Storage: S3

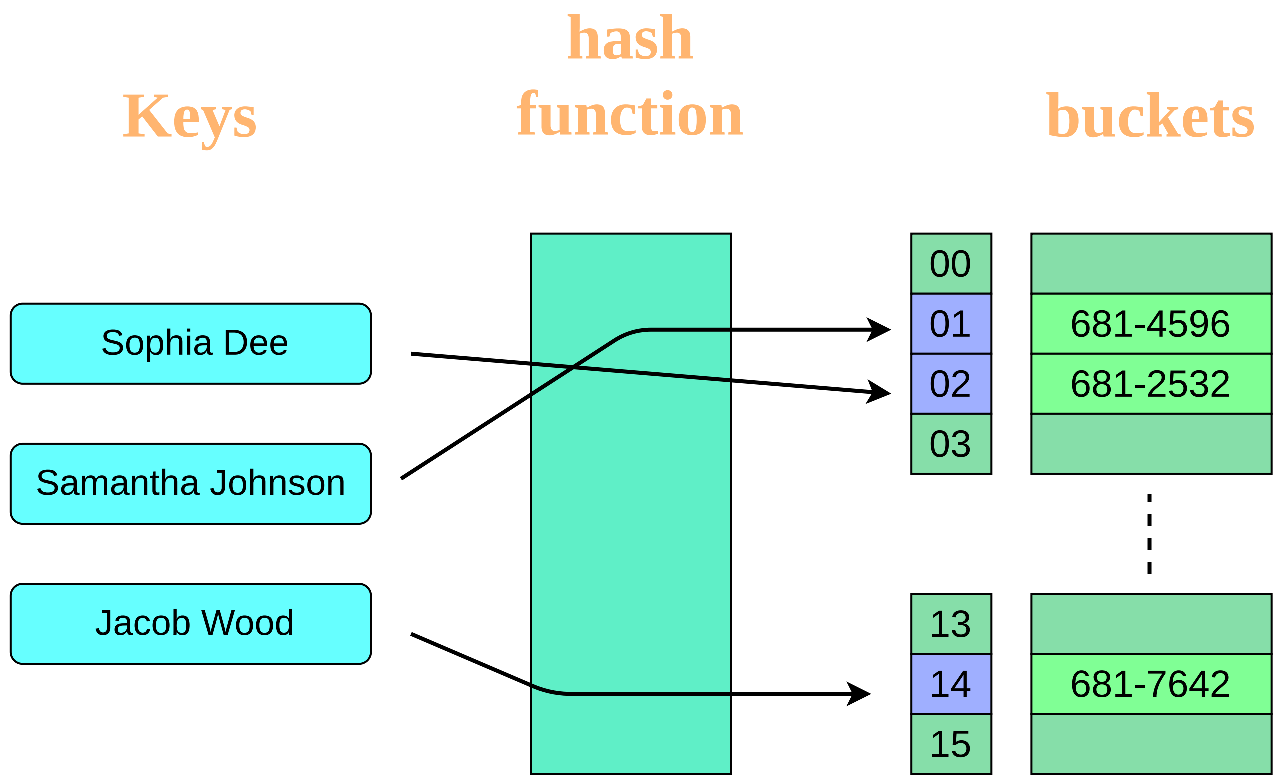
The features and attributes of S3 make it a top choice for storage.

We'll cover the following

If you're storing data—whatever it is—S3 should be the very first thing to consider using.

Hashtable in the cloud

Fundamentally, you can think of S3 as a highly-durable hash table in the cloud. The key can be any string, and the value any blob of data up to 5 TB. When you upload or download S3 objects, there's an initial delay of around 20 ms before the data gets streamed at a rate of around 90 MB/s. You can have as many parallel uploads and downloads as you want, thus, the infinite bandwidth. You can also store as many objects as you want and use as much volume as you need, without either having to provision capacity in advance or experiencing any performance degradation as the scale increases.



Hashtable of users and their phone numbers

Following are the different features of S3:

Features of S3#

1- Highly durable

2- Very easy to use

3- Infinite bandwidth

4- Infinite storage space

5- Zero capacity management

Storage cost#

S3 storage costs $23.55/TB/month using the default storage class. It's almost impossible to beat S3's storage costs when you factor in the in-built redundancy.

