Amazon Simple Storage Service

Getting Started Guide

Amazon Simple Storage Service: Getting Started Guide

Table of Contents

This guide is no longer being updated. For current information and instructions, see the new Amazon S3 User Guide.

Getting started with Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. You can accomplish these tasks using the AWS Management Console, which is a simple and intuitive web interface.

Amazon S3 stores data as objects within buckets. An object is a file and any optional metadata that describes the file. To store a file in Amazon S3, you upload it to a bucket. When you upload a file as an object, you can set permissions on the object and any metadata.

Buckets are containers for objects. You can have one or more buckets. You can control access for each bucket, deciding who can create, delete, and list objects in it. You can also choose the geographical Region where Amazon S3 will store the bucket and its contents and view access logs for the bucket and its objects.

This guide introduces you to Amazon S3 and explains how to use the AWS Management Console to complete the following tasks:

- Setting up Amazon S3 (p. 2)
- Creating a bucket (p. 5)
- Uploading an object to a bucket (p. 6)
- Downloading an object (p. 7)
- Copying an object to a folder (p. 8)
- Deleting objects and buckets (p. 9)

For information about Amazon S3 features, pricing, and frequently asked questions, see the Amazon S3 product page.

Setting up Amazon S3

When you sign up for AWS, your AWS account is automatically signed up for all services in AWS, including Amazon S3. You are charged only for the services that you use.

With Amazon S3, you pay only for what you use. For more information about Amazon S3 features and pricing, see Amazon S3. If you are a new Amazon S3 customer, you can get started with Amazon S3 for free. For more information, see AWS Free Tier.

To get started with Amazon S3, follow these steps:

Topics

Sign up for AWS

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create an IAM user

When you first create an Amazon Web Services (AWS) account, you begin with a single sign-in identity. That identity has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user*. When you sign in, enter the email address and password that you used to create the account.

> **Important**
> We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks. To view the tasks that require you to sign in as the root user, see AWS Tasks That Require Root User.

If you signed up for AWS but have not created an IAM user for yourself, follow these steps.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   Note
   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
2. In the navigation pane, choose **Users** and then choose **Add user**.
3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note
    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.
12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

Sign in as an IAM user

After you create an IAM user, you can sign in to AWS with your IAM user name and password.

Before you sign in as an IAM user, you can verify the sign-in link for IAM users in the IAM console. On the IAM Dashboard, under **IAM users sign-in link**, you can see the sign-in link for your AWS account. The URL for your sign-in link contains your AWS account ID without dashes (-).

If you don't want the URL for your sign-in link to contain your AWS account ID, you can create an account alias. For more information, see Creating, Deleting, and Listing an AWS Account Alias in the *IAM User Guide*.

To sign in as an AWS user

1. Sign out of the AWS Management Console.
2. Enter your sign-in link.

   Your sign-in link includes your AWS account ID (without dashes) or your AWS account alias:

   ```
   https://aws_account_id_or_alias.signin.aws.amazon.com/console
   ```
3. Enter the IAM user name and password that you just created.

   When you're signed in, the navigation bar displays "*your_user_name @ your_aws_account_id*".

Creating a bucket

Now that you've signed up for AWS you're ready to create a bucket using the AWS Management Console. Every object in Amazon S3 is stored in a bucket. Before you can store data in Amazon S3, you must create a bucket.

> **Note**
> You are not charged for creating a bucket. You are charged only for storing objects in the bucket and for transferring objects in and out of the bucket. The charges that you incur through following the examples in this guide are minimal (less than $1). For more information about storage charges, see Amazon S3 pricing.

To create a bucket using the AWS Command Line Interface, see create-bucket in the *AWS CLI Command Reference*.

To create a bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose **Create bucket**.

   The **Create bucket** page opens.
3. In **Bucket name**, enter a DNS-compliant name for your bucket.

   The bucket name must:

   - Be unique across all of Amazon S3.
   - Be between 3 and 63 characters long.
   - Not contain uppercase characters.
   - Start with a lowercase letter or number.

   After you create the bucket, you can't change its name. For information about naming buckets, see Rules for Bucket Naming in the *Amazon Simple Storage Service Developer Guide*.

