STORAGE DEVELOPER CONFERENCE

SDC 23

BY Developers FOR Developers

Beyond S3 Compatibility Claims

A Deep Dive into Real-World Incompatibilities

Gregory Touretsky, Seagate

Is '100% compatibility' with Amazon S3 just a myth? Let's find out

The fine print of S3 compatibility: What vendors won't tell you

Amazon S3

aws.amazon.com/blogs/aws/amazon_s3/

aws

Contact Us Support My Account S

Products Solutions Pricing Documentation Learn Partner Network

Blog Home Topics Edition

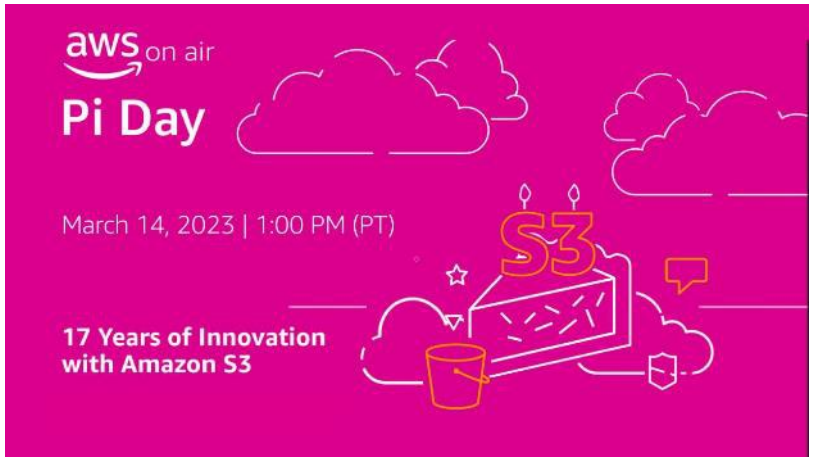
AWS News Blog

Amazon S3

by Jeff Barr | on 14 MAR 2006 | Permalink | Share

Earlier today we rolled out Amazon S3, our reliable, highly scalable, low-latency data storage service.

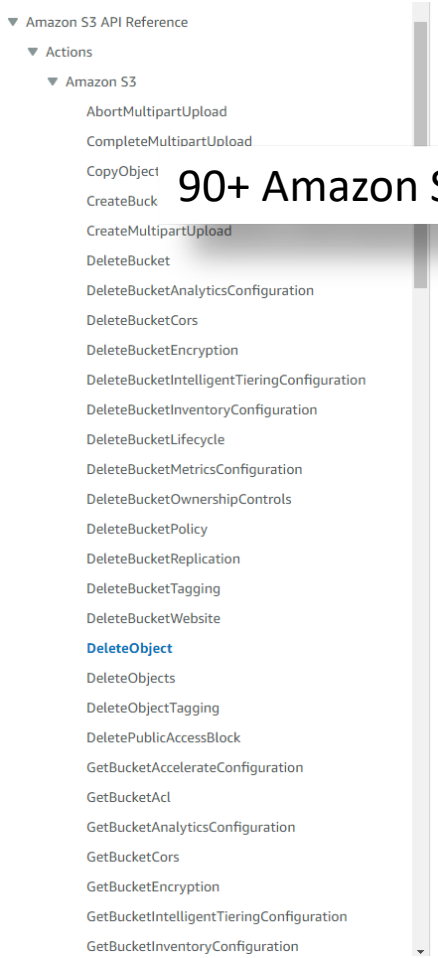
Using SOAP and REST interfaces, developers can easily store any number of blocks of data in S3. Each block can be up to 5 GB in length, and is associated with a user-defined key and additional key/value metadata pairs.



