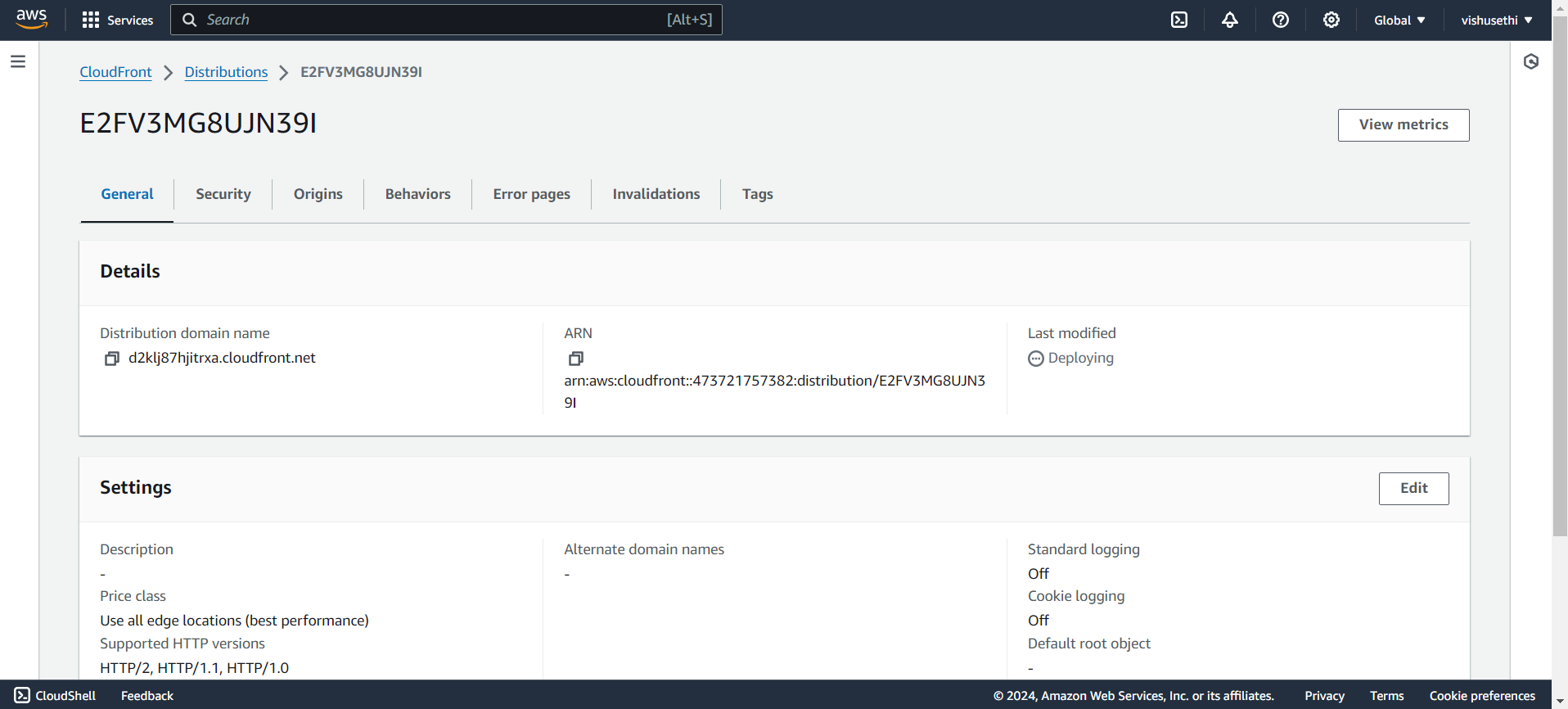


**Task-2 Creating cloud-front distribution**

**Objective: Set up CloudFront to distribute and serve content from the S3 buckets with enhanced features.**

**Steps Followed:**

* Created CloudFront distribution
* Choose primary bucket for origin domain
* Created a new origin access identity
* Made required changes in the settings of “Default behaviour”
* Changed the cache policy to “Caching disabled”

