AWS Snowball

User Guide

Table of Contents

This guide is for the Snowball (50 TB or 80 TB of storage space). If you are looking for documentation for the Snowball Edge, see the AWS Snowball Edge Developer Guide.

What Is an AWS Snowball Appliance?

AWS Snowball is a service that accelerates transferring large amounts of data into and out of AWS using physical storage appliances, bypassing the Internet. Each AWS Snowball appliance type can transport data at faster-than internet speeds. This transport is done by shipping the data in the appliances through a regional carrier. The appliances are rugged shipping containers, complete with E Ink shipping labels.

With a Snowball, you can transfer hundreds of terabytes or petabytes of data between your on-premises data centers and Amazon Simple Storage Service (Amazon S3). AWS Snowball uses Snowball appliances and provides powerful interfaces that you can use to create jobs, transfer data, and track the status of your jobs through to completion. By shipping your data in Snowballs, you can transfer large amounts of data at a significantly faster rate than if you were transferring that data over the Internet, saving you time and money.

> **Note**
> There are many options for transferring your data into AWS. Snowball is intended for transferring large amounts of data. If you want to transfer less than 10 terabytes of data between your on-premises data centers and Amazon S3, Snowball might not be your most economical choice.

Snowball uses Snowball appliances shipped through your region's carrier. Each Snowball is protected by AWS Key Management Service (AWS KMS) and made physically rugged to secure and protect your data while the Snowball is in transit. In the US regions, Snowballs come in two sizes: 50 TB and 80 TB. All other regions have 80 TB Snowballs only.

Snowball Features

Snowball with the Snowball appliance has the following features:

- You can import and export data between your on-premises data storage locations and Amazon S3.
- Snowball has an 80 TB model available in all regions, and a 50 TB model only available in the US regions.
- Encryption is enforced, protecting your data at rest and in physical transit.
- You don't have to buy or maintain your own hardware devices.
- You can manage your jobs through the AWS Snowball Management Console, or programmatically with the job management API.
- You can perform local data transfers between your on-premises data center and a Snowball. These transfers can be done through the Snowball client, a standalone downloadable client, or programmatically using Amazon S3 REST API calls with the downloadable Amazon S3 Adapter for Snowball. For more information, see Transferring Data with a Snowball (p. 51).
- The Snowball is its own shipping container, and its E Ink display changes to show your shipping label when the Snowball is ready to ship. For more information, see Shipping Considerations for AWS Snowball (p. 73).
- For a list of regions where the Snowball appliance is available, see AWS Snowball in the *AWS General Reference*.

> **Note**
> Snowball doesn't support international shipping or shipping between regions outside of the US. For more information on shipping restrictions, see Region-Based Shipping Restrictions (p. 74).

Prerequisites for Using AWS Snowball

Before transferring data into Amazon S3 using Snowball, you should do the following:

- Create an AWS account and an administrator user in AWS Identity and Access Management (IAM). For more information, see Creating an IAM User for Snowball (p. 79).
- If you are importing data, do the following:
  - Confirm that the files and folders to transfer are named according to the Object Key Naming Guidelines for Amazon S3. Any files or folders with names that don't meet these guidelines won't be imported into Amazon S3.
  - Plan what data you want to import into Amazon S3. For more information, see How to Transfer Petabytes of Data Efficiently (p. 36).
- If you are exporting data, do the following:
  - Understand what data will be exported when you create your job. For more information, see Using Export Ranges (p. 40).
  - For any files with a colon (`:`) in the file name, change the file names in Amazon S3 before you create the export job to get these files. Files with a colon in the file name fail export to Microsoft Windows Server.

Tools and Interfaces

Snowball uses the AWS Snowball Management Console and the job management API for creating and managing jobs. To perform data transfers on the Snowball appliance locally, use the Snowball client or the Amazon S3 Adapter for Snowball. To learn more about using these in detail, see the following topics:

- Using the AWS Snowball Management Console (p. 40)
- Using an AWS Snowball Appliance (p. 45)
- Transferring Data with a Snowball (p. 51)

We also recommend that you check out the job management API for AWS Snowball. For more information, see AWS Snowball API Reference.

Services Related to AWS Snowball

This guide assumes that you are an Amazon S3 user.

Are You a First-Time User of AWS Snowball?

If you are a first-time user of the Snowball service with the Snowball appliance, we recommend that you read the following sections in order:

1. To learn more about the different types of jobs, see Jobs for Standard Snowball Appliances (p. 11).
2. For an end-to-end overview of how Snowball works with the Snowball appliance, see How AWS Snowball Works with the Standard Snowball Appliance (p. 6).
3. When you're ready to get started, see Getting Started with AWS Snowball (p. 16).

Pricing

For information about the pricing and fees associated with the AWS Snowball, see AWS Snowball Pricing.

AWS Snowball Device Differences

The Snowball and the Snowball Edge are two different devices. This guide is for the Snowball. If you are looking for documentation for the Snowball Edge, see the AWS Snowball Edge Developer Guide. Both devices allow you to move huge amounts of data into and out of Amazon S3, they both have the same job management API, and they both use the same console. However, the two devices differ in hardware specifications, some features, what transfer tools are used, and price.

AWS Snowball Use Case Differences

Following is a table that shows the different use cases for the different AWS Snowball devices:

Use case	Snowball	Snowball Edge
Import data into Amazon S3	✓	✓
Copy data directly from HDFS	✓	
Export from Amazon S3	✓	✓
Durable local storage		✓
Use in a cluster of devices		✓
Use with AWS Greengrass (IoT)		✓
Transfer files through NFS with a GUI		✓

AWS Snowball Hardware Differences

Following is a table that shows how the devices differ from each other, physically. For information on specifications for the Snowball, see AWS Snowball Specifications (p. 91). For information on specifications for the Snowball Edge, see AWS Snowball Edge Specifications.

Snowball	Snowball Edge
	

Each device has different storage capacities, as follows:

Storage capacity (usable capacity)	Snowball	Snowball Edge
50 TB (42 TB) - US regions only	✓	
80 TB (72 TB)	✓	
100 TB (83 TB)		✓
100 TB Clustered (45 TB per node)		✓

Each device has the following physical interfaces for management purposes:

Physical interface	Snowball	Snowball Edge
E Ink display – used to track shipping information and configure your IP address.	✓	✓
LCD display – used to manage connections and provide some administrative functions.		✓

AWS Snowball Tool Differences

The following outlines the different tools used with the AWS Snowball devices, and how they are used:

Snowball Tools

Snowball client with Snowball

- Must be downloaded from the AWS Snowball Tools Download page and installed on a powerful workstation that you own.
- Can transfer data to or from the Snowball. For more information, see Using the Snowball Client (p. 52).
- Encrypts data on your powerful workstation before the data is transferred to the Snowball.

Amazon S3 Adapter for Snowball with Snowball

- Must be downloaded from the AWS Snowball Tools Download page and installed on a powerful workstation that you own.
- Can transfer data to or from the Snowball. For more information, see Transferring Data with the Amazon S3 Adapter for Snowball (p. 65).
- Encrypts data on your powerful workstation before the data is transferred to the Snowball.

Snowball Edge Tools

Snowball client with Snowball Edge

- Must be downloaded from the AWS Snowball Tools Download page and installed on a computer that you own.
- Must be used to unlock the Snowball Edge or the cluster of Snowball Edge devices. For more information, see Using the Snowball Client.
- Can't be used to transfer data.

Amazon S3 Adapter for Snowball with Snowball Edge

- Is already installed on the Snowball Edge by default. It does not need to be downloaded or installed.
- Can transfer data to or from the Snowball Edge. For more information, see Using the Amazon S3 Adapter.
- Encrypts data on the Snowball Edge while the data is transferred to the device.

File interface with Snowball Edge

- Is already installed on the Snowball Edge by default. It does not need to be downloaded or installed.
- Can transfer data by dragging and dropping files up to 150 GB in size from your computer to the buckets on the Snowball Edge through an easy-to-configure NFS mount point. For more information, see Using the File Interface for the AWS Snowball Edge.
- Encrypts data on the Snowball Edge while the data is transferred to the device.

AWS Greengrass console with Snowball Edge

- With a Snowball Edge, you can use the AWS Greengrass console to update your AWS Greengrass group and the core running on the Snowball Edge.

Differences Between Items Provided for the Snowball and Snowball Edge

The following outlines the differences between the network adapters, cables used, and cables provided for the Snowball and Snowball Edge.

Network Interface	Snowball Support	Snowball Edge Support	Cables Provided with Device
RJ45	✓	✓	Only provided with Snowball
SFP+	✓	✓	Only provided with Snowball
SFP+ (with optic connector)	✓	✓	No cables provided for either device. No optic connector provided for Snowball Edge devices. An optic connector is provided with each Snowball
QSFP		✓	No cables or optics provided

For more information on the network interfaces, cables, and connectors that work with the different device types, see the following topics:

- Supported Network Hardware (p. 91) for Snowballs in this guide.
- Supported Network Hardware in the *AWS Snowball Edge Developer Guide.*

AWS Snowball Other Differences

For other differences, including FAQs and pricing information, see:

- https://aws.amazon.com/snowball
- https://aws.amazon.com/snowball-edge

How AWS Snowball Works with the Standard Snowball Appliance

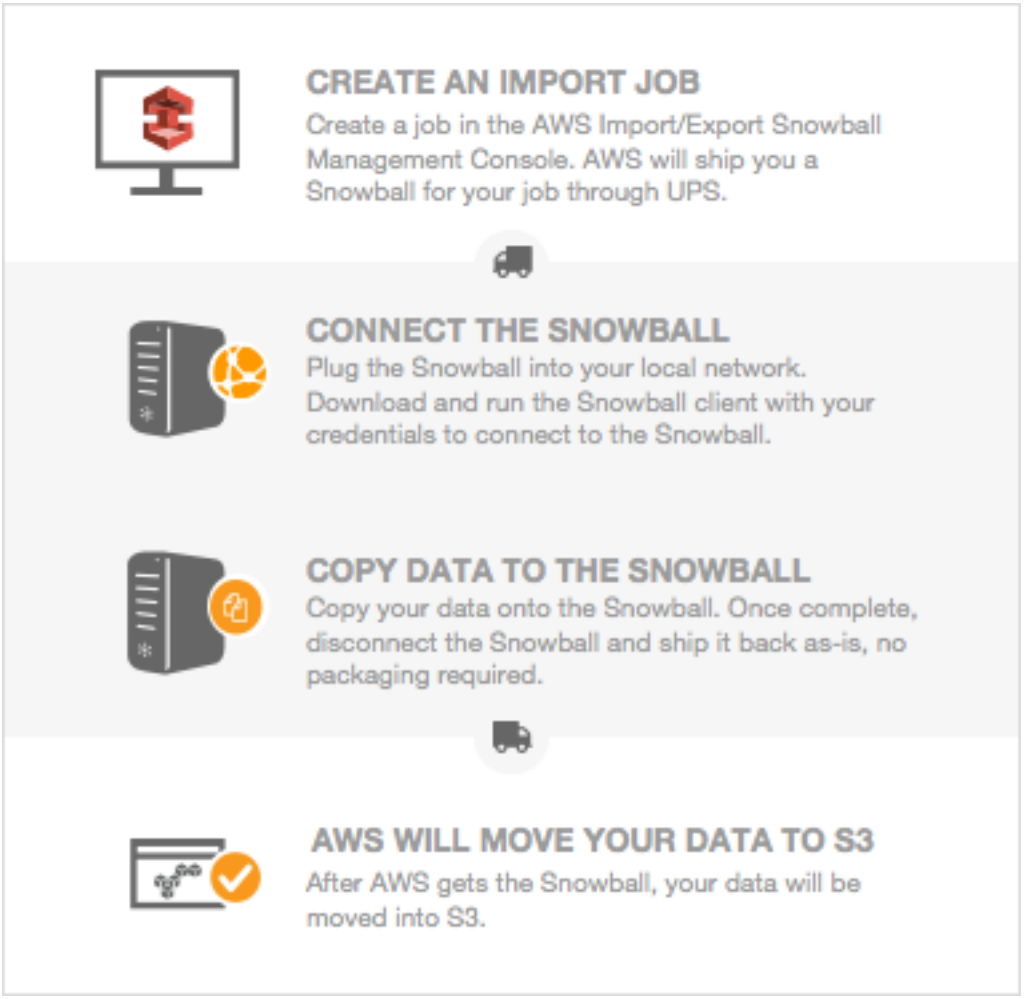
Following, you can find information on how AWS Snowball works, including concepts and its end-to-end implementation.

Topics

- How It Works: Concepts (p. 7)
- How It Works: Implementation (p. 9)

How It Works: Concepts

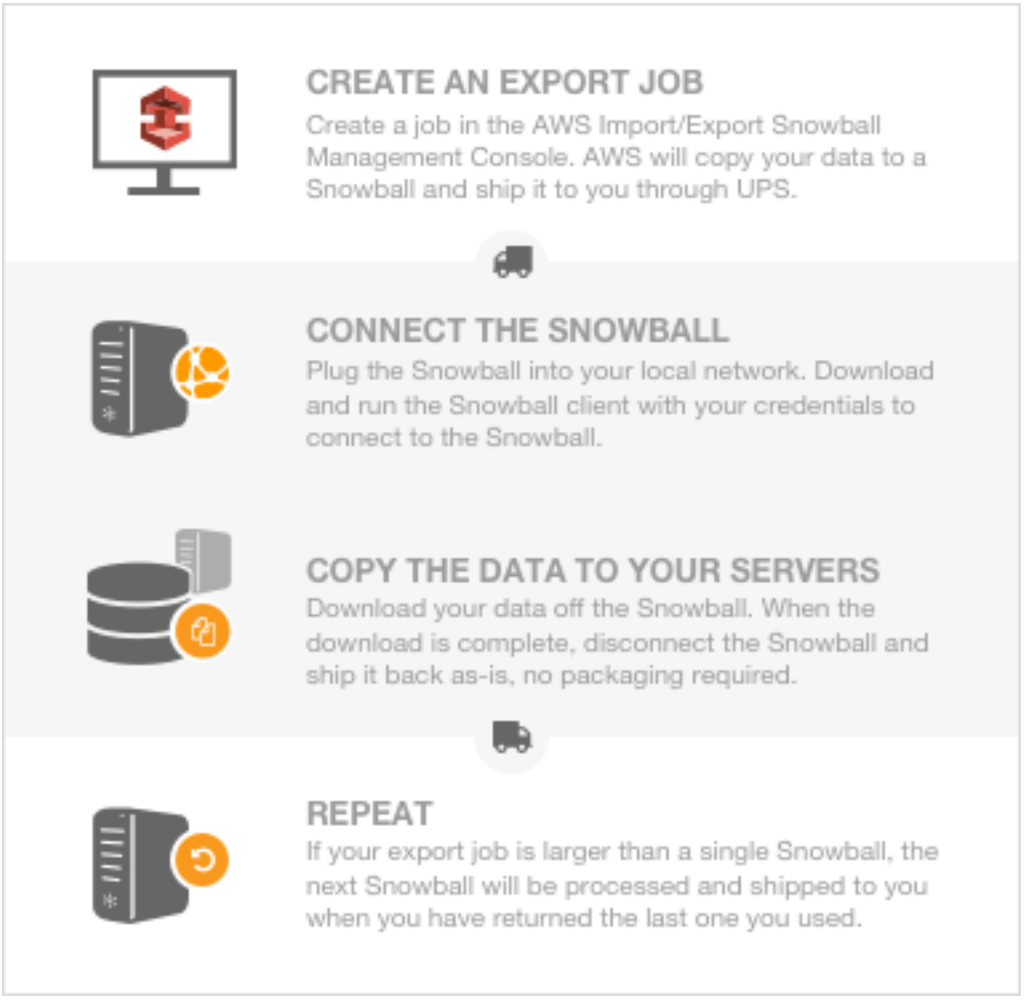
How Import Works



Each import job uses a single Snowball appliance. After you create a job in the AWS Snowball Management Console or the job management API, we ship you a Snowball. When it arrives in a few days, you'll connect the Snowball to your network and transfer the data that you want imported into Amazon S3 onto that Snowball using the Snowball client or the Amazon S3 Adapter for Snowball.

When you're done transferring data, ship the Snowball back to AWS, and we'll import your data into Amazon S3.

How Export Works



Each export job can use any number of Snowball appliances. After you create a job in the AWS Snowball Management Console or the job management API, a listing operation starts in Amazon S3. This listing operation splits your job into parts. Each job part can be up to about 80 TB in size, and each job part has exactly one Snowball associated with it. After your job parts are created, your first job part enters the **Preparing Snowball** status.

Soon after that, we start exporting your data onto a Snowball. Typically, exporting data takes one business day. However, this process can take longer. Once the export is done, AWS gets the Snowball ready for pickup by your region's carrier. When the Snowball arrives at your data center or office in a few days, you'll connect the Snowball to your network and transfer the data that you want exported to your servers by using the Snowball client or the Amazon S3 Adapter for Snowball.

When you're done transferring data, ship the Snowball back to AWS. Once we receive a returned Snowball for your export job part, we perform a complete erasure of the Snowball. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards. This step marks the completion of that particular job part. If there are more job parts, the next job part now is prepared for shipping.

> **Note**
> The listing operation is a function of Amazon S3. You are billed for it as you are for any Amazon S3 operation, even if you cancel your export job.

How It Works: Implementation

The following are overviews of how the Snowball is implemented for importing and exporting data. Both overviews assume that you'll use the AWS Snowball Management Console to create your job and the Snowball client to locally transfer your data. If you'd rather work programmatically, to create jobs you can use the job management API for Snowball. For more information, see AWS Snowball API Reference. To transfer your data programmatically, you can use the Amazon S3 Adapter for Snowball. For more information, see Transferring Data with the Amazon S3 API Adapter for Snowball (p. 65).

End-to-End Import Implementation

1. **Create an import job** – Sign in to the AWS Snowball Management Console and create a job. The status of your job is now **Job created**, and we have queued your job request for processing. If there's a problem with your request, you can cancel your job at this point.
2. **A Snowball is prepared for your job** – We prepare a Snowball for your job, and the status of your job is now **Preparing Snowball**. For security purposes, data transfers must be completed within 90 days of the Snowball being prepared.
3. **A Snowball is shipped to you by your region's carrier** – The carrier takes over from here, and the status of your job is now **In transit to you**. You can find your tracking number and a link to the tracking website on the AWS Snowball Management Console. For information on who your region's carrier is, see Shipping Carriers (p. 74).
4. **Receive the Snowball** – A few days later, your region's carrier delivers the Snowball to the address that you provided when you created the job, and the status of your job changes to **Delivered to you**. When the Snowball arrives, you'll notice that it didn't arrive in a box, because the Snowball is its own shipping container.
5. **Get your credentials and download the Snowball client** – Get ready to start transferring data by getting your credentials, your job manifest, and the manifest's unlock code, and then downloading the Snowball client.
   - The Snowball client is the tool that you'll use to manage the flow of data from your on-premises data source to the Snowball. You can download the Snowball client from the AWS Snowball Tools Download page.
   - The manifest is used to authenticate your access to the Snowball, and it is encrypted so that only the unlock code can decrypt it. You can get the manifest from the AWS Snowball Management Console when the Snowball is on-premises at your location.
   - The unlock code is a 29-character code that also appears when you get your manifest. We recommend that you write it down and keep it separate from the manifest to prevent unauthorized access to the Snowball while it's at your facility. The unlock code is visible when you get your manifest.
6. **Install and set up the Snowball client** – Install the Snowball client on the computer workstation that has your data source mounted on it.
7. **Position the hardware** – Move the Snowball into your data center and open it following the instructions on the case. Connect the Snowball to power and your local network.
8. **Power on the Snowball** – Next, power on the Snowball by pressing the power button above the E Ink display. Wait a few minutes, and the **Ready** screen appears.
9. **Start the Snowball client** – When you start the Snowball client on your workstation, type the IP address of the Snowball, the path to your manifest, and the unlock code. The Snowball client decrypts the manifest and uses it to authenticate your access to the Snowball.
10. **Transfer data** – Use the Snowball client to transfer the data that you want to import into Amazon S3 from your data source into the Snowball.
11. **Prepare the Snowball for its return trip** – After your data transfer is complete, power off the Snowball and unplug its cables. Secure the Snowball's cables into the cable caddie on the inside of the Snowball's back panel and seal the Snowball. Now the Snowball is ready to be returned.

12. **Your region's carrier returns the Snowball to AWS** – While the carrier has the Snowball for shipping, the status for the job becomes **In transit to AWS**.
13. **AWS gets the Snowball** – The Snowball arrives at AWS, and the status for your job becomes **At AWS**. On average, it takes about a day for AWS to begin importing your data into Amazon S3.
14. **AWS imports your data into Amazon S3** – When import starts, your job's status changes to **Importing**. The import can take a few days. At this point, if there are any complications or issues, we contact you through email.

    Once the import is complete, your job status becomes **Completed**, and a PDF file of your job completion report becomes available for download from the AWS Snowball Management Console.
15. **Your imported data now resides in Amazon S3** – With the import complete, the data that you transferred is now in Amazon S3.

Now that you know how an import job works, you're ready to create your first job. For more information, see Importing Data into Amazon S3 with AWS Snowball (p. 16).

For more information about the job management API for Snowball, see AWS Snowball API Reference.

End-to-End Export Implementation

1. **Create an export job** – Sign in to the AWS Snowball Management Console and create a job. This process begins a listing operation in Amazon S3 to determine the amount of data to be transferred, and also any optional ranges for objects within your buckets that your job will transfer. Once the listing is complete, the AWS Snowball Management Console creates all the job parts that you'll need for your export job. At this point, you can cancel your job if you need to.

   Note
   The listing operation is a function of Amazon S3. You are billed for it as you are for any Amazon S3 operation, even if you cancel your export job.
2. **A Snowball is prepared for your job part** – Soon after your job parts are created, your first job part enters the **Preparing Snowball** status. For security purposes, data transfers must be completed within 90 days of the Snowball being prepared. When the Snowball is prepared, the status changes to **Exporting**. Typically, exporting takes one business day; however, this process can take longer. Once the export is done, the job status becomes **Preparing shipment**, and AWS gets the Snowball ready for pickup.
3. **A Snowball is shipped to you by your region's carrier** – The carrier takes over from here, and the status of your job is now **In transit to you**. You can find your tracking number and a link to the tracking website on the AWS Snowball Management Console. For information on who your region's carrier is, see Shipping Carriers (p. 74).
4. **Receive the Snowball** – A few days later, the carrier delivers the Snowball to the address you provided when you created the job, and the status of your first job part changes to **Delivered to you**. When the Snowball arrives, you'll notice that it didn't arrive in a box, because the Snowball is its own shipping container.
5. **Get your credentials and download the Snowball client** – Get ready to start transferring data by getting your credentials, your job manifest, and the manifest's unlock code, and then downloading the Snowball client.
   - The Snowball client is the tool that you'll use to manage the flow of data from the Snowball to your on-premises data destination. You can download the Snowball client from the AWS Snowball Tools Download page.
   - The manifest is used to authenticate your access to the Snowball, and it is encrypted so that only the unlock code can decrypt it. You can get the manifest from the AWS Snowball Management Console when the Snowball is on-premises at your location.
   - The unlock code is a 29-character code that also appears when you get your manifest. We recommend that you write it down and keep it separate from the manifest to prevent unauthorized

access to the Snowball while it's at your facility. The unlock code is visible when you get your manifest.

6. **Install and set up the Snowball client** – Install the Snowball client on the computer workstation that has your data source mounted on it.
7. **Position the hardware** – Move the Snowball into your data center and open it following the instructions on the case. Connect the Snowball to power and your local network.
8. **Power on the Snowball** – Next, power on the Snowball by pressing the power button above the E Ink display. Wait a few minutes, and the **Ready** screen appears.
9. **Start the Snowball client** – When you start the Snowball client on your workstation, type the IP address of the Snowball, the path to your manifest, and the unlock code. The Snowball client decrypts the manifest and uses it to authenticate your access to the Snowball.
10. **Transfer data** – Use the Snowball client to transfer the data that you want to export from the Snowball appliance into your on-premises data destination.
11. **Prepare the Snowball for its return trip** – After your data transfer is complete, power off the Snowball and unplug its cables. Secure the Snowball's cables into the cable caddie on the inside of the Snowball's back panel and seal the Snowball. The Snowball is now ready to be returned.
12. **Your region's carrier returns the Snowball to AWS** – When the carrier has the Snowball, the status for the job becomes **In transit to AWS**. At this point, if your export job has more job parts, the next job part enters the **Preparing Snowball** status.
13. **We erase the Snowball** – Once we receive a returned Snowball we perform a complete erasure of the Snowball. This erasure follows the NIST 800-88 standards.

Now that you know how an export job works, you're ready to create your first job. For more information, see Exporting Data from Amazon S3 with Snowball (p. 24).

Jobs for Standard Snowball Appliances

A job in AWS Snowball (Snowball) is a discrete unit of work, defined when you create it in the console or the job management API. Jobs have types, details, and statuses. Each of those elements is covered in greater detail in the sections that follow.

Topics

Job Types

There are two different job types: import jobs and export jobs. Both of the Snowball job types are summarized following, including the source of the data, how much data can be moved, and the result you can expect at successful job completion. Although these two types of jobs have fundamental differences, they share some common details The source can be local to your data center or office, or it can be an Amazon S3 bucket.

