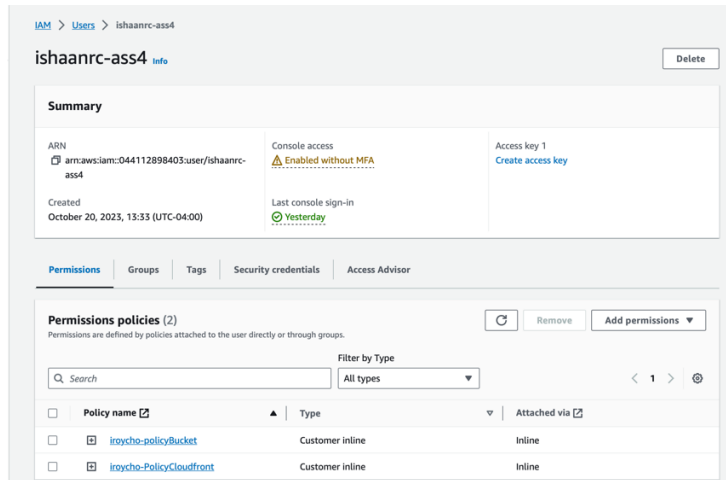


CS 551: Assignment 4

Ishaan Roychowdhury

Part 1: IAM - please submit the json file text of your IAM policies

Here I made a new non-root user and added 2 inline policies (1 for S3 buckets and 1 for CloudFront):



Below is the JSON file text of my IAM policies:

```
(base) ishaanrc@ishaan ~ % aws iam list-user-policies --user-name ishaanrc-ass4
{
  "PolicyNames": [
    "iroycho-policyBucket",
    "iroycho-PolicyCloudfront"
  ]
}
(base) ishaanrc@ishaan ~ % aws iam get-user-policy --user-name ishaanrc-ass4 --policy-name iroycho-policyBucket
{
  "UserName": "ishaanrc-ass4",
  "PolicyName": "iroycho-policyBucket",
  "PolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "s3:*",
          "s3-object-lambda:*"
        ],
        "Resource": "*"
      }
    ]
  }
}
(base) ishaanrc@ishaan ~ % aws iam get-user-policy --user-name ishaanrc-ass4 --policy-name iroycho-PolicyCloudfront
{
  "UserName": "ishaanrc-ass4",
  "PolicyName": "iroycho-PolicyCloudfront",
  "PolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": "cloudfront:*",
        "Resource": "*"
      }
    ]
  }
}
```

Below is the same on AWS console:

[IAM](#) > [Users](#) > [ishaanrc-ass4](#) > Edit policy

Step 1
Modify permissions in iroycho-policyBucket

Step 2
Review and save

Modify permissions in iroycho-policyBucket [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using

Policy editor

Visual

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "s3:*",  
8         "s3-object-lambda:*"  
9       ],  
10      "Resource": "*"   
11    }  
12  ]  
13 }
```

[IAM](#) > [Users](#) > [ishaanrc-ass4](#) > Edit policy

Step 1
Modify permissions in iroycho-PolicyCloudfront

Step 2
Review and save

Modify permissions in iroycho-PolicyCloudfront [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual **JSON** Actions

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "VisualEditor0",  
6       "Effect": "Allow",  
7       "Action": "cloudfront:*",  
8       "Resource": "*"   
9     }  
10  ]  
11 }
```

Edit statement
VisualEditor0 [Remove](#)