   > **Important**
   > Avoid including sensitive information, such as account numbers, in the bucket name. The bucket name is visible in the URLs that point to the objects in the bucket.
4. In **Region**, choose the AWS Region where you want the bucket to reside.

   Choose a Region close to you to minimize latency and costs and address regulatory requirements. Objects stored in a Region never leave that Region unless you explicitly transfer them to another Region. For a list of Amazon S3 AWS Regions, see AWS Service Endpoints in the *Amazon Web Services General Reference*.
5. In **Bucket settings for Block Public Access**, keep the values set to the defaults.

   By default Amazon S3 blocks all public access to your buckets. We recommend that you leave all Block Public Access settings enabled. For more information about blocking public access, see Using Amazon S3 Block Public Access in the *Amazon Simple Storage Service Developer Guide*.
6. Choose **Create bucket**.

You've created a bucket in Amazon S3.

To add an object to your bucket, see Uploading an object to a bucket (p. 6).

Uploading an object to a bucket

Now that you've created a bucket, you're ready to upload an object to it. An object can be any kind of file: a text file, a photo, a video, and so on.

To upload an object to a bucket

1. In the **Buckets** list, choose the name of the bucket that you want to upload your object to.
2. On the **Objects** tab for your bucket, choose **Upload**.
3. Under **Files and folders**, choose **Add files**.
4. Choose a file to upload, and then choose **Open.**
5. Choose **Upload**.

You've successfully uploaded an object to your bucket.

To view your object, see Downloading an object (p. 7).

Downloading an object

Now that you've uploaded an object to a bucket, you can view information about your object and download the object to your local computer.

To download an object from a bucket

1. In the **Buckets** list, choose the name of the bucket that you created.
2. In the **Objects** list, choose the name of the object that you uploaded.

   The object overview opens.
3. On the **Details** tab, review information about your object.
4. To download the object to your computer, choose **Object actions** and choose **Download**.

You've successfully downloaded your object.

To copy and paste your object within Amazon S3, see Copying an object to a folder (p. 8).

Copying an object to a folder

You've already added an object to a bucket and downloaded the object. In this tutorial, you create a folder and copy your object to it.

To copy an object to a folder

1. In the **Buckets** list, choose your bucket name.
2. Choose **Create folder** and configure a new folder:

    a. Enter a folder name (for example, `favorite-pics`).

    b. For the folder encryption setting, choose **None**.

    c. Choose **Save**.
3. Navigate to the Amazon S3 bucket or folder that contains the objects that you want to copy.
4. Select the check box to the left of the names of the objects that you want to copy.
5. Choose **Actions** and choose **Copy** from the list of options that appears.

    Alternatively, choose **Copy** from the options in the upper right.
6. Choose the destination folder:

    a. Choose **Browse S3**.

    b. Choose the option button to the left of the folder name.

    To navigate into a folder and choose a subfolder as your destination, choose the folder name.

    c. Choose **Choose destination**.

    The path to your destination folder appears in the **Destination** box. In **Destination**, you can alternately enter your destination path, for example, s3://*bucket-name*/*folder-name*/.
7. In the bottom right, choose **Copy**.

    Amazon S3 moves your objects to the destination folder.

To delete an object and a bucket in Amazon S3, see Deleting objects and buckets (p. 9).

Deleting objects and buckets

When you no longer need an object or a bucket, we recommend that you delete them to prevent further charges. If you completed this getting started walkthrough as a learning exercise and do not plan to use your bucket or objects, we recommend that you delete your bucket so that charges no longer accrue. Before you delete your bucket, you must empty the bucket or delete the objects in the bucket. After you delete your objects and bucket, they are no longer available.

If you want to continue to use the same bucket name, we recommend that you delete the objects or empty the bucket but do not delete the bucket. After you delete a bucket, the name becomes available to reuse. However, another account might create a bucket with the same name before you have a chance to reuse it.

Topics

Emptying your bucket

If you plan to delete your bucket, you must first empty your bucket, which deletes all the objects in the bucket.

To empty a bucket

1. In the **Buckets** list, select the bucket that you want to empty, and then choose **Empty**.
2. To confirm that you want to empty the bucket and delete all the objects in it, in **Empty bucket**, enter the name of the bucket.