Import into Amazon S3

An *import job* is the transfer of 80 TB or less of your data (located in an on-premises data source), copied onto a single Snowball, and then moved into Amazon S3. For import jobs, Snowballs and jobs have a one-to-one relationship, meaning that each job has exactly one Snowball associated with it. If you need additional Snowballs, you can create new import jobs or clone existing ones.

Your data source for an import job should be on-premises. In other words, the storage devices that hold the data to be transferred should be physically located at the address that you provided when you created the job.

You can import any number of directories, files, and objects for each import job, provided the amount of data you're importing fits within a single Snowball. In the US regions, Snowballs come in two sizes: 50 TB and 80 TB. All other regions have 80 TB Snowballs only.

When you import files, each file becomes an object in Amazon S3 and each directory becomes a prefix. If you import data into an existing bucket, any existing objects with the same names as newly imported objects will be overwritten.

When the import has been processed and verified, AWS performs a complete erasure of the Snowball. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards.

After your import is complete, you can download a job report. This report alerts you to any objects that failed the import process. You can find additional information in the success and failure logs.

Important
Don't delete your local copies of the transferred data until you can verify the results of the job completion report and review your import logs.

Export from Amazon S3

An *export job* is the transfer of any amount of data (located in Amazon S3), copied onto any number of Snowballs, and then moved one Snowball at a time into your on-premises data destination. When you create an export job, it's split into job parts. Each job part is no more than 80 TB in size, and each job part has exactly one Snowball associated with it.

Your data source for an export job is one or more Amazon S3 buckets. Once the data for a job part is moved from Amazon S3 to a Snowball, you can download a job report. This report will alert you to any objects that failed the transfer to the Snowball. You can find more information in your job's success and failure logs.

You can export any number of objects for each export job, using as many Snowballs as it takes to complete the transfer. Snowballs for an export job's job parts are delivered one after another, with subsequent Snowballs shipping out to you once the previous job part has entered the **In transit to AWS** status.

When you copy objects into your on-premises data destination from a Snowball, those objects are saved as files. If you copy objects into a location that already holds files, any existing files with the same names will be overwritten.

When AWS receives a returned Snowball, we perform a complete erasure of the Snowball. This erasure follows the NIST 800-88 standards.

Important
Don't change, update, or delete the exported Amazon S3 objects until you can verify that all of your contents for the entire job have been copied to your on-premises data destination.

When you create an export job, you can choose to export an entire Amazon S3 bucket or a specific range of objects keys. For more information, see Using Export Ranges (p. 40).

Job Details

Each import or export job for Snowball is defined by the details that you specify when it's created. The following list describes all the details of a job.

- **Job name** – A name for the job, containing alphanumeric characters, spaces, and any Unicode special characters.

- **Job type** – The type of job, either import or export.
- **Job ID** – A unique 39-character label that identifies your job. The job ID appears at the bottom of the shipping label that appears on the E Ink display, and in the name of a job's manifest file.
- **Created date** – The date that you created this job.
- **Shipping speed** – Speed options are based on region. For more information, see Shipping Speeds (p. 76).
- **IAM role ARN** – This Amazon Resource Name (ARN) is the AWS Identity and Access Management (IAM) role that is created during job creation with write permissions for your Amazon S3 buckets. The creation process is automatic, and the IAM role that you allow Snowball to assume is only used to copy your data between your Amazon S3 buckets and the Snowball. For more information, see Creating an IAM Role for Snowball (p. 81).
- **AWS KMS key** – In Snowball, AWS Key Management Service (AWS KMS) encrypts the keys on each Snowball. When you create your job, you also choose or create an ARN for an AWS KMS encryption key that you own. For more information, see AWS Key Management Service in Snowball (p. 84).
- **Snowball capacity** – In the US regions, Snowballs come in two sizes: 50 TB and 80 TB. All other regions have the 80 TB Snowballs only.
- **Storage service** – The AWS storage service associated with this job, in this case Amazon S3.
- **Resources** – The AWS storage service resources associated with your job. In this case, these are the Amazon S3 buckets that your data is transferred to or from.

Job Statuses

Each job has a *status*, which changes to denote the current state of the job.

Job Status	Description	Job Type That Status Applies To
Job created	Your job has just been created. This status is the only one during which you can cancel a job or its job parts, if the job is an export job.	Both
Preparing Snowball	AWS is preparing a Snowball for your job.	Both
Exporting	AWS is exporting your data from Amazon S3 onto a Snowball.	Export
Preparing shipment	AWS is preparing to ship a Snowball to you.	Both
In transit to you	The Snowball has been shipped to the address you provided during job creation.	Both
Delivered to you	The Snowball has arrived at the address you provided during job creation.	Both
In transit to AWS	You have shipped the Snowball back to AWS.	Both
At AWS	Your shipment has arrived at AWS. If you're importing data, your import typically begins within a day of its arrival.	Both
Importing	AWS is importing your data into Amazon Simple Storage Service (Amazon S3).	Import
Completed	Your import job or export job part has completed successfully.	Both

Job Status	Description	Job Type That Status Applies To
Canceled	Your job has been canceled. You can only cancel Snowball import jobs during the **Job created** status.	Both

Setting Up Your AWS Access

Before you use AWS Snowball (Snowball) for the first time, you need to complete the following tasks:

1. Sign Up for AWS (p. 14).
2. Create an IAM User (p. 14).

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including AWS Import/Export. You are charged only for the services that you use. For more information about pricing and fees for Snowball, see AWS Snowball Pricing. Snowball is not free to use; for more information on what AWS services are free, see AWS Free Usage Tier.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open https://aws.amazon.com/, and then choose **Create an AWS Account**.

   Note
   This might be unavailable in your browser if you previously signed into the AWS Management Console. In that case, choose **Sign in to a different account**, and then choose **Create a new AWS account**.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM User

Services in AWS, such as AWS Import/Export, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. AWS recommends not using the root credentials of your AWS account to make requests. Instead, create an AWS Identity and Access Management (IAM) user, and grant that user full access. We refer to these users as administrator users.

You can use the administrator user credentials, instead of root credentials of your account, to interact with AWS and perform tasks, such as to create an Amazon S3 bucket, create users, and grant them permissions. For more information, see Root Account Credentials vs. IAM User Credentials in the *AWS General Reference* and IAM Best Practices in *IAM User Guide*.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console.

To create an IAM user for yourself and add the user to an Administrators group

1. Use your AWS account email address and password to sign in as the *AWS account root user* to the IAM console at https://console.aws.amazon.com/iam/.

   Note
   We strongly recommend that you adhere to the best practice of using the **`Administrator`** 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 of the console, choose **Users**, and then choose **Add user**.
3. For **User name**, type **`Administrator`**.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to select a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions for user** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, type **`Administrators`**.
9. For **Filter**, choose **Job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. 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.
12. 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 to restrict users' permissions to specific AWS resources, go to Access Management and Example Policies.

To sign in as this new IAM user, sign out of the AWS Management Console, then use the following URL, where *`your_aws_account_id`* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Type 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*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, choose **Create Account Alias** and type an alias, such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **AWS Account Alias** on the dashboard.

If you're going to create Snowball jobs through an IAM user that is not an administrator user, that user needs certain permissions to use the AWS Snowball Management Console effectively. For more information on those permissions, see Creating an IAM User for Snowball (p. 79).

Next Step

Getting Started with AWS Snowball (p. 16)

Getting Started with AWS Snowball

With AWS Snowball (Snowball), you can transfer hundreds of terabytes or petabytes of data between your on-premises data centers and Amazon Simple Storage Service (Amazon S3). Following, you can find general instructions for creating and completing your first data transfer job. You can find more information on specific components of Snowball later in this documentation. For an overview of the service as a whole, see How AWS Snowball Works with the Standard Snowball Appliance (p. 6).

Both sets of instructions assume that you'll use the AWS Snowball Management Console to create your job and the Snowball client to locally transfer your data. If you'd rather work programmatically, to create jobs you can use the job management API for Snowball. For more information, see AWS Snowball API Reference. To transfer your data programmatically, you can use the Amazon S3 Adapter for Snowball. For more information, see Transferring Data with the Amazon S3 Adapter for Snowball (p. 65).

Sign Up for AWS

If you already have an AWS account, go ahead and skip to the next section: Create an Administrator IAM User (p. 16). Otherwise, see Sign Up for AWS (p. 14).

Create an Administrator IAM User

If you already have an administrator AWS Identity and Access Management (IAM) user account, go ahead and skip to one of the sections listed following. If you don't have an administrator IAM user, we recommend that you create one and not use the root credentials of your AWS account to make requests. To do so, see Create an IAM User (p. 14).

> **Important**
> There is no free tier for Snowball. To avoid unwanted charges and delays, read through the relevant import or export section following before you start creating your jobs.

Next:

- Importing Data into Amazon S3 with AWS Snowball (p. 16)
- Exporting Data from Amazon S3 with Snowball (p. 24)

Importing Data into Amazon S3 with AWS Snowball

The process for importing data into Amazon S3 with Snowball has the following steps.

Topics

Create an Import Job

To create an import job from the console

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console.
2. Choose **Create Job**.
3. Plan your job.

   In this optional step, you determine the number of jobs you need to create to finish transferring all the data you want to import into Amazon S3. The answer you provide helps you better plan for your data transfer.

   Once you've finished this page, choose **Next**.

   > **Note**
   > If you're performing a petabyte scale data transfer, we recommend that you read How to Transfer Petabytes of Data Efficiently (p. 36) before you create your first job.

4. Give shipping details.

   On this page, you provide the shipping address that you want the Snowball for this job delivered to. In some regions you choose your shipping speed as well. For more information, see Shipping Speeds (p. 76).

   Once you've finished this page, choose **Next**.
5. Give job details.

   On this page, specify the details of your job. These details include the name of your import job, the region for your destination Amazon S3 bucket, the specific Amazon S3 bucket to receive your imported data, and the storage size of the Snowball. If you don't already have an Amazon S3 bucket, you can create one on this page. If you create a new Amazon S3 bucket for your destination, note that the Amazon S3 namespace for buckets is shared universally by all AWS users as a feature of the service. Use a bucket name that is specific and clear for your usage.

   Once you've finished this page, choose **Next**.
6. Set security.

   On this page, you specify the following:

   - The Amazon Resource Name (ARN) for the IAM role that Snowball assumes to import your data to your destination S3 bucket when you return the Snowball.
   - The ARN for the AWS Key Management Service (AWS KMS) master key to be used to protect your data within the Snowball. For more information, see Security in AWS Snowball (p. 77).

   Once you've finished this page, choose **Next**.
7. Set notifications.

   On this page, specify the Amazon Simple Notification Service (Amazon SNS) notification options for your job and provide a list of comma-separated email addresses to receive email notifications for this job. You can also choose which job status values trigger these notifications. For more information, see Snowball Notifications (p. 90).

   Once you've finished this page, choose **Next**.
8. Review.

   On the next page, review the information you've provided. To make changes, choose the **Edit** button next to the step to change in the navigation pane, or choose **Back**.

Important
Review this information carefully, because incorrect information can result in unwanted delays.

Once your job is created, you're taken to the job dashboard, where you can view and manage your jobs. The last job you created is selected by default, with its **Job status** pane open.

Note
The **Job created** status is the only status during which you can cancel a job.

For more information on managing jobs from the AWS Snowball Management Console and tracking job status, see Using the AWS Snowball Management Console (p. 40). Jobs can also be created and managed with the job management API. For more information, see the AWS Snowball API Reference.

After you created your first import job, AWS processes the information you provided and prepares a Snowball specifically for your import job into Amazon S3. During the processing stage, if there's an issue with your job, we contact you by email. Otherwise, we ship a Snowball to the address you provided when you created the job. Shipping can take a few days, but you can track the shipping status of the Snowball we prepared for your job. In your job's details, you'll see a link to the tracking webpage with your tracking number provided.

Next: Receive the AWS Snowball Appliance (p. 18)

Receive the AWS Snowball Appliance

When you receive the Snowball appliance, you'll notice that it doesn't come in a box. The Snowball is its own physically rugged shipping container. When the Snowball first arrives, inspect it for damage or obvious tampering. If you notice anything that looks suspicious about the Snowball, don't connect it to your internal network. Instead, contact AWS Support and inform them of the issue so that a new Snowball can be shipped to you.

Important
The Snowball is the property of AWS. Tampering with a Snowball is a violation of the AWS Acceptable Use Policy. For more information, see http://aws.amazon.com/aup/.

Before you connect the Snowball to your network and begin transferring data, it's important to cover a few basic elements of your data transfer.

- **The Snowball** – The following is what the Snowball will look like.

- **Data source** – This device holds the data that you want to transfer from your on-premises data center into Amazon S3. It can be a single device, such as a hard drive or USB stick, or it can be separate sources of data within your data center. The data source or sources must be mounted onto your workstation in order to transfer data from them.
- **Workstation** – This computer hosts your mounted data source. You'll use this workstation to transfer data to the Snowball. We highly recommend that your workstation be a powerful computer, able to meet high demands in terms of processing, memory, and networking. For more information, see Workstation Specifications (p. 93).

Next: Connect the AWS Snowball Appliance to Your Local Network (p. 20)

Connect the AWS Snowball Appliance to Your Local Network

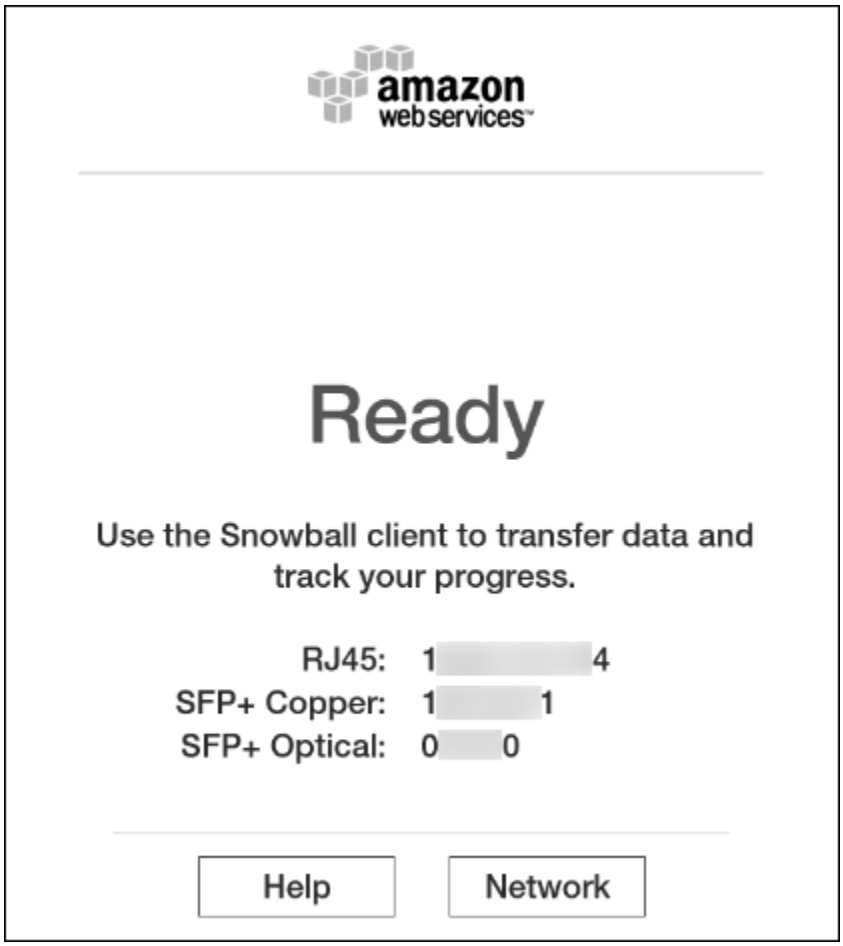
In this step, you'll connect the Snowball to your network. The Snowball appliance has two panels, a front and a back, which are opened by latches and flipped up to rest on the top of the Snowball. Open the front panel first, flip it on top of the Snowball, and then open the back panel, flipping it up to rest on the first. Doing this gives you access to the touch screen on the E Ink display embedded in the front side of the Snowball, and the power and network ports in the back.

Remove the cables from the cable catch, and plug the Snowball into power. Each Snowball has been engineered to support data transfer over RJ45, SFP+ copper, or SFP+ optical 10 gigabit Ethernet. For SFP + optical, you'll have to use your own cable, connected to the SFP+ optical adapter in one of the SFP+ ports. For more information on cables and ports, see Supported Network Hardware (p. 91). Choose a networking option, and plug the Snowball into your network. Power on the Snowball by pressing the power button above the E Ink display.

1. Connect the powered-off Snowball to your network.

   Note
   We recommend that you set up your network connections so that there are as few hops as possible between the data source, the workstation, and the Snowball.

2. Attach the power cable to the back of the Snowball, and then plug it in to a reliable source of power. Then press the power button, located above the E Ink display, and wait for the E Ink display to read **Ready**.
3. When the Snowball is ready, the E Ink display shows the following screen.



At this point, you can change the default network settings through the E Ink display by choosing **Network**. To learn more about specifying network settings for the Snowball, see Changing Your IP Address (p. 49).

Make a note of the IP address shown, because you'll need it to configure the Snowball client.

> **Important**
> To prevent corrupting your data, do not disconnect the Snowball or change its network settings while transferring data.

The Snowball is now connected to your network.

Next: Transfer Data (p. 21)

Transfer Data

The following section discuss the steps involved in transferring data. These steps involve getting your credentials, downloading and installing the Snowball client tool, and then transferring data from your data source into the Snowball using the Snowball client.

> **Note**
> You can also transfer data programmatically with the Amazon S3 Adapter for Snowball. For more information, see Transferring Data with the Amazon S3 Adapter for Snowball (p. 65).

Topics

Get Your Credentials

Each AWS Snowball job has a set of credentials that you must get from the AWS Snowball Management Console or the job management API to authenticate your access to the Snowball. These credentials are an encrypted manifest file and an unlock code. The manifest file contains important information about the job and permissions associated with it. Without it, you won't be able to transfer data. The unlock code is used to decrypt the manifest. Without it, you won't be able to communicate with the Snowball.

> **Note**
> You can only get your credentials after the Snowball appliance has been delivered to you. After the appliance has been returned to AWS, the credentials for your job are no longer available.

To get your credentials by using the console

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console at AWS Snowball Management Console.
2. In the AWS Snowball Management Console, search the table for the specific job to download the job manifest for, and then choose that job.
3. Expand that job's **Job status** pane, and select **View job details**
4. In the details pane that appears, expand **Credentials**. Make a note of the unlock code (including the hyphens), because you'll need to provide all 29 characters to transfer data. Choose **Download manifest** in the dialog box and follow the instructions to download the job manifest file to your computer. The name of your manifest file includes your **Job ID**.

   > **Note**
   > As a best practice, we recommend that you don't save a copy of the unlock code in the same location in the workstation as the manifest for that job. For more information, see Best Practices for AWS Snowball (p. 33).

Now that you have your credentials, you're ready to transfer data.

Install the AWS Snowball Client

The Snowball client is one of the tools that you can use transfer from your on-premises data source to the Snowball. You can download the Snowball client for your operating system from AWS Snowball Tools Download page.

Use the AWS Snowball Client

In this step, you'll run the Snowball client from the workstation first to authenticate your access to the Snowball for this job, and then to transfer data.

To authenticate your access to the Snowball, open a terminal or command prompt window on your workstation and type the following command:

```
snowball start –i [Snowball IP Address] –m [Path/to/manifest/file] –u [29
character unlock code]
```

Following is an example of the command to configure the Snowball client.

```
snowball start -i 192.0.2.0 -m /Downloads/JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-
manifest.bin  -u 12345-abcde-12345-ABCDE-12345
```

In this example, the IP address for the Snowball is 192.0.2.0, the job manifest file that you downloaded is `JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin`, and the 29 character unlock code is `12345-abcde-12345-ABCDE-12345`.

When you've entered the preceding command with the right variables for your job, you get a confirmation message. This message means that you're authorized to access the Snowball for this job.

Now you can begin transferring data onto the Snowball. Similarly to how Linux allows you to copy files and folders with the copy (or `cp`) command, the Snowball client also uses a `cp` command. As in Linux, when you use the copy command you'll provide the values of two paths in your command. One path represents the source location of the data to be copied, and the second path represents the destination where the data will be pasted. When you're transferring data, destination paths to the Snowball must start with the `s3://` root directory identifier.

During data transfer, you'll notice that there is at least one folder at the root level of the Snowball. This folder and any others at this level have the same names as the destination buckets that were chosen when this job was created. Data cannot be transferred directly into the root directory; it must instead go into one of the bucket folders or into their subfolders.

To transfer data using the Snowball client, open a terminal or command prompt window on your workstation and type the following command:

```
snowball cp [options] [path/to/data/source] s3://[path/to/data/destination]
```

Following is an example of the command to copy data using the client to the Snowball.

```
snowball cp --recursive /Logs/April s3://MyBucket/Logs
```

For more information on using the Snowball client tool, see Using the Snowball Client (p. 52). Use the Snowball client commands to finish transferring your data into the Snowball. When you finish, it's time to prepare the Snowball for its return trip.

Stop the AWS Snowball Client, and Power Off the Snowball

When you've finished transferring data on to the Snowball, prepare it for its return trip to AWS. To prepare it, run the `snowball stop` command in the terminal of your workstation. Running this command stops all communication to the Snowball from your workstation and performs local cleanup operations in the background. When that command has finished, power off the Snowball by pressing the power button above the E Ink display.

Disconnect the Appliance

Disconnect the Snowball cables. Secure the Snowball's cables into the cable caddie on the inside of the Snowball back panel and seal the Snowball. When the return shipping label appears on the Snowball's E Ink display, you're ready to drop it off with your region's carrier to be shipped back to AWS. To see who your region's carrier is, see Shipping Carriers (p. 74).

> **Important**
> Don't delete your local copies of the transferred data until the import into Amazon S3 is successful at the end of the process and you can verify the results of the data transfer.

Next:

Return the Appliance (p. 24)

Return the Appliance

The prepaid shipping label on the E Ink display contains the correct address to return the Snowball. For information on how to return your Snowball, see Shipping Carriers (p. 74). The Snowball will be delivered to an AWS sorting facility and forwarded to the AWS data center. The carrier will automatically report back a tracking number for your job to the AWS Snowball Management Console. You can access that tracking number, and also a link to the tracking website, by viewing the job's status details in the console, or by making calls to the job management API.

> **Important**
> Unless personally instructed otherwise by AWS, never affix a separate shipping label to the Snowball. Always use the shipping label that is displayed on the Snowball's E Ink display.

Additionally, you can track the status changes of your job through the AWS Snowball Management Console, by Amazon SNS notifications if you selected that option during job creation, or by making calls to the job management API. For more information on this API, see AWS Snowball API Reference. The final status values include when the Snowball has been received by AWS, when data import begins, and when the import job is completed.

Next: Monitor the Import Status (p. 24)

Monitor the Import Status

You can track the status of your job at any time through the AWS Snowball Management Console or by making calls to the job management API. For more information this API, see AWS Snowball API Reference. Whenever the Snowball is in transit, you can get detailed shipping status information from the tracking website using the tracking number you obtained when your region's carrier received the Snowball.

To monitor the status of your import job in the console, sign in to the AWS Snowball Management Console. Choose the job you want to track from the table, or search for it by your chosen parameters in the search bar above the table. Once you select the job, detailed information appears for that job within the table, including a bar that shows real-time status of your job.

Once your package arrives at AWS and the Snowball is delivered to processing, your job status changes from **In transit to AWS** to **At AWS**. On average, it takes a day for your data import into Amazon S3 to begin. When it does, the status of your job changes to **Importing**. From this point on, it takes an average of two business days for your import to reach **Completed** status.

Now your first data import job into Amazon S3 using Snowball is complete. You can get a report about the data transfer from the console. To access this report from the console, select the job from the table, and expand it to reveal the job's detailed information. Choose **Get report** to download your job completion report as a PDF file. For more information, see Getting Your Job Completion Report and Logs in the Console (p. 42).

Next: Where Do I Go from Here? (p. 32)

Exporting Data from Amazon S3 with Snowball

The AWS Snowball Management Console is where you'll create and manage jobs to export data from Amazon S3. The process for export data from Amazon S3 with Snowball has the following steps.

Topics

- Connect the AWS Snowball Appliance to Your Local Network (p. 28)
- Transfer Data (p. 29)
- Return the Appliance (p. 32)
- Repeat the Process (p. 32)

Create an Export Job

To create an export job from the console

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console at AWS Snowball Management Console.
2. Choose **Create Job**.
3. Plan your job.

   In this step, you'll choose your job type. For an export job, choose **Export**.

   Once you've finished this page, choose **Next**.

   > **Note**
   > If you're performing a petabyte scale data transfer, we recommend that you read How to Transfer Petabytes of Data Efficiently (p. 36) before you create your first job.
4. Give shipping details.

   On the next page, you'll provide the shipping address that you want the Snowball for this job delivered to. In some regions you choose your shipping speed as well. For more information, see Shipping Speeds (p. 76).

   Once you've finished this page, choose **Next**.
5. Give job details.