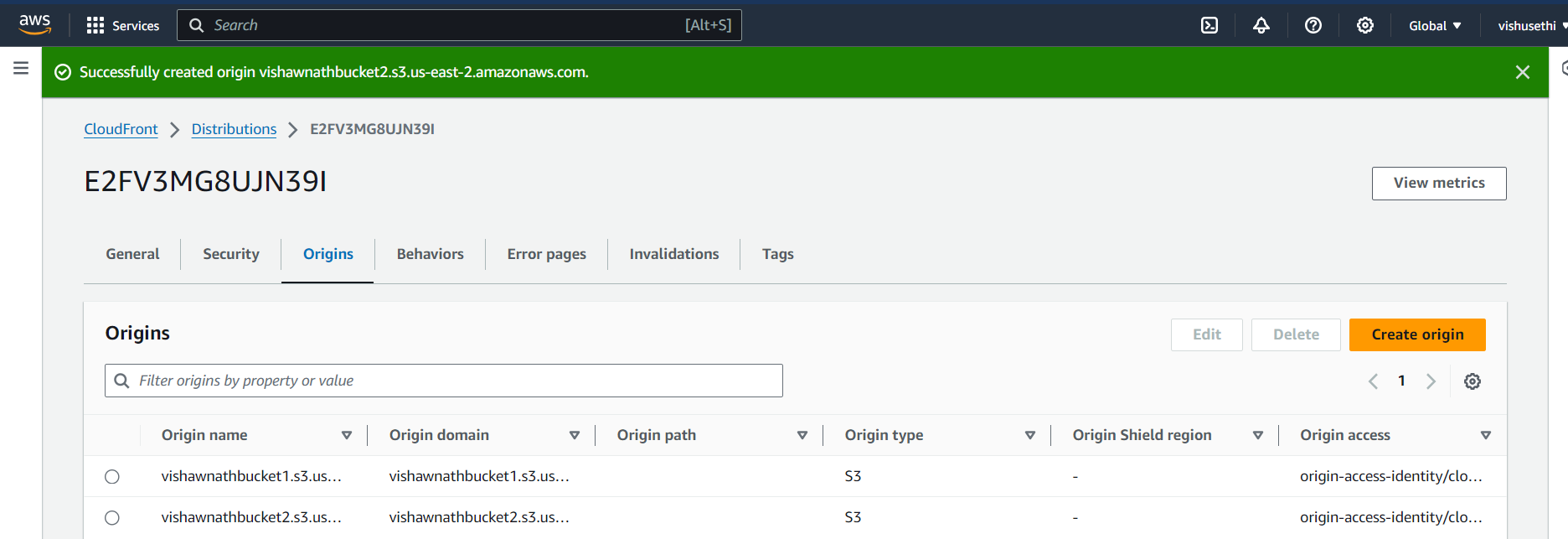


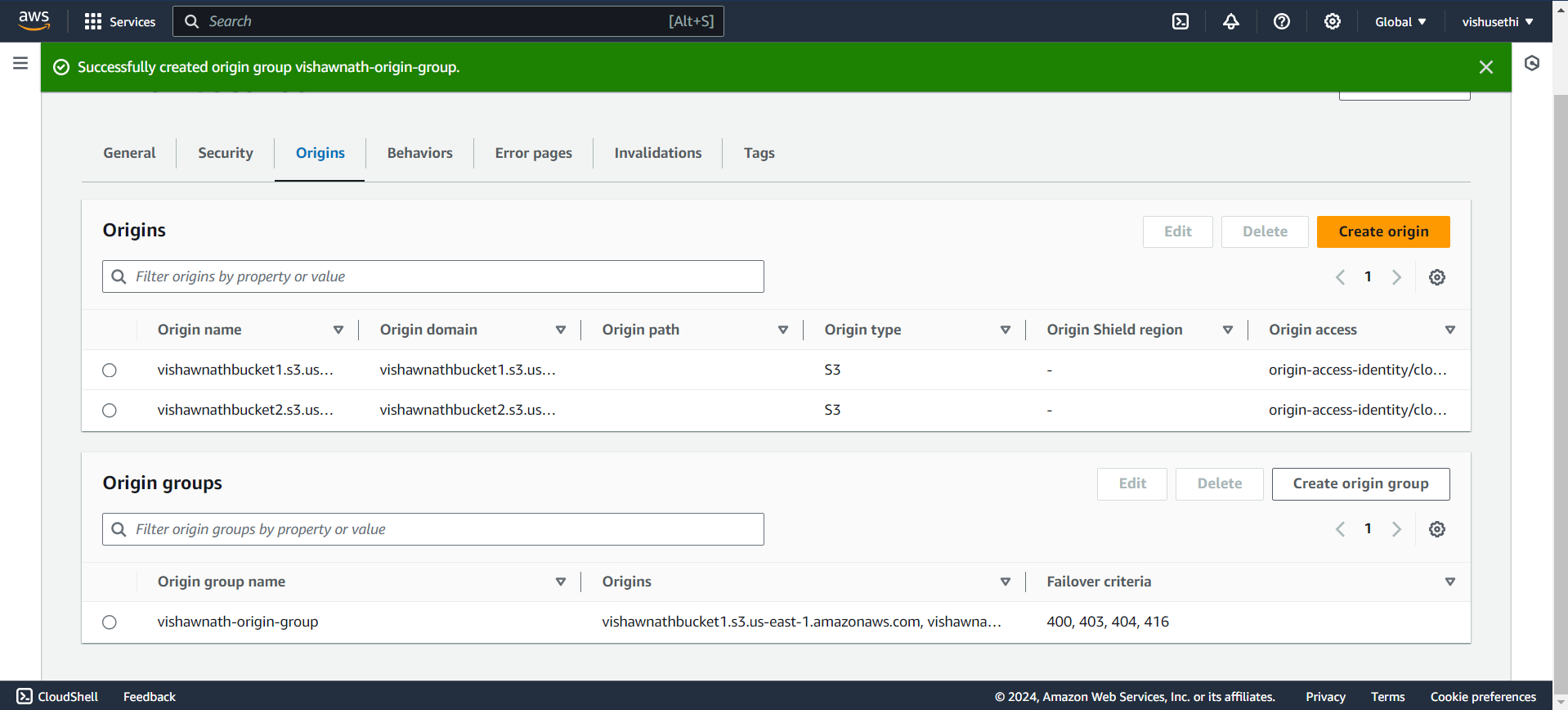
**Task-3 Configuring CloudFront for failover**