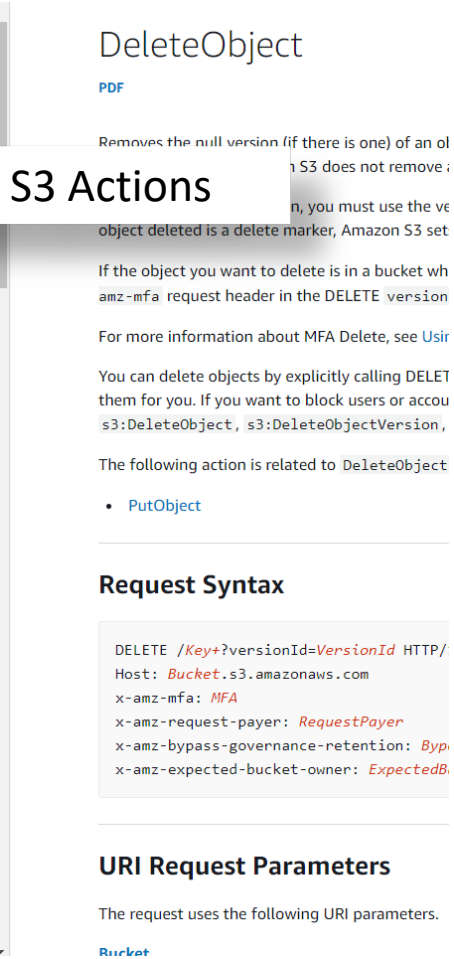
- >280 Trillion objects
- >100M requests per second
- 4B checksum calculations per second