   On the next page, specify the details of your job. These details include the name of your export job, the region that your source Amazon S3 buckets reside in, the buckets that you want to export data from, and the storage size for the Snowballs that will be used with this job. We recommend that you let AWS decide on the Snowball sizes for each job part, as we will optimize for cost efficiency and speed for each job part. When you create an export job in the AWS Snowball Management Console, you can choose to export an entire Amazon S3 bucket or a specific range of objects and prefixes. For more information, see Using Export Ranges (p. 40).

   > **Important**
   > When selecting what data to export, keep in mind that objects with trailing slashes in their names (`/` or `\`) will not be transferred. Before exporting any objects with trailing slashes, update their names to remove the slash.

   Once you've finished this page, choose **Next**.
6. Set security.

   On the next page, you'll specify the Amazon Resource Name (ARN) for the AWS Identity and Access Management role that Snowball assumes to export your data from your source Amazon S3 buckets, and also the AWS Key Management Service (AWS KMS) master key ARN to be used to protect your data within the Snowball. For more information, see Security in AWS Snowball (p. 77).

   Once you've finished this page, choose **Next**.
7. Set notifications.

   On the next page, specify the Amazon Simple Notification Service (Amazon SNS) notification options for your job and provide a list of comma-separated email addresses to receive email

notifications for this job. You can also choose which job status values trigger these notifications. For more information, see Snowball Notifications (p. 90).

Once you've finished this page, choose **Next**.

8. Review.

   On the next page, review the information you've provided. To make changes, choose the **Edit** button next to the step to change in the navigation pane, or choose **Back**.

   Important
   Review this information carefully, because incorrect information can result in unwanted delays.

Once your job is created, you're taken to the job dashboard, where you can view and manage your jobs. The newest job you created is selected by default, though this a temporary placeholder. When the Amazon S3 listing operation completes in the background, this newest job will be replaced with the number of job parts necessary to complete your job.

Note
At this point, until the job enters the **Preparing Snowball** status, you have the option of canceling the job and its job parts. If you think that you might want to cancel a job, we suggest that you use Amazon SNS notifications to track when the job is created.

For more information on managing jobs from the AWS Snowball Management Console and tracking job status, see Using the AWS Snowball Management Console (p. 40).

Once the Snowball is prepared, the status for your first job part will become **Exporting**. Exporting typically takes one business day; however, this can take longer on occasion.

Once Exporting has completed, the Snowball for your job part enters the **Preparing shipment** status, followed quickly by the **In transit to you** status. Shipping can take a few days, and you can track the shipping status of the Snowball we prepared for your job. In your job's details, you'll see a link to the tracking webpage with your tracking number provided.

Now that your export job is on its way, you can get from the console a report of the data transfer from Amazon S3 to the Snowball, and also success and failure logs. To access the report or the logs, select the job from the table, and expand it to reveal the job's detailed information. Choose **Get report** to download your job report. For more information, see Getting Your Job Completion Report and Logs in the Console (p. 42).

Next: Receive the AWS Snowball Appliance (p. 26)

Receive the AWS Snowball Appliance

When you receive the Snowball appliance, you'll notice that it doesn't come in a box. The Snowball is its own physically rugged shipping container. When the Snowball first arrives, inspect it for damage or obvious tampering. If you notice anything that looks suspicious about the Snowball, don't connect it to your internal network. Instead, contact AWS Support and inform us of the issue so that a new Snowball can be shipped to you.

Important
The Snowball is the property of AWS. Tampering with a Snowball is a violation of the AWS Acceptable Use Policy. For more information, see http://aws.amazon.com/aup/.

Before you connect the Snowball to your network and begin transferring data, it's important to cover a few basic components of Snowball data transfer.

- **The Snowball** – The following is what the Snowball will look like.

- **Data destination** – This on-premises device will hold the data that you want to transfer from the Snowball. It can be a single device, such as a hard drive or USB stick, or it can be separate destinations of data within your data center. The data destination must be mounted onto your workstation in order to transfer data to it.
- **Workstation** – This computer hosts your mounted data destination. You'll use this workstation to receive data from the Snowball. We highly recommend that your workstation be a powerful computer, able to meet high demands in terms of processing, memory, and networking. For more information, see Workstation Specifications (p. 93).

Next: Connect the AWS Snowball Appliance to Your Local Network (p. 28)

Connect the AWS Snowball Appliance to Your Local Network

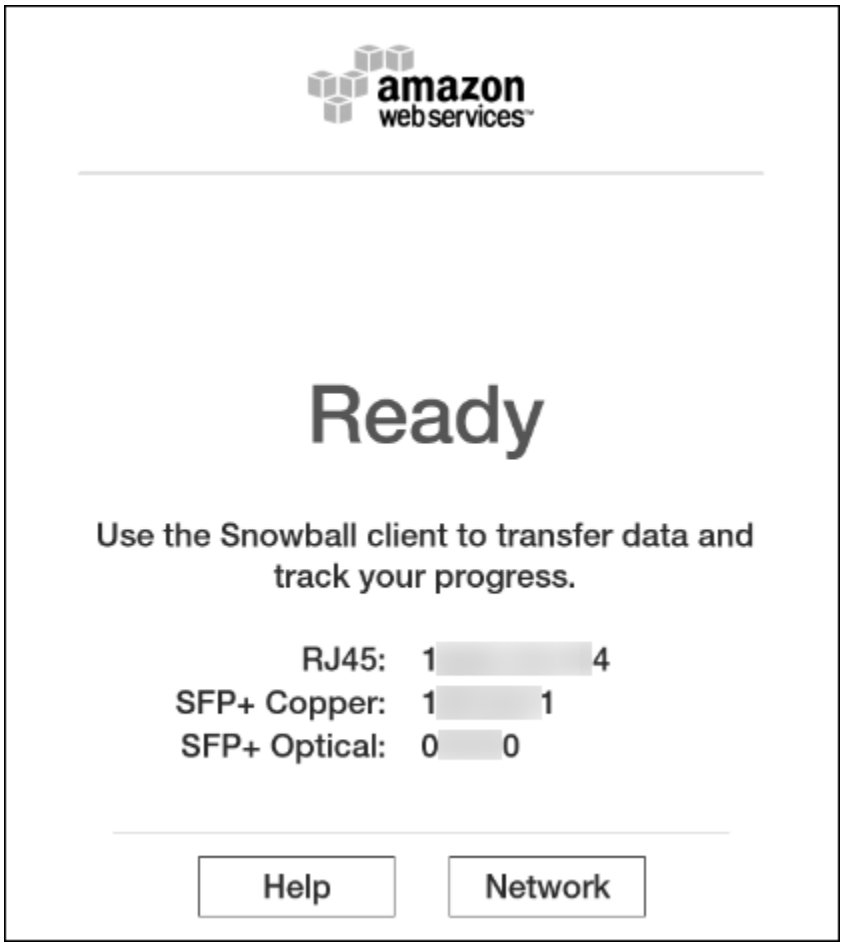
In this step, you'll connect the Snowball to your network. The Snowball appliance has two panels, a front and a back, which are opened by latches and flipped up to rest on the top of the Snowball. Open the front panel first, flip it on top of the Snowball, and then open the back panel, flipping it up to rest on the first. Doing this gives you access to the touch screen on the E Ink display embedded in the front side of the Snowball, and the power and network ports in the back.

Remove the cables from the cable catch, and plug the Snowball into power. Each Snowball has been engineered to support data transfer over RJ45, SFP+ copper, or SFP+ optical 10 gigabit Ethernet. For SFP + optical, you'll have to use your own cable, connected to the SFP+ optical adapter in one of the SFP+ ports. For more information on cables and ports, see Supported Network Hardware (p. 91). Choose a networking option, and plug the Snowball into your network. Power on the Snowball by pressing the power button above the E Ink display.

1. Connect the powered-off Snowball to your network.

   Note
   We recommend that you set up your network connections so that there are as few hops as possible between the data source, the workstation, and the Snowball.

2. Attach the power cable to the back of the Snowball, and then plug it in to a reliable source of power. Then press the power button, located above the E Ink display, and wait for the E Ink display to read **Ready**.
3. When the Snowball is ready, the E Ink display shows the following screen.



At this point, you can change the default network settings through the E Ink display by choosing **Network**. To learn more about specifying network settings for the Snowball, see Changing Your IP Address (p. 49).

Make a note of the IP address shown, because you'll need it to configure the Snowball client.

> **Important**
> To prevent corrupting your data, do not disconnect the Snowball or change its network settings while transferring data.

The Snowball is now connected to your network.

Next: Transfer Data (p. 29)

Transfer Data

Following, you can find information about getting your credentials, downloading and installing the Snowball client tool, and then transferring data from the Snowball to your on-premises data destination using the Snowball client.

Topics

- Use the AWS Snowball Client (p. 31)
- Disconnect the AWS Snowball Appliance (p. 31)

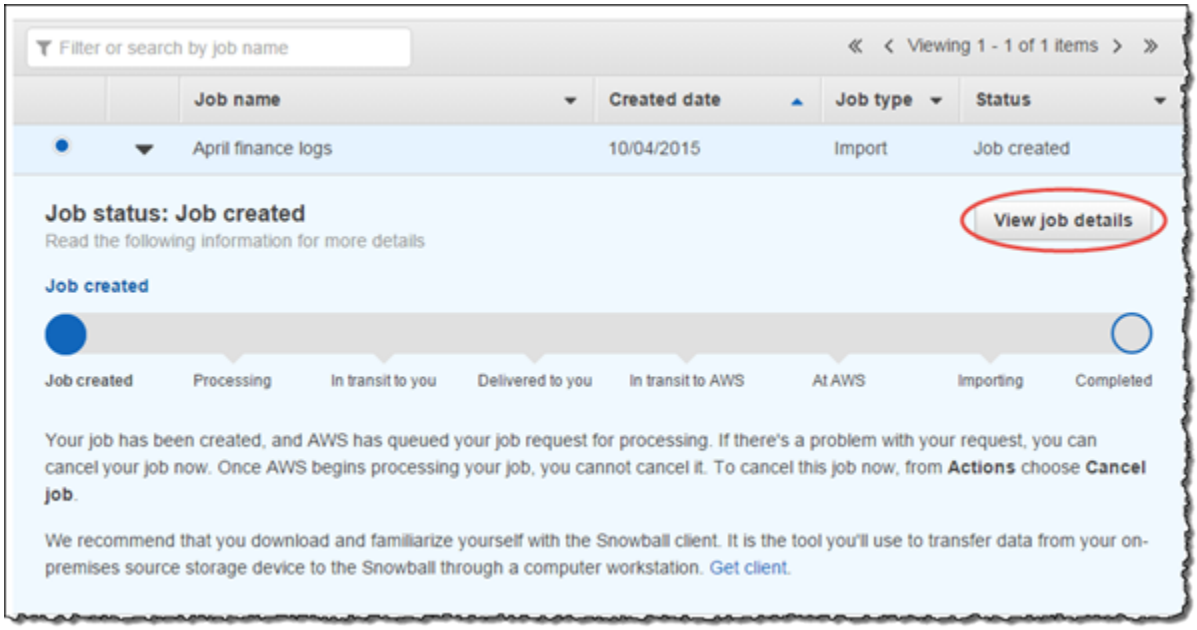
Get Your Credentials

Each AWS Snowball job has a set of credentials that you must get to authenticate your access to the Snowball. These credentials are an encrypted manifest file and an unlock code. The manifest file contains important information about the job and permissions associated with it. Without it, you won't be able to transfer data. The unlock code is used to decrypt the manifest. Without it, the you won't be able to communicate with the Snowball.

> **Note**
> You can only get your credentials after the Snowball appliance has been delivered to you.

To get your credentials from the console

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console.
2. In the AWS Snowball Management Console, search the table for the specific job part to download the job manifest for, and then choose that job.
3. Expand that job part's **Job status** pane, and select **View job details**.



> **Note**
> Each job part has its own unique set of credentials. You won't be able to unlock a Snowball for one job part with the credentials of a different job part, even if both job parts belong to the same export job.

4. In the details pane that appears, expand **Credentials**. Make a note of the unlock code (including the hyphens), because you'll need to provide all 29 characters to run the Snowball client.
5. Choose **Download manifest** in the dialog box, and then follow the instructions to download the job manifest file to your computer. The name of your manifest file includes your job part ID.

> **Note**
> As a best practice, we recommend that you don't save a copy of the unlock code in the same location in the workstation as the manifest for that job. For more information, see Best Practices for AWS Snowball (p. 33).

Now that you have your credentials, you're ready to use the Snowball client to transfer data.

Install the AWS Snowball Client

The Snowball client is one of the tools that you can use to manage the flow of data from your on-premises data source to the Snowball. You can download the Snowball client for your operating system from AWS Snowball Tools Download page.

Use the AWS Snowball Client

In this step, you'll run the Snowball client from the workstation first to authenticate your access to the Snowball for this job, and then to transfer data.

To authenticate your access to the Snowball, open a terminal or command prompt window on your workstation and type the following command:

```
snowball start –i [Snowball IP Address] –m [Path/to/manifest/file] –u [29 character unlock code]
```

Following is an example of the command to configure the Snowball client.

```
snowball start -i 192.0.2.0 -m /Downloads/JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin  -u 12345-abcde-12345-ABCDE-12345
```

In this example, the IP address for the Snowball is 192.0.2.0, the job manifest file that you downloaded is `JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin`, and the 29-character unlock code is `12345-abcde-12345-ABCDE-12345`.

When you've entered the preceding command with the right variables for your job part, you get a confirmation message. This message means that you're authorized to access the Snowball for this job. If you perform the `snowball ls` command, you'll notice that there is at least one folder at the root level of the Snowball. This folder and any others at this level have the same names as the source S3 buckets that were chosen when this job was created.

Now you can begin transferring data from the Snowball. Similarly to how Linux allows you to copy files and folders with the copy (or `cp`) command, the Snowball client also uses a `cp` command. As in Linux, when you use the copy command you'll provide the values of two paths in your command. One path represents the source location of the data to be copied, and the second path represents the destination where the data will be pasted. When you're transferring data, source paths from the Snowball must start with the `s3://` root directory identifier.

Following is an example of the command to copy data using the client from the Snowball

```
snowball cp --recursive s3://MyBucket/Logs /Logs/April
```

Use the Snowball client commands to finish transferring your data from the Snowball. For more information on using the Snowball client, see Using the Snowball Client (p. 52).

Disconnect the AWS Snowball Appliance

When you've finished transferring data from the Snowball, prepare it for its return trip to AWS. First, disconnect the Snowball cables. Secure the Snowball's cables into the cable caddie on the inside of the Snowball back panel, and then seal the Snowball.

When the return shipping label appears on the Snowball's E Ink display, it's ready to be returned.

Next:

Return the Appliance (p. 32)

Return the Appliance

The prepaid shipping label on the E Ink display contains the correct address to return the Snowball. For information on how to return your Snowball, see Shipping Carriers (p. 74). The Snowball will be delivered to an AWS sorting facility and forwarded to the AWS data center. Your region's carrier will automatically report back a tracking number for your job to the AWS Snowball Management Console. You can access that tracking number, and also a link to the tracking website, by viewing the job's status details in the console.

> **Important**
> Unless personally instructed otherwise by AWS, never affix a separate shipping label to the Snowball. Always use the shipping label that is displayed on the Snowball's E Ink display.

When your region's carrier gets the Snowball, the status for the job becomes **In transit to AWS**. At this point, if your export job has more job parts, the next job part enters the **Preparing Snowball** status.

Next: Repeat the Process (p. 32)

Repeat the Process

Once we receive a returned Snowball for your export job part, we perform a complete erasure of the Snowball. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards. This step marks the completion of that particular job part. If there are more job parts, the next job part is being prepared to be shipped out.

You can monitor the status of all your jobs and job parts from the AWS Snowball Management Console.

Next: Where Do I Go from Here? (p. 32)

Where Do I Go from Here?

Now that you've read through the getting started section and begun your first data transfer job, you can learn more about using the Snowball tools and interfaces detail from the following topics:

- Using the AWS Snowball Management Console (p. 40)
- Using an AWS Snowball Appliance (p. 45)
- Transferring Data with a Snowball (p. 51)

We also recommend that you checkout the job management API for AWS Snowball. For more information, see AWS Snowball API Reference

If you're importing data into Amazon S3 for the first time, you might want to learn more about what you can do with your data once it's there. For more information, see the *Amazon S3 Getting Started Guide*.

Best Practices for AWS Snowball

Following, you can find information to help you get the maximum benefit from and satisfaction with AWS Snowball (Snowball).

Security Best Practices for AWS Snowball

Following are approaches that we recommend to promote security when using Snowball:

- If you notice anything that looks suspicious about the Snowball, don't connect it to your internal network. Instead, contact AWS Support, and a new Snowball will be shipped to you.
- We recommend that you don't save a copy of the unlock code in the same location in the workstation as the manifest for that job. Saving the unlock code and manifest separately helps prevent unauthorized parties from gaining access to the Snowball. For example, you might take this approach: First, save a copy of the manifest to the workstation. Then, email the unlock code to the AWS Identity and Access Management (IAM) user to perform the data transfer from the workstation. This approach limits access to the Snowball to individuals who have access to files saved on the workstation and also that IAM user's email address.
- Whenever you transfer data between your on-premises data centers and a Snowball, logs are automatically generated and saved to your workstation. These logs are saved in plaintext format and can contain file name and path information for the files that you transfer. To protect this potentially sensitive information, we strongly suggest that you delete these logs after the job that the logs are associated with enters **Completed** status. For more information about logs, see Snowball Logs (p. 64).

Network Best Practices for AWS Snowball

Following are approaches that we recommend for using Snowball with a network:

- Your workstation should be the local host for your data. For performance reasons, we don't recommend reading files across a network when using Snowball to transfer data. If you must transfer data across a network, batch the local cache before copying to the Snowball so the copy operation can go as fast as possible.
- Because the workstation is considered to be the bottleneck for transferring data, we highly recommend that your workstation be a powerful computer, able to meet high demands in terms of processing, memory, and networking. For more information, see Workstation Specifications (p. 93).
- You can run simultaneous instances of the Snowball client in multiple terminals, each using the copy operation to speed up your data transfer. For more information about using the Snowball client see Commands for the Snowball Client (p. 56).
- To prevent corrupting your data, don't disconnect the Snowball or change its network settings while transferring data.
- Files must be in a static state while being copied. Files that are modified while they are being transferred are not imported into Amazon S3.

Resource Best Practices for AWS Snowball

Following are approaches that we recommend for working with Snowball and your data resources, along with a few additional important points:

- The 10 free days for performing your on-premises data transfer start the day after the Snowball arrives at your data center, and stop when you ship the appliance back out.
- The **Job created** status is the only status in which you can cancel a job. When a job changes to a different status, it can't be canceled.
- For import jobs, don't delete your local copies of the transferred data until the import into Amazon S3 is successful at the end of the process. As part of your process, be sure to verify the results of the data transfer.
- We recommend that you have no more than 500,000 files or directories within each directory.

Performance for AWS Snowball

Following, you can find information about AWS Snowball performance. Here, we discuss performance in general terms, because on-premises environments each have a different way of doing things—different network technologies, different hardware, different operating systems, different procedures, and so on.

The following table outlines how your network's transfer rate impacts how long it takes to fill a Snowball with data. Transferring smaller files without batching them into larger files reduces your transfer speed due to increased overhead.

Rate (MB/s)	42-TB Transfer Time	72-TB Transfer Time
800	14 hours	1 day
450	1.09 days	1.8 days
400	1.16 days	2.03 days
300	1.54 days	2.7 days
277	1.67 days	2.92 days
200	2.31 days	4 days
100	4.63 days	8.10 days
60	8 days	13 days
30	15 days	27 days
10	46 days	81 days

The following describes how to determine when to use Snowball instead of data transfer over the internet, and how to speed up transfer from your data source to the Snowball.

Speeding Up Data Transfer

In general, you can improve the transfer speed from your data source to the Snowball in the following ways, ordered from largest to smallest positive impact on performance:

1. **Use the latest Mac or Linux Snowball client** – The latest Snowball clients for Mac and Linux both support the Advanced Encryption Standard New Instructions (AES-NI) extension to the x86 instruction set architecture. This extension offers improved speeds for encrypting or decrypting data during transfers between the Snowball and your Mac or Linux workstations. For more information on AES-NI, including supported hardware, see AES instruction set on Wikipedia.

2. **Batch small files together** – Each copy operation has some overhead because of encryption. Therefore, performing many transfers on individual files has slower overall performance than transferring the same data in larger files. You can significantly improve your transfer speed for small files by batching them in a single `snowball cp` command. Batching of small files is enabled by default. During the import process into Amazon S3, these batched files are automatically extracted to their original state. For more information, see Options for the snowball cp Command (p. 60).
3. **Perform multiple copy operations at one time** – If your workstation is powerful enough, you can perform multiple `snowball cp` commands at one time. You can do this by running each command from a separate terminal window, in separate instances of the Snowball client, all connected to the same Snowball.
4. **Copy from multiple workstations** – You can connect a single Snowball to multiple workstations. Each workstation can host a separate instance of the Snowball client.
5. **Transfer directories, not files** – Because there is overhead for each `snowball cp` command, we don't recommend that you queue a large number of individual copy commands. Queuing many commands has a significant negative impact on your transfer performance.

   For example, say that you have a directory called C:\\MyFiles that only contains three files, file1.txt, file2.txt, and file3.txt. Suppose that you issue the following three commands.

   ```
   snowball cp C:\\MyFiles\file1.txt s3://mybucket
   snowball cp C:\\MyFiles\file2.txt s3://mybucket
   snowball cp C:\\MyFiles\file3.txt s3://mybucket
   ```

   In this scenario, you have three times as much overhead as if you transferred the entire directory with the following copy command.

   ```
   Snowball cp -r C:\\MyFiles\* s3://mybucket
   ```
6. **Don't perform other operations on files during transfer** – Renaming files during transfer, changing their metadata, or writing data to the files during a copy operation has a significant negative impact on transfer performance. We recommend that your files remain in a static state while you transfer them.
7. **Reduce local network use** – Your Snowball communicates across your local network. Because of this, reducing other local network traffic between the Snowball, the switch it's connected to, and the workstation that hosts your data source can improve data transfer speeds.
8. **Eliminate unnecessary hops** – We recommend that you set up your Snowball, your data source, and your workstation so that they're the only machines communicating across a single switch. Doing so can result in a significant improvement of data transfer speeds.

Experimenting to Get Better Performance

Your performance results will vary based on your hardware, your network, how many and how large your files are, and how they're stored. Therefore, we suggest that you experiment with your performance metrics if you're not getting the performance that you want.

First, attempt multiple copy operations until you see a reduction in overall transfer performance. Performing multiple copy operations at once can have a significantly positive impact on your overall transfer performance. For example, suppose that you have a single `snowball cp` command running in a terminal window, and you note that it's transferring data at 30 MB/second. You open a second terminal window, and run a second `snowball cp` command on another set of files that you want to transfer. You see that both commands are performing at 30 MB/second. In this case, your total transfer performance is 60 MB/second.

Now, suppose that you connect to the Snowball from a separate workstation. You run the Snowball client from that workstation to execute a third `snowball cp` command on another set of files that

you want to transfer. Now when you check the performance, you note that all three instances of the `snowball cp` command are operating at a performance of 25 MB/second, with a total performance of 75 MB/second. Even though the individual performance of each instance has decreased in this example, the overall performance has increased.

Experimenting in this way, using the techniques listed in Speeding Up Data Transfer (p. 34), can help you optimize your data transfer performance.

Performance Considerations for HDFS Data Transfers

When getting ready to transfer data from a Hadoop Distributed File System (HDFS) cluster (version 2.x) into a Snowball, we recommend that you follow the guidance in the previous section, and also the following tips:

- **Don't copy the entire cluster over in a single command** – Transferring an entire cluster in a single command can cause performance issues, including slow transfers, "flipped" bits, and missing or corrupted data on the Snowball. We recommend that in this case you separate the data transfer into multiple parts.
- **Don't transfer a large number of small files** – Suppose that you have a large number of files, say over 1000, and those files are small, say under 1 MB each in size. In this case, transferring them all at once has a negative impact on your performance. This performance degradation is due to per-file overhead when you transfer data from HDFS clusters.

  If you must transfer a large number of small files, we recommend that you find a method of collecting them into larger archive files, and then transferring those. However, these archives are what is imported into Amazon S3. If you want the files in their original state, take them out of the archives after importing the archives.

  Important
  The `--batch` option for the Snowball client's copy command is not supported for HDFS data transfers.

How to Transfer Petabytes of Data Efficiently

When transferring petabytes of data, we recommend that you plan and calibrate your data transfer between the Snowball you have on-site and your workstation according to the following guidelines. Small delays or errors can significantly slow your transfers when you work with large amounts of data.

Topics

Planning Your Large Transfer

To plan your petabyte-scale data transfer, we recommend the following steps:

Step 1: Understand What You're Moving to the Cloud

Before you create your first job for Snowball, you should make sure that you know what data you want to transfer, where it is currently stored, and the destination you want to transfer it to. For data transfers that are a petabyte in scale or larger, doing this administrative housekeeping makes your life much easier when your Snowballs start to arrive.

You can keep this data in a spreadsheet or on a whiteboard—however it works best for you to organize the large amount of content you plan to move to the cloud. If you're migrating data into the cloud for the first time, we recommend that you design a cloud migration model. For more information, see the whitepaper A Practical Guide to Cloud Migration on the AWS Whitepapers website.