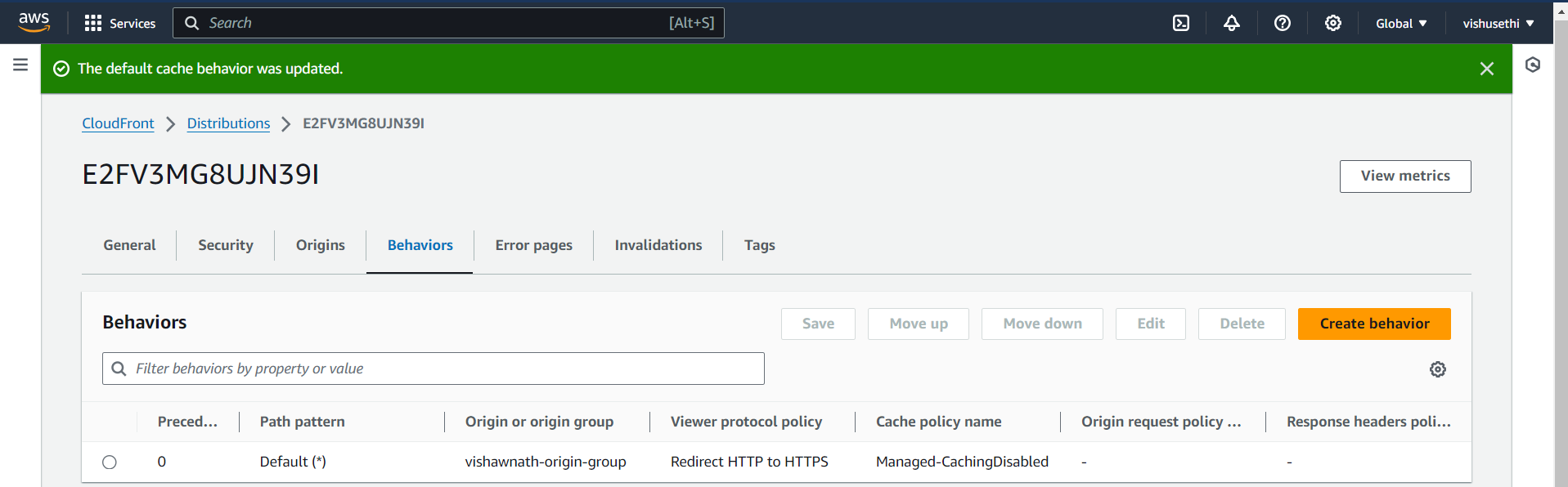
**Objective: Ensure high availability by configuring failover between primary and secondary S3 buckets.**

