S3 Performance Analysis Report

Group Members:

- Clement Owusu Bempah
- Marie-Pearl Otoo
- Michael Aboagye
- Samuel Kofi Asare Dwumah
- Vera Sekyere

Contents

Introduction

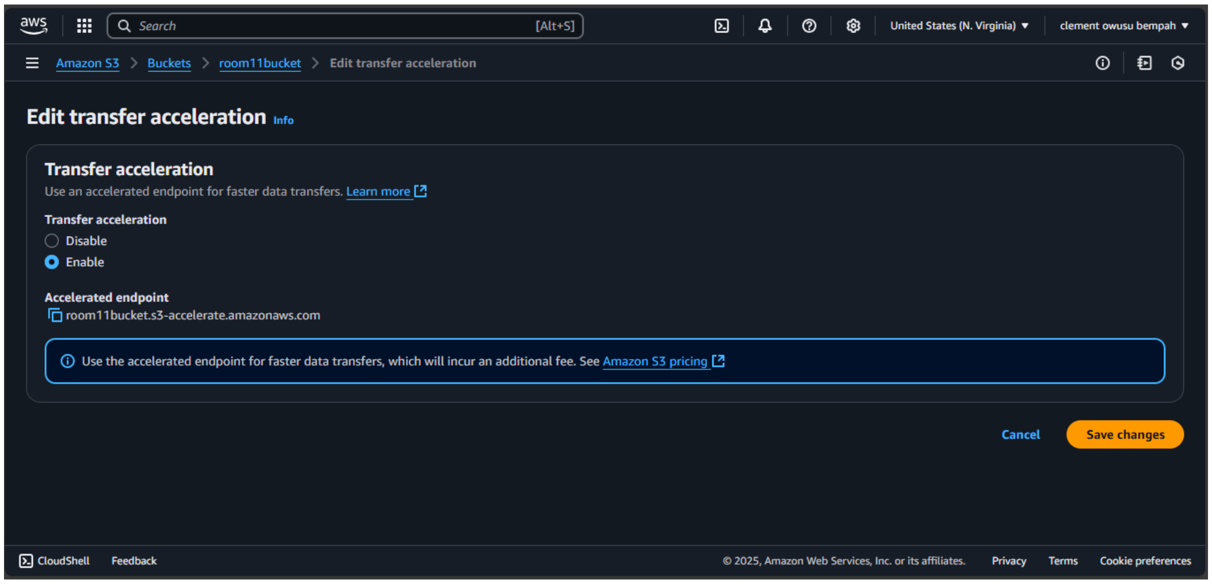
This report summarizes key findings from an analysis of Amazon S3 performance, focusing on metrics, tools, and features used, and provides actionable recommendations for improvement. The analysis covers S3 Transfer Acceleration, activity metrics (GetRequests, PutRequests, BytesDownloaded), and status code metrics (200OKStatusCount, 403ForbiddenErrorCount, 404NotFoundErrorCount).

1. Transfer Acceleration Review

S3 Transfer Acceleration leverages Amazon CloudFront's globally distributed edge locations to optimize data transfers to and from S3 buckets. When enabled, data is routed through the nearest edge location, which then uses optimized network paths to transfer the data to the S3 bucket. This significantly improves upload and download speeds, especially over long geographical distances, by reducing latency and leveraging Amazon's high-speed backbone network.

Enabling Transfer Acceleration:
As demonstrated in the provided screenshots, Transfer Acceleration can be enabled directly from the S3 bucket properties in the AWS Management Console. Navigate to the bucket, select the "Properties" tab, and then locate the "Transfer acceleration"

section to enable it.
Alternatively, it can be configured programmatically via the AWS CLI or SDKs using commands like aws configure set default.s3.use_accelerate_endpoint true for uploads and downloads.