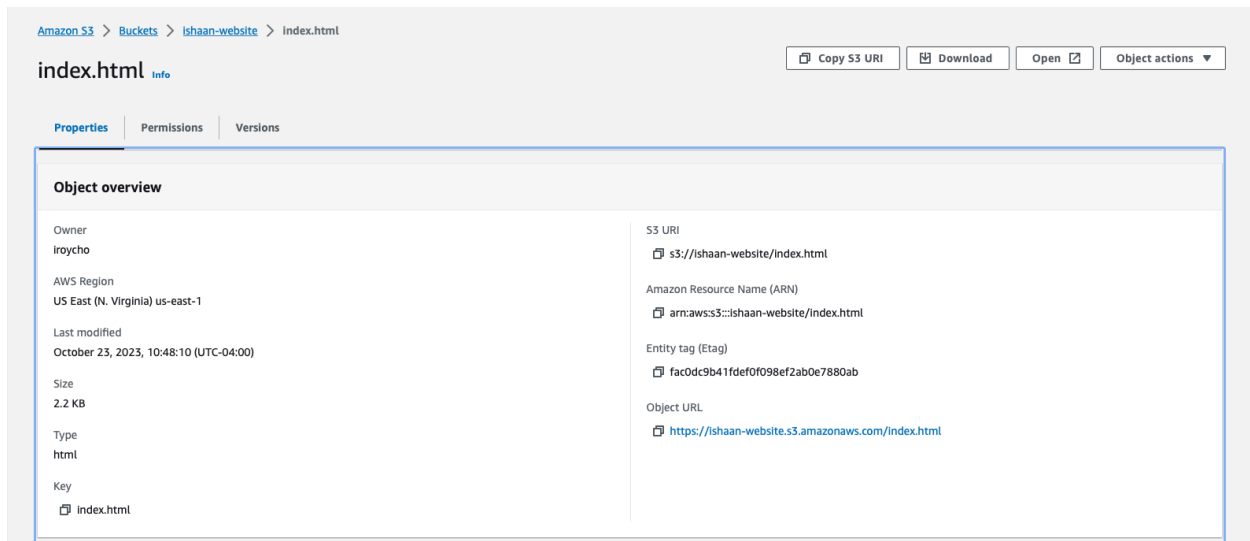
Add actions
Choose a service

Included
CloudFront

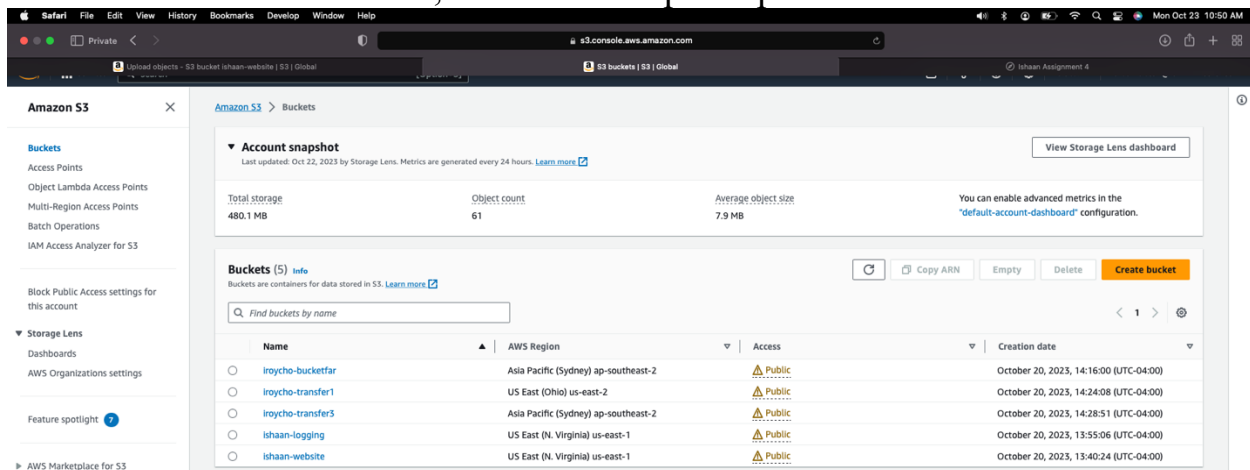
Available
AMP
API Gateway
API Gateway V2
ASC
Access Analyzer
Account
Activate
Alexa for Business

Part 2 S3 - please submit the amazon resource name (arn) of your website bucket and screenshot of your buckets page - you should have at least these 4 buckets by the end of this assignment - 1 log bucket, 1 static website, 2 transfer buckets.

ARN: arn:aws:s3:::ishaan-website/index.html



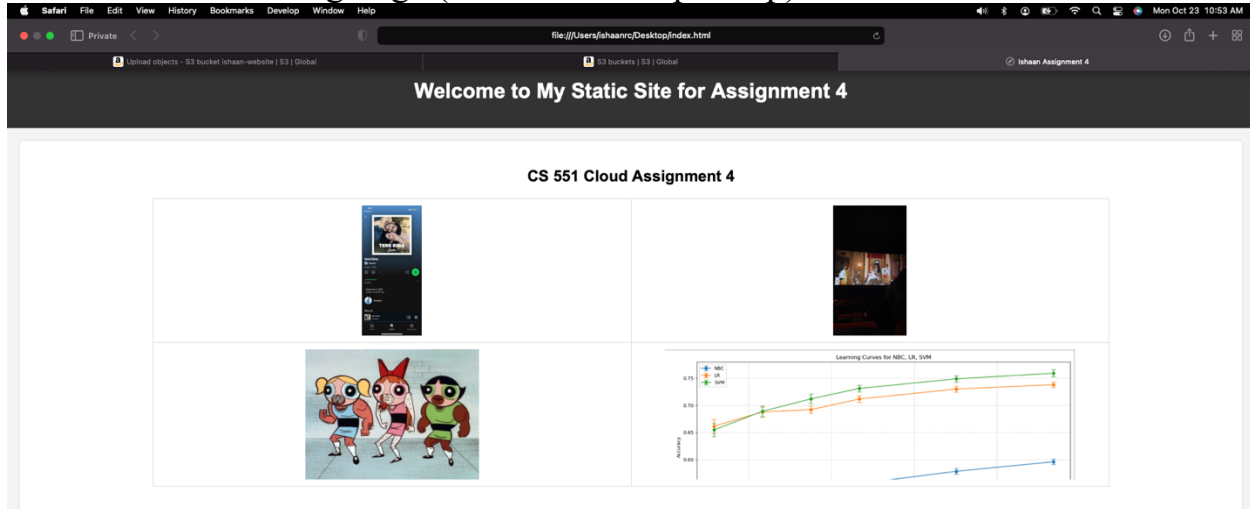
Screenshot of active buckets, with timestamp at top:



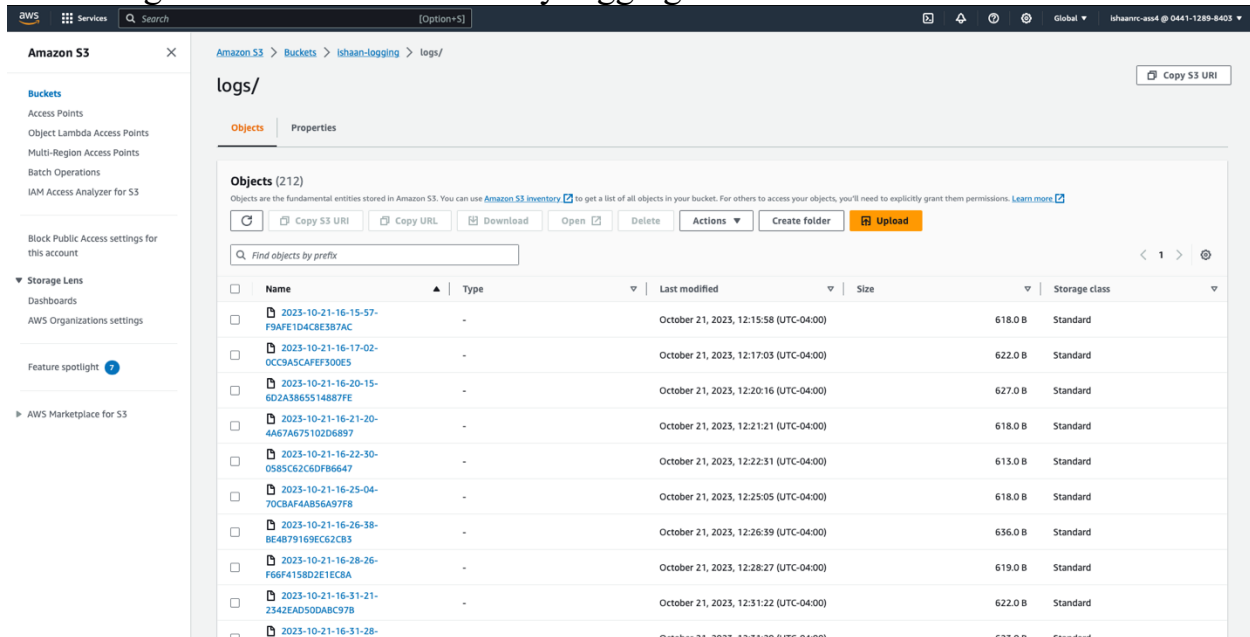
Part 3 Static website - please submit the website url, screenshot of your website landing page, list of log files screenshot inside your log bucket, any log file screenshot with requests made your webpage, the index.html, custom error.html file used as the error page for your static website.

