Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3) is a secure, scalable, and cost-effective object storage service offered by Amazon Web Services (AWS). It provides a vast amount of storage for any type of data, including images, videos, documents, and code. S3 is designed for durability, scalability, and high availability, making it an ideal solution for storing and retrieving any amount of data from anywhere in the world.

Key Components of S3

- **Buckets:** A bucket is a container that holds objects. Each object is identified by a unique key, and buckets can store an unlimited number of objects.
- **Objects:** Objects are the basic units of storage in S3. Each object is a file that can be up to 5 TB in size.
- **Access Control Lists (ACLs): ACLs** determine who can access and modify objects in a bucket. You can grant different levels of access to different users and groups.
- **Security:** S3 is a very secure storage service. It uses AES-256 encryption to protect your data at rest and in transit.
- **Durability:** S3 is designed for durability. Your data is stored across multiple data centers, and S3 has a 99.999999999% (11 nines) durability guarantee.
- **Scalability:** S3 can scale to any size. You can store as much data as you need, and S3 will automatically adjust to accommodate your changing needs.
- **Cost-Effectiveness:** S3 is a very cost-effective storage service. You only pay for the storage you use, and there are no upfront costs or minimum commitments.
- Easy to Use: S3 is a very easy-to-use storage service. You can create buckets, upload objects, and manage permissions using the AWS Management Console, the AWS CLI, or the Amazon S3 API.

Key Features:

1. **Durability and Availability:**
   - S3 is designed for 99.999999999% (11 nines) durability of objects over a given year.
   - It provides high availability by replicating data across multiple devices and facilities within a region.

2. **Storage Classes:**

   - S3 offers different storage classes to optimize costs and performance, including STANDARD, INTELLIGENT_TIERING, ONEZONE_IA, GLACIER, and DEEP_ARCHIVE.

3. **Access Control:**

   - Access to S3 buckets and objects is controlled through a combination of bucket policies, IAM (Identity and Access Management) policies, Access Control Lists (ACLs), and query string authentication.
   - Fine-grained access control allows specifying who can access objects, and how (e.g., read, write, delete).

4. **Versioning:**

   - S3 supports versioning, allowing you to preserve, retrieve, and restore every version of every object stored in a bucket. This helps with data recovery and compliance requirements.

5. **Logging and Monitoring:**

   - S3 provides server access logging to capture detailed information about requests made against your bucket.
   - AWS Cloud Watch metrics and S3 event notifications can be used for monitoring and alerting.

6. **Data Transfer Acceleration:**

   - S3 Transfer Acceleration (Amazon S3 Transfer Acceleration) uses the CloudFront globally distributed edge locations to accelerate transfers to and from S3.

7. **Multipart Upload:**

   - Large objects can be uploaded in parts using the Multipart Upload feature. This helps in improving performance, reliability, and the ability to pause and resume uploads.

8. **Server-Side Encryption:**

   - S3 supports server-side encryption to encrypt data at rest. You can choose from different encryption options, including SSE-S3, SSE-KMS, and SSE-C.

9. **Lifecycle Management:**

    - S3 provides lifecycle policies to automatically transition objects between storage classes or delete them when they are no longer needed.

10. **Event Notifications:**

    - S3 can generate event notifications when certain events occur in your bucket (e.g., object creation, deletion). These events can trigger workflows or Lambda functions.

Amazon S3 is a highly reliable and scalable storage service, making it suitable for various use cases such as backup and restore, data archiving, content distribution, and more. The flexibility and rich feature set of S3 make it a fundamental building block in many AWS architectures.