Amazon S3 > Buckets > room11bucket > Edit transfer acceleration

Edit transfer acceleration Info

Transfer acceleration
Use an accelerated endpoint for faster data transfers. Learn more

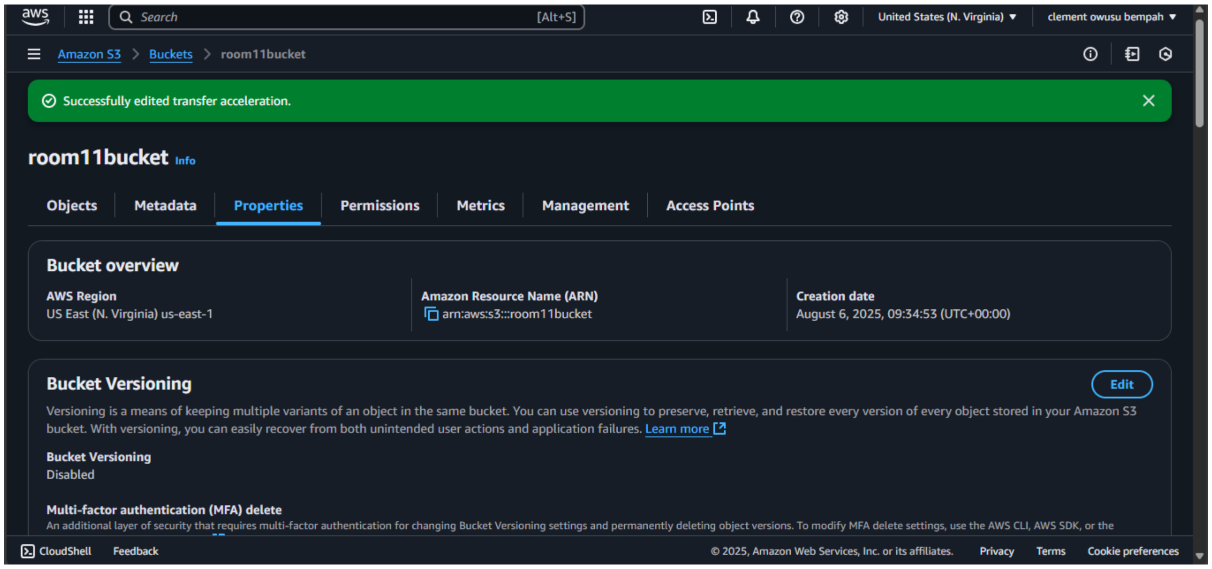
Transfer acceleration
Disable [✓ Enable]

Accelerated endpoint
room11bucket.s3-accelerate.amazonaws.com

Use the accelerated endpoint for faster data transfers, which will incur an additional fee. See Amazon S3 pricing

Cancel | Save changes

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Amazon S3 > Buckets > room11bucket

Successfully edited transfer acceleration.

room11bucket Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Bucket overview

AWS Region	Amazon Resource Name (ARN)	Creation date
US East (N. Virginia) us-east-1	arn:aws:s3:::room11bucket	August 6, 2025, 09:34:53 (UTC+00:00)

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Disabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the

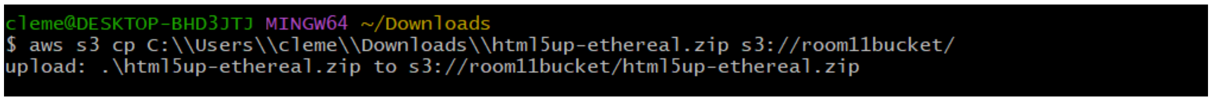
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Pros:

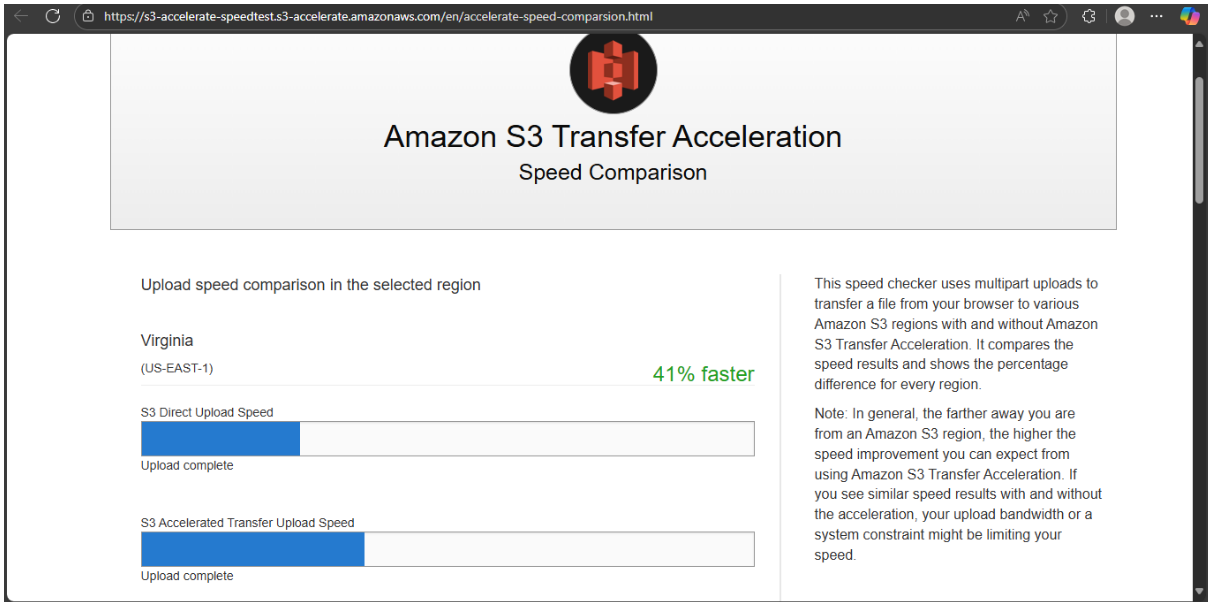
- **Faster Transfers:** Substantially improves data transfer speeds, particularly for users geographically distant from the S3 bucket's region.
- **Improved User Experience:** Reduces wait times for large file uploads and downloads, leading to a better experience for end-users.
- **Global Reach:** Benefits applications with a globally dispersed user base.

Cons:

- **Additional Cost:** Transfer Acceleration incurs an additional fee, which is based on the amount of data transferred through the accelerated endpoint.
- Not Always Beneficial: For users located close to the S3 bucket's region, the performance improvement might be minimal and not justify the extra cost. The speed comparison tool showed a 41% faster upload in Virginia (US-EAST-1), indicating its effectiveness in certain scenarios.



```
cleme@DESKTOP-BHD3JTJ MINGW64 ~/Downloads
$ aws s3 cp C:\\Users\\cleme\\Downloads\\html5up-ethereal.zip s3://room11bucket/
upload: .\html5up-ethereal.zip to s3://room11bucket/html5up-ethereal.zip
```


https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html

Amazon S3 Transfer Acceleration

Speed Comparison

Upload speed comparison in the selected region

Virginia

(US-EAST-1)