When you're done with this step, you know the total amount of data that you're going to move into the cloud.

Step 2: Prepare Your Workstations

When you transfer data to a Snowball, you do so through the Snowball client, which is installed on a physical workstation that hosts the data that you want to transfer. Because the workstation is considered to be the bottleneck for transferring data, we highly recommend that your workstation be a powerful computer, able to meet high demands in terms of processing, memory, and networking.

For large jobs, you might want to use multiple workstations. Make sure that your workstations all meet the suggested specifications to reduce your total transfer time. For more information, see Workstation Specifications (p. 93).

Step 3: Calculate Your Target Transfer Rate

It's important to estimate how quickly you can transfer data to the Snowballs connected to each of your workstations. This estimated speed equals your target transfer rate. This rate is the rate at which you can expect data to move into a Snowball given the realities of your local network architecture.

By reducing the hops between your workstation running the Snowball client and the Snowball, you reduce the time it takes for each transfer. We recommend hosting the data that you want transferred onto the Snowball on the workstation that you transfer the data through.

To calculate your target transfer rate, download the Snowball client and run the `snowball test` command from the workstation that you transfer the data through. If you plan on using more than one Snowball at a time, run this test from each workstation. For more information on running the test, see Testing Your Data Transfer with the Snowball Client (p. 53).

While determining your target transfer speed, keep in mind that it is affected by a number of factors including local network speed, file size, and the speed at which data can be read from your local servers. The Snowball client copies data to the Snowball as fast as conditions allow. It can take as little as a day to fill a 50 TB Snowball depending on your local environment. You can copy twice as much data in the same amount of time by using two 50 TB Snowballs in parallel. Alternatively, you can fill an 80 TB Snowball in two and a half days.

Step 4: Determine How Many Snowballs You Need

Using the total amount of data you're going to move into the cloud, found in step 1, determine how many Snowballs you need to finish your large-scale data migration. Remember that Snowballs come in 50 TB (42 usable) and 80 TB (72 usable) sizes so that you can determine this number effectively. You can move a petabyte of data in as little as 14 Snowballs.

Step 5: Create Your Jobs Using the AWS Snowball Management Console

Now that you know how many Snowballs you need, you can create an import job for each appliance. Because each Snowball import job involves a single Snowball, you create multiple import jobs. For more information, see Create an Import Job (p. 17).

Step 6: Separate Your Data into Transfer Segments

As a best practice for large data transfers involving multiple jobs, we recommend that you separate your data into a number of smaller, manageable data transfer segments. If you separate the data this way, you can transfer each segment one at a time, or multiple segments in parallel. When planning your segments, make sure that all the sizes of the data for each segment combined fit on the Snowball for this job. When segmenting your data transfer, take care not to copy the same files or directories multiple times. Some examples of separating your transfer into segments are as follows:

- You can make 10 segments of 4 TB each in size for a 50 TB Snowball.
- For large files, each file can be an individual segment.
- Each segment can be a different size, and each individual segment can be made of the same kind of data—for example, batched small files in one segment, large files in another segment, and so on. This approach helps you determine your average transfer rate for different types of files.

> **Note**
> Metadata operations are performed for each file transferred. Regardless of a file's size, this overhead remains the same. Therefore, you get faster performance out of batching small files together. For implementation information on batching small files, see Options for the snowball cp Command (p. 60).

Creating these data transfer segments makes it easier for you to quickly resolve any transfer issues, because trying to troubleshoot a large transfer after the transfer has run for a day or more can be complex.

When you've finished planning your petabyte-scale data transfer, we recommend that you transfer a few segments onto the Snowball from your workstation to calibrate your speed and total transfer time.

Calibrating a Large Transfer

You can calibrate a large transfer by running the `snowball cp` command with a representative set of your data transfer segments. In other words, choose a number of the data segments that you defined following last section's guidelines and transfer them to a Snowball. At the same time, make a record of the transfer speed and total transfer time for each operation.

> **Note**
> You can also use the `snowball test` command to perform calibration before receiving a Snowball. For more information about using that command, see Testing Your Data Transfer with the Snowball Client (p. 53).

While the calibration is being performed, monitor the workstation's CPU and memory utilization. If the calibration's results are less than the target transfer rate, you might be able to copy multiple parts of your data transfer in parallel on the same workstation. In this case, repeat the calibration with additional data transfer segments, using two or more instances of the Snowball client connected to the same Snowball. Each running instance of the Snowball client should be transferring a different segment to the Snowball.

Continue adding additional instances of the Snowball client during calibration until you see diminishing returns in the sum of the transfer speed of all Snowball client instances currently transferring data. At

this point, you can end the last active instance of Snowball client and make a note of your new target transfer rate.

Important
Your workstation should be the local host for your data. For performance reasons, we don't recommend reading files across a network when using Snowball to transfer data.

If a workstation's resources are at their limit and you aren't getting your target rate for data transfer onto the Snowball, there's likely another bottleneck within the workstation, such as the CPU or disk bandwidth.

When you complete these steps, you should know how quickly you can transfer data by using one Snowball at a time. If you need to transfer your data faster, see Transferring Data in Parallel (p. 39).

Transferring Data in Parallel

Sometimes the fastest way to transfer data with Snowball is to transfer data in parallel. Parallelization involves one or more of the following scenarios:

- Using multiple instances of the Snowball client on a single workstation with a single Snowball
- Using multiple instances of the Snowball client on multiple workstations with a single Snowball
- Using multiple instances of the Snowball client on multiple workstations with multiple Snowballs

If you use multiple Snowball clients with one workstation and one Snowball, you only need to run the `snowball start` command once, because you run each instance of the Snowball client from the same user account and home directory. The same is true for the second scenario, if you transfer data using a networked file system with the same user across multiple workstations. In any scenario, follow the guidance provided in Planning Your Large Transfer (p. 36).

Using the AWS Snowball Management Console

All jobs for AWS Snowball are created and managed from either the AWS Snowball Management Console or the job management API for AWS Snowball. The following provides an overview of how to use the AWS Snowball Management Console.

Topics

> **Note**
> For information on creating your first job in the console, see Create an Import Job (p. 17) or Create an Export Job (p. 25) in the Getting Started chapter.

Cloning an Import Job in the Console

When you first create an import job, you might discover that you need more than one Snowball. Because each Snowball is associated with a single import job, requiring more than one Snowball means that you need to create more than one job. When creating additional jobs, you can go through the job creation wizard again, or you can clone an existing job.

Cloning a job means recreating it exactly, except for an automatically modified name. Cloning is a simple process.

To clone a job

1. In the AWS Snowball Management Console, choose your job from the table.
2. For **Actions**, choose **Clone job**.
3. The **Create job** wizard opens to the last page, **Step 6: Review**.
4. Review the information and make any changes you want by choosing the appropriate **Edit** button.
5. To create your cloned job, choose **Create job**.

Cloned jobs are named in the format ***Job Name*-clone-*number***. The number is automatically appended to the job name and represents the number of times you've cloned this job after the first time you clone it. For example, **AprilFinanceReports-clone** represents the first cloned job of **AprilFinanceReports** job, and **DataCenterMigration-clone-42** represents the forty-second clone of the **DataCenterMigration** job.

Using Export Ranges

When you create an export job in the AWS Snowball Management Console, you can choose to export an entire Amazon S3 bucket or a specific range of objects keys. Object key names uniquely identify objects in a bucket. If you choose to export a range, you define the length of the range by providing either an inclusive range beginning, an inclusive range ending, or both.

Ranges are UTF-8 binary sorted. UTF-8 binary data is sorted in the following way:

- The numbers 0-9 come before both uppercase and lowercase English characters.
- Uppercase English characters come before all lowercase English characters.
- Lowercase English characters come last when sorted against uppercase English characters and numbers.
- Special characters are sorted among the other character sets.

For more information on the specifics of UTF-8 sort order, see https://en.wikipedia.org/wiki/UTF-8.

Export Range Examples

Assume you have a bucket containing the following objects, sorted in UTF-8 binary order.

- 01
- Aardvark
- Aardwolf
- Aasvogel/apple
- Aasvogel/banana
- Aasvogel/cherry
- Banana
- Car

Specified Range Beginning	Specified Range Ending	Objects in the Range That Will Be Exported
(none)	(none)	All of the objects in your bucket
(none)	Aasvogel	01 Aardvark Aardwolf Aasvogel/apple Aasvogel/banana Aasvogel/cherry
(none)	Aasvogel/banana	01 Aardvark Aardwolf Aasvogel/apple Aasvogel/banana
Aasvogel	(none)	Aasvogel/apple Aasvogel/banana Aasvogel/cherry

Specified Range Beginning	Specified Range Ending	Objects in the Range That Will Be Exported
		Banana Car
Aardwolf	(none)	Aardwolf Aasvogel/apple Aasvogel/banana Aasvogel/cherry Banana Car
Aar	(none)	Aardvark Aardwolf Aasvogel/apple Aasvogel/banana Aasvogel/cherry Banana Car
car	(none)	No objects will be exported, and you'll get an error message when you try to create the job. Note that car is sorted below Car according to UTF-8 binary values.
Aar	Aarrr	Aardvark Aardwolf

Getting Your Job Completion Report and Logs in the Console

Whenever data is imported into or exported out of Amazon S3, you'll get a downloadable PDF job report. For import jobs, this report becomes available at the very end of the import process. For export jobs, your job report typically becomes available for you while the Snowball for your job part is being delivered to you.

The job report provides you insight into the state of your Amazon S3 data transfer. The report includes details about your job or job part for your records. The job report also includes a table that provides a high-level overview of the total number of objects and bytes transferred between the Snowball and Amazon S3.

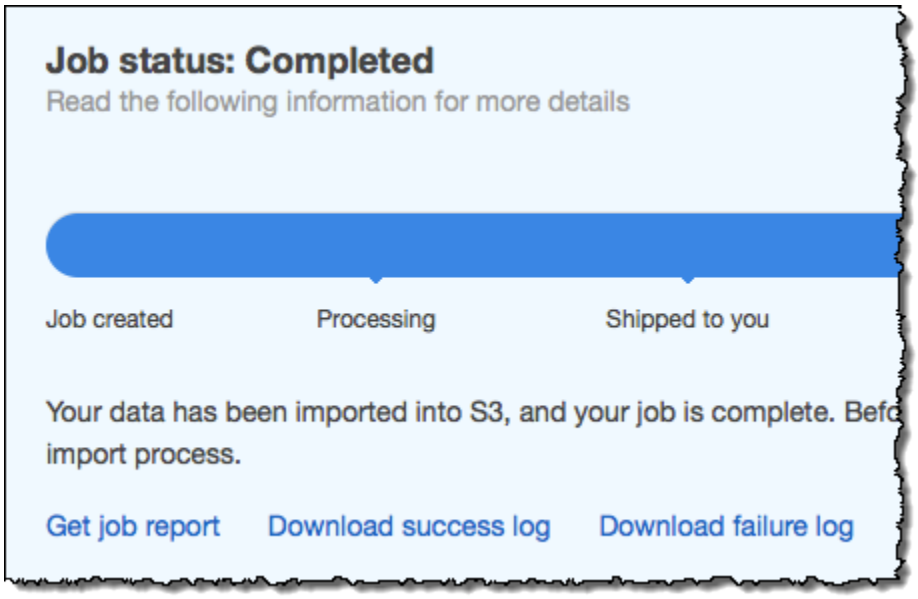
For deeper visibility into the status of your transferred objects, you can look at the two associated logs: a success log and a failure log. The logs are saved in comma-separated value (CSV) format, and the name of each log includes the ID of the job or job part that the log describes.

You can download the report and the logs from the AWS Snowball Management Console.

To get your job report and logs

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console.
2. Select your job or job part from the table and expand the status pane.

   Three options appear for getting your job report and logs: **Get job report**, **Download success log**, and **Download failure log**.



3. Choose your download.

The following list describes the possible values for the report.

- **Completed** – The transfer was completed successfully. You can find more detailed information in the success log.
- **Completed with errors** – Some or all of your data was not transferred. You can find more detailed information in the failure log.

Canceling Jobs in the Console

If you need to cancel a job for any reason, you can do so before it enters the **Preparing Snowball** status. You can only cancel jobs can when they have **Job created** status. Once a job begins processing, you can no longer cancel it.

To cancel a job

1. Sign in to the AWS Management Console and open the AWS Snowball Management Console.
2. Search for and choose your job from the table.
3. From **Actions**, choose **Cancel job**.

You have now canceled your job.

Using an AWS Snowball Appliance

Following, you can find an overview of the Snowball appliance, the physically rugged appliance protected by AWS Key Management Service (AWS KMS) that you use to transfer data between your on-premises data centers and Amazon Simple Storage Service (Amazon S3). This overview includes images of the Snowball, instructions for preparing the appliance for data transfer, and networking best practices to help optimize your data transfer.

For information on transferring data to or from a Snowball, see Transferring Data with a Snowball (p. 51).

When the Snowball first arrives, inspect it for damage or obvious tampering.

> **Warning**
> If you notice anything that looks suspicious about the Snowball, don't connect it to your internal network. Instead, contact AWS Support, and a new Snowball will be shipped to you.

The following is what the Snowball looks like.

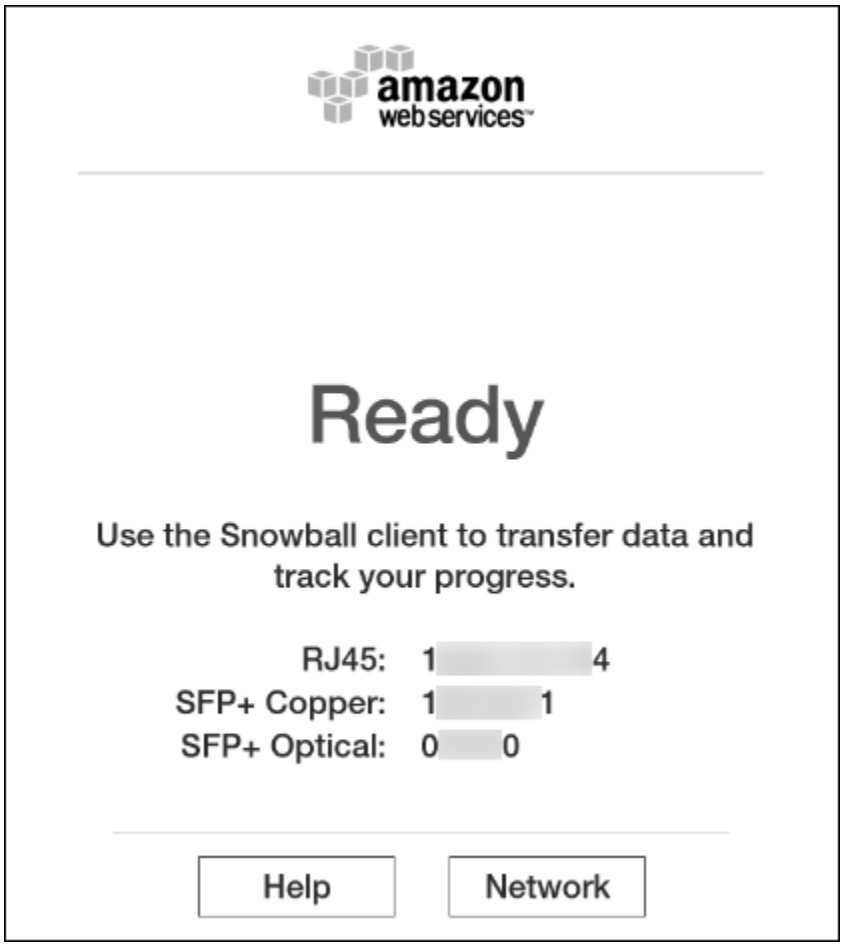
It has two panels, a front and a back, which are opened by latches and flipped up to rest on the top of the Snowball.

Open the front panel first, flip it on top of the Snowball, and then open the back panel second, flipping it up to rest on the first.

Doing this gives you access to the touch screen on the E Ink display embedded in the front side of the Snowball, and the power and network ports in the back.

Remove the cables from the cable catch, and plug the Snowball into power. Each Snowball has been engineered to support data transfer over RJ45, SFP+ copper, or SFP+ optical 10 gigabit Ethernet. Choose a networking option, and plug the Snowball into your network. Power on the Snowball by pressing the power button above the E Ink display.

You'll hear the Snowball internal fans start up, and the display changes from your shipping label to say **Preparing**. Wait a few minutes, and the **Ready** screen appears. When that happens, the Snowball is ready to communicate with the Snowball client data transfer tool and accept your incoming traffic.



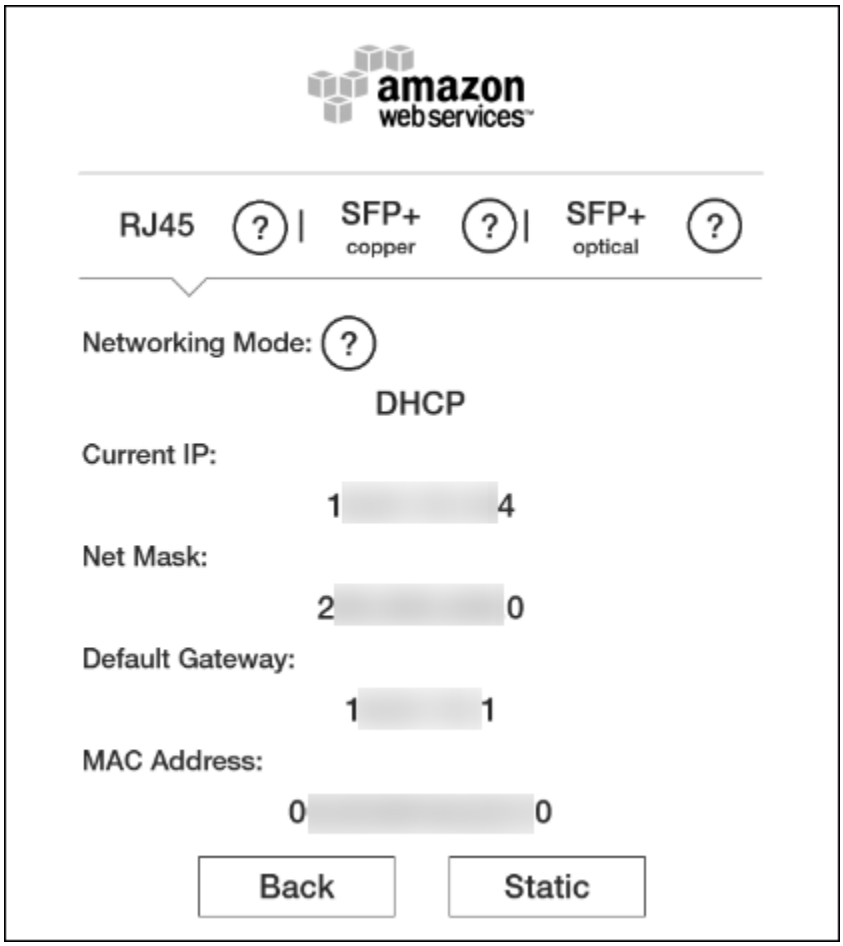
The E Ink display is used to provide you with shipping labels and with the ability to manage the IP address that the Snowball uses to communicate across your local network.

Changing Your IP Address

You can change your IP address to a different static address, which you provide by following this procedure.

To change the IP address of a Snowball

1. On the E Ink display, tap **Network**. The following screen appears, which shows you the current network settings for the Snowball.



2. At the top of the page, either **RJ45**, **SFP+ Copper**, or **SFP+ Optical** has been highlighted. This value represents the type of connection that Snowball is currently using to communicate on your local network. Here, on the active **DHCP** tap, you see your current network settings. To change to a static IP address, tap **Static**.

On this page, you can change the IP address and network mask to your preferred values.

Transferring Data with a Snowball

When locally transferring data between a Snowball and your on-premises data center, you can use the Amazon S3 Adapter for Snowball or the Snowball client:

- Snowball client (p. 52) – A standalone terminal application that you run on your local workstation to do your data transfer. You don't need any coding knowledge to use the Snowball client. It provides all the functionality you need to transfer data, including handling errors and writing logs to your local workstation for troubleshooting and auditing.
- Amazon S3 Adapter for Snowball (p. 65) – A tool that transfers data programmatically using a subset of the Amazon Simple Storage Service (Amazon S3) REST API, including support for AWS Command Line Interface (AWS CLI) commands and the Amazon S3 SDKs.

Note
If you're performing a petabyte-scale data transfer, we recommend that you read How to Transfer Petabytes of Data Efficiently (p. 36) before you create your first job.

Transferring Data with the Snowball Client

The Snowball client is a standalone terminal application that you run on your local workstation to do your data transfer. You don't need any coding knowledge to use the Snowball client. It provides all the functionality you need to transfer data, including handling errors and writing logs to your local workstation for troubleshooting and auditing. For information on best practices to improve your data transfer speeds, see Best Practices for AWS Snowball (p. 33).

You can download and install the Snowball client from the AWS Snowball Tools Download page. Once there, find the installation package for your operating system, and follow the instructions to install the Snowball client. Running the Snowball client from a terminal in your workstation might require using a specific path, depending on your operating system:

- **Linux** – The Snowball must be run from the ~/snowball-client-linux-*build_number*/bin/ directory.
- **Mac** – The **install.sh** script creates symbolic links (symlinks) in addition to copying folders from the Snowball client .tar file to the /usr/local/bin/snowball directory. If you run this script, you can then run the Snowball client from any directory, as long as the /usr/local/bin is a path in your bash_profile. You can verify your path with the `echo $PATH` command.
- **Windows** – Once the client has been installed, you can run it from any directory without any additional preparation.

> **Note**
> We recommend that you use the latest version of the Linux or Mac Snowball client, as they both support the Advanced Encryption Standard New Instructions (AES-NI) extension to the x86 instruction set architecture. This offers improved speeds for encrypting or decrypting data during transfers between the Snowball and your Mac or Linux workstations. For more information on AES-NI, including supported hardware, see AES instruction set on Wikipedia.

Topics

Using the Snowball Client

Following, you can find an overview of the Snowball client, one of the tools that you can use to transfer data between your on-premises data center and the Snowball. The Snowball client supports transferring the following types of data to and from a Snowball.

Sources of data that can be imported with the Snowball client are as follows:

- Files or objects hosted in locally mounted file systems.
- Files or objects from a Hadoop Distributed File System (HDFS) cluster. Currently, only HDFS 2.x clusters are supported.

> **Note**
> Each file or object that is imported must be less than or equal to 5 TB in size.

Because the computer workstation from which or to which you make the data transfer is considered to be the bottleneck for transferring data, we highly recommend that your workstation be a powerful computer. It should be able to meet high demands in terms of processing, memory, and networking. For more information, see Workstation Specifications (p. 93).

Topics

Testing Your Data Transfer with the Snowball Client

You can use the Snowball client to test your data transfer before it begins. Testing is useful because it can help you identify the most efficient method of transferring your data. The first 10 days that the Snowball is on-site at your facility are free, and you'll want to test your data transfer ahead of time to prevent fees starting on the eleventh day.

You can download the Snowball client from the tools page at any time, even before you first log in to the AWS Snowball Management Console. You can also use the Snowball client to test your data transfer job before you create the job, or any time thereafter. You can test the Snowball client without having a manifest, an unlock code, or a Snowball.

To test data transfer using the Snowball client

1. Download and install the Snowball client from the AWS Snowball Tools Download page.
2. Ensure that your workstation can communicate with your data source across the local network. We recommend that you have as few hops as possible between the two.
3. Run the Snowball client's test command and include the path to the mounted data source in your command as follows.

```
snowball test [OPTION...] [path/to/data/source]
```

Example

```
snowball test --recursive --time 5 /Logs/2015/August
```

Example

```
snowball test -r -t 5 /Logs/2015/August
```

In the preceding example, the first command tells the Snowball client to run the test recursively through all the folders and files found under **`/Logs/2015/August`** on the data source for 5 minutes. The second command tells the Snowball client to report real-time transfer speed data for the duration of the test.

Note
The longer the test command runs, the more accurate the test data you get back.

Authenticating the Snowball Client to Transfer Data

Before you can transfer data with your downloaded and installed Snowball client, you must first run the `snowball start` command. This command authenticates your access to the Snowball. For you to run this command, the Snowball you use for your job must be on-site, plugged into power and network, and turned on. In addition, the E Ink display on the Snowball's front must say **Ready**.

To authenticate the Snowball client's access to a Snowball

1. Obtain your manifest and unlock code.

   a. Get the manifest from the AWS Snowball Management Console or the job management API. Your manifest is encrypted so that only the unlock code can decrypt it. The Snowball client compares the decrypted manifest against the information that was put in the Snowball when it was being prepared. This comparison verifies that you have the right Snowball for the data transfer job you're about to begin.

   b. Get the unlock code, a 29-character code that also appears when you download your manifest. We recommend that you write it down and keep it in a separate location from the manifest that you downloaded, to prevent unauthorized access to the Snowball while it's at your facility.

2. Locate the IP address for the Snowball on the Snowball's E Ink display. When the Snowball is connected to your network for the first time, it automatically creates a DHCP IP address. If you want to use a different IP address, you can change it from the E Ink display. For more information, see Using an AWS Snowball Appliance (p. 45).