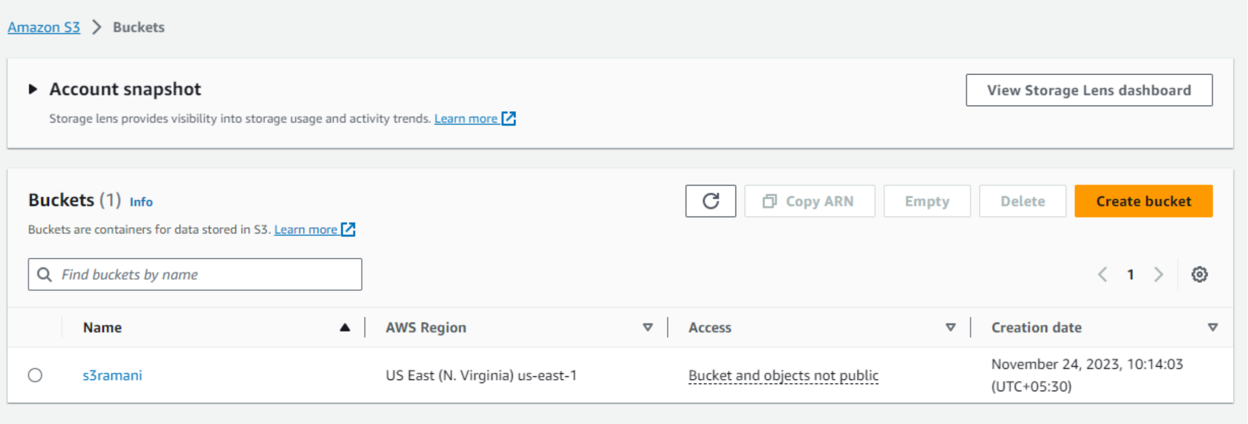
Common Use Cases for S3

- **Website Hosting:** S3 is a popular choice for hosting static websites. It is a very cost-effective and scalable way to host a website, and S3 is integrated with Amazon Cloud Front, a content delivery network (CDN) that can help to improve the performance of your website.
- **Data Backup and Archiving:** S3 is a great way to back up and archive data. It is a very durable and cost-effective storage solution, and S3 is integrated with AWS Glacier, a low-cost storage service for infrequently accessed data.
- **Data Storage for Applications:** S3 is a popular choice for storing data for applications. It is a very scalable and durable storage solution, and S3 can be accessed from anywhere in the world.

Creating a Bucket in S3

Steps to Create a Bucket in S3 (AWS Management Console):

1. **Sign in to AWS:**
    - Go to the AWS Management Console.
    - Sign in to your AWS account.
2. **Navigate to S3:**
    - In the AWS Management Console, find and select the "S3" service. You can use the search bar if needed.
3. **Click "Create Bucket":**
    - In the S3 dashboard, click the "Create bucket" button.
4. **Configure Bucket:**
    - Fill in the following details:
        - **Bucket Name:** Enter a globally unique name for your bucket. Follow DNS naming conventions (lowercase letters, numbers, hyphens, and periods).
        - **Region:** Choose the AWS region where you want to create the bucket. This is the region where your data will be stored.
5. **Configure Options (Optional):**
    - You can configure additional options, such as setting up versioning, server access logging, and tags. These are optional and can be configured based on your requirements.
6. **Set Permissions (Optional):**
    - You can configure bucket permissions during the creation process. You can set up public access, add bucket policies, or configure access control lists (ACLs).
7. **Review Configuration:**
    - Review the configuration settings for your bucket.
8. **Create Bucket:**
    - Click the "Create bucket" button to create your S3 bucket.



Amazon S3 > Buckets

▸ Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

View Storage Lens dashboard

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Copy ARN | Empty | Delete | Create bucket

Find buckets by name

< 1 >

	Name ▲	AWS Region ▽	Access ▽	Creation date ▽
○	s3ramani	US East (N. Virginia) us-east-1	Bucket and objects not public	November 24, 2023, 10:14:03 (UTC+05:30)

Creating an Object in S3

Steps to Create an Object in S3 (AWS Management Console):

1. **Sign in to AWS:**
    - Go to the AWS Management Console.
    - Sign in to your AWS account.
2. **Navigate to S3:**
    - In the AWS Management Console, find and select the "S3" service. You can use the search bar if needed.
3. **Select the Bucket:**
    - In the S3 dashboard, click on the name of the bucket where you want to create the object.
4. **Click "Upload":**
    - In the bucket overview, click the "Upload" button.
5. **Add Files:**
    - Click the "Add files" button to select the file(s) you want to upload.
    - Alternatively, you can drag and drop files directly into the upload window.
6. **Configure Upload Options (Optional):**
    - You can configure additional options such as setting access permissions, adding tags, and configuring storage class.