41% faster

S3 Direct Upload Speed

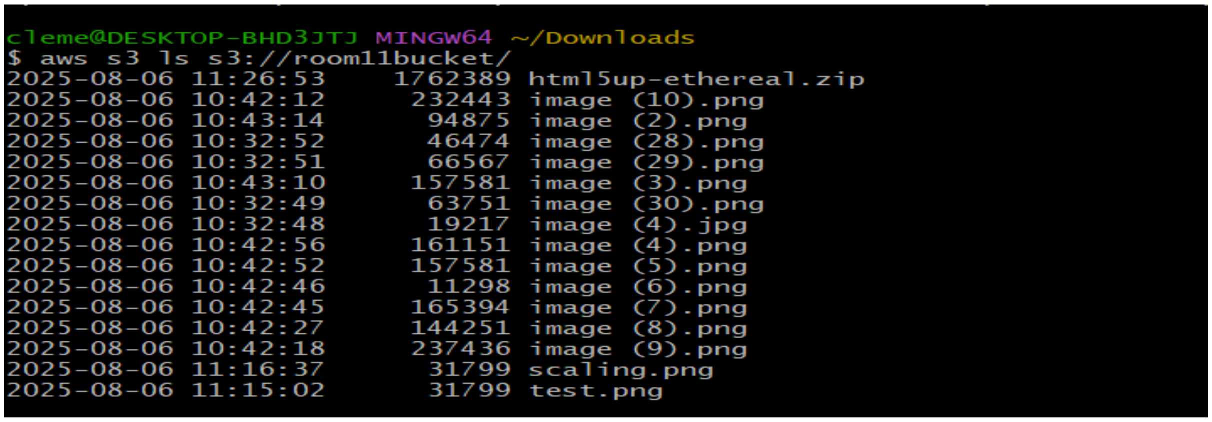
Upload complete

S3 Accelerated Transfer Upload Speed

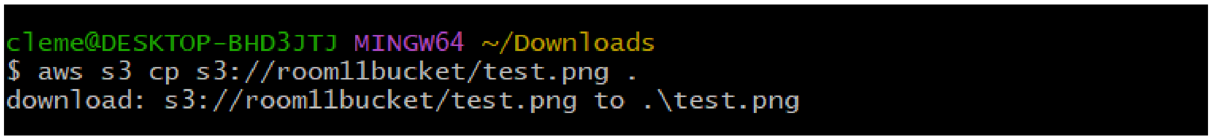
Upload complete

This speed checker uses multipart uploads to transfer a file from your browser to various Amazon S3 regions with and without Amazon S3 Transfer Acceleration. It compares the speed results and shows the percentage difference for every region.

Note: In general, the farther away you are from an Amazon S3 region, the higher the speed improvement you can expect from using Amazon S3 Transfer Acceleration. If you see similar speed results with and without the acceleration, your upload bandwidth or a system constraint might be limiting your speed.



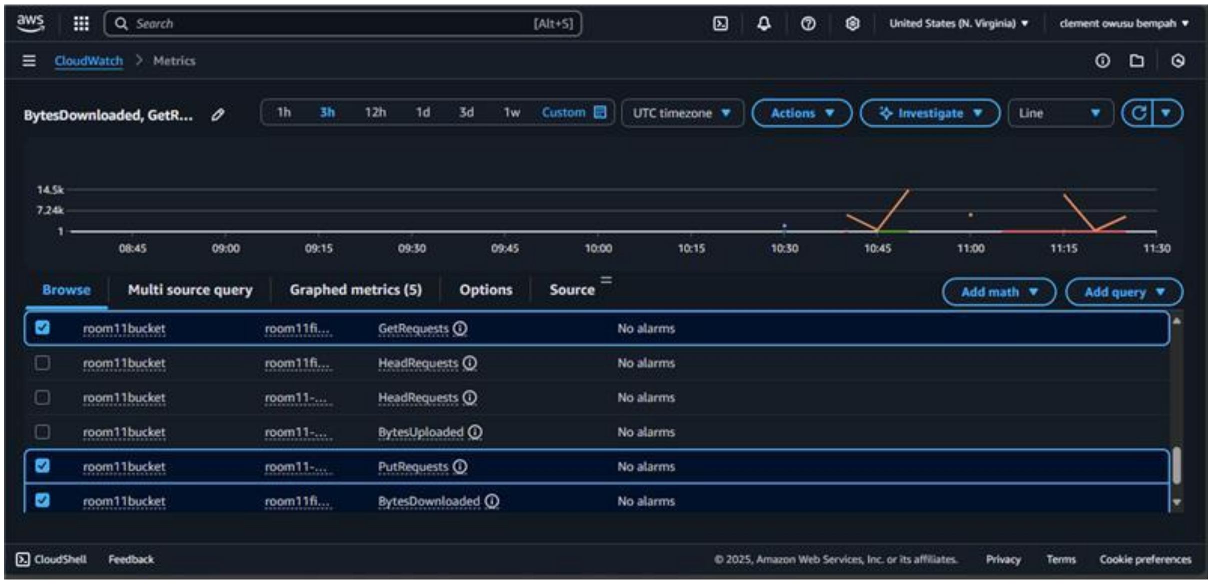
```
cleme@DESKTOP-BHD3JTJ MINGW64 ~/Downloads
$ aws s3 ls s3://room11bucket/
2025-08-06 11:26:53    1762389 html5up-ethereal.zip
2025-08-06 10:42:12     232443 image (10).png
2025-08-06 10:43:14      94875 image (2).png
2025-08-06 10:32:52      46474 image (28).png
2025-08-06 10:32:51      66567 image (29).png
2025-08-06 10:43:10     157581 image (3).png
2025-08-06 10:32:49      63751 image (30).png
2025-08-06 10:32:48      19217 image (4).jpg
2025-08-06 10:42:56     161151 image (4).png
2025-08-06 10:42:52     157581 image (5).png
2025-08-06 10:42:46      11298 image (6).png
2025-08-06 10:42:45     165394 image (7).png
2025-08-06 10:42:27     144251 image (8).png
2025-08-06 10:42:18     237436 image (9).png
2025-08-06 11:16:37      31799 scaling.png
2025-08-06 11:15:02      31799 test.png
```



```
cleme@DESKTOP-BHD3JTJ MINGW64 ~/Downloads
$ aws s3 cp s3://room11bucket/test.png .
download: s3://room11bucket/test.png to .\test.png
```