   > **Important**
   > Emptying the bucket cannot be undone. Objects added to the bucket while the empty bucket action is in progress will be deleted.
3. To empty the bucket and delete all the objects in it, and choose **Empty**.

   An **Empty bucket: Status** page opens that you can use to review a summary of failed and successful object deletions.
4. To return to your bucket list, choose **Exit**.

Deleting an object

If you want to choose which objects you delete without emptying all the objects from your bucket, you can delete an object.

1. In the **Buckets** list, choose the name of the bucket that you want to delete an object from.
2. Select the check box to the left of the names of the objects that you want to delete.
3. Choose **Actions** and choose **Delete** from the list of options that appears.

   Alternatively, choose **Delete** from the options in the upper right.

4. Enter `delete` if asked to confirm that you want to delete these objects.
5. Choose **Delete objects** in the bottom right and Amazon S3 deletes the specified objects.

Deleting your bucket

After you empty your bucket or delete all the objects from your bucket, you can delete your bucket.

1. To delete a bucket, in the **Buckets** list, select the bucket.
2. Choose **Delete**.
3. To confirm deletion, in **Delete bucket**, enter the name of the bucket.

   Important
   Deleting a bucket cannot be undone. Bucket names are unique. If you delete your bucket, another AWS user can use the name. If you want to continue to use the same bucket name, don't delete your bucket. Instead, empty and keep the bucket.
4. To delete your bucket, choose **Delete bucket**.

For more information about using Amazon S3, see Where do I go from here? (p. 11)

Where do I go from here?

In the preceding examples, you learned how to perform some basic Amazon S3 tasks. For more in-depth information, see one of the following Amazon S3 guides:

- The Amazon Simple Storage Service Console User Guide to learn more about using the Amazon S3 console.
- The Amazon Simple Storage Service Developer Guide to find detailed information about Amazon S3 features and code examples to support those features.
- The Amazon Simple Storage Service API Reference to find details about the Amazon S3 REST API.

The following topics explain various ways in which you can gain a deeper understanding of Amazon S3 so that you can implement it in your applications.

Topics

Common use scenarios

The AWS Solutions site lists many of the ways you can use Amazon S3. The following list summarizes some of those ways.

- **Backup and storage** – Provide data backup and storage services for others.
- **Application hosting** – Provide services that deploy, install, and manage web applications.
- **Media hosting** – Build a redundant, scalable, and highly available infrastructure that hosts video, photo, or music uploads and downloads.
- **Software delivery** – Host your software applications that customers can download.

For more information, see AWS Solutions.

Considerations going forward

This section introduces you to topics you should consider before launching your own Amazon S3 product.

Topics

AWS account and security credentials

When you signed up for the service, you created an AWS account using an email address and password. Those are your AWS account root user credentials. As a best practice, you should not use your root user credentials to access AWS. Nor should you give your credentials to anyone else. Instead, create individual users for those who need access to your AWS account. First, create an AWS Identity and Access Management (IAM) administrator user for yourself and use it for your daily work. For details, see Creating your first IAM admin user and group in the *IAM User Guide*. Then create additional IAM users for other people. For details, see Creating your first IAM delegated user and group in the *IAM User Guide*.

If you're an account owner or administrator and want to know more about IAM, see the product description at https://aws.amazon.com/iam or the technical documentation in the IAM User Guide.

Security

Amazon S3 provides authentication mechanisms to secure data stored in Amazon S3 against unauthorized access. Unless you specify otherwise, only the AWS account owner can access data uploaded to Amazon S3. For more information about how to manage access to buckets and objects, go to Identity and Access Management in Amazon S3 in the *Amazon Simple Storage Service Developer Guide*.

You can also encrypt your data before uploading it to Amazon S3.

AWS integration

You can use Amazon S3 alone or in concert with one or more other Amazon products. The following are the most common products used with Amazon S3:

- Amazon EC2
- Amazon EMR
- Amazon SQS
- Amazon CloudFront

Pricing

Learn the pricing structure for storing and transferring data on Amazon S3. For more information, see Amazon S3 pricing.