**Steps Followed:**

* In distribution, created origin
* Select secondary bucket
* In origin access, select previously created OAI
* Created origin group
* In failover criteria select all 4xx errors and created origin group
* In the behaviour tab of distribution, click edit and in origin and origin groups select previously created origin group & hit save





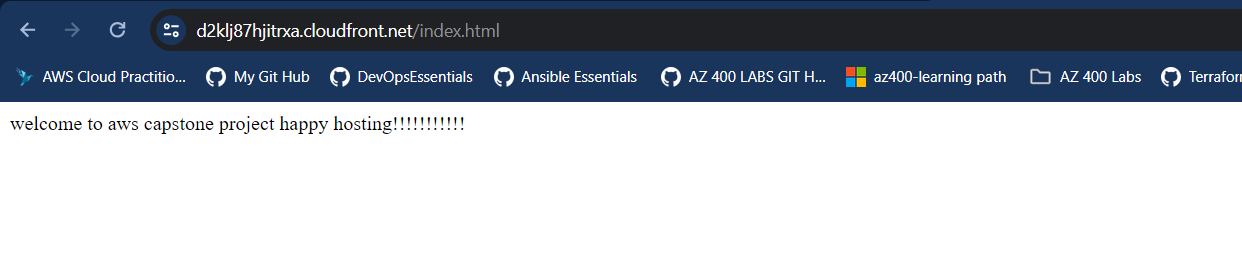


**Task-4 Testing Failover**

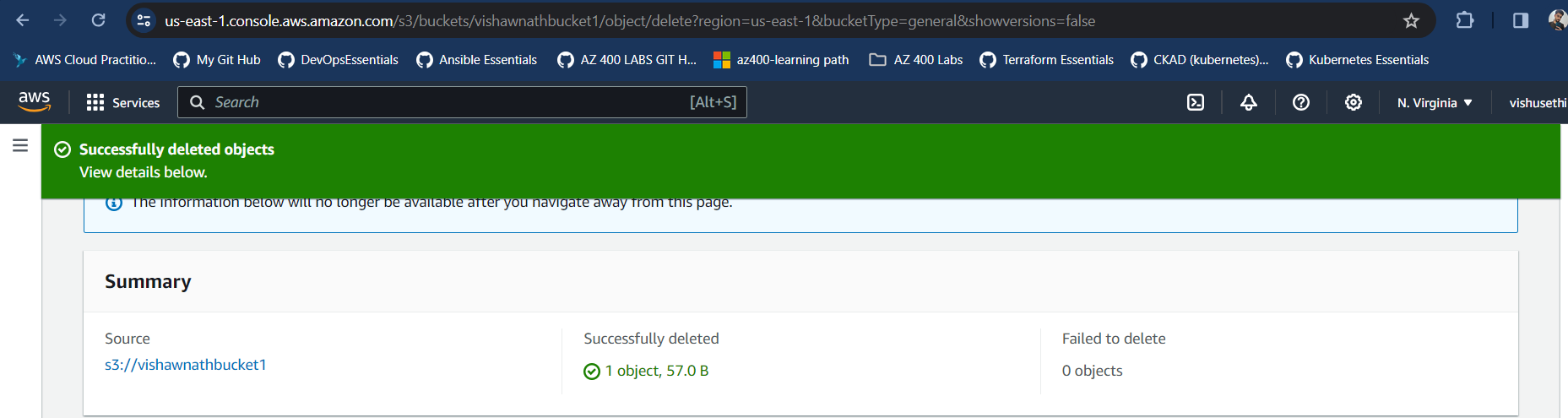
**Objective: Validate that the failover mechanism works as expected when the primary bucket is unavailable.**