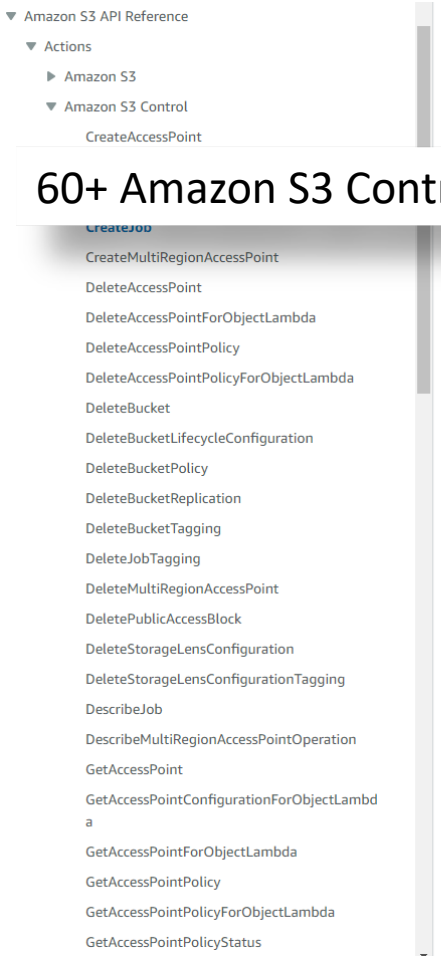
SDC 23

Amazon S3 API

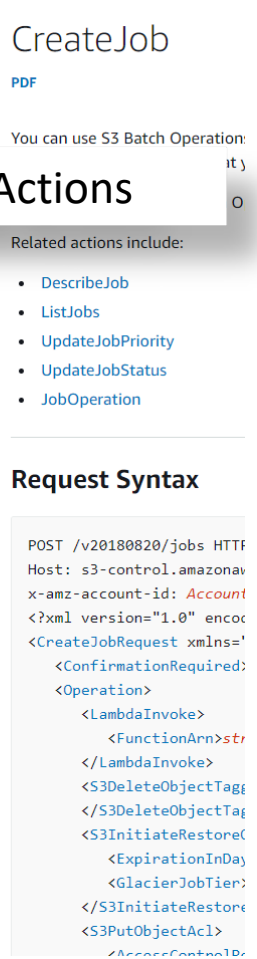


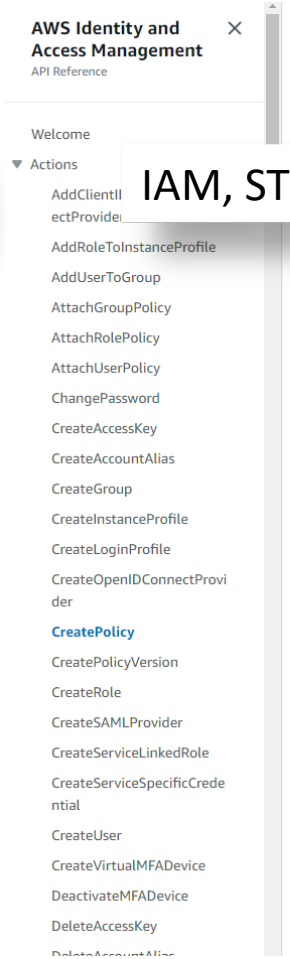
90+ Amazon S3 Actions



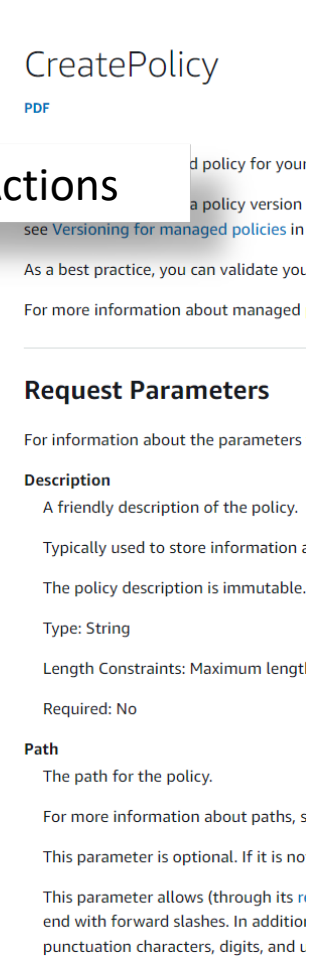


60+ Amazon S3 Control Actions





IAM, STS Actions



S3-Compatible Storage

Cloud Services

. . .

Systems and Software

. . .

SDC 23

Official Incompatibilities

ibm.com/docs/en/coss/3.10.2?topic=development-rest-api-reference

IBM Documentation | Search in IBM Cloud Object Storage System 3.10.2

IBM Cloud Object Storage System

Show full table of contents

Filter on titles

System Cloud Storage Object API

List Parts (GET) Returns a list of parts that are associated wi

The following operations are not supported in the IBM COS implementation of the S3 API:

- DELETE Bucket analytics
- DELETE Bucket inventory
- DELETE Bucket lifecycle
- DELETE Bucket metrics
- DELETE Bucket policy
- DELETE Bucket replication
- DELETE Bucket website
- GET Bucket accelerate

ECS 3.6.2 Data Access Guide | De

dell.com/support/manuals/en-us/ecs-appliance-software-with-encr...

Unsupported S3 APIs

Table 3. Unsupported S3 APIs

FEATURE	NOTES
DELETE Bucket tagging	-
DELETE Bucket website	-
GET Bucket location	ECS is only aware of a single Virtual Data Center (VDC).

n is only defined for

github.com/scality/cloudserver/blob/development/8.7/constants.js#L98

```
98      unsupportedQueries: [
99          'accelerate',
100         'analytics',
101         'inventory',
102         'logging',
103         'metrics',
104         'policyStatus',
105         'publicAccessBlock',
106         'requestPayment',
107         'torrent',
108     ],
109
110     // Headers supported by AWS that we do not currently support.
111     unsupportedHeaders: [
112         'x-amz-server-side-encryption-customer-algorithm',
113         'x-amz-server-side-encryption-context',
114         'x-amz-server-side-encryption-customer-key',
115         'x-amz-server-side-encryption-customer-key-md5',
116     ],
```

Use metadata search queries

The metadata search feature provides a rich query language that enables objects that have indexed metadata to be searched.