At first:

- You can start with the default storage class and ignore all the other classes.

Unless you're storing several terabytes in S3, it is almost never worth bothering with them. In general, you can spare yourself the trouble of understanding all the implications of the different storage classes until you really need to start saving money from S3 storage costs.

Apart from the storage classes:

- S3 also offers a reduced redundancy option, but this one should definitely *never* be used.

This is a legacy feature that has been around since 2010 (before storage classes existed), and it is currently more expensive than the default storage class, but with no benefits and lower availability (in theory).

Request pricing#

While S3 storage costs are practically unbeatable, request pricing can become expensive in certain situations. With S3, you pay $5/million uploads and $0.40/million downloads.

- When your S3 usage is the result of a human operation—such as somebody uploading a file, or requesting an image from a website—this cost tends to be acceptable. Serving a million people is a big deal, and paying $5 or $0.40 for that level of scale is hardly expensive.
- However, when your S3 usage is driven by other computers, this can change quickly. If you're touching S3 objects at a high frequency (millions of times a day), request pricing becomes an important aspect of S3's viability for your use case.

Let's do a fun exercise: Use the below widget to calculate the request pricing cost when $5 is the cost of a million uploads. You just have to put the *number* of **upload** requests (in millions) required for serving a web page, in the first cell. It will automatically generate the cost in the next cell.

No. of upload requests (in millions)	Request pricing cost ($)
2	*f* 10

Here's another activity: Use the below widget to calculate the request pricing cost when $0.40 is the cost of a million downloads. You just have to put the *number* of **download** requests (in millions) required for serving a web page, in the first cell. It will automatically generate the cost in the next cell.

No. of download requests (in millions)	Request pricing cost ($)
10	*f* 4

NOTE: It is very important to note that the prices mentioned above are only for the purpose of understanding cost comparisons. These prices are subject to change anytime by AWS. Therefore, most current prices should be referenced for final business decisions.

Limitations#

- One limitation of S3 is that you cannot append to objects.

- If you have something that's changing rapidly (such as a log file), you have to buffer updates on your side for a while, and then periodically flush chunks to S3 as new objects. This buffering can reduce your overall data durability because the buffered data will typically sit in a single place without any replication.

- A solution for this issue is to buffer data in a durable queue, such as SQS or Kinesis streams, as we'll see later.

Q S3 is a highly-durable hash table in the cloud. The key can be any string, and the value any blob of data up to ______ TB?

○ A) 7TB

○ B) 10TB

○ C) 3TB

○ D) 5TB

Submit Answer

Reset Quiz

Hosting static websites#

As far as hosting static websites on S3 is concerned:

- It doesn't support HTTPS when used as a static website host, which is a problem.

Web browsers will display a warning, and search engines will penalize you in the rankings.

- You could set up HTTPS using CloudFront, but it's probably much more trouble than it's worth.

Nowadays, there are plenty of static website hosts outside of AWS that offer a much better hosting experience for static websites.

Bucket names#

Finally, a note on one of S3's quirks.

- Bucket names are globally unique across all AWS customers and across all AWS regions.

This can be quite inconvenient because, if you try to rely on a naming convention for your bucket names, you might find that someone else has already taken the name you want.

- A common mitigation is to always add your AWS account ID to the bucket name, which makes conflicts much less likely.

Apart from the inconvenience, the global bucket namespace also has an important security implication.

- If your application tries to use an S3 bucket without checking its owner, you might find yourself uploading data to someone else's bucket.

Luckily, S3 has an API to check if you own the bucket, and this should always be done before interacting with an existing S3 bucket.

> The following example sets bucket_exists to true if a bucket with the name my-bucket already exists. The `region:` parameter to `Resource` has no effect on the result.

```
s3 = Aws::S3::Resource.new(region: 'us-west-2')
bucket_exists = s3.bucket('my-bucket').exists?
```

In the next lesson, we will take a look at EC2 and it’s functionalities.

← Back
Database: DynamoDB

Next →
Compute: EC2

✓ Mark as Completed

Report an Issue