3. Execute the `snowball start` command to authenticate your access to the Snowball with the Snowball's IP address and your credentials, as follows:

```
snowball start -i [IP Address] -m [Path/to/manifest/file] -u [29 character unlock
              code]
```

   Example

```
snowball start -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234
```

Schemas for Snowball Client

The Snowball client uses schemas to define what kind of data is transferred between your on-premises data center and a Snowball. You declare the schemas whenever you issue a command.

Sources for the Snowball Client Commands

Transferring file data from a local mounted file system requires that you specify the source path, in the format that works for your OS type. For example, in the command `snowball ls C:\\User\Dan \CatPhotos s3://MyBucket/Photos/Cats`, the source schema specifies that the source data is standard file data.

For importing data directly from a Hadoop Distributed File System (HDFS) to a Snowball, you specify the Namenode URI as the source schema, which has the `hdfs://`*`IP Address`*`:`*`port`* format. For example:

```
snowball cp -n hdfs://192.0.2.0:9000/ImportantPhotos/Cats
s3://MyBucket/Photos/Cats
```

Destinations for the Snowball Client

In addition to source schemas, there are also destination schemas. Currently, the only supported destination schema is `s3://`. For example, in the command `snowball cp -r /Logs/April s3:// MyBucket/Logs`, the content in `/Logs/April` is copied recursively to the `MyBucket/Logs` location on the Snowball using the `s3://` schema.

Importing Data from HDFS

You can import data into Amazon S3 from your on-premises Hadoop Distributed File System (HDFS) through a Snowball. You perform this import process by using the Snowball client. Importing from HDFS

is not supported with the Amazon S3 Adapter for Snowball. Following, you can find information about how to prepare for and perform HDFS data transfer.

Although you can write HDFS data to a Snowball, you can't write Hadoop data from a Snowball to your local HDFS. As a result, export jobs are not supported for HDFS.

If you have a large number of small files, say under a megabyte each in size, then transferring them all at once has a negative impact on your performance. This performance degradation is due to per-file overhead when you transfer data from HDFS clusters.

> **Important**
> The `batch` option for the Snowball client's copy command is not supported for HDFS data transfers. If you must transfer a large number of small files from an HDFS cluster, we recommend that you find a method of collecting them into larger archive files, and then transferring those. However, these archives are what is imported into Amazon S3. If you want the files in their original state, take them out of the archives after importing the archives.

Preparing for Transferring Your HDFS Data with the Snowball Client

Before you transfer your HDFS (version 2.x) data, do the following:

- **Confirm the Kerberos authentication settings for your HDFS cluster** – The Snowball client supports Kerberos authentication for communicating with your HDFS in two ways: with the Kerberos login already on the host system and with authentication through specifying a principal and keytab in the `snowball cp` command. The following HDFS/Kerberos encryption types are known to work with Snowball:
  - des3-cbc-sha1-kd
  - aes-128-cts-hmac-sha1-96
  - 256-cts-hmac-sha1-96
  - rc4-hmac (arcfour-hmac)

  Alternatively, you can copy from an unsecured HDFS cluster.
- **Confirm that your workstation has the Hadoop client 2.x version installed on it** – To use the Snowball client, your workstation needs to have the Hadoop client 2.x installed, running, and able to communicate with your HDFS 2.x cluster.
- **Confirm the location of your site-specific configuration files** – If you are using site-specific configuration files, you need to use the `--hdfsconfig` parameter to pass the location of each XML file.
- **Confirm your Namenode URI** – Each HDFS 2.x cluster has a Namenode.core-site.xml file. This file includes a `property` element with the name of `fs.defaultFS` and a value of *`IP Address`*`:`*`port`*, for example `hdfs://192.0.2.0:9000`. You use this value, the Namenode URI, as a part of the source schema when you run Snowball client commands to perform operations on your HDFS cluster. For more information, see Sources for the Snowball Client Commands (p. 54).

  > **Note**
  > Currently, only HDFS 2.X clusters are supported with Snowball. You can still transfer data from an HDFS 1.x cluster by staging the data that you want to transfer on a workstation, and then copying that data to the Snowball with the standard `snowball cp` commands and options.

When you have confirmed the information listed previously, identify the Amazon S3 bucket that you want your HDFS data imported into.

After your preparations for the HDFS import are complete, you can begin. If you haven't created your job yet, see Importing Data into Amazon S3 with AWS Snowball (p. 16) until you reach Use the AWS Snowball Client (p. 22). At that point, return to this topic.

Before Transferring Data from HDFS

Before using the Snowball client to copy HDFS (version 2.x) data, take the following steps:

1. To transfer data from an HDFS cluster, get the latest version of the Snowball client. You can download and install the Snowball client from the AWS Snowball Tools Download page. There you can find the installation package for your operating system. Follow the instructions to install the Snowball client.
2. Ensure that your HDFS cluster is running, and accessible from the workstation that you've installed the Snowball client on.

Transferring Data from HDFS

Now you're ready to transfer data from your HDFS (version 2.x) cluster. For more information on all the Snowball client copy command options, including those specific to HDFS, see Options for the snowball cp Command (p. 60).

If you encounter performance issues while transferring data from your HDFS 2.x cluster to a Snowball, see Performance Considerations for HDFS Data Transfers (p. 36).

After Transferring Data from HDFS

Once you've finished transferring data from your HDFS (version 2.x) cluster, you can validate the data on the Snowball with the following steps:

1. Use the `snowball validate` command to verify the number of uploaded files and confirm that they were uploaded correctly.
2. List all the files at the destination path or paths to confirm that the HDFS file or files were copied. For example, you can use the following command:

```
snowball ls s3://bucket-name/destination-path
```

Commands for the Snowball Client

Following, you can find information on Snowball client commands that help you manage your data transfer into Amazon Simple Storage Service (Amazon S3). You can have multiple instances of the Snowball client in different terminal windows connected to a single Snowball.

Topics

During data transfer, at least one folder appears at the root level of the Snowball. This folder and any others at this level have the same names as the Amazon S3 buckets that you chose when this job was created. You can't write data to the root level of the Snowball. All data must be written into one of the bucket folders or into their subfolders.

You can work with files or folders with spaces in their names, like `my photo.jpg` or `My Documents`. However, make sure that you handle the spaces properly in the client commands. For more information, see the following examples:

- **Linux and Mac version of the client** – `snowball ls s3://mybucket/My\ Folder/my\ photo.jpg`
- **Windows version of the client** – `snowball ls "s3://mybucket/My Documents/my photo.jpg"`

> **Note**
> Before transferring data into Amazon S3 using Snowball, you should make sure that the files and folders that you're going to transfer are named according to the Object Key Naming Guidelines for Amazon S3.

If you're having trouble using the Snowball client, see Troubleshooting for a Standard Snowball (p. 98).

Copy Command for the Snowball Client

The `snowball cp` command copies files and folders between the Snowball and your data source. For details on options for the Snowball copy command (`snowball cp`), see Options for the snowball cp Command (p. 60). In addition to supporting command options, transferring data with the Snowball client supports schemas to define what type of data is being transferred. For more information on schemas, see Schemas for Snowball Client (p. 54).

Usage

```
snowball cp [OPTION...] SRC... s3://DEST
```

Import examples

```
snowball cp --recursive /Logs/April s3://MyBucket/Logs
```

```
snowball cp -r /Logs/April s3://MyBucket/Logs
```

Export examples

```
snowball cp --recursive s3://MyBucket/Logs/ /Logs/April
```

```
snowball cp -r s3://MyBucket/Logs/ /Logs/April
```

For details on options for the Snowball copy command (`snowball cp`), see Options for the snowball cp Command (p. 60).

List Command for the Snowball Client

The `snowball ls` command lists the Snowball contents in the specified path. You can't use this command to list the contents on your workstation, your data source, or other network locations outside of the Snowball.

Usage

```
snowball ls [OPTION...] s3://DEST
```

Example

```
snowball ls s3://MyBucket/Logs/April
```

Make Directory Command for the Snowball Client

The `snowball mkdir` command creates a new subfolder on the Snowball. You can't create a new folder at the root level. The root level is reserved for bucket folders.

Usage

```
snowball mkdir [OPTION...] s3://DEST
```

Example

```
snowball mkdir s3://MyBucket/Logs/April/ExpenseReports
```

Retry Command for the Snowball Client

The `snowball retry` command retries the `snowball cp` command for all the files that didn't copy the last time `snowball cp` was executed. The list of files that weren't copied is saved in a plaintext log in your workstation's temporary directory. The exact path to that log is printed to the terminal if the `snowball cp` command fails to copy a file.

Example Usage

```
snowball retry
```

Remove Command for the Snowball Client

The `snowball rm` command deletes files and folders on the Snowball. This operation can take some time to complete if it removes a large number of files or directories, such as with `snowball rm -r`, which deletes everything on the device. If you run the `snowball ls` command afterwards, it shows you the state of the device when the deletion is completed.

However, the amount of storage reported by the `snowball status` command may show you the amount of storage remaining before the `snowball rm` command was issued. If this happens, try the `snowball status` command in an hour or so to see the new remaining storage value.

Usage

```
snowball rm [OPTION...] s3://DEST
```

Examples

```
snowball rm --recursive s3://MyBucket/Logs/April
```

```
snowball rm -r s3://MyBucket/Logs/April
```

Start Command for the Snowball Client

The `snowball start` command authenticates your access to the Snowball with the Snowball's IP address and your credentials. After you run a `snowball start` command, you can execute any number of `snowball cp` commands.

Usage

```
snowball start -i IP Address -m Path/to/manifest/file -u 29 character unlock code
```

Example

```
snowball start -i 192.0.2.0 -m /user/tmp/manifest -u 01234-abcde-01234-ABCDE-01234
```

Status Command for the Snowball Client

The `snowball status` command returns the status of the Snowball.

Example Usage

```
snowball status
```

Example Output

```
Snowball Status: SUCCESS
S3 Endpoint running at: http://192.0.2.0:8080
Total Size: 72 TB
Free Space: 64 TB
```

Stop Command for the Snowball Client

The `snowball stop` command stops communication from the current instance of the Snowball client to the Snowball.

You can use this command to make sure that all other commands are stopped between the data source server and the Snowball. If you have multiple instances of the client connected to a single Snowball, you use this command for each instance when you're ready to stop transferring data. You can run this command to stop one instance of the client while still copying data with another instance.

Example Usage

```
snowball stop
```

Test Command for the Snowball Client

The `snowball test` command tests your data transfer before it begins. For more information, see Testing Your Data Transfer with the Snowball Client (p. 53).

Example Usage

```
snowball test
```

Validate Command for the Snowball Client

Unless you specify a path, the `snowball validate` command validates all the metadata and transfer statuses for the objects on the Snowball. If you specify a path, then this command validates the content

pointed to by that path and its subdirectories. This command lists files that are currently in the process of being transferred as incomplete for their transfer status.

Doing this for import jobs helps ensure that your content can be imported into AWS without issue.

This command might take some time to complete, and might appear to be stuck from time to time. This effect is common when there are lots of files, and even more so when files are nested within many subfolders. We recommend that you run this command with the `verbose` option.

Example Usage

```
snowball -v validate
```

Version Command for the Snowball Client

The `snowball version` command displays the Snowball client version on the terminal.

Example Usage

```
snowball version
```

Using the Verbose Option

Whenever you execute a Snowball client command, you can use the `verbose` option for additional information. This additional information is printed to the terminal while the command is running.

Using the `verbose` option helps you to better understand what each command is doing. It also helps you troubleshoot issues you might encounter with the Snowball client.

The verbose option is off by default. You can turn it on by specifying the option while running a command, as in the following examples.

```
snowball -v cp /Logs/April/logs1.csv s3://MyBucket/Logs/April/logs1.csv
```

```
snowball --verbose ls s3://MyBucket/Logs/April/
```

Options for the snowball cp Command

Following, you can find information about `snowball cp` command options and also syntax guidelines for using this command. You use this command to transfer data from your workstation to a Snowball.

Command Option	Description
`-b`, `--batch`	String. Significantly improves the transfer performance for small files by batching them into larger `.snowballarchives` files. Batching is on by default. You can change the following defaults to specify when a file is included in a batch: • By default, files that are 1 MB or smaller are included in batches. You can change this setting by specifying the `--batchFileSizeInKBLimit` option with a new maximum file size, in kilobytes. Maximum file sizes range from 100 KB to 1 MB. Files that are larger than the specified maximum file size are transferred to the Snowball as individual files and not included in any batches.

Command Option	Description
	• By default, batches hold up to 10,000 files. This limit can be changed by setting the `--batchNumOfFiles` option. The number of files in a batch can range from 5,000 to 100,000 files. During import into Amazon S3, batches are extracted and the original files are imported into Amazon S3. Only `.snowballarchives` files that were created during the copy command with this option are extracted automatically during import.
`--checksum`	On and set to false by default. Calculates a checksum for any source and destination files with the same name, and then compares the checksums. This command option is used when a copy operation is resumed. Using this option adds computational overhead during your copy operation. **Note** When this option isn't used, a faster comparison of just file names and dates occurs when you resume as copy operation.
`-f`, `--force`	On and set to false by default. This command option has two uses: • When used with a copy command, `-f` overwrites any existing content on the destination that matches the path and name of the content being transferred. • When used after a copy command is run, `-f` overrides the `--resume` command option. Instead, your copy operation is performed from the beginning again, overwriting any existing content on the destination with the same path and name. **Note** The preceding use cases are not mutually exclusive. We recommend that you use `-f` with care to prevent delays in data transfer.
`-h`, `--help`	On and set to false by default. Displays the usage information for the `snowball cp` command in the terminal.
`--nobatch`	String. Disables automatic batching of small files. If you're copying a directory, and you use this option, you must also use the `--recursive` option. This option is hidden. For performance reasons, we don't recommend that you use it unless your use case requires it.
`-r`, `--recursive`	On and set to false by default. Recursively traverses directories during the `snowball cp` command's operation.
`-s`, `--stopOnError`	On and set to false by default. Stops the `snowball cp` command's operation if it encounters an error.

In addition to the previously defined Snowball client copy command options, there are some options specific to transferring data from an HDFS cluster. The following table describes those options. For more information on transferring from an HDFS cluster, see Importing Data from HDFS (p. 54).

Important
The `--batch` option for the Snowball client's copy command is not supported for HDFS data transfers. If you must transfer a large number of small files from an HDFS cluster, we recommend that you find a method of collecting them into larger archive files, and then transferring those. However, these archives are what is imported into Amazon S3. If you want the files in their original state, take them out of the archives after importing the archives.

HDFS-Specific Command Option	Description
`--hdfsconfig`	Used with the `hdfs://` import schema, this option sets the path to a custom XML configuration file on the server running your HDFS cluster. This option must be repeated if you have multiple configuration files. For example, the following specifies two configuration files. `--hdfsconfig src/core/Namenode-site.xml --hdfsconfig / hdfs/corp/conf/hdfs-site.xml`
`-k`	On and set to false by default. Used with the `hdfs://` import schema and the `-p` option, this option sets the path to the keytab file used to authenticate the Snowball client's connection to the HDFS cluster before copying data to a Snowball. **Note** You must have both the principal and the keytab registered with the Kerberos authentication server used to authenticate the HDFS cluster. If you recently ran the `kinit` command on your terminal, then you don't need to specify this option.
`-n`	On and set to false by default. Used with the `hdfs://` import schema, this option copies data from a nonsecure HDFS cluster.
`-p`	On and set to false by default. Used with the `hdfs://` import schema and the `-k` option, this option sets the principal used to authenticate the Snowball client's connection to the HDFS cluster before then copying data to a Snowball. **Note** You must have both the principal and the keytab registered with the Kerberos authentication server used to authenticate the HDFS cluster. If you recently ran the `kinit` command on your terminal, then you don't need to specify this option.

Syntax for the snowball cp Command

Copying data with the Snowball client's `snowball cp` command uses a syntax that is similar to Linux `cp` command syntax. However, there are some notable differences. In the following topics, you can find a reference for the syntax used by the `snowball cp` command. Failure to follow this syntax can lead to unexpected results when copying data to or from a Snowball.

When copying data, define a source path and a destination path, as in the following example.

```
snowball cp [source path] [destination path]
```

When copying a directory, if you also want to copy the contents of the source directory, you use the `-r` option to recursively copy the contents.

Syntax for Copying a File

- **Copying a file to a nonexistent destination with no trailing slash** – Copies the source file to a new file at the destination.

  ```
  snowball cp /tmp/file1 s3://bucket-name/dir1/file2
  ```

  In the preceding example, the source file file1 is copied to the Snowball with the new file name of file2.
- **Copying a file to a nonexistent destination with a trailing slash** – Creates a new directory at the destination, and copies the file into that new directory.

  ```
  snowball cp /tmp/file3 s3://bucket-name/dir2/
  ```

  In the preceding example, the dir2 directory does not exist until this command is executed. Because `dir2/` has a trailing slash in this example, dir2 is created as a directory, and the path to file3 on the Snowball is s3://bucket-name/dir2/file3.
- **Copying a file to an existing destination file** – Fails unless you specify the `-f` option to overwrite the existing destination file.

  ```
  snowball cp -f /tmp/file4 s3://bucket-name/dir3/file5
  ```

  In the preceding example, the destination file file5 already exists before the command was executed. By executing this command with the `-f` option, file5 is overwritten by the contents of file4, with a destination path of s3://bucket-name/dir3/file5.
- **Copying a file to an existing destination directory** – Copies the file into the existing destination directory.

  ```
  snowball cp /tmp/file6 s3://bucket-name/dir4/
  ```

  The preceding example copies file6 into s3://bucket-name/dir4/.

  Note
  If file6 already exists in s3://bucket-name/dir4/ when this command is executed, the command fails. You can force the destination file6 to be overwritten by the source file6 by using the `snowball cp` command with the `-f` option.
- **Copying a file to a bucket on Snowball with or without a trailing slash** – Copies the file into the root level directory on the Snowball that shares the name of an Amazon S3 bucket.

  ```
  snowball cp /tmp/file7 s3://bucket-name
  ```

  The preceding example copies file7 into s3://bucket-name/file7.

  Note
  If file7 already exists in s3://bucket-name when this command is executed, the command fails. You can force the destination file7 to be overwritten by the source file7 by using the `snowball cp` command with the `-f` option.

Syntax for Copying a Directory

- **Copying a directory to a new destination with or without a trailing slash** – Specify the source path and the destination path.

```
snowball cp -r /tmp/dir1 s3://bucket-name/dir2/
```

```
snowball cp -r /tmp/dir1 s3://bucket-name/dir2
```

  The preceding examples both do the same thing. They both create the new directory dir2 and recursively copy the contents of dir1 to it.

- **Copying a directory to a destination directory that already exists** – Only the unique contents from the source directory make it into the destination directory, unless the `snowball cp` command is used with the `-f` option to force the entire destination directory to be overwritten.

```
snowball cp -r /tmp/dir3 s3://bucket-name/dir4/
```

  In the preceding example, only the unique contents from the source directory make it into the destination directory, dir4.

```
snowball cp -r -f /tmp/dir3 s3://bucket-name/dir4/
```

  In the preceding example, the destination directory dir4 is overwritten with the contents in the source dir3 directory.

- **Copying a directory to a destination file that already exists** – This operation fails, unless you use the `snowball cp` command with the `-f` option. In this case, the operation succeeds, because the destination file is overwritten with a copy of the source directory of the same name.

```
snowball cp -r -f /tmp/dir5 s3://bucket-name/dir6
```

  In the preceding example, dir6 on the Snowball is actually a file. Usually this command fails in this case, because the source dir5 is a directory. However, because the `-f` is used, the file dir6 is forcibly overwritten as a directory with the contents from the source dir5.

- **Copying a directory to a bucket on a Snowball** – Specify the bucket name in the destination path.

```
snowball cp -r /tmp/dir7 s3://bucket-name/
```

  Note
  If dir7 already exists in s3://bucket-name when this command is executed, the command copies over the unique content from the source directory into the destination directory. You can force the destination dir7 to be overwritten by the source dir7 by using the `snowball cp` command with the `-f` option.

Snowball Logs

When you transfer data between your on-premises data centers and a Snowball, the Snowball client automatically generates a plaintext log and saves it to your workstation. If you encounter unexpected errors during data transfer to the Snowball, make a copy of the associated log files. Include them along with a brief description of the issues that you encountered in a message to AWS Support.

Logs are saved in the following locations, based on your workstation's operating system:

- **Windows** – C:/Users/*<username>*/.aws/snowball/logs/
- **Mac** – /Users/*<username>*/.aws/snowball/logs/
- **Linux** – /home/*<username>*/.aws/snowball/logs/

Logs are saved with the file name snowball_*<year>*_*<month>*_*<date>*_*<hour>*. The hour is based on local system time for the workstation and uses a 24-hour clock.

Example Log Name

```
snowball_2016_03_28_10.log
```

Each log has a maximum file size of 5 MB. When a log reaches that size, a new file is generated, and the log is continued in the new file. If additional logs start within the same hour as the old log, then the name of the first log is appended with `.1` and the second log is appended with `.2`, and so on.

> **Important**
> Logs are saved in plaintext format and contain file name and path information for the files that you transfer. To protect this potentially sensitive information, we strongly suggest that you delete these logs once the job that the logs are associated with enters the **completed** status.

Transferring Data with the Amazon S3 Adapter for Snowball

The Amazon S3 Adapter for Snowball is a programmatic tool that you can use to transfer data between your on-premises data center and a Snowball. It replaces the functionality of the Snowball client. As with the Snowball client, you need to download the adapter's executable file from the AWS Snowball Tools Download page, install it, and run it from your computer workstation. When programmatically transferring data to a Snowball, all data goes through the Amazon S3 Adapter for Snowball, without exception.

We highly recommend that your workstation be a powerful computer. It should be able to meet high demands in terms of processing, memory, and networking. For more information, see Workstation Specifications (p. 93).

Downloading and Installing the Amazon S3 Adapter for Snowball

You can download and install the Amazon S3 Adapter for Snowball from the AWS Snowball Tools Download page. Once there, find the installation package for your operating system, and follow the instructions to install the Amazon S3 Adapter for Snowball. Running the adapter from a terminal in your workstation might require using a specific path, depending on your operating system.

To install the adapter, first download the snowball-adapter-*operating_system*.zip file from the AWS Snowball Tools Download page. Unzip the file, and navigate the extracted folder. For the Mac and Linux versions of the adapter, to make the snowball-adapter file executable, change the permissions on this file within the bin directory with the following commands.

```
chmod +x snowball-adapter
```

To confirm the adapter was installed successfully

1. Open a terminal window on the workstation with the installed adapter.
2. Navigate to the directory where you installed the snowball-adapter-*operating_system* subdirectory.

3. Navigate to snowball-adapter-*operating_system*/bin.
4. Type the following command to confirm that the adapter was installed correctly: `./snowball-adapter --help`.

   If the adapter was successfully installed, its usage information appears in the terminal.

Installing the adapter also adds the snowball subdirectory to your .aws directory. Within this snowball directory, you can find the logs and config subdirectories. Their contents are as follows:

- The logs directory is where you find the log files for your data transfers to the Snowball through the Amazon S3 Adapter for Snowball.
- The config directory contains two files:
  - The snowball-adapter-logging.config file contains the configuration settings for the log files written to the ~/.aws/snowball/logs directory.
  - The snowball-adapter.config file contains the configuration settings for the adapter itself.

Note
The .aws directory is located at ~/.aws in Linux, OS X, or Unix, or at C:\User*USERNAME*\.aws on Windows.

Using the Amazon S3 Adapter for Snowball

Following, you can find an overview of the Amazon S3 Adapter for Snowball, which allows you to programmatically transfer data between your on-premises data center and the Snowball using Amazon S3 REST API actions. This Amazon S3 REST API support is limited to a subset of actions, meaning that you can use the subset of supported Amazon S3 AWS CLI commands or one of the AWS SDKs to transfer data.

If your solution uses the AWS SDK for Java version 1.11.0 or newer, you must use the following `S3ClientOptions`:

- `disableChunkedEncoding()` – Indicates that chunked encoding is not supported with the adapter.
- `setPathStyleAccess(true)` – Configures the adapter to use path-style access for all requests.

For more information, see Class S3ClientOptions.Builder in the Amazon AppStream SDK for Java.

Topics

Starting the Amazon S3 Adapter for Snowball

To use the Amazon S3 Adapter for Snowball, start it in a terminal on your workstation and leave it running while transferring data.

Note
Because the computer workstation from which or to which you make the data transfer is considered to be the bottleneck for transferring data, we highly recommend that your workstation be a powerful computer. It should be able to meet high demands in

terms of processing, memory, and networking. For more information, see Workstation Specifications (p. 93).

Before you start the adapter, you need the following information:

- **The Snowball's IP address** – Providing the IP address of the Snowball when you start the adapter tells the adapter where to send your transferred data. You can get this IP address from the E Ink display on the Snowball itself.
- **The job's manifest file** – The manifest file contains important information about the job and permissions associated with it. Without it, you won't be able to access the Snowball. It's an encrypted file that you can download after your job enters the `WithCustomer` status. The manifest is decrypted by the unlock code. You can get the manifest file from the console, or programmatically from calling a job management API action.
- **The job's unlock code** – The unlock code a string of 29 characters, including 4 dashes. It's used to decrypt the manifest. You can get the unlock code from the AWS Snowball Management Console (p. 22), or programmatically from the job management API.
- **Your AWS credentials** – Every interaction with the Snowball is signed with the AWS Signature Version 4 algorithm. For more information, see Signature Version 4 Signing Process. When you start the Amazon S3 Adapter for Snowball, you specify the AWS credentials that you want to use to sign this communication. By default, the adapter uses the credentials specified in the *home directory*/.aws/credentials file, under the [default] profile. For more information on how this Signature Version 4 algorithm works locally with the Amazon S3 Adapter for Snowball, see Authorization with the Amazon S3 API Adapter for Snowball (p. 85).

