Amazon Web Services:

Introduction to Amazon S3

Created by Zarsha Mian

Amazon Web Services:

Introduction to Amazon S3

Table of Contents

Introduction to Amazon S3

Amazon Simple Storage Service, otherwise known as Amazon S3, is storage for the Internet designed to make web-scale computing easier for developers. Amazon S3 is used to store and retrieve any amount of data from anywhere on the web at any time. There are many benefits of using Amazon S3, including increased speed, reliability, and scalability.

There is certain terminology associated with Amazon S3 that you must be familiar with in order to better understand the service and its functions:

- Bucket: container of objects stored in Amazon S3. Buckets are used to store data that can be uploaded or downloaded by other users. Permissions can be set for your Amazon S3 bucket to grant or deny access to certain users or groups.
- Object: entities stored in Amazon S3, such as images and files, consisting of object data and metadata. Objects can be uniquely identified within a bucket by a key (name) and version ID.
- Metadata: name-value pairs that describe an object.
- Key: unique identifier of an object within a bucket. The bucket, key, and version ID can be used to uniquely identify an object, and each object is assigned only one key.
- Region: geographical region where Amazon S3 will store the buckets that you create. Objects stored in a region will never leave that region unless you transfer them to another region.

Amazon S3 is inexpensive compared to other storage providers. Rather than forcing you to purchase a predetermined amount of storage and network transfer capacity, Amazon S3 only charges you for what you use. Before using Amazon S3, you must register with the service. Your payment method will be charged at the end of the month.

For more information about Amazon S3's features, such as storage classes, access control lists, versioning, go to https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html.

Creating and Configuring S3 Bucket for Root Domain

As mentioned earlier, buckets are containers that store data in Amazon S3, such as images, documents, and the HTML files that contain the pages of your website. The bucket you will create in this section will be referred to as your root domain bucket since it will have your domain's name. In order to create a bucket, you must follow these steps:

1. Open the **Amazon S3 console** at https://console.aws.amazon.com/s3/.
2. Choose *Create Bucket*.
3. Under **General Configuration**, in the *Bucket Name* field, enter the name of your domain, such as example.com.
4. Choose the *region* closest to your users, such as "US East (Ohio) us-east-2" in the *Region* field. **NOTE:** Make note of the region you choose for your bucket since you will need this information later on. Remember, objects stored in a region will not leave that region unless you transfer them to another region.
5. In the **Bucket settings for Block Public Access** section, uncheck the *Block all public access* box in order to grant public access to your bucket. **NOTE:** You will add a bucket policy later on to limit access to the bucket.
6. In the new prompt that appears **below the Block Public Access** section, check the box that says that you *acknowledge that the current settings might result in the bucket and its objects becoming public*.
7. In the **Advanced Settings** section, decide whether or not to enable *Object Lock*. By default, this setting is set to Disabled. An explanation of Object Lock is provided in the section.
8. Click *Create Bucket*.

You will be returned to the **Amazon S3 Buckets** page, and the top of the page will display an alert that says *"Successfully created bucket "your-bucket-name"*. Your new bucket will appear in the **list of buckets**.

Now that your domain bucket has successfully been created, we will configure it to host a website. This will allow users to access your website by using your domain name.

9. In the **list of S3 buckets**, choose the *name* of the bucket you just created.
10. Choose the **Properties** tab.
11. Click *Static website hosting*.
12. Choose the *"Use this bucket to host a website"* option. **NOTE:** Make note of your endpoint link. You will need to use it later on.
13. In the *Index Document* field, enter the name of the file that contains the main page for your website. **NOTE:** The file name "index.html" is a good option. Later on, you will create this HTML file and upload it to your bucket.
14. In the *Error Document* field, enter the name of the file that contains the error message or page for your website. **NOTE:** This step is not required, and you may leave this field blank if you choose.
15. Click *Save*.