Website Link: <https://ishaan-website.s3.amazonaws.com/index.html>

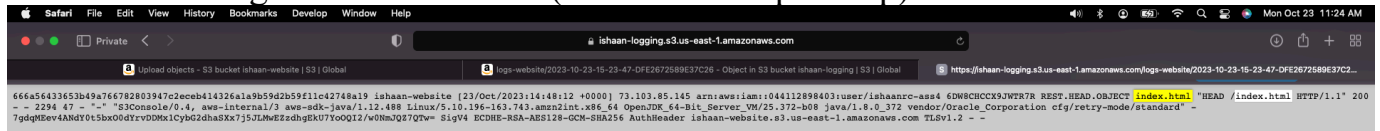
Screenshot of Landing Page (with Timestamp at top):



List of log files screenshot inside my logging bucket :



Screenshot of log file for index.html(with timestamp on top):

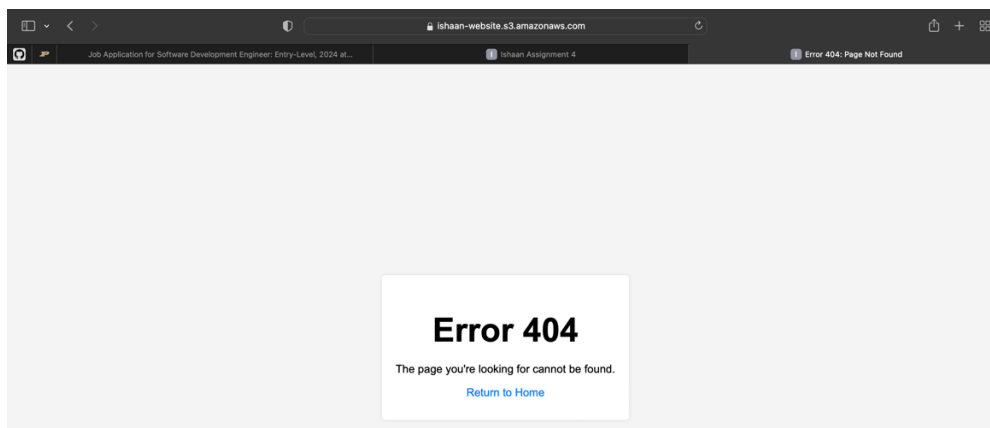


Content of log file for index.html:

```
666a56433653b49a766782803947c2eceb414326a1a9b59d2b59f11c42748a19 ishaan-website [23/Oct/2023:14:48:12 +0000] 73.103.85.145 arn:aws:iam::044112898403:user/ishaanrc-ass4 6DW8CHCCX9JWTR7R REST.HEAD.OBJECT index.html "HEAD /index.html HTTP/1.1" 200 - - 2294 47 - "-" "S3Console/0.4, aws-internal/3 aws-sdk-java/1.12.488 Linux/5.10.196-163.743.amzn2int.x86_64 OpenJDK_64-Bit_Server_VM/25.372-b08 java/1.8.0_372 vendor/Oracle_Corporation cfg/retry-mode/standard" - 7gdqMEev4ANDY0t5bxO0dYrvDDMx1CybG2dhaSXx7j5JLMwEZzdHgEkU7YoOQI2/w0NmJQZ7QTW= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader ishaan-website.s3.us-east-1.amazonaws.com TLSv1.2 - -
```

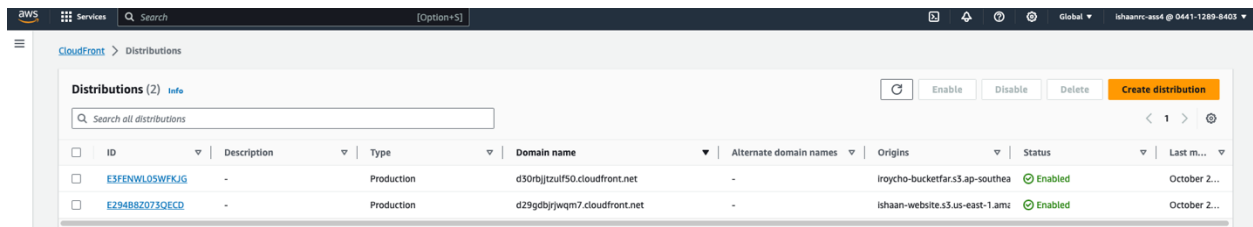
Other logs when searching for custom error.html and screenshot of error:

```
ZPKf/oK9mLVTBSonL8D2qW2ZHtJ81kB6mKBYAL3oSxqIW4GnB2R2R4AmE0xo4jPKtXAWV900GYC= - ECDHE-RSA-AES128-GCM-SHA256 - ishaan-website.s3.amazonaws.com TLSv1.2 - - 666a56433653b49a766782803947c2eceb414326a1a9b59d2b59f11c42748a19 ishaan-website [23/Oct/2023:14:55:49 +0000] 40.94.36.58 - MD8W0M9JWGGHFK4V REST.GET.OBJECT styles.css "GET /styles.css HTTP/1.1" 403 AccessDenied 243 - 45 - "https://ishaan-website.s3.amazonaws.com/error.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36" - RSgu/lgBmFwAufaGebOB3xilVRBmseG6ae8hYx/zqjgsKQEMr+4K/Z6im3IMVHrq46izG3DwSrE= - ECDHE-RSA-AES128-GCM-SHA256 - ishaan-website.s3.amazonaws.com TLSv1.2 - - 666a56433653b49a766782803947c2eceb414326a1a9b59d2b59f11c42748a19 ishaan-website [23/Oct/2023:14:55:50 +0000] 40.94.36.58 - V7XZPXEXZBMA55RH REST.GET.OBJECT 8BC40049-EB63-4011-B14A-5E146F189B19_4_5005_c.jpeg "GET /8BC40049-EB63-4011-B14A-5E146F189B19_4_5005_c.jpeg HTTP/1.1" 404 - 54883 54883 41 40 "https://ishaan-website.s3.amazonaws.com/error.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36" - hXh/7ZcXwtNCZhW7R9bstysxgK4fleo4y+7W9ggBsFuD0HZofmtjutxBLenDi3R/9NXrrxqjqCE= - ECDHE-RSA-AES128-GCM-SHA256 - ishaan-website.s3.amazonaws.com TLSv1.2 -
```



Part 4 Cloudfront - please submit the cloudfront url origin and distribution endpoint.