Once you have the preceding information, you're ready to start the adapter on your workstation. The following procedure outlines this process.

To start the adapter

1. Open a terminal window on the workstation with the installed adapter.
2. Navigate to the directory where you installed the snowball-adapter-*operating_system* directory.
3. Navigate to the bin subdirectory.
4. Type the following command to start the adapter: `./snowball-adapter -i` *Snowball IP address* `-m` *path to manifest file* `-u` *29 character unlock code*.

   Note
   If you don't specify any AWS credentials when starting the adapter, the default profile in the *home directory*`/.aws/credentials` file is used.

The Amazon S3 Adapter for Snowball is now started on your workstation. Leave this terminal window open while the adapter runs. If you're going to use the AWS CLI to transfer your data to the Snowball, open another terminal window and run your AWS CLI commands from there.

Getting the Status of a Snowball Using the Adapter

You can get a Snowball's status by initiating a `HEAD` request to the Amazon S3 Adapter for Snowball. You receive the status response in the form of an XML document. The XML document includes storage information, latency information, version numbers, and more.

You can't use the AWS CLI or any of the AWS SDKs to retrieve status in this. However, you can easily test a `HEAD` request over the wire by running a `curl` command on the adapter, as in the following example.

```
curl -H "Authorization Header" -X HEAD http://192.0.2.0:8080
```

Note
When requesting the status of a Snowball, you must add the authorization header. For more information, see Signing AWS Requests with Signature Version 4.

An example of the XML document that this request returns follows.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Status xsi:schemaLocation="http://s3.amazonaws.com/doc/2006-03-01/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
    <snowballIp>192.0.2.0</snowballIp>
    <snowballPort>8080</snowballPort>
    <snowballId>012EXAMPLE01</snowballId>
    <totalSpaceInBytes>77223428091904</totalSpaceInBytes>
    <freeSpaceInBytes>77223428091904</freeSpaceInBytes>
    <jobId>JID850f06EXAMPLE-4EXA-MPLE-2EXAMPLEab00</jobId>
    <snowballServerVersion>1.0.1</snowballServerVersion>
    <snowballServerBuild>2016-08-22.5729552357</snowballServerBuild>
    <snowballAdapterVersion>Version 1.0</snowballAdapterVersion>
    <snowballRoundTripLatencyInMillis>1</snowballRoundTripLatencyInMillis>
</Status>
```

Options for the Amazon S3 Adapter for Snowball

Following, you can find information on Amazon S3 Adapter for Snowball options that help you configure how the adapter communicates with a Snowball.

Note
Before transferring data into Amazon S3 using Snowball, make sure that the files and directories that you're going to transfer are named according to the Object Key Naming Guidelines.

Option	Description	Usage and Example

`-a`
 	
`--aws-profile-name`
	The AWS profile name that you want to use to sign requests to the Snowball. By default, the adapter uses the credentials specified in the *home directory*/.aws/credentials file, under the [default] profile. To specify a different profile, use this option followed by the profile name.	`snowball-adapter -a`
		
		`snowball-adapter -a Lauren`

`-s`
 	
`--aws-secret-key`
	The AWS secret key that you want to use to sign requests to the Snowball. By default, the adapter uses the key present in the default profile specified in the *home directory*/.aws/credentials file, under the [default] profile. To specify a different profile, use this option, followed by a secret key. The `--aws-profile-name` option takes precedence if both options are specified.	`snowball-adapter -s`
		
		`snowball-adapter -s wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

`-h`
 	
`--help`
	Usage information for the adapter.	`snowball-adapter -h`

`-i`
 	
`--ip`
	The Snowball's IP address, which can be found on the Snowball's E Ink display.	`snowball-adapter -i`
		
		`snowball-adapter -i 192.0.2.0`

`-m`	The path to the manifest file for this job. You can get the manifest file from the AWS Snowball	`snowball-adapter -m`

Option	Description	Usage and Example
`--manifest`	Management Console (p. 22), or programmatically from the job management API.	`snowball-adapter -m ~/Downloads/manifest.bin`
`-u` `--unlockcode`	The unlock code for this job. You can get the unlock code from the AWS Snowball Management Console (p. 22), or programmatically from the job management API.	`snowball-adapter -u` `snowball-adapter -u 01234-abcde-01234-ABCDE-01234`
`-ssl` `--ssl-enabled`	A value that specifies whether or not the Secure Socket Layer (SSL) protocol is used for communicating with the adapter. If no additional certification path or private key is provided, then a self-signed certificate and key are generated in the *home directory*/.aws/snowball/config directory.	`snowball-adapter -ssl` `snowball-adapter -ssl`
`-c` `--ssl-certificate-path`	The path to the certificate to use for the SSL protocol when communicating with the adapter.	`snowball-adapter -c` `~/.aws/snowball/myssl/certs`
`-k` `--ssl-private-key-path`	The path to the private key to use for the SSL protocol when communicating with the adapter.	`snowball-adapter -k` `~/.aws/snowball/myssl/keys`

Using the Adapter with Amazon S3 Commands for the AWS CLI

In the following, you can find how to specify the Amazon S3 Adapter for Snowball as the endpoint for applicable AWS CLI commands. You can also find what Amazon S3 AWS CLI commands are supported for transferring data to the Snowball with the adapter.

Note
For information on installing and setting up the AWS CLI, including specifying what regions you want to make AWS CLI calls against, see the AWS Command Line Interface User Guide.

Specifying the Adapter as the AWS CLI Endpoint

When you use the AWS CLI to issue a command to the Snowball, specify that the endpoint is the Amazon S3 Adapter for Snowball, as shown following.

```
aws s3 ls --endpoint http://<IP address for the S3 Adapter>:8080
```

By default, the adapter runs on port 8080. You can specify a different port by changing the contents in the snowball-adapter.config file described in Downloading and Installing the Amazon S3 Adapter for Snowball (p. 65).

Supported AWS CLI Amazon S3 Commands

Following, you can find a description of the subset of AWS CLI commands and options for Amazon S3 that the AWS Snowball Edge appliance supports. If a command or option isn't listed following, it's not

supported. You can declare some unsupported options, like `--sse` or `--storage-class`, along with a command. However, these are ignored and have no impact on how data is imported.

- cp Copies a file or object to or from the Snowball.
  - `--dryrun` (boolean) The operations that would be performed using the specified command are displayed without being run.
  - `--quiet` (boolean) Operations performed by the specified command are not displayed.
  - `--include` (string) Don't exclude files or objects in the command that match the specified pattern. See Use of Exclude and Include Filters in the *AWS CLI Command Reference* for details.
  - `--exclude` (string) Exclude all files or objects from the command that matches the specified pattern.
  - `--follow-symlinks | --no-follow-symlinks` (boolean) Symbolic links (symlinks) are followed only when uploading to S3 from the local file system. Amazon S3 doesn't support symbolic links, so the contents of the link target are uploaded under the name of the link. When neither option is specified, the default is to follow symlinks.
  - `--only-show-errors` (boolean) Only errors and warnings are displayed. All other output is suppressed.
  - `--recursive` (boolean) The command is performed on all files or objects under the specified directory or prefix. Currently, this option is only supported for uploading data to a Snowball.
  - `--storage-class` (string) The type of storage to use for the object. Valid choices are: `STANDARD` | `REDUCED_REDUNDANCY` | `STANDARD_IA`. Defaults to `STANDARD`.
  - `--metadata` (map) A map of metadata to store with the objects in Amazon S3. This map is applied to every object that is part of this request. In a sync, this functionality means that files that aren't changed don't receive the new metadata. When copying between two Amazon S3 locations, the metadata-directive argument defaults to `REPLACE` unless otherwise specified.

    Important
    Syncing from one directory on a Snowball to another directory on the same Snowball is not supported. Syncing from one Snowball to another Snowball is not supported.
- ls Lists objects on the Snowball.
  - `--human-readable` (boolean) File sizes are displayed in human-readable format.
  - `--summarize` (boolean) Summary information is displayed (number of objects, total size).
- rm Deletes an object on the Snowball.
  - `--dryrun` (boolean) The operations that would be performed using the specified command are displayed without being run.
  - `--include` (string) Don't exclude files or objects in the command that match the specified pattern. For details, see Use of Exclude and Include Filters in the *AWS CLI Command Reference*.
  - `--exclude` (string) Exclude all files or objects from the command that matches the specified pattern.
  - `--only-show-errors` (boolean) Only errors and warnings are displayed. All other output is suppressed.
  - `--quiet` (boolean) Operations performed by the specified command are not displayed.

Supported REST API Actions

Following, you can find REST API actions that you can use with the Snowball.

Topics

Supported REST API Actions for Snowball

HEAD Snowball

Description

Currently, there's only one Snowball REST API operation, which can be used to return status information for a specific device. This operation returns the status of a Snowball. This status includes information that can be used by AWS Support for troubleshooting purposes.

You can't use this operation with the AWS SDKs or the AWS CLI. We recommend that you use `curl` or an HTTP client. The request doesn't need to be signed for this operation.

Request

In the below example, the IP address for the Snowball is 192.0.2.0. Replace this value with the IP address of your actual device.

```
curl -X HEAD http://192.0.2.0:8080
```

Response

```
<Status xsi:schemaLocation="http://s3.amazonaws.com/doc/2006-03-01/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
    <snowballIp>127.0.0.1</snowballIp>
    <snowballPort>8080</snowballPort>
    <snowballId>device-id</snowballId>
    <totalSpaceInBytes>499055067136</totalSpaceInBytes>
    <freeSpaceInBytes>108367699968</freeSpaceInBytes>
    <jobId>job-id</jobId>
    <snowballServerVersion>1.0.1</snowballServerVersion>
    <snowballServerBuild>DevBuild</snowballServerBuild>
    <snowballClientVersion>Version 1.0</snowballClientVersion>
    <snowballRoundTripLatencyInMillis>33</snowballRoundTripLatencyInMillis>
</Status>
```

Supported REST API Actions for Amazon S3

Following, you can find the list of Amazon S3 REST API actions that are supported for using the Amazon S3 Adapter for Snowball. The list includes links to information about how the API actions work with Amazon S3. The list also covers any differences in behavior between the Amazon S3 API action and the Snowball counterpart. All responses coming back from a Snowball declare `Server` as `AWSSnowball`, as in the following example.

```
HTTP/1.1 200 OK
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagkKdBX2z7Il/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Fri, 08 2016 21:34:56 GMT
Server: AWSSnowball
```

- GET Bucket (List Objects) version 1 – In this implementation of the GET operation, the following is true:
  - Pagination is not supported.
  - Markers are not supported.
  - Delimiters are not supported.
  - When the list is returned, the list is not sorted.
  - Only version 1 is supported. GET Bucket (List Objects) Version 2 is not supported.

  - The Snowball adapter is not optimized for large list operations. For example, you might have a case with over a million objects per folder where you want to list the objects after you transfer them to the device. In this type of case, we recommend that you order a Snowball Edge for your job instead.
- GET Service
- HEAD Bucket
- HEAD Object
- GET Object – When an object is uploaded to a Snowball using `GET Object`, an entity tag (ETag) is not generated unless the object was uploaded using multipart upload. The ETag is a hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. The ETag might or might not be an MD5 digest of the object data. For more information on ETags, see Common Response Headers in the *Amazon Simple Storage Service API Reference.*
- PUT Object – When an object is uploaded to a Snowball using `PUT Object`, an ETag is not generated unless the object was uploaded using multipart upload.
- DELETE Object
- Initiate Multipart Upload – In this implementation, initiating a multipart upload request for an object already on the Snowball first deletes that object and then copies it in parts to the Snowball.
- List Multipart Uploads
- Upload Part
- Complete Multipart Upload
- Abort Multipart Upload

Note
Any Amazon S3 REST API actions not listed here are not supported. Using any unsupported REST API actions with your Snowball Edge returns an error message saying that the action is not supported.

Shipping Considerations for AWS Snowball

Following, you can find information about how shipping is handled for AWS Snowball, and a list that shows each AWS Region that is supported. The shipping rate you choose for a job applies to both sending and receiving the Snowball or Snowballs used for that job. For information on shipping charges, see AWS Snowball Pricing.

Topics

When you create a job, you specify a shipping address and shipping speed. This shipping speed doesn't indicate how soon you can expect to receive the Snowball from the day you created the job. It only shows the time that the appliance is in transit between AWS and your shipping address. That time doesn't include any time for processing. Processing time depends on factors including job type (exports take longer than imports, typically) and job size (80-TB models take longer than 50-TB models, typically). Also, carriers generally only pick up outgoing Snowballs once a day. Thus, processing before shipping can take a day or more.

> **Note**
> Snowball devices can only be used to import or export data within the AWS Region where the devices were ordered.

Preparing a Snowball for Shipping

The following explains how to prepare a Snowball appliance and ship it back to AWS.

To prepare a Snowball for shipping

1. Make sure that you've finished transferring all the data for this job to or from the Snowball.
2. Power off the Snowball by pressing the power button above the digital display.

   > **Note**
   > If you've powered off and unplugged the Snowball, and your shipping label doesn't appear after about a minute, contact AWS Support.
3. Disconnect and stow the cables the Snowball was sent with. The back panel of the Snowball has a cable caddie that holds the cables safely during the return trip.
4. Close the two panels on the back and front of the Snowball, pressing in until you hear and feel them click.

You don't need to pack the Snowball in a container, because it is its own physically rugged shipping container. The E Ink display on the front of the Snowball changes to your return shipping label when the Snowball is turned off.

Region-Based Shipping Restrictions

Before you create a job, you should sign in to the console from the AWS Region that your Amazon S3 data is housed in. A few shipping restrictions apply, as follows:

- For data transfers in US regions, we don't ship Snowballs outside of the United States.
- We don't ship Snowballs between non-US regions—for example, from EU (Ireland) to EU (Frankfurt), or from Asia Pacific (Mumbai) to Asia Pacific (Sydney).
- For data transfers in Asia Pacific (Sydney), we only ship Snowballs within Australia.
- For data transfers in Asia Pacific (Mumbai), we only ship Snowballs within India.
- For data transfers in Japan, we only ship Snowballs within Japan.
- For data transfers in the EU regions, we only ship Snowballs to EU member countries listed following: Austria, Belgium, Bulgaria, Croatia, Republic of Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, and the UK.
- For data transfers in the Asia Pacific (Singapore) region, we only ship Snowballs to Singapore.

Note
AWS doesn't ship Snowballs to post office boxes.

Shipping an AWS Snowball Appliance

The prepaid shipping label contains the correct address to return the Snowball. For information on how to return your Snowball, see Shipping Carriers (p. 74). The Snowball is delivered to an AWS sorting facility and forwarded to the AWS data center. Package tracking is available through your region's carrier. You can track status changes for your job by using the AWS Snowball Management Console.

Important
Unless personally instructed otherwise by AWS, don't affix a separate shipping label to the Snowball. Always use the shipping label that is displayed on the Snowball digital display.

Shipping Carriers

When you create a job, you provide the address that you want the Snowball shipped to. The carrier that supports your region handles the shipping of Snowballs from AWS to you, and from you back to AWS. Whenever a Snowball is shipped, you get a tracking number. You can find each job's tracking number and a link to the tracking website from the AWS Snowball Management Console's job dashboard, or by using API calls to the job management API. Following is the list of supported carriers for Snowball by region:

- For India, Amazon Logistics is the carrier.
- For Japan, Schenker-Seino Co., Ltd., is the carrier.
- For all other regions, UPS is the carrier.

AWS Snowball Pickups in the EU, US, Canada, and Singapore

In the EU, US, Canada, and Singapore, keep the following information in mind for UPS to pick up a Snowball:

- You arrange for UPS to pick up the Snowball by scheduling a pickup with UPS directly, or take the Snowball to a UPS package drop-off facility to be shipped to AWS. To schedule a pickup with UPS, you need a UPS account.

- The prepaid UPS shipping label on the E Ink display contains the correct address to return the Snowball.
- The Snowball is delivered to an AWS sorting facility and forwarded to the AWS data center. UPS automatically reports back a tracking number for your job.

Important
Unless personally instructed otherwise by AWS, never affix a separate shipping label to the Snowball. Always use the shipping label that is displayed on the Snowball's E Ink display.

AWS Snowball Pickups in Brazil

In Brazil, keep the following information in mind for UPS to pick up a Snowball:

- When you're ready to return a Snowball, call 0800-770-9035 to schedule a pickup with UPS.
- Snowball is available domestically within Brazil, which includes 26 states and the Distrito Federal.
- If you have a Cadastro Nacional de Pessoa Juridica (CNPJ) tax ID, be sure that you know this ID before you create your job.
- You should issue the appropriate document to return the Snowball device. Confirm with your tax department which of the documents following is required in your state, according to your ICMS registration:
  - **Within São Paulo –** A non-ICMS declaration and an Electronic Tax Invoice (NF-e) are usually required.
  - **Outside São Paulo** – The following are usually required:
    - A non-ICMS declaration
    - A nota fiscal avulsa
    - An Electronic Tax Invoice (NF-e)

Note
For non-ICMS taxpayer declaration, we recommend that you generate four copies of your declaration: one for your records, the other three for transport.

AWS Snowball Pickups in Australia

In Australia, if you're shipping a Snowball back to AWS, send an email to snowball-pickup@amazon.com with **Snowball Pickup Request** in the subject line so we can schedule the pickup for you. In the body of the email, include the following information:

- **Job ID** – The job ID associated with the Snowball that you want returned to AWS.
- **AWS Account ID** – The ID for the AWS account that created the job.
- **Postcode** – The postcode for the address where we originally shipped the Snowball to you.
- **Earliest Pickup Time** (your local time) – The earliest time of day that you want the Snowball picked up.
- **Latest Pickup Time** (your local time) – The latest time of day that you want the Snowball picked up.
- **Email Address** – The email address that you want the pickup request confirmation sent to.
- **Special Instructions** (optional) – Any special instructions for picking up the Snowball.

Soon, you get a follow-up email from UPS to the email address you specified with more information about your pending pickup, scheduled for the soonest available date.

AWS Snowball Pickups in India

In India, Amazon Logistics picks up the Snowball device at the end of your 10 free days.

Important
When using a Snowball in India, remember to file all relevant tax paperwork with your state.

If you need the Snowball for longer than 10 days, send an email to snowballpickupindia@amazon.com with **Snowball Pickup Request** in the subject line so we can reschedule the pickup for you. In the body of the email, include the following information:

- **Job ID** – The job ID associated with the Snowball that you want returned to AWS.
- **Shipment ID** – The shipment ID can be found on the E Ink display, directly above the barcode.
- **Package ID** – The Package ID can be found on the E Ink display, directly below the barcode.
- **AWS Account ID** – The ID for the AWS account that created the job.
- **Pincode** – The pincode for the address where we originally shipped the Snowball to you.
- **Requested Pick-up Date** (your local time) – The day that you want the Snowball picked up.
- **Email Address** – The email address that you want the pickup request confirmation sent to.
- **Special Instructions** (optional) – Any special instructions for picking up the Snowball.

The pickup is automatically updated within 24 hours of your request, because there is no email confirmation.

AWS Snowball Pickups in Japan

In Japan, Schenker-Seino handles your pickups. When you are ready to return your device, you can schedule a pickup on the Schenker-Seino booking website: https://track.seino.co.jp/CallCenterPlusOpen/PickupOpen.do. Keep the following in mind when returning a device:

- You arrange for Schenker-Seino to pick up the Snowball by scheduling a pickup with them directly.
- Find the self-adhesive paper return-shipping label in the pouch attached to the device and apply it over the existing paper shipping label on the side of the device. Don't apply the paper label on the doors, inside the doors, on the bottom of the device, or on the E Ink display.
- The Snowball is delivered to an AWS sorting facility and forwarded to the AWS data center. Schenker-Seino automatically reports back a tracking number for your job.

Shipping Speeds

Each country has different shipping speeds available. These shipping speeds are based on the country in which you're shipping a Snowball. Shipping speeds are as follows:

- **Australia** – When shipping within Australia, you have access to express shipping. Typically, Snowballs shipped express are delivered in about a day.
- **Brazil** – When shipping within Brazil, you have access to UPS Domestic Express Saver shipping, which delivers within two business days during commercial hours. Shipping speeds might be affected by interstate border delays.
- **European Union (EU)** – When shipping to any of the countries within the EU, you have access to express shipping. Typically, Snowballs shipped express are delivered in about a day. In addition, most countries in the EU have access to standard shipping, which typically takes less than a week, one way.
- **India** – When shipping within India, Snowballs are sent out within 7 working days of AWS receiving all related tax documents.
- **Japan** – When shipping within Japan, you have access to the standard shipping speed.
- **United States of America (US) and Canada** – When shipping within the US or Canada, you have access to one-day shipping and two-day shipping.
- **Singapore** – When shipping within Singapore, you have access to Domestic Express Saver shipping.

Security in AWS Snowball

Following, you can find information on security considerations for working with AWS Snowball. Security is a significant concern when transporting information of any level of classification, and Snowball has been designed with this concern in mind.

Topics

Encryption in AWS Snowball

When you're using a standard Snowball to import data into S3, all data transferred to a Snowball has two layers of encryption:

1. A layer of encryption is applied in the memory of your local workstation. This layer is applied whether you're using the Amazon S3 Adapter for Snowball or the Snowball client. This encryption uses AES GCM 256-bit keys, and the keys are cycled for every 60 GB of data transferred.
2. SSL encryption is a second layer of encryption for all data going onto or off of a standard Snowball.

AWS Snowball uses server side-encryption (SSE) to protect data at rest.

Server-Side Encryption in AWS Snowball

AWS Snowball supports server-side encryption with Amazon S3–managed encryption keys (SSE-S3). Server-side encryption is about protecting data at rest, and SSE-S3 has strong, multifactor encryption to protect your data at rest in Amazon S3. For more information on SSE-S3, see Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3) in the *Amazon Simple Storage Service Developer Guide*.

Currently, Snowball doesn't support server-side encryption with AWS KMS–managed keys (SSE-KMS) or server-side encryption with customer-provided keys (SSE-C). However, you might want to use either of these SSE types to protect data that has been imported. Or you might already use one of those two SSE types and want to export. In these cases, keep the following in mind:

- **Import** – If you want to use SSE-KMS or SSE-C to encrypt the objects that you've imported into S3, copy those objects into another bucket that has SSE-KMS encryption established as a part of that bucket's bucket policy.
- **Export** – If you want to export objects that are encrypted with SSE-KMS or SSE-C, first copy those objects to another bucket that either has no server-side encryption, or has SSE-S3 specified in that bucket's bucket policy.

Enabling SSE-S3 for Data Imported into Amazon S3 from a Snowball

Use the following procedure in the Amazon S3 Management Console to enable SSE-S3 for data being imported into Amazon S3. No configuration is necessary in the AWS Snowball Management Console or on the Snowball device itself.

To enable SSE-S3 encryption for the data that you're importing into Amazon S3, simply update the bucket policies for all the buckets that you're importing data into. You update the policies to deny upload object (`s3:PutObject`) permission if the upload request doesn't include the `x-amz-server-side-encryption` header.

To enable SSE-S3 for data imported into Amazon S3

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose the bucket that you're importing data into from the list of buckets.
3. Choose **Permissions**.
4. Choose **Bucket Policy**.
5. In **Bucket policy editor**, enter the following policy. Replace all the instances of *YourBucket* in this policy with the actual name of your bucket.

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [
    {
      "Sid": "DenyIncorrectEncryptionHeader",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::YourBucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    },
    {
      "Sid": "DenyUnEncryptedObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::YourBucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption": "true"
        }
      }
    }
  ]
}
```

6. Choose **Save**.

You've finished configuring your Amazon S3 bucket. When your data is imported into this bucket, it is protected by SSE-S3. Repeat this procedure for any other buckets, as necessary.

Authorization and Access Control in AWS Snowball

You must have valid credentials to create Snowball jobs. You use these credentials to authenticate your access. A requester with valid credentials must also have permissions from the resource owner to access resources from the resource owner. For example, you can use the AWS Identity and Access Management (IAM) service to create users in your account. IAM users have valid credentials to make requests, but by default they don't have permissions to access any resources. Following, you can find information on how to authenticate requests and manage permissions to access Snowball resources.

Note
The following contains information specific to the AWS Snowball Management Console and Snowball client. If you're planning on programmatically creating jobs and transferring data, see AWS Snowball API Reference.

Authentication

Every Snowball job must be authenticated. You do this by creating and managing the IAM users in your account. Using IAM, you can create and manage users and permissions in AWS.

Snowball users must have certain IAM-related permissions to access the AWS Snowball Management Console to create jobs. An IAM user that creates an import or export job must also have access to the right Amazon Simple Storage Service (Amazon S3) resources, such as the Amazon S3 buckets to be used for the job.