The next step of the process is to attach a bucket policy to your bucket. The policy mentioned in this example will allow everyone full access to the contents of your bucket. If you wish to learn more about the best practices for securing the files in your S3 bucket and the risks involved in granting public access, go to https://aws.amazon.com/premiumsupport/knowledge-center/secure-s3-resources/.

To attach a bucket policy to your bucket, follow these steps:

16. Choose the **Permissions** tab.
17. Choose *Bucket Policy*.
18. Enter your bucket policy into the text editor. If you do not have a bucket policy of your own, copy and paste the following bucket policy into the provided bucket policy editor.

```
{
"Version":"2012-10-17",
"Statement":[{
"Sid":"AddPerm",
"Effect":"Allow",
"Principal":"*",
"Action":[
"s3:GetObject"
],
"Resource":[
"arn:aws:s3:::your-domain-name/*"
]
}]
```

}

This policy grants everyone on the Internet permission to get the files in the S3 bucket associated with your domain name. Under **Resource**, be sure to replace the value *your-domain-name* with the name of your bucket, such as example.com. If you fail to do so, or if the bucket name in the bucket policy does not match your bucket name, you will get an error message that says *"Policy has invalid resource"*.

19. Click the *Save* button.
20. You will receive a warning that indicates that *"This bucket has public access"*. In **Bucket Policy**, a Public label will appear.

NOTE: If you receive an *Error – Access Denied* warning and the bucket policy editor does not allow you to save the bucket policy, check your account-level and bucket-level *block public access* settings to confirm that you allow public access to your bucket.

On the **Buckets** page, find the *name* of your root domain bucket in the **buckets** list. In the **Access** column, it will show your bucket to be Public due to the permissions granted in the bucket policy.

Creating and Configuring S3 Bucket for Subdomain

The next step is to create a subdomain bucket. If you want your users to be able to use *www.your-domain-name* to access your website, you will need to create a second S3 bucket that is configured to route traffic to the first bucket, your root domain bucket.

Typically, websites redirect *your-domain-name* to *www.your-domain-name*, but because of the way that S3 works, you must set up the redirection in the opposite direction. This means that *www.your-domain-name* will redirect to *your-domain-name*. **Please note that this process is optional but highly recommended.**

In order to create an S3 bucket for your subdomain, follow the following steps:

1. Follow the previously mentioned steps 1 through 4 from **Create and Configure S3 Bucket for Root Domain.**
2. In the **Bucket settings for Block Public Access** section, leave the default settings in place. The *Block all public access* box is checked.
3. Follow previously mentioned steps 7 and 8 from **Create and Configure S3 Bucket for Root Domain**. Your new bucket will successfully be created and will appear in the **list of buckets**.

Now you will configure this subdomain bucket to redirect traffic to the first bucket you made, your root domain bucket.

4. In the **list of S3 buckets**, click the *name* of the subdomain bucket that you just created.
5. Choose the **Properties** tab.
6. Click *Static website hosting*.
7. Choose the *Redirect requests* option.
8. In the *Target bucket or domain* field, enter the name of the first bucket you created, otherwise known as the root domain bucket. **NOTE:** The bucket name you enter is the name of the bucket that you want to redirect requests to.
9. In the *Protocol* field, enter http. **NOTE:** Amazon S3 doesn't support HTTPS connections for website endpoints, and your bucket is configured as a website endpoint.
10. Click *Save*.

These changes will now enable redirections from your subdomain bucket to your domain bucket.

Configure Logging for Website Traffic

If you wish to track the number of visitors accessing your website, you can enable server access logging for your root domain bucket, which has been configured as a static website. To enable logging for your static website bucket:

1. Open the **Amazon S3 console**.
2. Click the *Create Bucket* button. We will be creating a new bucket for logging.
3. In the *Bucket Name* field, distinguish your bucket as a logging bucket. A name like logs.example.com for your logging bucket is suggested.
4. Make sure that the *Region* field has the same region as your root domain bucket.
5. In the **Bucket settings for Block Public Access** section, leave the default settings in place. The *Block all public access* box is checked.
6. Choose your settings for the **Advanced Settings** section. By default, the *Object Lock* is set to Disable.
7. Click the *Create Bucket* button.