Screenshot of Cloudfront distributions:



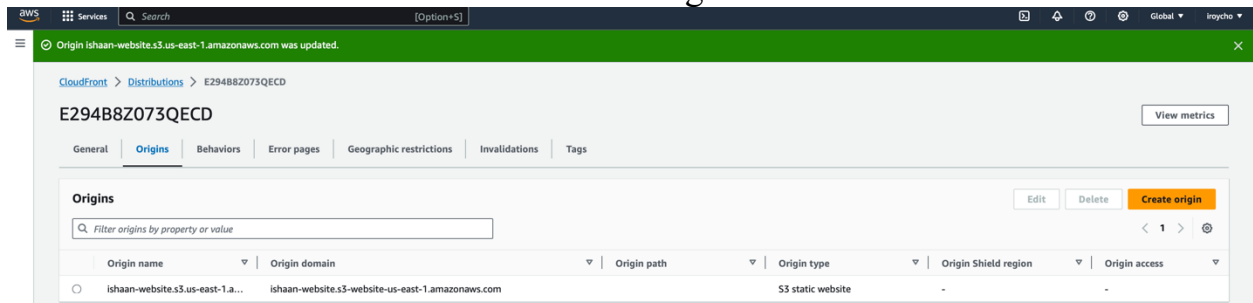
The screenshot shows the AWS CloudFront console with the 'Distributions' tab selected. There are two distributions listed. The second distribution, E294B8Z073QECD, is the one of interest.

ID	Description	Type	Domain name	Alternate domain names	Origins	Status	Last modified
E3FENWLO5WFKJG	-	Production	d3orbjztul50.cloudfront.net	-	lroycho-bucketfar.s3.ap-southea	Enabled	October 2...
E294B8Z073QECD	-	Production	d29gdbjrjwqm7.cloudfront.net	-	ishaan-website.s3.us-east-1.amz	Enabled	October 2...

In this screenshot, we will look at the second distribution we made that takes care of our website made in parts above. Our distribution endpoint as we can see in the screenshot is:

d29gdbjrjwqm7.cloudfront.net

Screenshot of details of distribution under origins tab:



The screenshot shows the details of the distribution E294B8Z073QECD, specifically the 'Origins' tab. There is one origin listed.

Origin name	Origin domain	Origin path	Origin type	Origin Shield region	Origin access
ishaan-website.s3.us-east-1.a...	ishaan-website.s3-website-us-east-1.amazonaws.com		S3 static website	-	-

As we can see out cloudfront url origin is:

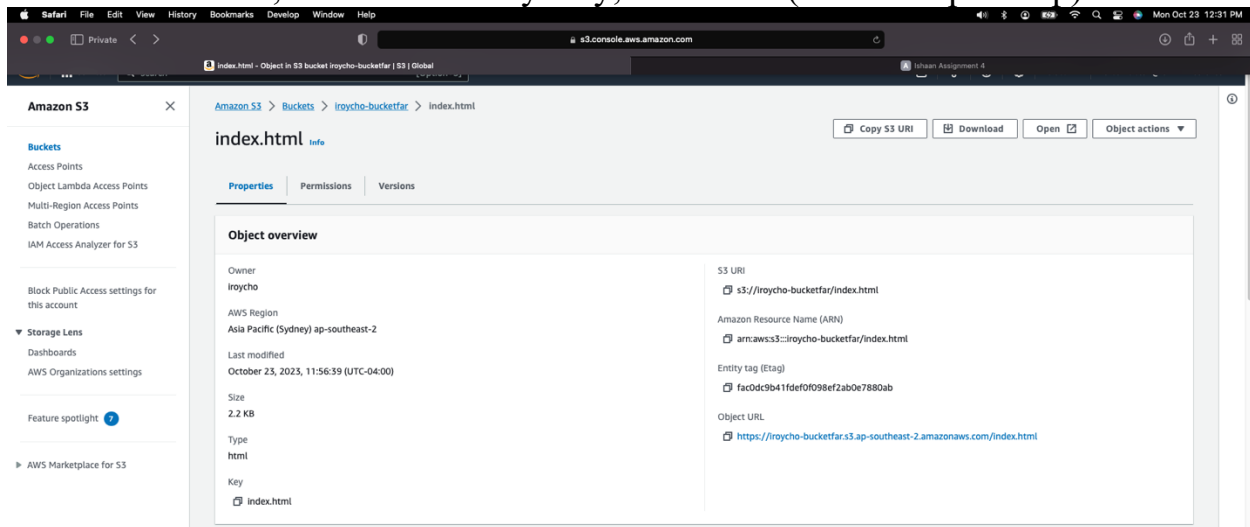
http://ishaan-website.s3-website-us-east-1.amazonaws.com

Part 5 CDN region - submit the S3 website url and CloudFront distribution for your static website in the different region. Include the time taken to load for two different regions, you can check this in the network tab.