Table 21. API Syntax

API Syntax	Response Body
`GET /{bucket}/?query={expression}&attributes={fieldname,…}&sorted={selector}&include_older_version={true|false}&max-keys={num_keys}&marker={marker value}`	`<BucketQueryResult xmlns:ns2="http://s3.amazonaws.com/doc/2006-03-01/"><Name>mybucket</Name><Marker/><IsTruncated>false</IsTruncated><MaxKeys>0</MaxKeys><ObjectMatches><object><objectName>file4</objectName><objectId>09998027b1b7fbb21f50e13fabb481a237ba2f60f352d437c8da3c7c1c8d7589</objectId><versionId>0</versionId><queryMds><type>SYSMD</type><mdMap>`

NOTE: Prefix capability is added to the metadata search. See Prefix capability in metadata search.

min.io/docs/minio/linux/operations/checklists/thresholds.html

are Checklist

hold

afte

hooting

ration

nsole

Unsupported S3 Bucket APIs

MinIO does not support the following API calls available in S3. These APIs are either redundant or only provide functionality within AWS S3.

- `BucketACL`, `ObjectACL` (use Policies)
- `BucketCORS` (CORS enabled by default on all buckets for all HTTP verbs)
- `BucketWebsite` (use `caddy` or `nginx`)
- `BucketAnalytics`, `BucketMetrics`, `BucketLogging` (use Bucket Notifications)
- `BucketRequestPayment`

Protocol Compliance Tools

- Home-grown
- 3rd party applications
- https://github.com/ceph/s3-tests
- https://github.com/splunk/s3-tests
- https://github.com/open-io/ceph-s3-tests
- https://github.com/minio/mint

Solution	Failed Mint tests
Minio	0
Backblaze B2	14
Google Cloud Storage – S3	15
AWS S3	12

Is Minio more S3-compatible than Amazon S3? ☺

SDC 23

What Is Behind the Endpoint?

```
aws s3 ls --debug |& grep "Response headers" | awk -F 'Server' '{print $2}'
```

Solution	Server Header
Amazon S3	AmazonS3
Google Cloud Storage (S3 compatible)	UploadServer
Ceph	-
Minio	MinIO
Wasabi	WasabiS3/7.12.1004-2023-02-17-7ff2f5bdd9 (head07)
Seagate Lyve Cloud	Seagate-LyveCloudS3
Backblaze B2	-

What Is in the Request?

- Example: UploadPart

Component	Example
Method	PUT
Bucket	mybucket
Host	us-east-1.lyvecloud.seagate.com
Key	myprefix/myobject
partNumber	12
uploadId	2e1c42be-fc1d-4055-bbce-d10a55a0a662
authorization	AWS4-HMAC-SHA256 Credential=REDUCTED/20230418/us-east-1/s3/aws4_request, SignedHeaders=content-length;host;user-agent;x-amz-content-sha256;x-amz-date, Signature=61b9391bc68984f634db8437779e76a8f609a5823b3ea0ac00a3df48e431d59c
user-agent	APN/1.0 Qumulo/1.0 S3Replication/6.0.2
…	