Your new bucket will appear in the **list of buckets**. You now have a root domain bucket, a subdomain bucket, and a logging bucket. Now that the logging bucket has been created, we will configure it to enable server access logging:

8. In the **Bucket list**, click the *name* of your logging bucket.
9. Click the *Create folder* button.
10. In the *New folder* field, enter the name of the folder for server access logging log files. For example, a suggested name input is logs.
11. Leave the default *encryption setting* for the object. By default, the None (Use bucket settings) setting is chosen.
12. Click the *Save* button. The new folder is created.
13. Choose the **Properties** tab.
14. Click *Server access logging*.
15. Click *Enable logging*.
16. In the *Target bucket* field, enter the name of the logging bucket, for example logs.example.com.

17. For the *Target prefix* field, enter the name of the folder you created for the log files followed by the delimiter (/). According to this example, the input would be logs/.
    NOTE: Setting the Target prefix allows you to group your log data files in the folder so that they are easy to locate.
18. Click the *Save* button. You will now be able to access your logs. Amazon S3 writes website access logs to your log bucket every 2 hours.
19. To view your logs, choose the **Overview** tab, and click on the log folder.

Uploading Index and Website Content

Earlier in step 13 of **Creating and Configuring S3 Bucket for Root Domain**, you entered the name of an index document, which is an HTML file, that contains the main page of your website. In this example, the index document name is index.html. In this section of the tutorial, you are going to upload your index document to your root domain bucket. You can also upload additional website content, as well. To configure the index document, follow these steps:

1. Create an index.html file. **NOTE:** You can use a text editor like TextEdit or Notepad to do this, but you will need to change the file extension from .txt (text file) to .html (HTML document).

   If you do not have an index.html file of your own, you can use the following template to create one:

```
<html>
<head>
  <title>My Website Home Page</title>
</head>
<body>
  <h1>This is a header.</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

2. Save the index file locally. **NOTE:** The name of this file must match your index document name.
3. In the **Buckets list**, choose the *name* of your root domain bucket, which is used to host a static website.
4. Click the *Upload* button.
5. In the **Select Files** section, either drag and drop your index file or click the *Add Files* button and choose your index HTML file.
6. Click *Next.*
7. In the **Set Permissions** section, manage users and set your object permissions. By default, the owner is granted permission to list, create, overwrite, and delete objects in the bucket, as well as permission to read and write to an access control list (ACL) for the bucket.

8. Optionally, add access for other AWS accounts to grant other users object permissions. You can leave the default settings if you do not wish to change the current permission settings.
9. In the *Manage public permissions* field, choose whether or not to grant public read access to the object. By default, the recommended setting is *Do not grant public read access to this object.*
10. Click *Next*.
11. In the **Set properties** section, choose a storage class based on your use case and access requirements. By default, the storage class is set to *Standard*, which is used for frequently accessed data.
12. Set **Encryption** settings. By default, None is selected. This means that S3 will encrypt objects as per bucket settings.
13. Add object metadata by selecting a key in the *Header* field and adding a value in the *Value* field, and then click *Save* to save the key-header pair. By default, no metadata is associated with the object.
14. Add tags to search, organize, and manage access by adding a key in the *Key* field and a value in the *Value* field. By default, no tags are associated with the object.
15. Click *Next.*
16. In the **Review** section, make sure that all of your settings are set according to your preferences. If not, click *Previous* to edit any incorrect sections.
17. Click *Upload*. The HTML file will appear in your bucket.

If you choose to upload additional website content, such as images, documents, or other files, follow steps 4 through 17.

The HTML files that you upload into your bucket will contain the pages of your website. With these files, you will be able to show images, links, redirections to other pages of your website or other websites, and other information that you wish to publicly share on your website. For example, a page with your contact information could be contact.html or a page with a successful upload message could be successful_upload.html.