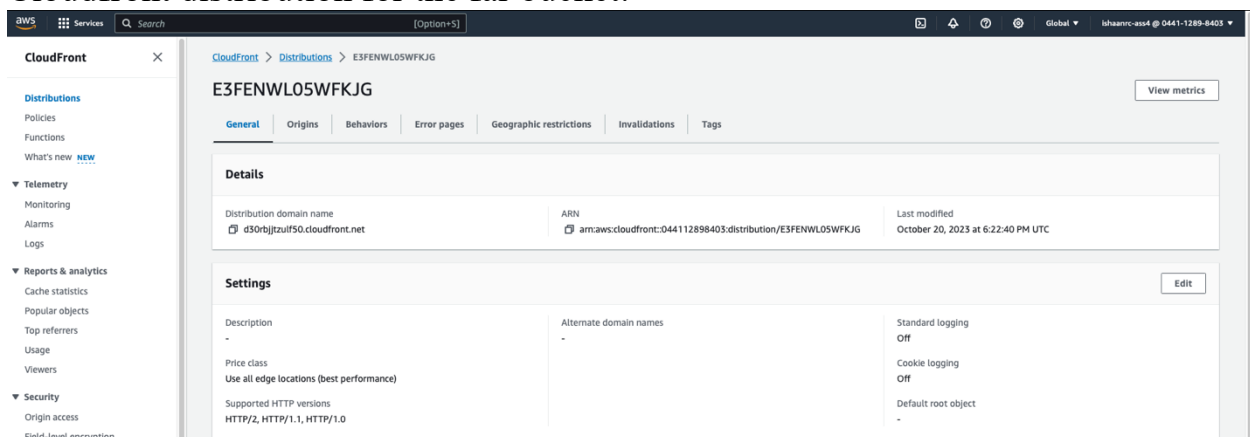
Website URL for static website in a different region:

<https://iroycho-bucketfar.s3.ap-southeast-2.amazonaws.com/index.html>

Screenshot of my new bucket with the same index.html file, but this time instead of North America, it is hosted in Sydney, Australia. (Timestamp on top)



Cloudfront distribution for the far bucket:

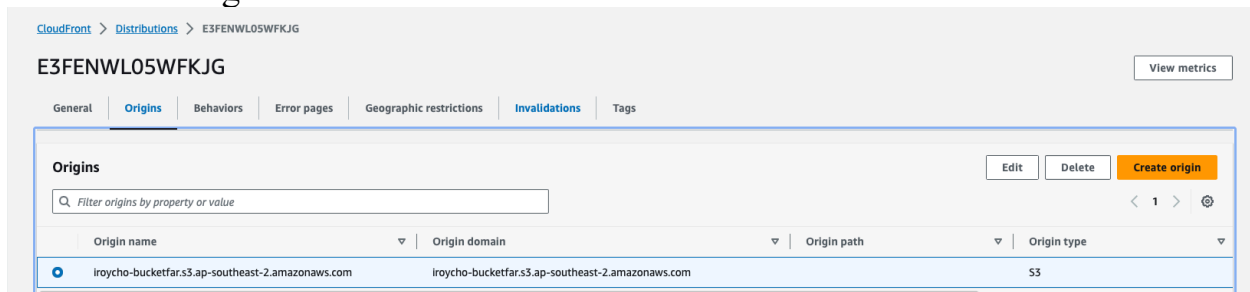


Distribution endpoint: d30rbjitzulf50.cloudfront.net

Cloudfront URL origin:

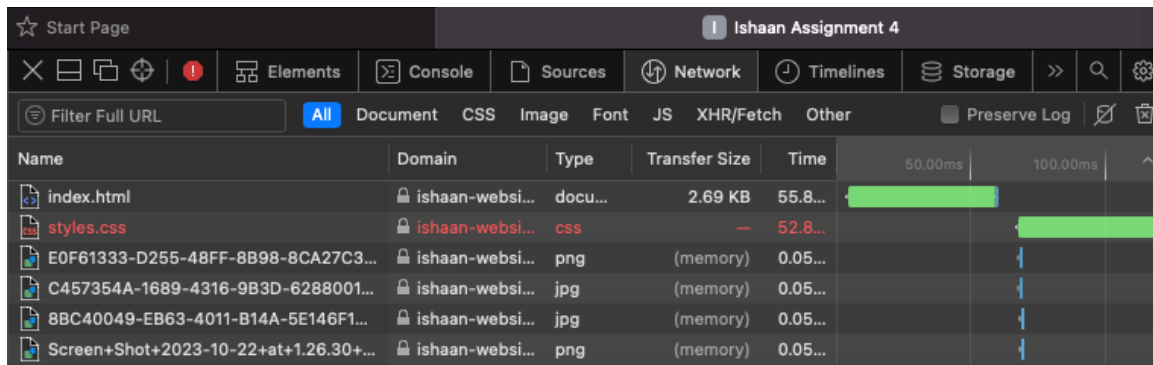
iroycho-bucketfar.s3.ap-southeast-2.amazonaws.com

Cloudfront origin details for the far bucket:

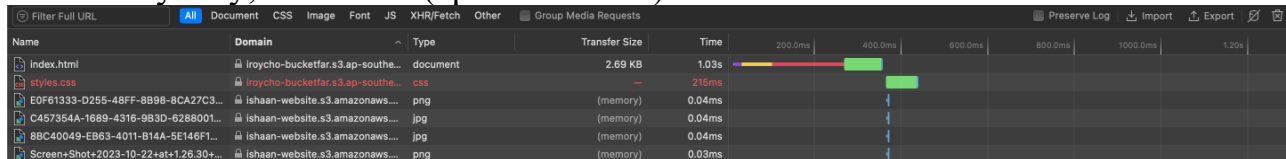


Comparison of load times for both the different regions:

This if for us-east-1:





This for Sydney, Australia: (ap-southeast-2)



We can see a stark difference in the 2. The one in us-east-1 loads the index.html in 55ms while index.html in Sydney takes 1.03 seconds to load. This is more than 18 times the time taken. This occurs primarily because of latency. The physical distance between me and the server can result in increased latency. Since I am sitting in US, the data will have to travel a longer distance to reach Sydney, resulting in higher latency.