**Steps Followed:**

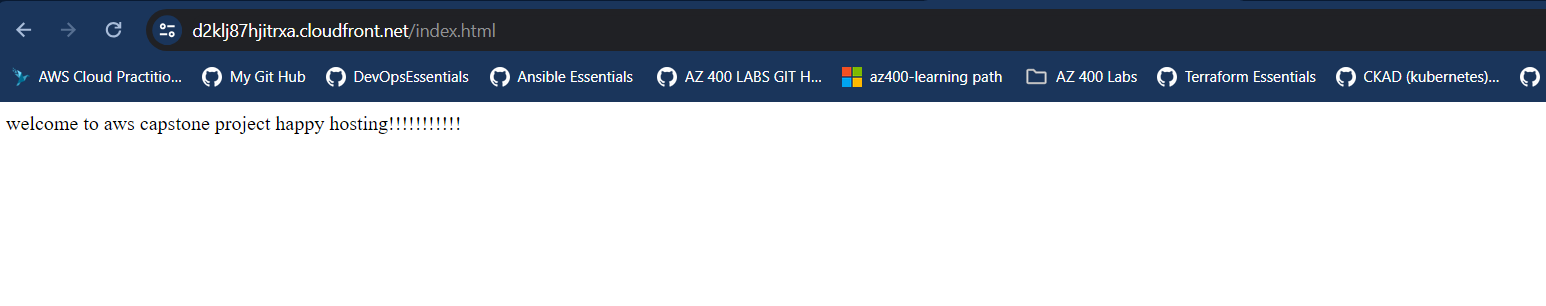
* Copy origin domain and copy/paste the domain in the browser tab by adding/index.html in front of the URL to observe the website
* Website observability



* Deleted index.html from primary s3 bucket



* Website is still accessible; it is fetching the index.html from the secondary region