2. Activity Metrics Analysis

Analyzing activity metrics like GetRequests, PutRequests, and BytesDownloaded provides insights into how an S3 bucket is being utilized and helps identify potential performance bottlenecks. The CloudWatch metrics dashboard shows the ability to monitor these metrics.



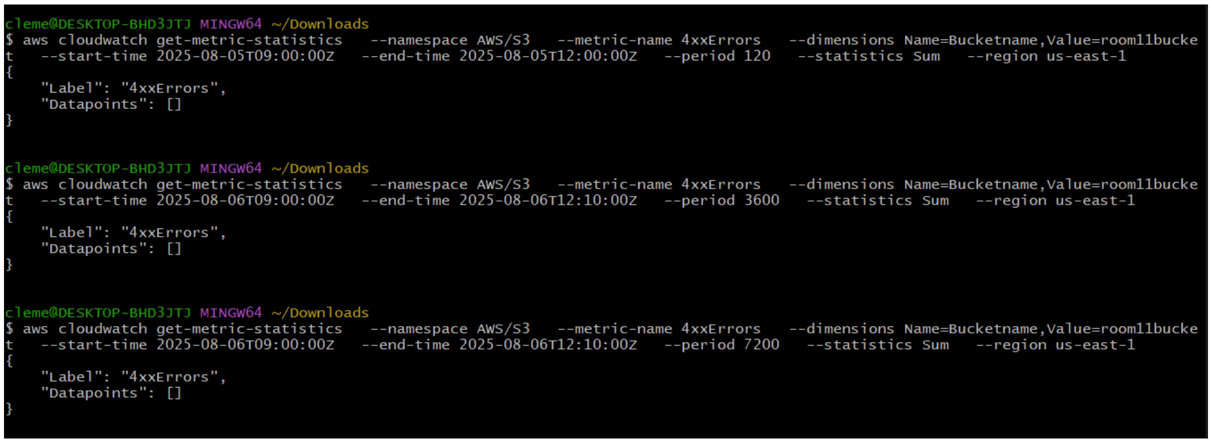
✓	room11bucket	room11fi...	GetRequests	No alarms
☐	room11bucket	room11fi...	HeadRequests	No alarms
☐	room11bucket	room11-...	HeadRequests	No alarms
☐	room11bucket	room11-...	BytesUploaded	No alarms
✓	room11bucket	room11-...	PutRequests	No alarms
✓	room11bucket	room11fi...	BytesDownloaded	No alarms

- **GetRequests**: Represents the number of times objects are retrieved from the bucket. A high volume of GetRequests could indicate frequent data consumption by applications or users.
- **PutRequests**: Indicates the number of times objects are uploaded or modified in the bucket. A high number here suggests frequent data ingestion or updates.
- **BytesDownloaded**: Shows the total bytes transferred out of the bucket. This metric is crucial for understanding data egress patterns and potential costs associated with data transfer out.