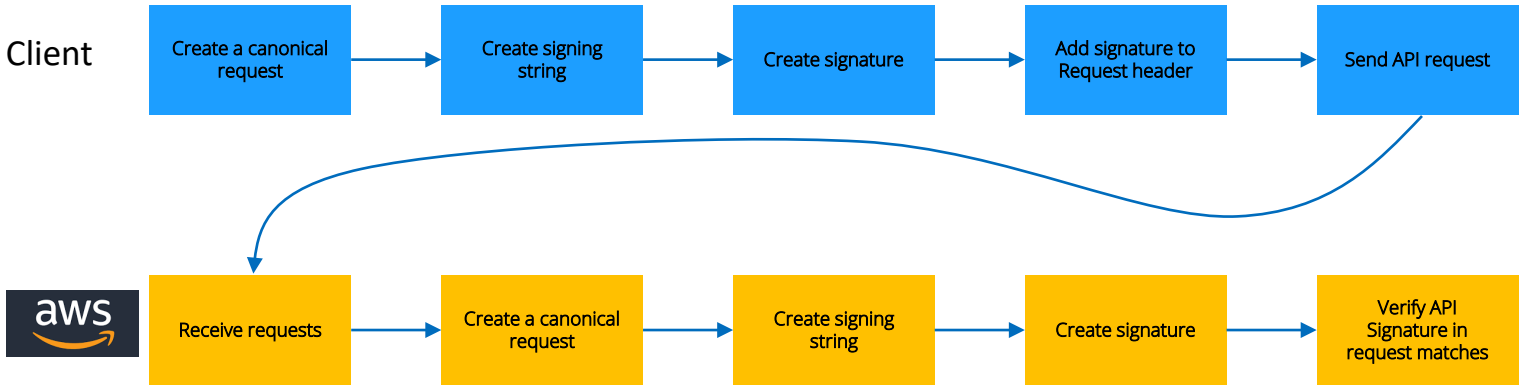
What Is in the Response?

- Example: UploadPart

Component	Example
Response code	HTTP/1.1 200
Date	Tue, 18 Apr 2023 04:25:17 GMT
Etag	"3ae3ac4887f98ddefde4eeb82e37280b"
Server	Seagate-LyveCloudS3
X-Amz-Request-Id	1756ECD65A5EE9C2
…	

Authentication / Signature

- Sig V2 – old, discontinued in 2020
  - But many client apps still use it!
- Sig V4 – current, in use since 2012:




```
#Create a Canonical Request
CanonicalRequest =
  HTTPRequestMethod + '\n' +
  CanonicalURI + '\n' +
  CanonicalQueryString + '\n' +
  CanonicalHeaders + '\n' +
  SignedHeaders + '\n' +
  HexEncode(Hash(RequestPayload))
```

Header names, sorted by lowercase character code, delimited by semi-colon

- Customer's app fails CopyObject requests: SignatureDoesNotMatch
- Root cause:
  - Server returns a mix of lower case and mixed-case header names,
    Ex: X-Amz-Server-Side-Encryption
  - Customer's app sorts headers case-sensitive → Signed headers string is not properly sorted

```
Authorization: AWS4-HMAC-SHA256
Credential=REDACTED/20230205/us-east-
1/s3/aws4_request,SignedHeaders=content-
type;host;user-agent;x-amz-server-side-encryption;x-
amz-acl;x-amz-content-sha256;x-amz-copy-source;x-
amz-date;x-amz-metadata-directive,
Signature=246998b7b32681af8d6dbfdf8754da20c4633509d2
0fc6391bfe291d0d4caba1
```


Object Key (Path)

- Up to 1,024 bytes long

- The following objects can coexist in a bucket:
  - mybucket/myfolder/obj
  - mybucket/myfolder/obj/
  - mybucket/myfolder/obj//
  - mybucket/myfolder/../obj/
  - mybucket/myfolder/

- Up to 1,024 bytes long
- "/" is interpreted as a directory
- Directory segments are limited to 255 bytes

- "//", ".", ".." are not allowed
- mybucket/myfolder and mybucket/myfolder/obj objects can't co-exist

```
$ aws --profile minio s3api put-object --bucket
bucket1 --key NameWith//Inside --body ~/empty

An error occurred (XMinioInvalidObjectName) when
calling the PutObject operation: Object name
contains unsupported characters
```


Complete-Multipart-Upload Response Caching

Complete Multipart Upload is an idempotent operation. After your first successful complete multipart upload, if you call the operation again within a short period, the operation will succeed.

```
$ aws s3api complete-multipart-upload --bucket A
--key B --uploadId XX --multipart-upload file
POST 200 None
$ aws s3api complete-multipart-upload --bucket A
--key B --uploadId XX --multipart-upload file
POST 200 None
```