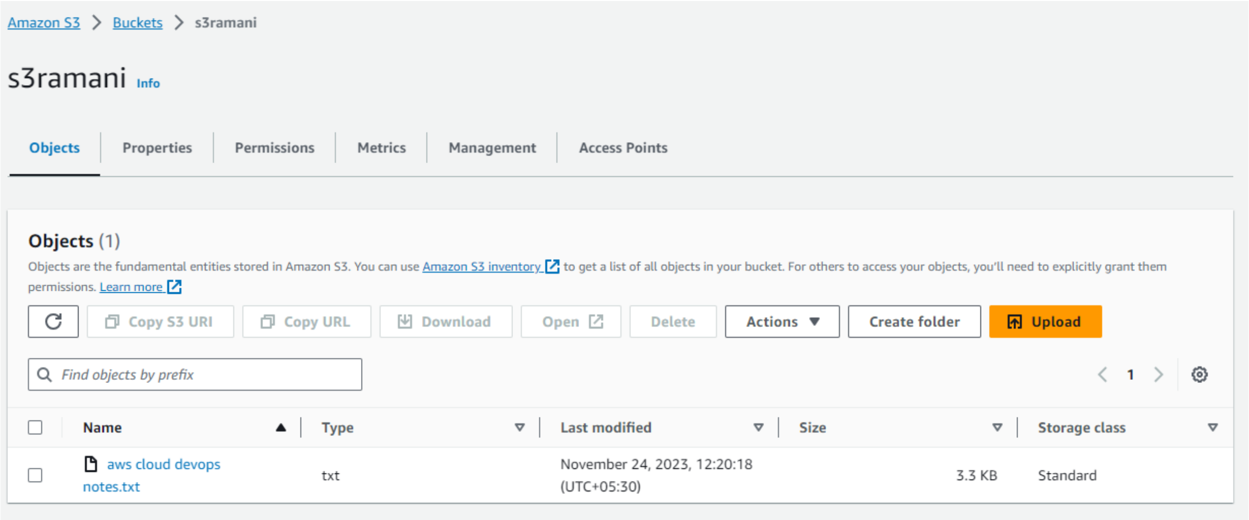
- These options are optional and can be configured based on your requirements.

7. **Review and Start Upload:**

   - Review the files you've added and click the "Upload" button to start the upload process.

8. **Monitor Upload Progress:**

   - Monitor the progress of the upload. Once complete, you will see a confirmation message.



Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	aws cloud devops notes.txt	txt	November 24, 2023, 12:20:18 (UTC+05:30)	3.3 KB	Standard

Additional Notes:

- **Object Key:**

  - The object key is the unique identifier for the object within the bucket. It can include a prefix (folder structure) and the actual file name.

  - For example, if your bucket is named "my-bucket" and you upload a file named "example.txt" to a folder named "documents," the object key might be "documents/example.txt."

- **Permissions:**

  - Pay attention to the permissions you set during the upload process. By default, objects inherit permissions from the bucket, but you can also configure specific permissions for individual objects.

To allow access to an object in Amazon S3 via its object URL, you need to ensure that the necessary permissions are configured. Permissions in S3 can be managed using a combination of bucket policies, object ACLs (Access Control Lists), and IAM (Identity and Access Management) policies. Below are the steps to grant public access to an object in S3 through its URL:

Granting Public Access to an Object:

1. **Object ACL (Access Control List):**

- **Navigate to the S3 Console:**
- **Select the Bucket and Object:**
  - Click on the bucket that contains the object.
  - Navigate to the object for which you want to grant public access.
- **Open the Object's "Actions" Menu:**
  - Click on the object, and in the "Actions" menu, select "Make public."
- **Confirm Making the Object Public:**
  - Confirm that you want to make the object public.

2. **Bucket Policy:**

- **Select the Bucket:**
  - Click on the bucket that contains the object.
- **Open the Bucket's "Permissions" Tab:**
  - Navigate to the "Permissions" tab.
- **Block Public Access:**
  - ➢ In the "Permissions" tab, click on "Block public access."
  - ➢ Uncheck options like "Block all public access" if they are checked.
  - ➢ Explicitly set permissions for individual objects or enable "Bucket settings for Block all public access" based on your needs.