Advanced Amazon S3 features

The examples in this guide show how to accomplish the basic tasks of creating a bucket, uploading and downloading data to and from it, and moving and deleting the data. The following table summarizes some of the most common advanced functionality offered by Amazon S3. Note that some advanced functionality is not available in the AWS Management Console and requires that you use the Amazon S3 API. All advanced functionality and how to use it is described in the Amazon Simple Storage Service Developer Guide.

Link	Functionality
Requester Pays Buckets	Learn how to configure a bucket so that a customer pays for the downloads they make.

Link	Functionality
Using BitTorrent With Amazon S3	Use BitTorrent, which is an open, peer-to-peer protocol for distributing files.
Versioning	Learn about Amazon S3 versioning capabilities.
Hosting Static Websites	Learn how to host a static website on Amazon S3.
Object Lifecycle Management	Learn how to manage the lifecycle of objects in your bucket. Lifecycle management includes expiring objects and archiving objects (transitioning objects to the `S3 S3 Glacier` storage class).

Access control best practices

Amazon S3 provides a variety of security features and tools. The following scenarios should serve as a guide to what tools and settings you might want to use when performing certain tasks or operating in specific environments. Proper application of these tools can help maintain the integrity of your data and help ensure that your resources are accessible to the intended users.

Topics

Creating a new bucket

When creating a new bucket, you should apply the following tools and settings to help ensure that your Amazon S3 resources are protected.

Block Public Access

S3 Block Public Access provides four settings to help you avoid inadvertently exposing your S3 resources. You can apply these settings in any combination to individual access points, buckets, or entire AWS accounts. If you apply a setting to an account, it applies to all buckets and access points that are owned by that account. By default, the **Block all public access** setting is applied to new buckets created in the Amazon S3 console.

For more information, see The meaning of "public" in the *Amazon Simple Storage Service Developer Guide*.

If the S3 Block Public Access settings are too restrictive, you can use AWS Identity and Access Management (IAM) identities to grant access to specific users rather than disabling all Block Public Access settings. Using Block Public Access with IAM identities helps ensure that any operation that is blocked by a Block Public Access setting is rejected unless the requesting user has been given specific permission.

For more information, see Block public access settings in the *Amazon Simple Storage Service Developer Guide*.

Grant access with IAM identities

When setting up accounts for new team members who require S3 access, use IAM users and roles to ensure least privileges. You can also implement a form of IAM multi-factor authentication (MFA) to support a strong identity foundation. Using IAM identities, you can grant unique permissions to users and specify what resources they can access and what actions they can take. IAM identities provide increased capabilities, including the ability to require users to enter login credentials before accessing shared resources and apply permission hierarchies to different objects within a single bucket.

For more information, see Example 1: Bucket owner granting its users bucket permissions in the *Amazon Simple Storage Service Developer Guide*.

Bucket policies

With bucket policies, you can personalize bucket access to help ensure that only those users you have approved can access resources and perform actions within them. In addition to bucket policies, you should use bucket-level Block Public Access settings to further limit public access to your data.

For more information, see Policies and Permissions in Amazon S3 in the *Amazon Simple Storage Service Developer Guide*.

When creating policies, avoid the use of wildcards in the `Principal` element because it effectively allows anyone to access your Amazon S3 resources. It's better to explicitly list users or groups that are allowed to access the bucket. Rather than including a wildcard for their actions, grant them specific permissions when applicable.

To further maintain the practice of least privileges, Deny statements in the `Effect` element should be as broad as possible and Allow statements should be as narrow as possible. Deny effects paired with the "`s3:*`" action are another good way to implement opt-in best practices for the users included in policy condition statements.

For more information about specifying conditions for when a policy is in effect, see Amazon S3 Condition Keys in the *Amazon Simple Storage Service Developer Guide*.

Buckets in a VPC setting

When adding users in a corporate setting, you can use a virtual private cloud (VPC) endpoint to allow any users in your virtual network to access your Amazon S3 resources. VPC endpoints enable developers to provide specific access and permissions to groups of users based on the network the user is connected to. Rather than adding each user to an IAM role or group, you can use VPC endpoints to deny bucket access if the request doesn’t originate from the specified endpoint.

For more information, see Example Bucket Policies for VPC Endpoints for Amazon S3 in the *Amazon Simple Storage Service Developer Guide*.