Based on the provided CloudWatch screenshot, GetRequests, PutRequests, and BytesDownloaded are being monitored. Without specific data points or trends over time, it's difficult to pinpoint exact bottlenecks. However, consistently high spikes in GetRequests or PutRequests might suggest a need for scaling application components that interact with S3 or optimizing data access patterns. For example, if GetRequests are consistently high from a single geographical region, enabling Transfer Acceleration for that region would be beneficial.

3. Status Code Metrics Review

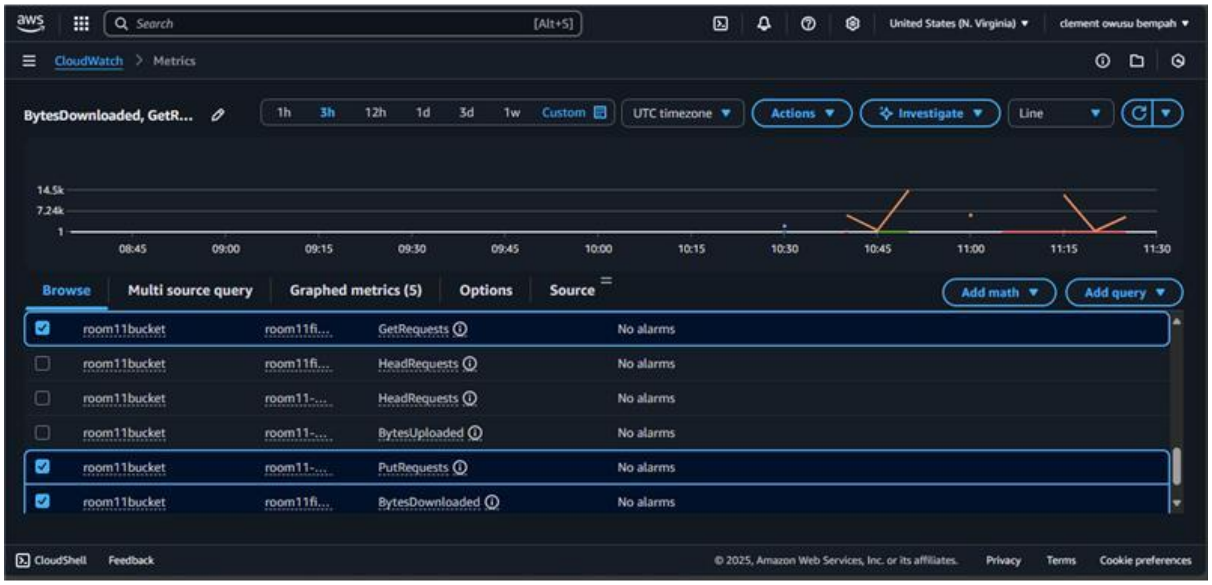
Monitoring status code metrics such as 200OKStatusCount, 403ForbiddenErrorCount, and 404NotFoundErrorCount is essential for understanding the success rate of requests and troubleshooting failed operations. The CloudWatch dashboard shows 4xxErrors being monitored.



```
cleme@DESKTOP-BHD3JTJ MINGW64 ~/Downloads
$ aws cloudwatch get-metric-statistics   --namespace AWS/S3   --metric-name 4xxErrors   --dimensions Name=Bucketname,Value=room11bucke
t   --start-time 2025-08-05T09:00:00Z   --end-time 2025-08-05T12:00:00Z   --period 120   --statistics Sum   --region us-east-1
{
    "Label": "4xxErrors",
    "Datapoints": []
}

cleme@DESKTOP-BHD3JTJ MINGW64 ~/Downloads
$ aws cloudwatch get-metric-statistics   --namespace AWS/S3   --metric-name 4xxErrors   --dimensions Name=Bucketname,Value=room11bucke
t   --start-time 2025-08-06T09:00:00Z   --end-time 2025-08-06T12:10:00Z   --period 3600   --statistics Sum   --region us-east-1
{
    "Label": "4xxErrors",
    "Datapoints": []
}

cleme@DESKTOP-BHD3JTJ MINGW64 ~/Downloads
$ aws cloudwatch get-metric-statistics   --namespace AWS/S3   --metric-name 4xxErrors   --dimensions Name=Bucketname,Value=room11bucke
t   --start-time 2025-08-06T09:00:00Z   --end-time 2025-08-06T12:10:00Z   --period 7200   --statistics Sum   --region us-east-1
{
    "Label": "4xxErrors",
    "Datapoints": []
}
```