- **Object ACL (Access Control List):**
    - After enabling public access, ensure that object ownership ACL enabled. By default, the owner of the bucket owns the objects within it.

3. Go to properties to enable object versioning

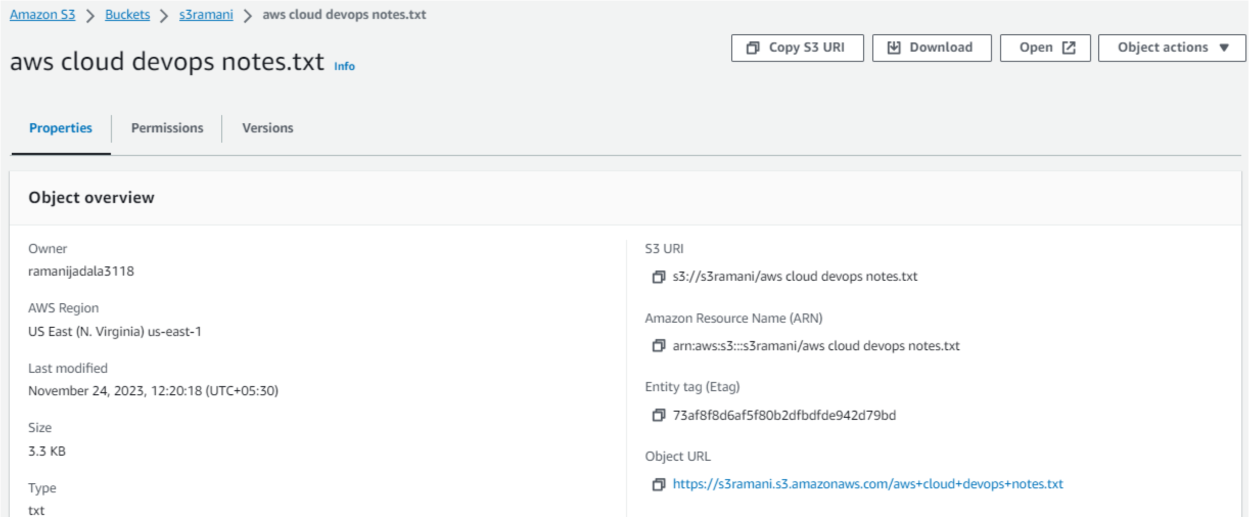
Important Note:

- **Security Implications:**
    - Granting public access to objects should be done cautiously. It means that anyone with the object URL can access the object. Ensure that you really need to make the object public, and consider using more fine-grained access controls if possible.

Once these steps are completed, users should be able to access the object through its URL, which typically follows the format:

https://s3.amazonaws.com/your-bucket-name/your-object-key

Replace "your-bucket-name" with your actual bucket name and "your-object-key" with the actual object key.



aws cloud devops notes.txt

Object overview

Owner
ramanijadala3118

AWS Region
US East (N. Virginia) us-east-1

Last modified
November 24, 2023, 12:20:18 (UTC+05:30)

Size
3.3 KB

Type
txt

S3 URI
s3://s3ramani/aws cloud devops notes.txt

Amazon Resource Name (ARN)
arn:aws:s3:::s3ramani/aws cloud devops notes.txt

Entity tag (Etag)
73af8f8d6af5f80b2dfbdfde942d79bd

Object URL
https://s3ramani.s3.amazonaws.com/aws+cloud+devops+notes.txt

Creating a static website using Amazon S3

Creating a static website using Amazon S3 involves uploading your HTML files and configuring the bucket to act as a website. Here's a step-by-step guide to creating a simple static website using index.html and error.html in an S3 bucket:

Step 1: Create HTML Files

Create two HTML files, **index.html** for the main page and **error.html** for a custom error page. For example:

Sample Html code

index.html

```
<!DOCTYPE html>

<html>

<body>

<h1 style="background-color:DodgerBlue;">Welcome to My Static Website!</h1>

<p>This is the main page.</p>

</body> </html>
```

error.html

```
 <!DOCTYPE html>

<html>

<body>

<h1>404 - Not Found</h1>

<p>Sorry, the page you're looking for does not exist.</p>

</body> </html>

</body>

</html>
```