MINIO

```
$ aws s3api complete-multipart-upload --bucket A
--key B --uploadId XX --multipart-upload file
POST 200 325
$ aws s3api complete-multipart-upload --bucket A
--key B --uploadId XX --multipart-upload file
POST 404 449
```


Don’t Be Fooled by “Success”

Complete-Multipart-Upload

Processing of a Complete Multipart Upload request could take several minutes to complete. After Amazon S3 begins processing the request, it sends an HTTP response header that specifies a 200 OK response. While processing is in progress, Amazon S3 periodically sends white space characters to keep the connection from timing out. A request could fail after the initial 200 OK response has been sent. This means that a 200 OK response can contain either a success or an error.

```
2023-03-15 21:40:29,089 - MainThread - urllib3.connectionpool - DEBUG - https://gt-test-
006.s3.us-east-1.amazonaws.com:443 "POST
/largeobjecttest?uploadId=szRCQ4o6dw8qjRjUjOe9WD2z2JbE5bHFuZvL27zUciZJW3um8GeIqYcPlLNu_GzUzuYChe
YCYpAaWdLZF3x3I8rAdVF_7U109PBm3nd_ATIntjyYqHOcVdbS6X8vmxNI HTTP/1.1" 200 None

21:40:29,096 - MainThread - botocore.parsers - DEBUG - Response body:b'<?xml version="1.0"
encoding="UTF-8"?>\n<Error><Code>EntityTooLarge</Code><Message>Your proposed upload exceeds the
maximum allowed size</Message>
<ProposedSize>5513664266240</ProposedSize><MaxSizeAllowed>5497558138880</MaxSizeAllowed><Request
Id>A1B9N8GCJ9368Z0N</RequestId><HostId>nXky785oI/qBz4qo1PO3M00bNF/SJSXiw6tLSEESNF1hVT2kEU2cKWKxb
fG5iTw4KlVBNOk5GoY=</HostId></Error>'
```


Get-object-attributes

```
$ aws s3api get-object-attributes --bucket gt-test-006 --key awscliv2.zip --object-attributes "ETag"

{
    "LastModified": "2023-04-20T01:01:27+00:00",
    "VersionId": "2m1EHmC6ALD_FVJ6HZBJ9znvBEbi6pNa",
    "ETag": "75c77163c337dfd5bb5a5f9f7a6473dd-1"
}
```

```
$ aws s3api get-object-attributes --bucket gt-test-006 --key awscliv2.zip --object-attributes "ETag"


Unable to parse response (not well-formed (invalid token): line 1, column 2), invalid XML received. Further retries may succeed:
b'PK\x03\x04\x14\x00\x00\x00\x00\x00F\x8f\xf4T\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x00\x00\x00aws/PK\x03\x04\x14\x00\x00\x00\x00\x00E\x8f\xf4T\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\t\x00\x00\x00aws/dist/PK\x03\x04\x14\x00\x00\x00\x08\x00\x1a\x8e\xf4TW\x92\xbe\n<\x02\x00\x00\xb
```


Head-bucket

```
$ aws s3api head-bucket --bucket gt-test-006 --expected-bucket-owner wronguser --debug
|& grep HEAD
urllib3.connectionpool - DEBUG - https://gt-test-006.s3.us-east-1.amazonaws.com:443
"HEAD / HTTP/1.1" 400 0
```

```
$ aws s3api head-bucket --bucket gt-test-006 --expected-bucket-owner wronguser --debug
|& grep HEAD

urllib3.connectionpool - DEBUG - http://10.0.0.83:9000 "HEAD /gt-test-006 HTTP/1.1" 200
0
```


Put-object – Unsupported CRC32C

```
$ ./warp get --access-key=REDACTED --secret-key= REDACTED --bucket=gt-test-001 --
concurrent=60 --host=ENDPOINT --obj.size=16MiB --tls --duration 300s --objects=2500 --
analyze.v
warp: <ERROR> upload error: The X-Amz-Checksum-Crc32c you specified did not match what
we received.


$ aws s3api put-object --bucket gt-test-001 --key myobject --body myobject.zip --
checksum-crc32-c 8KygcQ==
An error occurred (InvalidRequest) when calling the PutObject operation: Value for x-
amz-checksum-crc32c header is invalid.
```