✓	room11bucket	room11fi...	GetRequests	No alarms
☐	room11bucket	room11fi...	HeadRequests	No alarms
☐	room11bucket	room11-...	HeadRequests	No alarms
☐	room11bucket	room11-...	BytesUploaded	No alarms
✓	room11bucket	room11-...	PutRequests	No alarms
✓	room11bucket	room11fi...	BytesDownloaded	No alarms

- **200OKStatusCount**: Indicates successful requests. A consistently high count is desirable.
- **403ForbiddenErrorCount**: Signifies requests that were denied due to insufficient permissions.
- **404NotFoundErrorCount**: Represents requests for objects that do not exist in the bucket.

Causes of Failed Requests and Troubleshooting:

- **403ForbiddenErrorCount**:
  - **Causes:** Incorrect bucket policies, IAM user/role permissions, or ACLs. Public access blocks might also be enabled, preventing unauthorized access.
  - **Troubleshooting:**
    - Review bucket policies and IAM permissions attached to the users or roles

        attempting to access the bucket.
        - Check object ACLs if specific objects have different permissions.
        - Verify if S3 Public Access Block settings are too restrictive for the intended access.
        - Use AWS CloudTrail to log API calls and identify the exact permission issue.

- **404NotFoundErrorCount**:
    - **Causes:** Incorrect object keys (filenames), objects being deleted, or requests being made to the wrong bucket.
    - **Troubleshooting:**
        - Verify the object key (path and filename) in the application or request.
        - Check if the object was accidentally deleted. S3 Versioning (if enabled) could help recover deleted objects.
        - Ensure the request is targeting the correct S3 bucket and region.

The provided CloudWatch screenshot for 4xxErrors shows some activity around 11:15, indicating a few client-side errors occurred. Further investigation into the specific 403 or 404 counts during that period would be necessary to diagnose the exact cause.

4. Recommendations for S3 Performance Improvement

Based on the analysis, here are three actionable steps to improve S3 performance:

1. **Optimize for Geographic Distribution with Transfer Acceleration:**
    - **Actionable Step:** Enable S3 Transfer Acceleration for buckets that serve users or applications geographically distant from the bucket's region. Continuously monitor the cost implications and actual performance gains to ensure it provides a tangible benefit.
    - **Best Practice:** Use the S3 Transfer Acceleration Speed Comparison tool to evaluate potential gains before committing to its use for all traffic. For applications with a global user base, consider using Amazon CloudFront as a content delivery network (CDN) in conjunction with S3, as it offers more advanced caching and edge delivery capabilities, often at a lower cost for highly accessed content.
2. **Implement Object Lifecycle Management and Tiering:**
    - **Actionable Step:** Configure S3 Lifecycle rules to automatically transition objects to more cost-effective storage classes (S3 Standard-IA, S3 Glacier) based on access patterns and age. For frequently accessed data, ensure it remains in S3 Standard.
    - **Best Practice:** Analyze GetRequests and BytesDownloaded metrics over time to identify data that is accessed infrequently. For example, if data hasn't been accessed in 30 days, transition it to S3 Standard-IA. This optimizes storage

costs without significantly impacting performance for active data.

3. **Leverage S3 Event Notifications for Real-time Processing:**
    - **Actionable Step:** For applications that need to react to S3 object changes (new file uploads), configure S3 Event Notifications to trigger AWS Lambda functions. This allows for immediate processing of new data without polling the bucket, reducing latency and improving overall system responsiveness.
    - **Best Practice:** Use S3 Event Notifications for tasks like image resizing, data validation, or initiating data pipelines upon object creation or modification. This eliminates the need for applications to constantly check the bucket for new objects, reducing GetRequests and improving efficiency.

These recommendations combine configuration changes within S3 and AWS best practices to enhance performance, optimize costs, and improve the reliability of S3-based applications.