Step 2: Upload HTML Files to S3

1. **Navigate to S3:**
   - Go to the AWS Management Console.
   - Select the "S3" service.
2. **Create a New Bucket:**
   - Click on the "Create bucket" button.
   - Choose a unique name for your bucket and select a region.
   - Click through the rest of the setup, and create the bucket.
3. **Upload HTML Files:**
   - Inside your bucket, create a folder (e.g., "website").
   - Upload the **index.html** and **error.html** files to the "website" folder.

Step 3: Enable Static Website Hosting

1. **Navigate to Properties:**
   - In your S3 bucket, click on the "Properties" tab.
2. **Enable Static Website Hosting:**
   - Scroll down to the "Static website hosting" card.
   - Click on the "Edit" button.
   - Choose "Use this bucket to host a website."
   - Set **index.html** as the index document and **error.html** as the error document.
   - Click "Save changes."

Step 4: Access Your Static Website

1. **Find the Endpoint:**
   - In the "Static website hosting" card, you'll see an endpoint (e.g., **http://your-bucket-name.s3-website-your-region.amazonaws.com/**).

2. **Access Your Website:**

    - Open a web browser and navigate to the endpoint.

You should see your static website's main page. If you intentionally type an incorrect path, it should redirect to the custom error page.

Step 5: Optionally Configure Public Access

If you want your website to be publicly accessible, you might need to adjust the bucket policy or Access Control Lists (ACLs) to allow public access to the objects.

Important Note:

- Ensure that your S3 bucket, folder, and objects have the appropriate permissions for public access if you want the website to be accessible to the public.
- A basic setup for a static website in S3. Depending on your requirements, you might need additional configurations such as setting up a custom domain, using Cloud Front for content delivery, and securing your website using HTTPS.

Amazon S3 Lifecycle management

Amazon S3 Lifecycle management allows you to automate the transition and expiration of your objects over time. This can help you optimize storage costs and manage the lifecycle of your data. Here's an overview of how to set up and configure lifecycle management in Amazon S3:

Key Concepts:

1. **Transition Actions:**

    - **Storage Class Transitions:** Move objects between storage classes (e.g., from STANDARD to INTELLIGENT_TIERING).
    - **Expiration:** Define rules to delete objects after a specified number of days or on a specific date.

2. **Storage Classes:**

    - Amazon S3 offers different storage classes, each with different durability, availability, and cost characteristics (e.g., STANDARD, INTELLIGENT_TIERING, GLACIER).

Steps to Configure Lifecycle Management:

1. **Navigate to the S3 Console:**
    - Go to the AWS Management Console.
    - Select the "S3" service.
2. **Select the Bucket:**
    - Click on the bucket for which you want to configure lifecycle management.
3. **Navigate to the Management Tab:**
    - Click on the "Management" tab.
4. **Create a Lifecycle Rule:**
    - Click on the "Create lifecycle rule" button.
5. **Configure Rule Settings:**
    - **Name:** Provide a name for your rule.
    - **Prefix:** Optionally, you can set a prefix to filter objects to which the rule applies.
    - **Transitions:** Define storage class transitions based on the number of days since creation.
    - **Expirations:** Set up expiration actions based on the number of days since creation.
6. **Review and Create:**
    - Review your configuration and click "Create rule."

Example Lifecycle Rule:

Here's an example of a lifecycle rule that transitions objects to the INTELLIGENT_TIERING storage class after 30 days and deletes them after 365 days:

- **Name:** TransitionAndExpireRule
- **Prefix:** (leave blank for all objects)
- **Transitions:**
    - **Storage Class:** INTELLIGENT_TIERING

    - **Days After Creation:** 30

- **Expirations:**
    - **Days After Creation:** 365

Verification:

After setting up a lifecycle rule, you can monitor its effects in the S3 Management Console. Objects meeting the conditions specified in the rule will automatically transition between storage classes or be deleted based on your configuration.

Lifecycle management in S3 helps you automate data management tasks, ensuring that your storage is optimized for cost and performance over time.