To use AWS Snowball Management Console, the IAM user must meet the following conditions:

- The IAM account must be able to do the following:
  - List all of your Amazon S3 buckets and create new ones as needed.
  - Create Amazon Simple Notification Service (Amazon SNS) topics.
  - Select AWS Key Management Service (AWS KMS) keys.
  - Create IAM role Amazon Resource Names (ARNs).

  For more information on granting a user access to an Amazon S3 bucket, see Creating an IAM User for Snowball (p. 79).
- An IAM role must be created with write permissions for your Amazon S3 buckets. The role must also have a trust relationship with Snowball, so AWS can write the data in the Snowball to your designated Amazon S3 buckets. The job creation wizard for each job does this step automatically; you can also do it manually. For more information, see Creating an IAM Role for Snowball (p. 81).

Creating an IAM User for Snowball

If the account doing the work in the Snowball console is not the root account or administrator, you must use the IAM Management Console to grant the user the permissions necessary to create and manage jobs. The following procedure shows how to create a new IAM user for this purpose and give that user the necessary permissions through an inline policy.

If you are updating an existing IAM user, start with step 6.

To create a new IAM user for Snowball

1. Sign in to the AWS Management Console and open the IAM Management Console at https://console.aws.amazon.com/iam.
2. From the navigation pane, choose **Users**.
3. Choose **Create New Users**.

4. Type a name for the user, and choose **Create**.
5. On the screen that appears, you can download security credentials for the IAM user that you just created. Creating an IAM user generates an access key consisting of an access key ID and a secret access key, which are used to sign programmatic requests that you make to AWS. If you want to download these security credentials, choose **Download**. Otherwise, choose **close** to return to your list of users.

   Note
   After this access step, your secret key is no longer available through the AWS Management Console; you have the only copy. To protect your account, keep this information confidential and never email it. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.
6. To view the user details page, choose your user from the table.
7. Choose the **Permissions** tab, and then expand **Inline Policy.**
8. Choose the **click here** link to create a new inline policy.
9. Choose **Custom Policy**, and then choose **Select** to provide your own policy document.
10. Create a name you'll remember for your custom policy, like *JobCreation*.
11. Paste the following text into your custom **Policy Document**.

    Note
    If you're updating an existing user, review the following text carefully before you change their permissions, as you might inadvertently grant or disable permissions that you didn't intend to change.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                        "s3:ListBucket",
                        "s3:GetBucketPolicy",
                        "s3:GetBucketLocation",
                        "s3:ListBucketMultipartUploads",
                        "s3:ListAllMyBuckets",
                        "s3:CreateBucket"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                        "kms:DescribeKey",
                        "kms:ListAliases",
                        "kms:ListKeys",
                        "kms:CreateGrant"
            ],
            "Resource": [
                "*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
                "iam:AttachRolePolicy",
                "iam:CreatePolicy",
                "iam:CreateRole",
                "iam:ListRoles",
```

```
                "iam:ListRolePolicies",
                "iam:PutRolePolicy",
                "iam:PassRole"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                        "sns:CreateTopic",
                        "sns:GetTopicAttributes",
                        "sns:ListSubscriptionsByTopic",
                        "sns:ListTopics",
                        "sns:Subscribe",
                        "sns:SetTopicAttributes"
            ],
            "Resource": [
                "*"
            ]
         },
         {
            "Effect": "Allow",
            "Action": [
                "snowball:*",
                "importexport:*"
            ],
            "Resource": "*"
        }
    ]
}
```

12. Choose **Apply Policy** to finalize your new inline policy and return to the IAM **Users** page in the console.

The preceding procedure creates a user that can create and manage jobs in the Snowball console.

Creating an IAM Role for Snowball

An IAM role must be created with read and write permissions for your Amazon S3 buckets. The role must also have a trust relationship with Snowball, so AWS can write the data in the Snowball and in your Amazon S3 buckets, depending on whether you're importing or exporting data. Creating this role is done as a step in the job creation wizard for each job.

When creating a job in the AWS Snowball Management Console, creating the necessary IAM role occurs in step 4 in the **Permission** section. This process is automatic, and the IAM role that you allow Snowball to assume is only used to write your data to your bucket when the Snowball with your transferred data arrives at AWS. However, if you want to create an IAM role specifically for this purpose, the following procedure outlines that process.

To create the IAM role for your import job

1. On the AWS Snowball Management Console, choose **Create job**.
2. In the first step, fill out the details for your import job into Amazon S3, and then choose **Next**.
3. In the second step, under **Permission**, choose **Create/Select IAM Role**.
4. The IAM Management Console opens, showing the IAM role that AWS uses to copy objects into your specified Amazon S3 buckets.

   Once you've reviewed the details on this page, choose **Allow**.

5. You return to the AWS Snowball Management Console, where **Selected IAM role ARN** contains the Amazon Resource Name (ARN) for the IAM role that you just created.
6. Choose **Next** to finish creating your IAM role.

The preceding procedure creates an IAM role that has write permissions for the Amazon S3 buckets that you plan to import your data into The IAM role that is created has one of the following structures, depending on whether it's for an import or export job.

IAM Role ARN for an Import Job

```
        {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketPolicy",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

IAM Role ARN for an Export Job

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Access Control

As IAM resource owner, you have responsibility for access control and security for the Snowball console, Snowball, and other assets associated with using Snowball. These assets include such things as Amazon S3 buckets, the workstation that the data transfer goes through, and your on-premises data itself.

In some cases, we can help you grant and manage access control to the resources used in transferring your data with Snowball. In other cases, we suggest that you follow industry-wide best practices for access control.

Resource	Description	How to Control Access
AWS Snowball Management Console	The AWS Snowball Management Console is where you create and manage your data transfers between your on-premises data centers and Amazon S3 using discrete units of work called *jobs*. To access the console, see AWS Snowball Management Console.	You can control access to this resource by creating or managing your IAM users. For more information, see Creating an IAM User for Snowball (p. 79).
Amazon S3 buckets	All data in Amazon S3 is stored in units called *objects*. Objects are stored in containers called *buckets*. Any data that goes into Amazon S3 must be stored in a bucket.	To import data into an Amazon S3 bucket, the IAM user that created the import job must have read and write access to your Amazon S3 buckets. For more information on granting a user access to an Amazon S3 bucket, see How Amazon S3 Authorizes a Request for a Bucket Operation and Example 1: Bucket Owner Granting Its Users Bucket Permissions in the *Amazon Simple Storage Service Developer Guide*.
Snowball	A Snowball is a storage appliance that is physically rugged, protected by AWS Key Management Service (AWS KMS), and owned by Amazon. In the AWS Snowball service, all data transfers between Amazon S3 and your on-premises data center is done through a Snowball. You can only access a Snowball through the Snowball client, the data transfer tool. For you to access a Snowball, it must be connected to a physical workstation that has the Snowball client installed on it in your on-premises data center. With the Snowball client, you can access the Snowball by providing the job manifest and unlock code in the command that the Snowball client uses to start communication with the Snowball.	You can control access to the Snowball by careful distribution of a job's manifest and unlock code.
Manifest	The manifest is an encrypted file that you can download from the AWS Snowball Management Console after your job enters the **Processing** status. The manifest is decrypted by the unlock code, when you pass both values to the Snowball through the Snowball client when the client is started for the first time.	As a best practice, we recommend that you don't save a copy of the unlock code in the same location as the manifest for that job. Saving these separately helps prevent unauthorized parties from gaining access to the Snowball associated with that job. For example, you might save a copy of the manifest to the workstation, and email the code

Resource	Description	How to Control Access
		to the IAM user to perform the data transfer from the workstation. This approach limits those who can access the Snowball to individuals who have access to files saved on the workstation and also that IAM user's email address.
Unlock code	The unlock code is a 29-character code with 25 alphanumeric characters and 4 hyphens. This code decrypts the manifest when it is passed along with the manifest to the Snowball through the Snowball client when the client is started for the first time. You can see the unlock code in the AWS Snowball Management Console after your job enters the **Preparing Snowball** status. The code also appears in the dialog box when you download the manifest for a job. The unlock code appears on-screen only and is not downloaded.	Again, as a best practice we recommend that you don't save a copy of the unlock code in the same location as the manifest for that job. Saving these separately helps prevent unauthorized parties from gaining access to the Snowball associated with that job. For example, you might save a copy of the manifest to the workstation, and email the code to the IAM user to perform the data transfer from the workstation. This approach limits those who can access the Snowball to individuals who have access to files saved on the workstation and also that IAM user's email address.

AWS Key Management Service in Snowball

AWS Key Management Service (AWS KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS KMS uses hardware security modules (HSMs) to protect the security of your keys. Specifically, the Amazon Resource Name (ARN) for the AWS KMS key that you choose for a job in AWS Snowball is associated with a KMS key. That KMS key is used to encrypt the unlock code for your job. The unlock code is used to decrypt the top layer of encryption on your manifest file. The encryption keys stored within the manifest file are used to encrypt and de-encrypt the data on the device.

In Snowball, you can choose an existing KMS key. Specifying the ARN for an AWS KMS key tells Snowball which AWS KMS master key to use to encrypt the unique keys on the Snowball.

Your data is encrypted in the local memory of your workstation before it is transferred to the Snowball. The Snowball never contains any discoverable keys.

In Amazon S3, there is a server-side-encryption option that uses AWS KMS–managed keys (SSE-KMS). SSE-KMS is not supported with AWS Snowball. For more information on supported SSE in AWS Snowball, see Server-Side Encryption in AWS Snowball (p. 77).

Using the AWS-Managed Customer Master Key for Snowball

If you'd like to use the AWS-managed customer master key (CMK) for Snowball created for your account, use the following procedure.

To select the AWS KMS CMK for your job

1. On the AWS Snowball Management Console, choose **Create job**.
2. Choose your job type, and then choose **Next**.
3. Provide your shipping details, and then choose **Next**.
4. Fill in your job's details, and then choose **Next**.
5. Set your security options. Under **Encryption**, for **KMS key** either choose the AWS-managed CMK or a custom CMK that was previously created in AWS KMS, or choose **Enter a key ARN** if you need to enter a key that is owned by a separate account.

   Note
   The AWS KMS key ARN is a globally unique identifier for the AWS KMS CMK.
6. Choose **Next** to finish selecting your AWS KMS CMK.

Creating a Custom KMS Envelope Encryption Key

You have the option of using your own custom AWS KMS envelope encryption key with AWS Snowball. If you choose to create your own key, that key must be created in the same region that your job was created in.

To create your own AWS KMS key for a job, see Creating Keys in the *AWS Key Management Service Developer Guide*.

Authorization with the Amazon S3 API Adapter for Snowball

When you use the Amazon S3 Adapter for Snowball, every interaction is signed with the AWS Signature Version 4 algorithm by default. This authorization is used only to verify the data traveling from its source to the adapter. All encryption and decryption happens in your workstation's memory. Unencrypted data is never stored on the workstation or the Snowball.

When using the adapter, keep the following in mind:

- **You can disable signing** – After you've installed the adapter on your workstation, you can disable signing by modifying the snowball-adapter.config file. This file, saved to /.aws/snowball/config, has a value named `auth.enabled` set to true by default. If you change this value to `false`, you disable signing through the Signature Version 4 algorithm. You might not want to disable signing, because signing is used to prevent modifications or changes to data traveling between the adapter and your data storage. You can also enable HTTPS and provide your own certificate when communicating with the adapter. To do so, you start the adapter with additional options. For more information, see Options for the Amazon S3 Adapter for Snowball (p. 68).

  Note
  Data traveling to or from a Snowball is always encrypted, regardless of your signing solution.

- **The encryption key is not changed by what AWS credentials you use** – Because signing with the Signature Version 4 algorithm is only used to verify the data traveling from its source to the adapter, it never factors into the encryption keys used to encrypt your data on the Snowball.
- **You can use any AWS profile** – The Amazon S3 Adapter for Snowball never connects back to AWS to verify your AWS credentials, so you can use any AWS profile with the adapter to sign the data traveling between the workstation and the adapter.
- **The Snowball doesn't contain any AWS credentials** – You manage access control and authorization to a Snowball on-premises. No software on the Snowball or adapter differentiates access between one user and another. When someone has access to a Snowball, the manifest, and the unlock code, that person has complete and total access to the appliance and all data on it. We recommend that you plan physical and network access for the Snowball accordingly.

Other Security Considerations for Snowball

Following are some security points that we recommend you consider when using Snowball, and also some high-level information on other security precautions that we take when a Snowball arrives at AWS for processing.

We recommend the following security approaches:

- When the Snowball first arrives, inspect it for damage or obvious tampering. If you notice anything that looks suspicious about the Snowball, don't connect it to your internal network. Instead, contact AWS Support, and a new Snowball will be shipped to you.
- You should make an effort to protect your job credentials from disclosure. Any individual who has access to a job's manifest and unlock code can access the contents of the Snowball appliance sent for that job.
- Don't leave the Snowball sitting on a loading dock. Left on a loading dock, it can be exposed to the elements. Although the Snowball is rugged, weather can damage the sturdiest of hardware. Report stolen, missing, or broken Snowballs as soon as possible. The sooner such a Snowball issue is reported, the sooner another one can be sent to complete your job.

> **Note**
> The Snowball is the property of AWS. Tampering with a Snowball is a violation of the AWS Acceptable Use Policy. For more information, see http://aws.amazon.com/aup/.

We perform the following security steps:

- All objects transferred to the Snowball have their metadata changed. The only metadata that remains the same is `filename` and `filesize`. All other metadata is set as in the following example: `-rw-rw-r-- 1 root root [filesize] Dec 31 1969 [path/filename]`
- When a Snowball arrives at AWS, we inspect every appliance for any signs of tampering and to verify that no changes were detected by the Trusted Platform Module (TPM). AWS Snowball uses multiple layers of security designed to protect your data, including tamper-resistant enclosures, 256-bit encryption, and an industry-standard TPM designed to provide both security and full chain of custody for your data.
- Once the data transfer job has been processed and verified, AWS performs a software erasure of the Snowball appliance that follows the National Institute of Standards and Technology (NIST) guidelines for media sanitization.

Data Validation in AWS Snowball

Following, you'll find information on how Snowball validates data transfers, and the manual steps you can take to ensure data integrity during and after a job.

Topics

Checksum Validation of Transferred Data

When you copy a file from a local data source using the Snowball client or the Amazon S3 Adapter for Snowball, to the Snowball, a number of checksums are created. These checksums are used to automatically validate data as it's transferred.

At a high level, these checksums are created for each file (or for parts of large files). These checksums are never visible to you, nor are they available for download. The checksums are used to validate the integrity of your data throughout the transfer, and will ensure that your data is copied correctly.

When these checksums don't match, we won't import the associated data into Amazon S3.

Common Validation Errors

Validations errors can occur. Whenever there's a validation error, the corresponding data (a file or a part of a large file) is not written to the destination. The common causes for validation errors are as follows:

- Attempting to copy symbolic links.
- Attempting to copy files that are actively being modified. This will not result in a validation error, but it will cause the checksums to not match at the end of the transfer.
- Attempting to copy whole files larger than 5 TB in size.
- Attempting to copy part sizes larger than 5 GB in size.
- Attempting to copy files to a Snowball that is already at full data storage capacity.
- Attempting to copy files to a Snowball that doesn't follow the Object Key Naming Guidelines for Amazon S3.

Whenever any one of these validation errors occurs, it is logged. You can take steps to manually identify what files failed validation and why as described in the following sections:

- Manual Data Validation for Snowball During Transfer (p. 88) – Outlines how to check for failed files while you still have the Snowball on-premises.

- Manual Data Validation for Snowball After Import into Amazon S3 (p. 89) – Outlines how to check for failed files after your import job into Amazon S3 has ended.

Manual Data Validation for Snowball During Transfer

You can use manual validation to check that your data was successfully transferred to a Snowball Edge. You can also use manual validation if you receive an error after attempting to transfer data. Use the following section to find how to manually validate data on a Snowball Edge.

Check the failed-files log – Snowball client

When you run the Snowball client `copy` command, a log showing any files that couldn't be transferred to the Snowball is generated. If you encounter an error during data transfer, the path for the failed-files log will be printed to the terminal. This log is saved as a comma-separated values (.csv) file. Depending on your operating system, you find this log in one of the following locations:

- **Windows** – `C:/Users/<username>/AppData/Local/Temp/snowball-<random-character-string>/failed-files`
- **Linux** – `/tmp/snowball-<random-character-string>/failed-files`
- **Mac** – `/var/folders/gf/<random-character-string>/<random-character-string>/snowball-7464536051505188504/failed-files`

Use the --verbose option for the Snowball client copy command

When you run the Snowball client `copy` command, you can use the `--verbose` option to list all the files that are transferred to the Snowball. You can use this list to validate the content that was transferred to the Snowball.

Check the logs – Amazon S3 Adapter for Snowball

When you run the Amazon S3 Adapter for Snowball to copy data with the AWS CLI, logs are generated. These logs are saved in the following locations, depending on your file system:

- **Windows** – `C:/Users/<username>/.aws/snowball/logs/snowball_adapter_<year_month_date_hour>`
- **Linux** – `/home/.aws/snowball/logs/snowball_adapter_<year_month_date_hour>`
- **Mac** – `/Users/<username>/.aws/snowball/logs/snowball_adapter_<year_month_date_hour>`

Use the --stopOnError copy option

If you're transferring with the Snowball client, you can use this option to stop the transfer process in the event a file fails. This option stops the copy on any failure so you can address that failure before continuing the copy operation. For more information, see Options for the snowball cp Command (p. 60).

Run the Snowball client's validate command

The Snowball client's `snowball validate` command can validate that the files on the Snowball were all completely copied over to the Snowball. If you specify a path, then this command validates the content pointed to by that path and its subdirectories. This command lists files that are currently in the process of being transferred as incomplete for their transfer status. For more information on the validate command, see Validate Command for the Snowball Client (p. 59).

Manual Data Validation for Snowball After Import into Amazon S3

After an import job has completed, you have several options to manually validate the data in Amazon S3, as described following.

Check job completion report and associated logs

Whenever data is imported into or exported out of Amazon S3, you get a downloadable PDF job report. For import jobs, this report becomes available at the end of the import process. For more information, see Getting Your Job Completion Report and Logs in the Console (p. 42).

S3 inventory

If you transferred a huge amount of data into Amazon S3 in multiple jobs, going through each job completion report might not be an efficient use of time. Instead, you can get an inventory of all the objects in one or more Amazon S3 buckets. Amazon S3 inventory provides a .csv file showing your objects and their corresponding metadata on a daily or weekly basis. This file covers objects for an Amazon S3 bucket or a shared prefix (that is, objects that have names that begin with a common string).

Once you have the inventory of the Amazon S3 buckets that you've imported data into, you can easily compare it against the files that you transferred on your source data location. In this way, you can quickly identify what files where not transferred.

Use the Amazon S3 sync command

If your workstation can connect to the internet, you can do a final validation of all your transferred files by running the AWS CLI command `aws s3 sync`. This command syncs directories and S3 prefixes. This command recursively copies new and updated files from the source directory to the destination. For more information, see http://docs.aws.amazon.com/cli/latest/reference/s3/sync.html.

> **Important**
> If you specify your local storage as the destination for this command, make sure that you have a backup of the files you sync against. These files are overwritten by the contents in the specified Amazon S3 source.

Snowball Notifications

Snowball is designed to take advantage of the robust notifications delivered by Amazon Simple Notification Service (Amazon SNS). While creating a job, you can provide a list of comma-separated email addresses to receive email notifications for your job.

You can also choose from the status list which job status values trigger these notifications. For more information about the different job status values, see Job Statuses (p. 13).

You can configure Amazon SNS to send text messages for these status notifications from the Amazon SNS console. For more information, see Sending and Receiving SMS Notifications Using Amazon SNS.

> **Note**
> These notifications are optional, and are free if you're within your first million Amazon SNS requests for the month. For more information about Amazon SNS pricing, see https://aws.amazon.com/sns/pricing.

After you create your job, every email address that you specified to get Amazon SNS notifications receives an email from AWS Notifications asking for confirmation to the topic subscription. For each email address to receive additional notifications, a user of the account must confirm the subscription by choosing **Confirm subscription**.

The Amazon SNS notification emails are tailored for each triggering state, and include a link to the AWS Snowball Management Console.

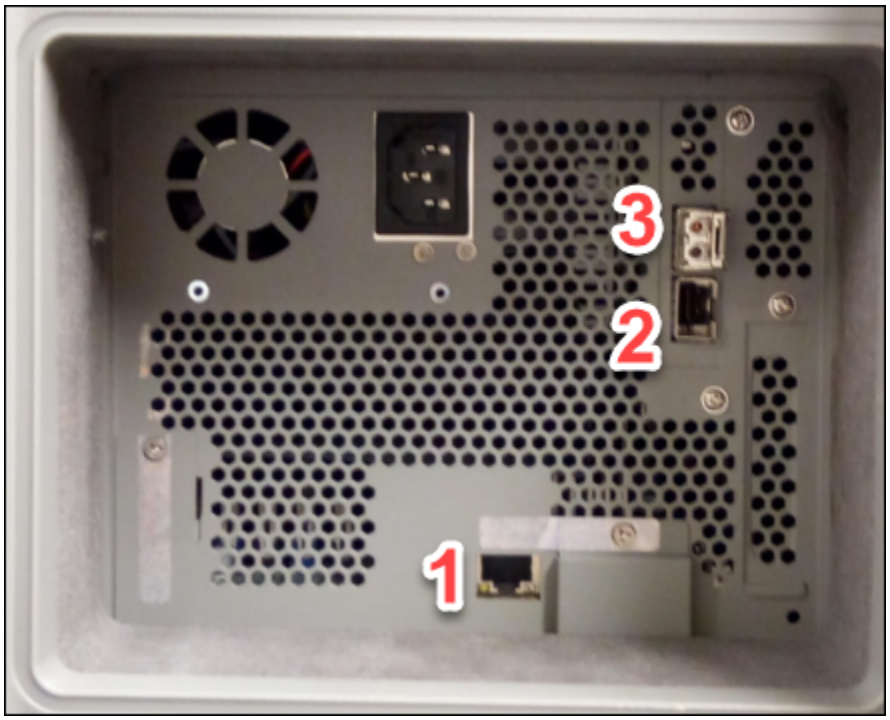
AWS Snowball Specifications

The following table outlines hardware specifications for the Snowball appliance.

Item	Specification
Storage capacity	50 TB Snowballs have 42 TB of usable space. 80 TB Snowballs have 72 TB of usable space.
On-board I/O 10-gigabit interfaces	Each Snowball supports RJ45 (Cat6), SFP+ Copper, and SFP+ Optical.
Cables	Each Snowball ships with RJ45 and copper SFP+ cables. For SFP+ optical, you need to use your own cable, connected to the SFP+ optical adapter in one of the SFP+ ports.
Thermal requirements	Snowballs are designed for office operations, and are ideal for data center operations.
Decibel output	On average, a Snowball produces 68 decibels of sound, typically quieter than a vacuum cleaner or living-room music.
Weight	47 pounds (21.3 Kg)
Height	19.75 inches (501 mm)
Width	12.66 inches (320 mm)
Length	21.52 inches (548 mm)
Power	In the US regions: NEMA 5–15p 100–220 volts. In all regions, a power cable is included.
Power consumption	200 watts.
Voltage	100 – 240V AC
Frequency	47/63 Hz
Power conversion efficiency	80 – 84% at 25C, 230Vac
Temperature range	0 – 40°C (operational)
Non-operational Altitude	Not specified
Operational Altitude	0 to 3,000m (0 to 10,000')

Supported Network Hardware

After you open the back panel of the Snowball, you see the network ports shown in the following photograph.



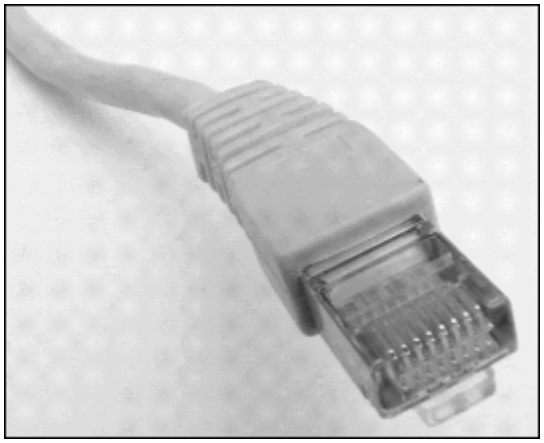
These ports support the following network hardware.

1. RJ45

This port provides 1Gbase-TX/10Gbase-TX operation. It is connected via UTP cable terminated with a RJ45 connector.

1G operation is indicated by a blinking amber light. 1G operation is not recommended for large-scale data transfers to the Snowball device, as it will dramatically increase the time it takes to transfer data.

10G operation is indicated by a blinking green light. It requires a Cat6A UTP cable with a maximum operating distance of 180 feet(55 meters).