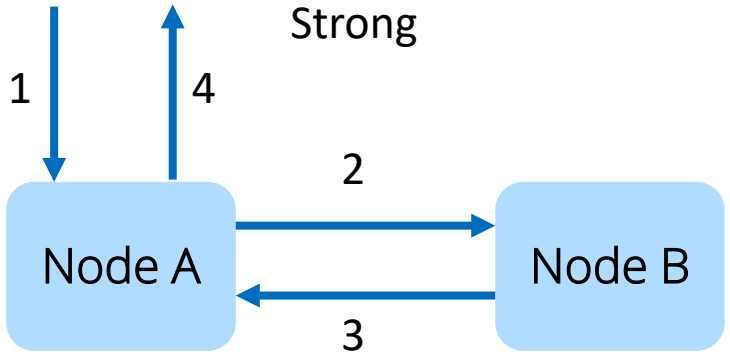

Performance

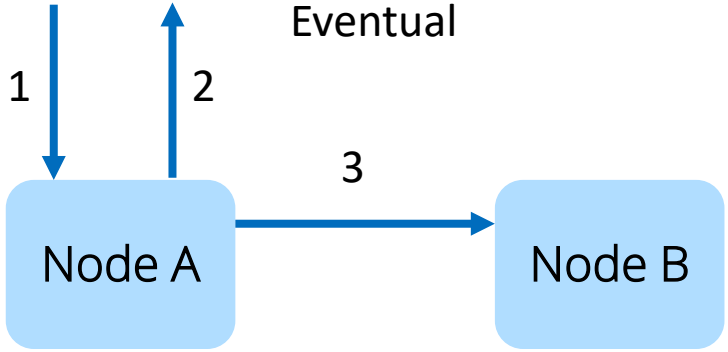
	aws	Azure	Google Cloud Storage	wasabi

Requests/ sec	• 3,500 PUTs / 5,500 GETs per sec per partitioned prefix
	• Request rates are allocated proportionally
		• Up to 500 requests/sec for a single blob (object)
		• Default max request rate per storage account: 20,000 requests / sec
			• Initial ~1,000 writes/sec, 5,000 reads/sec
			• Gradually autoscaling above limits, based on prefixes. Double rate every 20 min
			• Bucket
			  • Create/delete: 0.5/sec
			  • Update: 1/sec
				• S3 API: “fair use”, depends on storage volume
				• Account Control API: GET 1000/min, PUT 100/min, DELETE 10/min

Throughput	• single-instance … up to 100 Gb/s • aggregate … multiple Tbps	• Default max ingress per storage account: 10/25/60 Gbps (varies per region) • Default max egress per storage account : 50/120 Gbps	• Default egress quota: 200Gbps per region	• No details

Consistency Model (*)





aws

Google Cloud Storage

aws

Before Dec'2020

IBM Spectrum Scale

SwiftStack

(*) Data path. Access control, etc may vary

SDC 23

Summary

Complexity + nuance

Incompatibilities

Deep understanding of the S3 API

Thorough testing

S3 compatibility = customer adoption

Driving Compatibility and Collaboration in Cloud Storage

SNIA Cloud Storage TWG Work Item

- Foster Ecosystem Collaboration
  - Facilitate collaboration and knowledge sharing among S3 developers by establishing a platform for discussions, forums, and workshops

- Enable S3 Multi-Cloud Interoperability
- Establish Compliance Certification
  - Documentation
  - List of known incompatibilities
  - Standardized compatibility tests

BoF Session: Tuesday, 9/19 @ 8pm

gregory.touretsky@seagate.com

SDC 23

Beyond S3 Compatibility Claims

Please take a moment to rate this session.

Your feedback is important to us.