Storing and sharing data

Use the following tools and best practices to store and share your Amazon S3 data.

Versioning and Object Lock for data integrity

If you use the Amazon S3 console to manage buckets and objects, you should implement S3 Versioning and S3 Object Lock. These features help prevent accidental changes to critical data and enable you to roll back unintended actions. This capability is particularly useful when there are multiple users with full write and execute permissions accessing the Amazon S3 console.

For information about S3 Versioning, see Using versioning in the *Amazon Simple Storage Service Developer Guide*. For information about Object Lock, see Locking objects using S3 Object Lock in the *Amazon Simple Storage Service Developer Guide*.

Object lifecycle management for cost efficiency

To manage your objects so that they are stored cost effectively throughout their lifecycle, you can pair lifecycle policies with object versioning. Lifecycle policies define actions that you want S3 to take during an object's lifetime. For example, you can create a lifecycle policy that will transition objects to another storage class, archive them, or delete them after a specified period of time. You can define a lifecycle policy for all objects or a subset of objects in the bucket by using a shared prefix or tag.

For more information, see Object lifecycle management in the *Amazon Simple Storage Service Developer Guide*.

Cross-Region Replication for multiple office locations

When creating buckets that are accessed by different office locations, you should consider implementing S3 Cross-Region Replication. Cross-Region Replication helps ensure that all users have access to the resources they need and increases operational efficiency. Cross-Region Replication offers increased availability by copying objects across S3 buckets in different AWS Regions. However, the use of this tool increases storage costs.

For more information, see Replication in the *Amazon Simple Storage Service Developer Guide*.

Permissions for secure static website hosting

When configuring a bucket to be used as a publicly accessed static website, you need to disable all Block Public Access settings. It is important to only provide `s3:GetObject` actions and not `ListObject` or `PutObject` permissions when writing the bucket policy for your static website. This helps ensure that users cannot view all the objects in your bucket or add their own content.

For more information, see Setting permissions for website access in the *Amazon Simple Storage Service Developer Guide*.

Amazon CloudFront provides the capabilities required to set up a secure static website. Amazon S3 static websites only support HTTP endpoints. CloudFront uses the durable storage of Amazon S3 while providing additional security headers like HTTPS. HTTPS adds security by encrypting a normal HTTP request and protecting against common cyber attacks.

For more information, see Getting started with a secure static website in the *Amazon CloudFront Developer Guide*.

Sharing resources

There are several different ways that you can share resources with a specific group of users. You can use the following tools to share a set of documents or other resources to a single group of users, department, or office. Although they can all be used to accomplish the same goal, some tools might pair better than others with your existing settings.

User policies

You can share resources with a limited group of people using IAM groups and user policies. When creating a new IAM user, you are prompted to create and add them to a group. However, you can create and add users to groups at any point. If the individuals you intend to share these resources with are already set up within IAM, you can add them to a common group and share the bucket with their group within the user policy. You can also use IAM user policies to share individual objects within a bucket.

For more information, see Allowing an IAM User Access to One of Your Buckets in the *Amazon Simple Storage Service Developer Guide*.

Access control lists

As a general rule, we recommend that you use S3 bucket policies or IAM policies for access control. Amazon S3 access control lists (ACLs) are a legacy access control mechanism that predates IAM. If you already use S3 ACLs and you find them sufficient, there is no need to change. However, certain access control scenarios require the use of ACLs. For example, when a bucket owner wants to grant permission to objects, but not all objects are owned by the bucket owner, the object owner must first grant permission to the bucket owner. This is done using an object ACL.

For more information, see Example 3: Bucket owner granting its users permissions to objects it does not own in the *Amazon Simple Storage Service Developer Guide*.

Prefixes

When trying to share specific resources from a bucket, you can replicate folder-level permissions using prefixes. The Amazon S3 console supports the folder concept as a means of grouping objects by using a shared name prefix for objects. You can then specify a prefix within the conditions of an IAM user's policy to grant them explicit permission to access the resources associated with that prefix.

For more information, see Using folders in the *Amazon Simple Storage Service Console User Guide*.