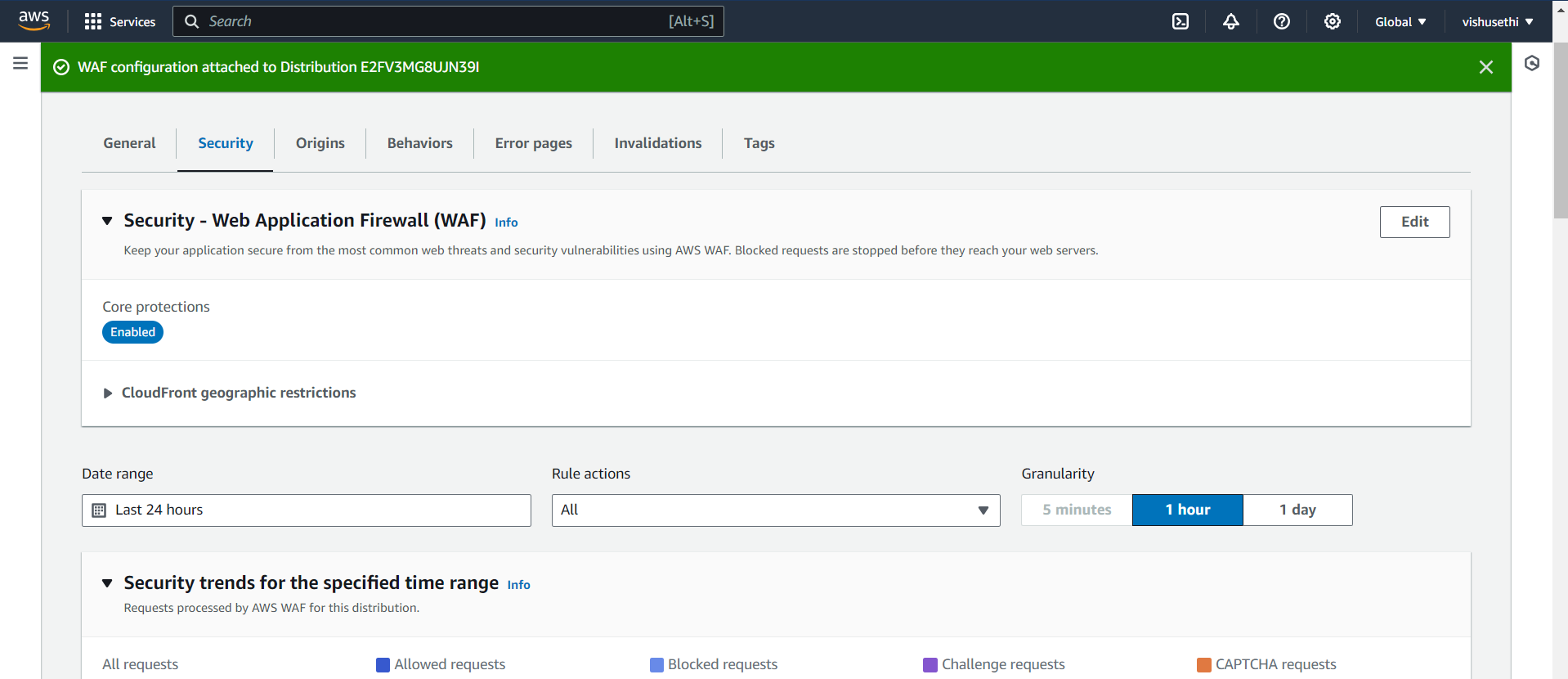


**Task-5 Adding WAF to CloudFront**

**Objective: Implement firewall protection to filter and allow/deny traffic based on predefined rules.**

**Steps Followed:**

* Enabled web application firewall in distributions > security

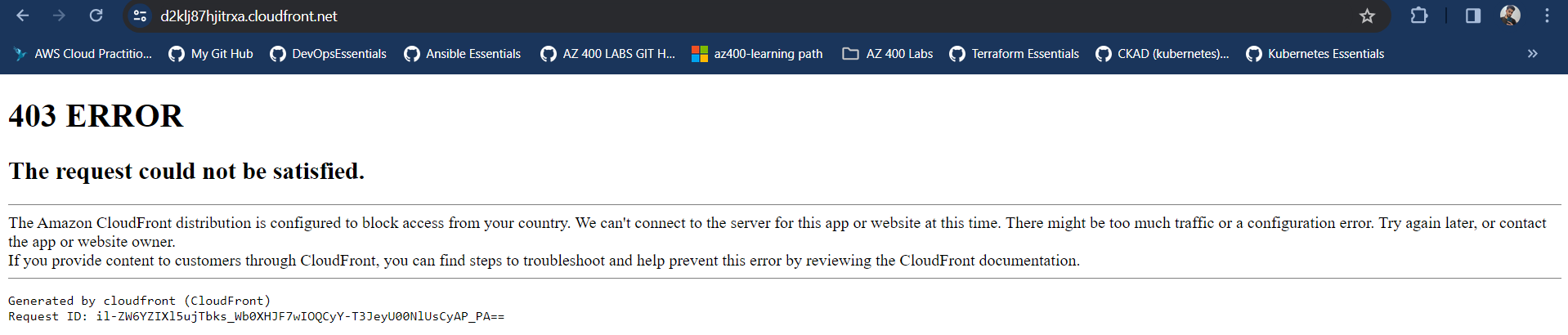


**Task-6 Implementing Geo-location**

**Objective: Restrict access to the website based on geographic location.**

**Steps Followed:**

* In distributions, in security tab edit the geographic restriction section
* Tried to access the website from the blocked location and got the error