Part 6 Transfer - please use the stated tool in your analysis and keep your answer terse. Include Accelerated S3 bucket speed test URLs and screenshots of the tool with speed difference for the two buckets. Include screenshots of both the transfer buckets with the object you uploaded, screenshot should have the file size

Screenshot of my 2 transfer accelerated buckets:

<input type="radio"/> iroycho-transfer1	US East (Ohio) us-east-2		October 20, 2023, 14:24:08 (UTC-04:00)
<input type="radio"/> iroycho-transfer3	Asia Pacific (Sydney) ap-southeast-2		October 20, 2023, 14:28:51 (UTC-04:00)

Speed Test URL for my transfer-acceleration-bucket 1:

<https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html?region=us-east-2&origBucketName=iroycho-transfer1>

Speed Test URL for my transfer-acceleration-bucket 2: (name of bucket is iroycho-transfer3)

<https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html?region=ap-southeast-2&origBucketName=iroycho-transfer3>

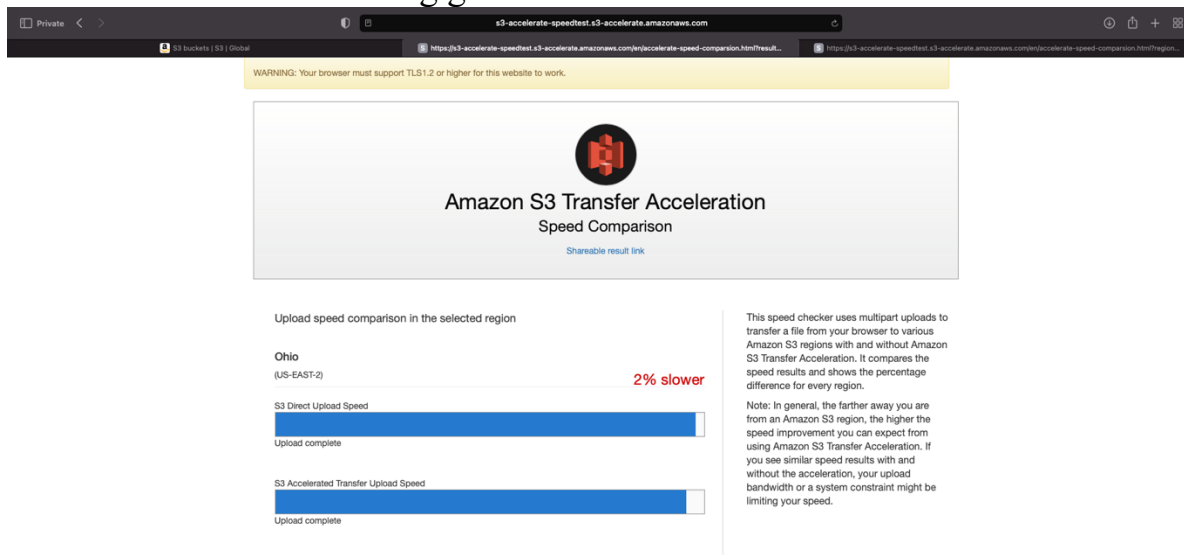
Result for bucket 1:

<https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html?result=00742-10118-06871-15847&identityId=unknown>

Result for bucket 2:

<https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html?result=25419-3494-26275-00636&identityId=unknown>

Screenshot of bucket 1 using given tool:

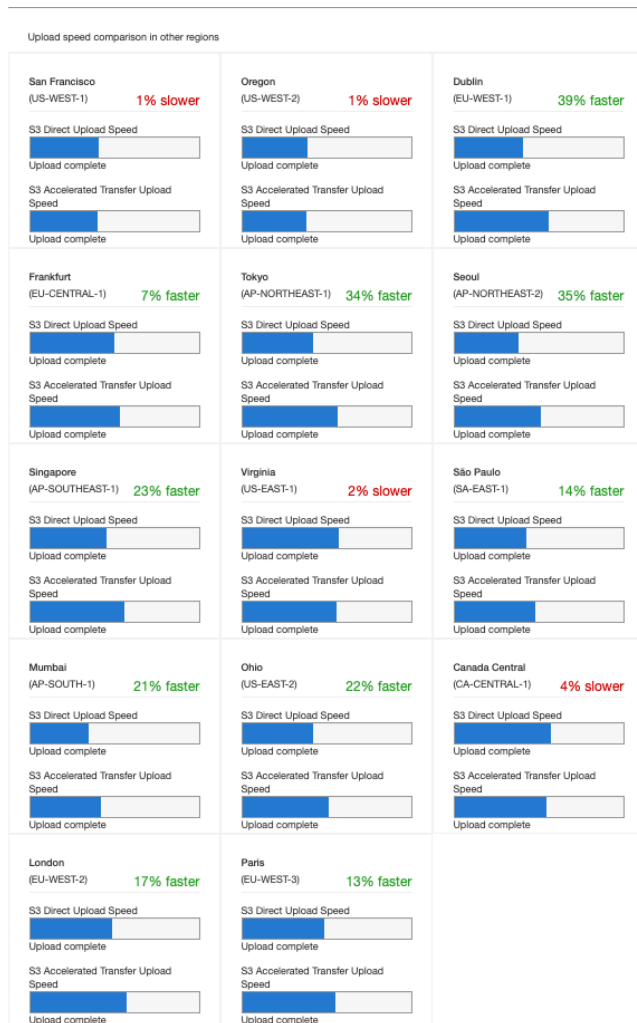
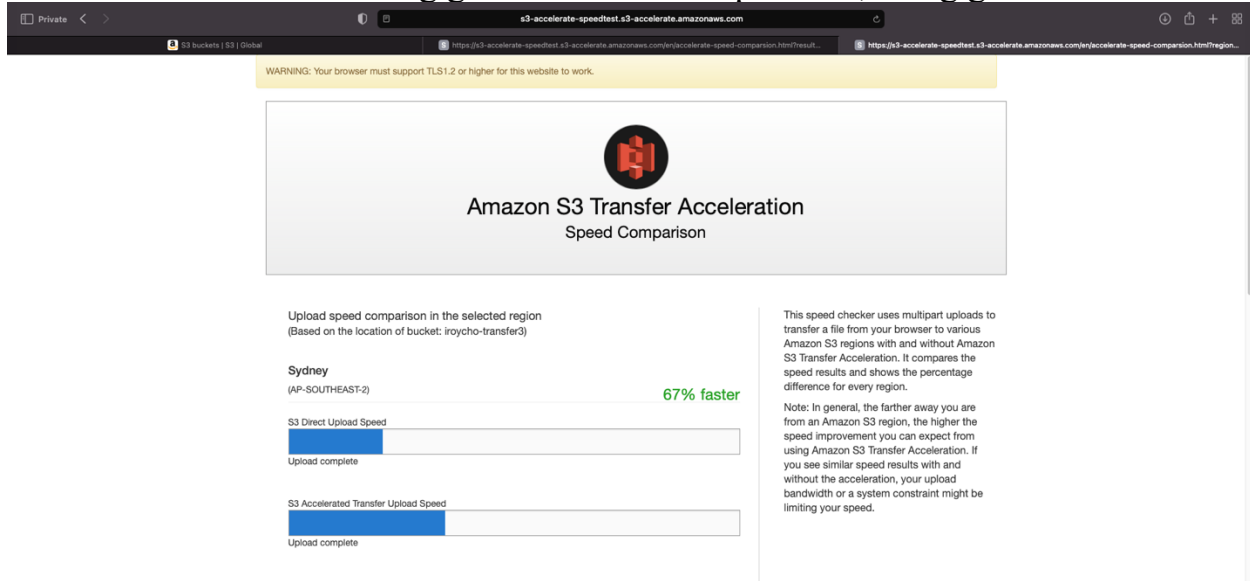