Tagging

If you use object tagging to categorize storage, you can share objects that have been tagged with a specific value with specified users. Resource tagging allows you to control access to objects based on the tags associated with the resource that a user is trying to access. To do this, use the `ResourceTag/key-name` condition within an IAM user policy to allow access to the tagged resources.

For more information, see Controlling access to AWS resources using resource tags in the *IAM User Guide*.

Protecting data

Use the following tools to help protect data in transit and at rest, both of which are crucial in maintaining the integrity and accessibility of your data.

Object encryption

Amazon S3 offers several object encryption options that protect data in transit and at rest. Server-side encryption encrypts your object before saving it on disks in its data centers and then decrypts it when you download the objects. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. When setting up server-side encryption, you have three mutually exclusive options:

- Amazon S3 managed keys (SSE-S3)
- Customer master keys (CMK) stored in AWS Key Management Service (SSE-KMS)
- Customer-provided keys (SSE-C)

For more information, see Protecting data using server-side encryption in the *Amazon Simple Storage Service Developer Guide*.

Client-side encryption is the act of encrypting data before sending it to Amazon S3. For more information, see Protecting data using client-side encryption in the *Amazon Simple Storage Service Developer Guide*.

Signing methods

Signature Version 4 is the process of adding authentication information to AWS requests sent by HTTP. For security, most requests to AWS must be signed with an access key, which consists of an access key ID and secret access key. These two keys are commonly referred to as your security credentials.

For more information, see Authenticating Requests (AWS Signature Version 4) and Signature Version 4 signing process.

Logging and monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon S3 solutions so that you can more easily debug a multi-point failure if one occurs. Logging can provide insight into any errors users are receiving, and when and what requests are made. AWS provides several tools for monitoring your Amazon S3 resources:

- Amazon CloudWatch
- AWS CloudTrail
- Amazon S3 Access Logs
- AWS Trusted Advisor

For more information, see Logging and monitoring in Amazon S3 in the *Amazon Simple Storage Service Developer Guide*.

Amazon S3 is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, a role, or an AWS service in Amazon S3. This feature can be paired with Amazon GuardDuty, which monitors threats against your Amazon S3 resources by analyzing CloudTrail management events and CloudTrail S3 data events. These data sources monitor different kinds of activity. For example, S3 related CloudTrail management events include operations that list or configure S3 projects. GuardDuty analyzes S3 data events from all of your S3 buckets and monitors them for malicious and suspicious activity.

For more information, see Amazon S3 protection in Amazon GuardDuty in the *Amazon GuardDuty User Guide*.

Development resources

To help you build applications using the language of your choice, we provide the following resources:

- **Sample Code and Libraries** – The AWS Developer Center has sample code and libraries written especially for Amazon S3.

  You can use these code samples as a means of understanding how to implement the Amazon S3 API. For more information, see the AWS Developer Center.
- **Tutorials** – Our Resource Center offers more Amazon S3 tutorials.

  These tutorials provide a hands-on approach for learning Amazon S3 functionality. For more information, see Articles & Tutorials.
- **Customer Forum** – We recommend that you review the Amazon S3 forum to get an idea of what other users are doing and to benefit from the questions they ask.

  The forum can help you understand what you can and can't do with Amazon S3. The forum also serves as a place for you to ask questions that other users or AWS representatives might answer. You can use the forum to report issues with the service or the API. For more information, see Discussion Forums.

Reference resources

The following list shows additional resources that you can use to further your understanding of Amazon S3.

- The Amazon Simple Storage Service Console User Guide describes all of the AWS Management Console functions related to Amazon S3.
- The Amazon Simple Storage Service Developer Guide provides a detailed discussion of the service.

  It includes an architectural overview, detailed concept descriptions, and procedures for using the API.
- The Amazon Simple Storage Service API Reference provides a detailed discussion of the actions and parameters in Amazon S3.
- The Service Health Dashboard shows you the status of the Amazon S3 web service.

  The dashboard shows you whether Amazon S3 (and all other AWS products) are functioning properly. For more information, see the Service Health Dashboard.

About this guide

This is the *Amazon Simple Storage Service Getting Started Guide*.

Amazon Simple Storage Service is frequently referred to within this guide as "Amazon S3." All copyrights and legal protections still apply.