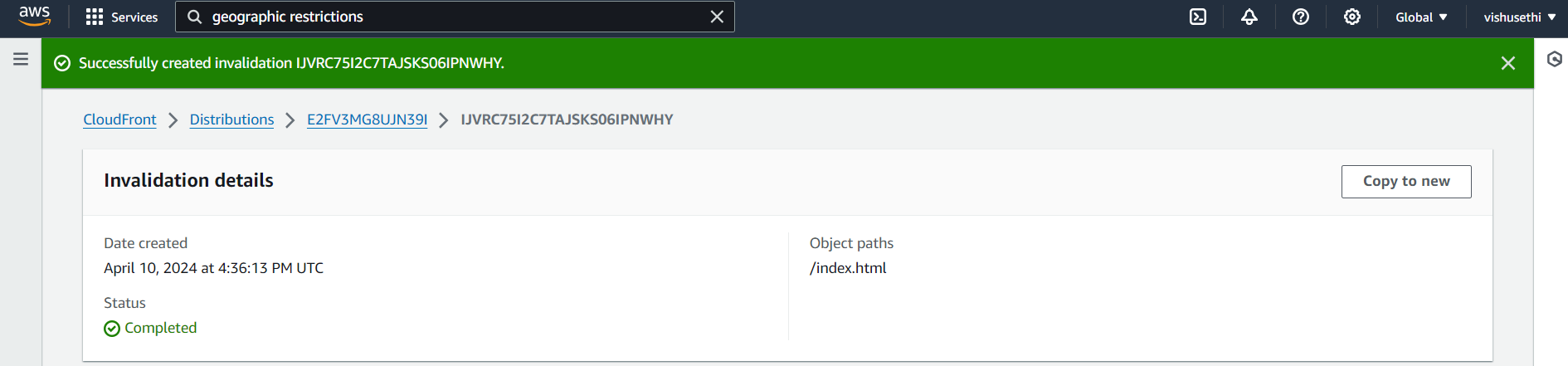


**Task-7 Invalidating data in cache**

**\*\*Objective: Ensure updated content is served by invalidating cached data in CloudFront.**

**Steps Followed:**

* Index.html has been invalidated and has been removed from the edge location

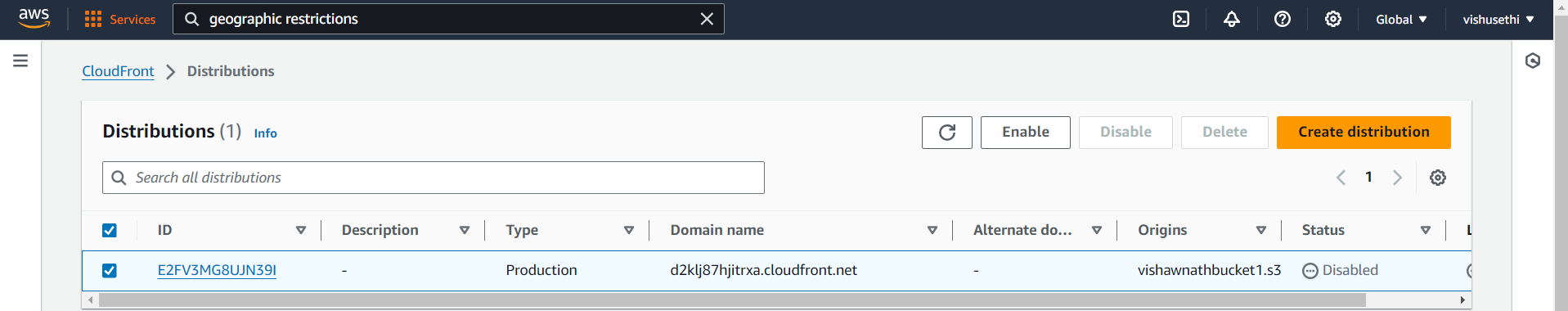


**Task-8 Deleting Resources**

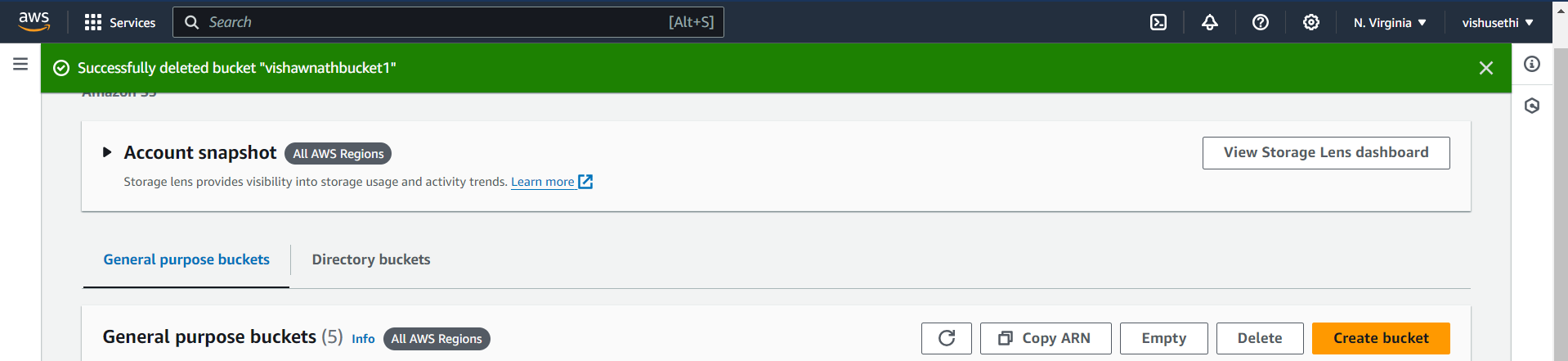
**\*\*Objective: Clean up and delete all resources created in AWS.**

**Steps Followed:**

* Disabled & deleted distribution



* Deleted Primary bucket



* Deleted secondary bucket