Screenshot of comparisons that I got from the tool: (For bucket 1)

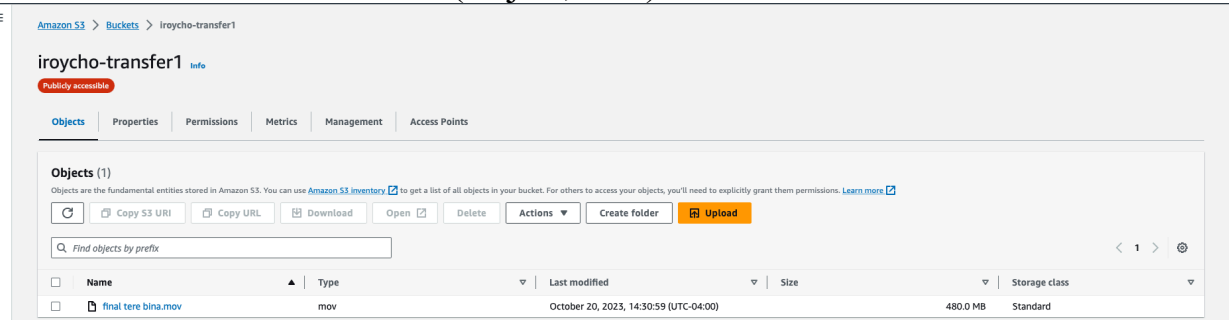
Upload speed comparison in other regions



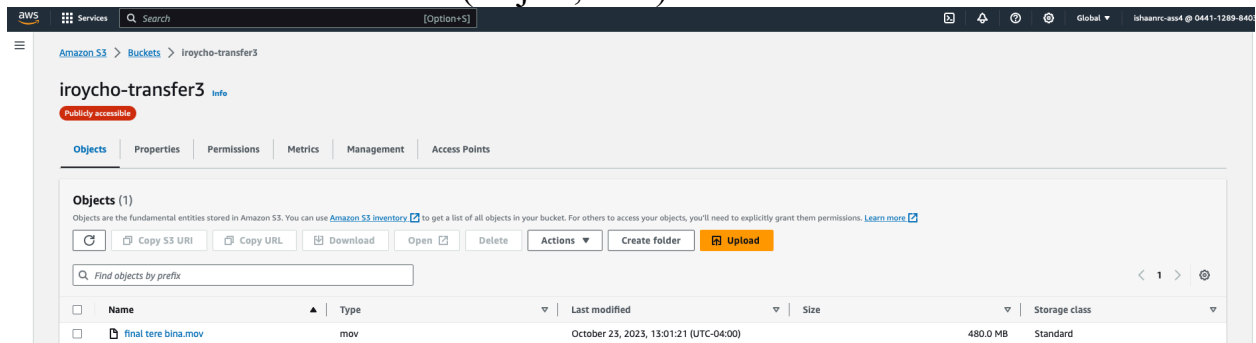
Screenshot of bucket 2 using given tool and comparisons, using given tool:



Screenshot of Bucket 1 details (Object, Size):



Screenshot of Bucket 2 details (Object, Size):



I uploaded a video close to 500 MB and did the analysis. Attached are the links for results that you can see for yourself.

Part 7 Free tier - your analysis of free tier resource usage, a screenshot of aws free tier page under billing with details which will have service and usage information.

A screenshot of my AWS Free Tier Resource Usage under billing:

Service	AWS Free Tier usage limit	Current usage	Forecasted usage	MTD actual usage %	MTD forecasted usage %
Amazon Elastic Compute Cloud	750.0 Hrs for free for 12 months as part of AWS Free Usage Tier (Global-BoxUsage:freetier.micro)	537 Hrs	724 Hrs	71.67%	96.59%
Amazon EC2 Container Registry (ECR)	0.5 GB-Mo for free for 12 months as part of AWS Free Usage Tier (Global-TimedStorage-ByteHrs)	0 GB-Mo	0 GB-Mo	65.54%	88.34%
Amazon Elastic Compute Cloud	30.0 GB-Mo for free for 12 months as part of AWS Free Usage Tier (Global-EBS:VolumeUsage)	6 GB-Mo	8 GB-Mo	19.10%	25.74%
Amazon Simple Storage Service	2000.0 Requests for free for 12 months as part of AWS Free Usage Tier (Global-Requests-Tier1)	305 Requests	411 Requests	15.25%	20.55%
AmazonCloudWatch	10.0 Metrics are always free per month as part of AWS Free Usage Tier (Global-CW:MetricMonitorUsage)	1 Metrics	1 Metrics	9.89%	13.33%
Amazon Simple Storage Service	20000.0 Requests for free for 12 months as part of AWS Free Usage Tier (Global-Requests-Tier2)	1,462 Requests	1,971 Requests	7.31%	9.85%
AmazonCloudWatch	10.0 Alarms are always free per month as part of AWS Free Usage Tier (Global-CW:AlarmMonitorUsage)	0 Alarms	0 Alarms	1.27%	1.71%
Amazon Simple Storage Service	5.0 GB-Mo for free for 12 months as part of AWS Free Usage Tier (Global-TimedStorage-ByteHrs)	0 GB-Mo	0 GB-Mo	0.60%	0.82%
AmazonCloudWatch	5.0 GB are always free per month as part of AWS Free Usage Tier (Global-DataProcessing-Bytes)	0 GB	0 GB	0.15%	0.21%
AWS Data Transfer	1.0 GB are always free per month as part of AWS Free Usage Tier (Global-DataTransfer-Regional-Bytes)	0 GB	0 GB	0.12%	0.16%

Below is my analysis:

1. Amazon Elastic Compute Cloud (EC2)

- **Free Tier Limit:** 750 hours per month for 12 months.
- **Current Usage:** 537 hours.
- **Forecasted Usage:** 724 hours.

This means I am close to my monthly usage. I should be under the 750 mark though, which means I will not have to pay anything.

2. Amazon EC2 Container Registry (ECR)

- **Free Tier Limit:** 0.5 GB-month for 12 months.
- **Current Usage:** 0 GB-month.

3. Amazon Elastic Compute Cloud (EBS:VolumeUsage)

- **Free Tier Limit:** 30 GB-month for 12 months.
- **Current Usage:** 6 GB-month.
- **Forecasted Usage:** 8 GB-month.

I am well within the limits for EBS usage. I will again not pay anything.

4. Amazon Simple Storage Service (S3) (Requests-Tier1)

- **Free Tier Limit:** 2000 requests per month for 12 months.
- **Current Usage:** 305 requests.
- **Forecasted Usage:** 411 requests.

Since I used S3 for this assignment, I used up a minor fraction of the allowed budget for requests. This means this is again 0 dollars.

5. Amazon Simple Storage Service (S3) (Global-Requests-Tier2)

- **Free Tier Limit:** 20000 requests for 12 months.
- **Current Usage:** 1,462 requests.
- **Forecasted Usage:** 1,971 requests.

Again, I am within the free tier limit for this S3 request tier. The rest are in the image, and I am well within the bounds. I have stopped and deleted all containers, repositories, S3 objects, Cloudfront distributions, etc. So, I should not be paying anything for this assignment.

Below is a screenshot of my costs: (It was \$3.19 since in the last assignment, I used Fargate since I had a M1 chip and x86 Docker Images weren't being built in the free tier of EC2)

Charges by service

Charges by account

Invoices

Savings

Taxes by service

Amazon Web Services, Inc. charges by service

info

Expand all

Total active services

7

Total pre-tax service charges in USD

USD 3.19

Q Filter by service name or region name

< 1 >

Description	Usage Quantity	Amount in USD
<div><div></div>Elastic Container Service</div>		USD 3.19
<div><div><div></div>US East (Ohio)</div></div>		USD 3.19
<div><div><div><div></div>Amazon Elastic Container Service USE2-Fargate-GB-Hours</div></div></div>		USD 0.79
<div><div><div>AWS Fargate - Memory - US East 2 (Ohio)</div></div></div>	178.188 hours	USD 0.79
<div><div><div><div></div>Amazon Elastic Container Service USE2-Fargate-vCPU-Hours:perCPU</div></div></div>		USD 2.40
<div><div><div>AWS Fargate - vCPU - US East 2 (Ohio)</div></div></div>	59.396 hours	USD 2.40
<div><div><div></div>CloudWatch</div></div>		USD 0.00
<div><div><div></div>Data Transfer</div></div>		USD 0.00
<div><div><div></div>EC2 Container Registry (ECR)</div></div>		USD 0.00
<div><div><div></div>Elastic Compute Cloud</div></div>		USD 0.00
<div><div><div></div>Simple Storage Service</div></div>		USD 0.00
<div><div><div></div>Virtual Private Cloud</div></div>		USD 0.00
<div><div>Total tax</div></div>		USD 0.00

This time as we can see our S3 costs are 0 dollars since we are within the free tier bounds as mentioned above.

Part 8: Public cloud vendors provide managed services and customer managed services. While not strictly related to the concepts in this course, It's important to distinguish the responsibilities of the public cloud vendor and the customer. S3 provides server-side encryption (SSE) that is either AWS managed or customer managed.

a)List the different SSE services.

1. **SSE-S3**: Server-Side Encryption with Amazon S3-Managed Keys.
2. **SSE-KMS**: Server-Side Encryption with AWS Key Management Service keys.
3. **DSSE-KMS**: Dual-Layer Server-Side Encryption with AWS KMS keys.
4. **SSE-C**: Server-Side Encryption with Customer-Provided Keys.

b)Please point out which services are customer managed and Amazon managed services.

- **Amazon Managed:**
 - **SSE-S3**: Amazon uses AES-256 encryption and handles the key management.
 - **SSE-KMS**: Uses AWS Key Management Service, which is managed by AWS but provides more control and flexibility to users over their encryption keys.
 - **DSSE-KMS**: This is an advanced option that provides dual-layer encryption with AWS KMS. Although, AWS manages the infrastructure, users have control over their KMS keys.
- **Customer Managed:**
 - **SSE-C**: The customer manages and provides their encryption keys.

c)What could be the trade-offs of having you the customer manage this versus AWS, List 4 tradeoffs.

1. **Complexity:**
 - **Customer-Managed**: Handling users handle their own keys introduces additional complexity regarding key creation, and secure storage.
 - **AWS-Managed**: AWS takes away a significant portion of this complexity, which is beneficial for users lacking key management expertise.

2. Key Losing/Stolen/Unintended Access Risk:

- **Customer-Managed:** The integrity of key security is on the customer. Losing a key would mean the corresponding encrypted data is irretrievable. Plus, there's a potential risk of unintended access if the key goes to someone else.
- **AWS-Managed:** AWS assumes the responsibility of key management and they ensure high levels of security, which minimizes key mismanagement chances.

3. Potential Added Cost v/s Convenience:

- **Customer Managed:** Managing keys on your own may potentially reduce costs, but it could lead to added expenses related to building, maintaining, and auditing a secure key management infrastructure.
- **AWS Managed:** AWS Key Management Service might incur additional costs based on the number of keys. However, it provides a default integration with most AWS services, making it convenient.

4. Scalability v/s Granularity:

- **Customer Managed:** While managing keys on your own gives you more granular control, it might be more challenging to scale the infrastructure quickly to accommodate rapid growth. The customer will need to anticipate scaling needs, which might use extra hardware, software, and human resources.
- **AWS Managed:** Using AWS's infrastructure for key management allows for easy scalability. As our organization grows and we require more keys, AWS can handle this growth without requiring any intervention. This scalability can be a significant advantage for organizations experiencing rapid growth.