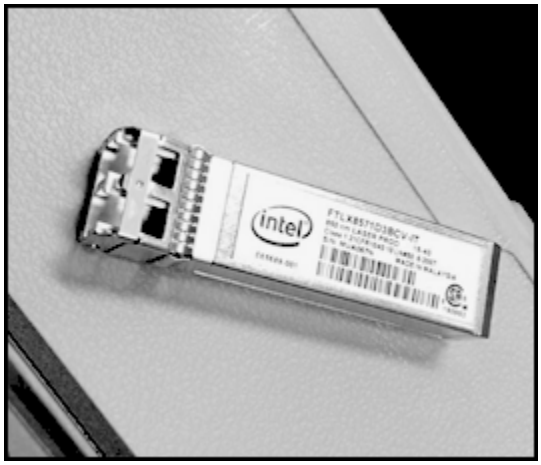
2. SFP+

This port provides an SFP+ interface compatible with both SFP+ transceiver modules and direct-attach copper (DAC) cables. You need to provide your own transceivers or DAC cables. Examples include:

- 10Gbase-LR (single mode fiber) transceiver
- 10Gbase-SR (multi-mode fiber) transceiver
- SFP+ DAC cable

3. SFP+

This port provides an SFP+ interface and a 10Gbase-SR transceiver that uses multi-mode fiber optic media with an LC connector.



Workstation Specifications

The workstation is the computer, server, or virtual machine that hosts your mounted data source. You connect the Snowball to this workstation to transfer your data. Because the workstation is considered the bottleneck for transferring data between the Snowball and the data source, we highly recommend that your workstation be a powerful computer, able to meet high demands for processing, memory, and networking.

We recommend that your workstation be a computer dedicated to the task of running the Snowball client or the Amazon S3 Adapter for Snowball while you're transferring data. Each instance of the client

or the adapter requires up to 7 GB of dedicated RAM for memory-intensive tasks, such as performing encryption.

Note
The hardware specifications for the workstations that are used to transfer data to and from a Snowball are for a powerful computer. These hardware specifications are mainly based on security requirements for the service. When data is transferred to a Snowball, a file is loaded into the workstation's memory. While in memory, that file is fully encrypted by either the Snowball client or the Amazon S3 Adapter for Snowball. Once the file is encrypted, chunks of the encrypted file are sent to the Snowball. At no point is any data stored to disk. All data is kept in memory, and only encrypted data is sent to the Snowball. These steps of loading into memory, encrypting, chunking, and sending to the Snowball are both CPU- and memory-intensive.

The following table outlines the suggested specifications for your workstation.

Item	Suggested Specification
Processing power	16 core CPU
Memory	16 gigabytes of RAM **Important** Each running instance of the client and/or adapter requires up to 7 GB of dedicated RAM for memory-intensive tasks, such as performing the `snowball cp` command.
Microsoft Windows support (64-bit only)	• Windows 7 • Windows 8 • Windows 10
Mac support	Mac OS X version 10.10 or higher
Linux support (64-bit only)	• Ubuntu version 12 or higher • Red Hat Enterprise Linux (RHEL) version 6 or higher
User interface support	• Keyboard • Mouse • Monitor
Network I/O support	• RJ45 • SFP+ Copper • SFP+ Optical

AWS Snowball Limits

Following, you can find information about limitations on using AWS Snowball (Snowball).

Important
When you transfer data into Amazon Simple Storage Service using a Snowball, keep in mind that individual Amazon S3 objects can range in size from a minimum of 0 bytes to a maximum of 5 terabytes (TB).

Regional Limitations for AWS Snowball

The AWS Snowball service has two device types, the standard Snowball and the Snowball Edge. The following table highlights which of these devices are available in which regions.

Note
The guide you're reading now is for the Snowball, which has 50 TB or 80 TB of storage space. If you are looking for documentation for the Snowball Edge, see the AWS Snowball Edge Developer Guide.

Region	Snowball Availability	Snowball Edge Availability
US East (Ohio)	50 TB and 80 TB	100 TB
US East (N. Virginia)	50 TB and 80 TB	100 TB
US West (N. California)	50 TB and 80 TB	100 TB
US West (Oregon)	50 TB and 80 TB	100 TB
Canada (Central)	80 TB only	100 TB
Asia Pacific (Mumbai)	80 TB only	Not available
Asia Pacific (Sydney)	80 TB only	100 TB
Asia Pacific (Tokyo)	80 TB only	100 TB
EU (Frankfurt)	80 TB only	100 TB
EU (Ireland)	80 TB only	100 TB
EU (London)	80 TB only	100 TB
South America (São Paulo)	80 TB only	100 TB

Limitations on Jobs in AWS Snowball

The following limitations exist for creating jobs in AWS Snowball:

- For security purposes, data transfers must be completed within 90 days of the Snowball being prepared.
- Currently, Snowball doesn't support server-side encryption with AWS KMS–managed keys (SSE-KMS) or server-side encryption with customer-provided keys (SSE-C). Snowball does support server-side

encryption with Amazon S3–managed encryption keys (SSE-S3). For more information on SSE-S3, see Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3) in the *Amazon Simple Storage Service Developer Guide.*

- In the US regions, Snowballs come in two sizes: 50 TB and 80 TB. All other regions have the 80 TB Snowballs only. If you're using Snowball to import data, and you need to transfer more data than will fit on a single Snowball, create additional jobs. Each export job can use multiple Snowballs.
- The default service limit for the number of Snowballs you can have at one time is 1. If you want to increase your service limit, contact AWS Support.
- All objects transferred to the Snowball have their metadata changed. The only metadata that remains the same is `filename` and `filesize`. All other metadata is set as in the following example: `-rw-rw-r-- 1 root root [filesize] Dec 31 1969 [path/filename]`

Limitations on Transferring On-Premises Data with a Snowball

The following limitations exist for transferring data to or from a Snowball appliance on-premises:

- Files must be in a static state while being copied. Files that are modified while they are being transferred will not be imported into Amazon S3.
- Jumbo frames are not supported—that is, Ethernet frames with more than 1500 bytes of payload.
- When selecting what data to export, keep in mind that objects with trailing slashes in their names (`/` or `\`) will not be transferred. Before exporting any objects with trailing slashes, update their names to remove the slash.
- When using the Amazon S3 Adapter for Snowball with the AWS CLI to transfer data, note that the `--recursive` option for the `cp` command is only supported for uploading data to a Snowball, not for transferring data from a Snowball.

Limitations on Shipping a Snowball

The following limitations exist for shipping a Snowball:

- AWS will not ship a Snowball to a post office box.
- AWS will not ship a Snowball between non-US regions—for example, from EU (Ireland) to EU (Frankfurt), or from Asia Pacific (Mumbai) to Asia Pacific (Sydney).
- Moving a Snowball to an address other than the one specified when the job was created is not allowed and is a violation of the AWS Service Terms.

For more information on shipping, see Shipping Considerations for AWS Snowball (p. 73).

Limitations on Processing Your Returned Snowball for Import

To connect your returned Snowball to one of our Snowball stations for import, the appliance must meet the following requirements:

- The Snowball appliance must not be compromised. Except for the two access panels in the front and the back, don't open the Snowball for any reason.

- The appliance must not be physically damaged. You can prevent damage by closing the two panels on the Snowball until the latches make an audible clicking sound.
- The Snowball's E Ink display must be visible, and must show the return label that was automatically generated when you finished transferring your data onto the Snowball.

Note
All Snowballs returned that do not meet these requirements are erased without work performed on them.

Troubleshooting for a Standard Snowball

The following can help you troubleshoot problems that you might have with an AWS Snowball (Snowball). If you're having trouble establishing a connection to a Snowball, see Why can't my AWS Snowball appliance establish a connection with the network? in the *AWS Knowledge Center*.

Troubleshooting Connection Problems

The following can help you troubleshoot issues you might have with connecting to your Snowball.

- Routers and switches that work at a rate of 100 megabytes per second won't work with a Snowball. We recommend that you use a switch that works at a rate of 1 GB per second (or faster).

Troubleshooting Data Transfer Problems

If you encounter performance issues while transferring data to or from a Snowball, see Performance for AWS Snowball (p. 34) for recommendations and guidance on improving transfer performance. The following can help you troubleshoot issues you might have with your data transfer to or from a Snowball.

- Data can't be transferred into the root folder of the Snowball. If you're having trouble transferring data into the Snowball, make sure that you're transferring data into a folder on the Snowball that is not the root folder.
- For security purposes, data transfers must be completed within 90 days of the Snowball being prepared. After 90 days, the Snowball becomes locked to additional on-premises data transfers. If the Snowball becomes locked during a data transfer, return the Snowball and create a new job to transfer the rest of your data. If the Snowball becomes locked during an import job, we can still transfer the existing data on the Snowball into Amazon S3.
- Objects transferred onto Snowballs have a maximum key length of 933 bytes. Key names that include characters that take up more than one byte each still have a maximum key length of 933 bytes. When determining key length, you include the file or object name and also its path or prefixes. Thus, files with short file names within a heavily nested path can have keys longer than 933 bytes. The bucket name is not factored into the path when determining the key length. Some examples follow.

Object Name	Bucket Name	Path Plus Bucket Name	Key Length
`sunflower-1.jpg`	`pictures`	`sunflower-1.jpg`	15 characters
`receipts.csv`	`MyTaxInfo`	`/Users/Eric/Documents/2016/January/`	47 characters
`bhv.1`	`$7$zWwwXKQj$gLAOoZCj$r8p`	`/.VfV/FqGC3QN$7BXys3KHYePfuIOMNjY83dVx`	135 characters

Object Name	Bucket Name	Path Plus Bucket Name	Key Length
		ugPYlxVg/evpcQEJLT/rSwZc$MlVVf/$hwefVISRqwepB$/BiiD/PPF$twRAjrD/fIMp/0NY	

If a key's length is larger than 933 bytes, you see the following error message when you try to copy the object to a Snowball:

```
Failed to copy the following file: <Name of object with a keylength over 933 bytes>
          PARENT_NOT_FOUND:
```

If you receive this error message, you can resolve the issue by reducing the object's key length.

- If you're using Linux and you can't upload files with UTF-8 characters to a Snowball, it might be because your Linux workstation doesn't recognize UTF-8 character encoding. You can correct this issue by installing the `locales` package on your Linux workstation and configuring it to use one of the UTF-8 locales like `en_US.UTF-8`. You can configure the `locales` package by exporting the environment variable `LC_ALL`, for example: `export LC_ALL=en_US.UTF-8`
- If you encounter unexpected errors during data transfer to the Snowball, we want to hear about it. Make a copy of your logs and include them along with a brief description of the issues that you encountered in a message to AWS Support. For more information about logs, see Snowball Logs (p. 64).

Troubleshooting Client Problems

The following can help you troubleshoot issues with the Snowball client.

- If you're having trouble using the Snowball client, type the command `snowball help` for a list of all available actions for that tool.
- Although you can run multiple instances of the Snowball client at the same time, each instance of the client requires up to 7 GB of dedicated RAM for memory-intensive tasks, such as performing the `snowball cp` command. If your workstation runs out of memory as it runs the Snowball client, you see a Java `OutOfMemoryError` exception returned in the terminal window. You can resolve this issue by freeing up resources on the workstation or increasing the amount of memory for your workstation, and then performing your Snowball client task again.
- If you encounter issues while transferring data to a Snowball using the client on a PC running Microsoft Windows Server, it might be due to the Data Deduplication feature in Windows. If you have the Data Deduplication feature turned on, we recommend that you use the Amazon S3 Adapter for Snowball with the AWS CLI to transfer data instead. For more information, see Transferring Data with the Amazon S3 Adapter for Snowball (p. 65).

Troubleshooting Snowball Client Validation Problems

When you transfer data, the copy operation first performs a precheck on the metadata for each file to copy. If any of the following attributes are true about a file's metadata, then the copy operation stops before it transfers any files:

- **The size of the file is greater than 5 TB** – Objects in Amazon S3 must be 5 TB or less in size, so files that are larger 5 TB in size can't be transferred to the Snowball. If you encounter this problem, separate the file into parts smaller than 5 TB, compress the file so that it's within the 5 TB limit, or otherwise reduce the size of the file, and try again.
- **The file is a symbolic link, and only contains a reference to another file or directory** – Symbolic links (or junctions) can't be transferred into Amazon S3.
- **There are permissions issues for access to the file** – For example, a user might be trying to read a file on the Snowball client when that user doesn't have read permissions for that file. Permissions issues result in precheck failures.
- **Object key length too large** –If an object's key length is larger than 933 bytes, it fails the precheck.

For a list of files that can't be transferred, check the terminal before data copying starts. You can also find this list in the `<temp directory>/snowball-<random-character-string>/failed-files` file, which is saved to your Snowball client folder on the workstation. For Windows, this temp directory would be located in `C:/Users/<username>/AppData/Local/Temp`. For Linux and Mac, the temp directory would be located in `/tmp`.

If you discover errors when you run the `snowball validate` command, identify the files that failed the transfer, resolve the issues that the error messages report, and then transfer those files again. If your validation command fails with the same error message, then you can use the `-f` option with the `snowball cp` command to force the copy operation and overwrite the invalid files.

HDFS Troubleshooting

When setting up a data transfer from your HDFS (version 2.x) cluster to a Snowball device, you may encounter Kerberos authentication errors. This can happen if you're not using one of the verified encryption types known to work with Snowball:

- des3-cbc-sha1-kd
- aes-128-cts-hmac-sha1-96
- 256-cts-hmac-sha1-96
- rc4-hmac (arcfour-hmac)

If you've encountered a Kerberos authentication issue, you can attempt to resolve it with one of the following workarounds:

- **Temporarily disable Kerberos** – If you disable Kerberos on your HDFS cluster, you should also disconnect any non-essential active connections to the cluster while transferring data. Once your transfer is complete, reactivate your Kerberos authentication.
- **Use a Snowball Edge with the file interface** – The Snowball Edge provides an NFS mount point through it's file interface feature. You could mount the Snowball Edge, and copy the files from your HDFS cluster. For more information on using the file interface, see Using the File Interface for the AWS Snowball Edge in the AWS Snowball Edge Developer Guide.

Troubleshooting Adapter Problems

If you're communicating with the Snowball through the Amazon S3 Adapter for Snowball using the AWS CLI, and you encounter an error that says `Unable to locate credentials. You can configure credentials by running "aws configure".` you need to configure your AWS credentials used by the CLI to run commands. For more information, see Configuring the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

Troubleshooting Import Job Problems

Sometimes files fail to import into Amazon S3. If the following issue occurs, try the actions specified to resolve your issue. If a file fails import, you might need to try importing it again. Importing it again might require a new job for Snowball Edge.

Files failed import into Amazon S3 due to invalid characters in object names

This problem occurs if a file or folder name has characters that aren't supported by Amazon S3. Amazon S3 has rules about what characters can be in object names. For more information, see Object Key Naming Guidelines.

Action to take

If you encounter this issue, you see the list of files and folders that failed import in your job completion report.

In some cases, the list is prohibitively large, or the files in the list are too large to transfer over the internet. In these cases, you should create a new Snowball import job, change the file and folder names to comply with Amazon S3 rules, and transfer the files again.

If the files are small and there isn't a large number of them, you can copy them to Amazon S3 through the AWS CLI or the AWS Management Console. For more information, see How Do I Upload Files and Folders to an S3 Bucket? in the *Amazon Simple Storage Service Console User Guide.*

Troubleshooting Export Job Problems

Sometimes files fail to export into your workstation. If the following issue occurs, try the actions specified to resolve your issue. If a file fails export, you might need to try exporting it again. Exporting it again might require a new job for Snowball Edge.

Files failed export to a Microsoft Windows Server

A file can fail export to a Microsoft Windows Server if it or a related folder is named in a format not supported by Windows. For example, if your file or folder name has a colon (`:`) in it, the export fails because Windows doesn't allow that character in file or folder names.

Action to take

1. Make a list of the names that are causing the error. You can find the names of the files and folders that failed export in your logs. For more information, see Getting Your Job Completion Report and Logs in the Console (p. 42).
2. Change the names of the objects in Amazon S3 that are causing the issue to remove or replace the unsupported characters.
3. If the list of names is prohibitively large, or if the files in the list are too large to transfer over the internet, create a new export job specifically for those objects.

   If the files are small and there isn't a large number of them, copy the renamed objects from Amazon S3 through the AWS CLI or the AWS Management Console. For more information, see How Do I Download an Object from an S3 Bucket? in the *Amazon Simple Storage Service Console User Guide.*

Job Management API Reference for AWS Snowball

The job management API for AWS Snowball is a network protocol based on HTTP (RFC 2616). For more information on this RFC, see HTTP (RFC 2616). on the IETF website. For each call to the job management API, you make an HTTP request to the region-specific job management API endpoint for the AWS Region where you want to manage jobs. The API uses JSON (RFC 4627) documents for HTTP request/response bodies.

> **Note**
> API calls made within the US regions for listing jobs or describing addresses will return all jobs or addresses within the US for that account, respectively.

The job management API for Snowball is an RPC model, in which there is a fixed set of operations and the syntax for each operation is known to clients without any prior interaction. Following, you can find a description of each API operation using an abstract RPC notation, with an operation name that does not appear on the wire. For each operation, the topic specifies the mapping to HTTP request elements.

The specific job management operation to which a given request maps is determined by a combination of the request's method (GET, PUT, POST, or DELETE) and which of the various patterns its Request-URI matches. If the operation is PUT or POST, Snowball extracts call arguments from the Request-URI path segment, query parameters, and the JSON object in the request body.

Although the operation name, such as `CreateJob`, doesn't appear on the wire, these operation names are meaningful in AWS Identity and Access Management (IAM) policies. The operation name is also used to name commands in command-line tools and elements of the AWS SDK APIs. For example, the AWS Command Line Interface (AWS CLI) command `create-job` maps to the `CreateJob` operation. The operation name also appears in CloudTrail logs for Snowball API calls.

For information on installing and setting up the AWS CLI, including specifying what regions you want to make AWS CLI calls against, see the AWS Command Line Interface User Guide.

> **Note**
> The job management API provides programmatic access to the same functionality available in the AWS Snowball Management Console, that is to create and manage jobs for Snowball. To actually transfer data locally with a Snowball appliance, you'll need to use the Snowball client or the Amazon S3 Adapter for Snowball. For more information, see Transferring Data with a Snowball (p. 51).

API Endpoint

The API endpoint is the Domain Name Service (DNS) name used as a host in the HTTP URI for the API calls. These API endpoints are region-specific and take the following form.

```
snowball.aws-region.amazonaws.com
```

For example, the Snowball API endpoint for the US West (Oregon) Region is the following.

```
snowball.us-west-2.amazonaws.com
```

For a list of AWS Regions that Snowball supports (where you can create and manage jobs), see AWS Import/Export in the *AWS General Reference*.

The region-specific API endpoint defines the scope of the Snowball resources that are accessible when you make an API call. For example, when you call the `ListJobs` operation using the preceding endpoint, you get a list of jobs in the US West (Oregon) Region that have been created in your account.

API Version

The version of the API being used for a call is identified by the first path segment of the request URI, and its form is a ISO 8601 date. The documentation describes API version 2016-06-30.

API Permission Policy Reference

The following policies are needed for creating jobs with the job management API for Snowball.

Role Trust Policy for Creating Jobs

Using the job management API to create jobs requires the following trust policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "importexport.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "AWSIE"
        }
      }
    }
  ]
}
```

Note
To learn more about trust policies, see Modifying a Role in the IAM User Guide.

Role Policy for Creating Import Jobs

Creating an import job requires the following role policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": "arn:aws:s3:::*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "snowball:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Role Policy for Creating Export Jobs

Creating an export job requires the following role policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
       "Effect": "Allow",
       "Action": [
         "snowball:*"
       ],
       "Resource": [
         "*"
       ]
    }
  ]
}
```

Amazon S3 Bucket Policy Principal for Creating Jobs

If the Amazon S3 buckets that you use with Snowball have bucket policies in place that require listing the role session name of the assumed role, then you'll need to specify a principal in those policies that identifies `AWSImportExport-Validation`. The following Amazon S3 bucket policy example demonstrates how to do so.

Example

```
{
 "Version": "2012-10-17",
```

```
 "Statement": {
  "Sid": "Allow AWS Snowball To Create Jobs",
  "Effect": "Deny",
  "NotPrincipal": {
   "AWS": [
    "arn:aws:iam::111122223333:role/rolename",
    "arn:aws:sts::111122223333:assumed-role/rolename/AWSImportExport-Validation",
    "arn:aws:iam::111122223333:root"
   ]
  },
  "Action": "S3:*",
  "Resource": ["arn:aws:s3:::examplebucket/*"]
 }
}
```

In this policy example, we deny access to all principals except the one named in the `NotPrincipal` element. For more information on how to use `NotPrincipal`, see NotPrincipal in the *IAM User Guide*.

Related Topics

- AWS Snowball API Reference

Document History

The following table describes the important changes to the documentation since the last release of AWS Snowball.

- **API version:** latest
- **Latest document update:** March 1, 2018

Change	Description	Date Changed
New AWS Region supported	AWS Snowball is now supported in the Asia Pacific (Singapore) region. For more information on shipping in this AWS Region, see Shipping Considerations for AWS Snowball (p. 73).	March 1, 2018
New AWS Region supported	AWS Snowball is now supported in the EU (Paris) region. For more information on shipping in this AWS Region, see Shipping Considerations for AWS Snowball (p. 73).	December 18, 2017
Improved transfer speed for small files	You can now automatically batch small files to improve transfer speed by using the `--batch` option of the Snowball client copy command. During the import process into Amazon S3, all files in batches are automatically extracted. For more information, see Options for the snowball cp Command (p. 60).	November 14, 2017
New AWS Region supported	AWS Snowball is now supported in the Asia Pacific (Tokyo) region, with region-specific shipping options. For more information, see Shipping Considerations for AWS Snowball (p. 73).	September 19, 2017
New AWS Region supported	AWS Snowball is now supported in the South America (São Paulo) region, with region-specific shipping options. For more information, see Shipping Considerations for AWS Snowball (p. 73).	August 8, 2017
New AWS Region supported	AWS Snowball is now supported in the Canada (Central) region, with region-specific shipping options. For more information, see Shipping Considerations for AWS Snowball (p. 73).	June 29, 2017
Documentation update	The right navigation has been updated for clarity and consistency, and a regional limitations section has been added. For more information, see Regional Limitations for AWS Snowball (p. 95).	May 8, 2017
Reading Hadoop Distributed File System (HDFS) custom configuration files is now supported.	You can now specify the location of your HDFS custom configuration XML files using the new `--hdfsconfig` option for the Snowball client `cp` command.	February 8, 2017

Change	Description	Date Changed
Importing data from a Hadoop Distributed File System (HDFS) cluster (version 2.x) is now supported.	You can now import data from a HDFS cluster (version 2.x) to Amazon S3 through a Snowball. For more information, see Importing Data from HDFS (p. 54).	September 30, 2016
Programmatic job management and data transfers are now supported.	You can now programmatically manage jobs and transfer data with Snowball. For more information on using the job management API for Snowball, see AWS Snowball API Reference. For more information on using the Amazon S3 Adapter for Snowball to call Amazon S3 REST API actions to transfer data with a Snowball, see Transferring Data with the Amazon S3 Adapter for Snowball (p. 65).	August 11, 2016
Snowball is now available from EU (Frankfurt) in the European Union.	You can now create and manage jobs from the EU (Frankfurt) AWS Management Console. For more information, see Shipping Considerations for AWS Snowball (p. 73).	July 25, 2016
Snowball is now available in India.	Snowball is now available in the Asia Pacific (Mumbai) region. For more information, see Shipping Considerations for AWS Snowball (p. 73).	June 27, 2016
Snowball is now available in new AWS Regions and has a new storage capacity option.	Snowball is now available in the following regions; US East (N. Virginia), US West (Oregon), US West (N. California), EU (Ireland), Asia Pacific (Sydney), and AWS GovCloud (US). For more information, see Shipping Considerations for AWS Snowball (p. 73). Snowball also has a new 80 TB model available in all regions, in addition to the 50 TB model only available in the US regions.	April 19, 2016
Introducing export for AWS Snowball	You can now use Snowball to export data from Amazon Simple Storage Service (Amazon S3).	February 29, 2016
Hardware update: SFP+ optical interface	The Snowball appliance has been updated to include a new SFP+ optical interface that offers slightly better signal integrity over its copper counterpart but otherwise shares the same high performance. If you received a Snowball before this date, it does not have this network interface option.	November 18, 2015
Introducing AWS Snowball	AWS Snowball is a data transfer service for importing huge amounts of data into Amazon Simple Storage Service (Amazon S3). With Snowball, you can import hundreds of terabytes or petabytes of data from your on-premises data centers into Amazon S3.	October 7, 2015

AWS Glossary

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

Numbers and Symbols

100-continue	A method that enables a client to see if a server can accept a request before actually sending it. For large PUT requests, this method can save both time and bandwidth charges.

A

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

AAD	See additional authenticated data.
access control list (ACL)	A document that defines who can access a particular bucket (p. 121) or object. Each bucket (p. 121) and object in Amazon S3 (p. 113) has an ACL. The document defines what each type of user can do, such as write and read permissions.
access identifiers	See credentials.
access key	The combination of an access key ID (p. 108) (like `AKIAIOSFODNN7EXAMPLE`) and a secret access key (p. 150) (like `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). You use access keys to sign API requests that you make to AWS.
access key ID	A unique identifier that's associated with a secret access key (p. 150); the access key ID and secret access key are used together to sign programmatic AWS requests cryptographically.
access key rotation	A method to increase security by changing the AWS access key ID. This method enables you to retire an old key at your discretion.

access policy language	A language for writing documents (that is, *policies* (p. 143)) that specify who can access a particular AWS resource (p. 147) and under what conditions.
account	A formal relationship with AWS that is associated with (1) the owner email address and password, (2) the control of resource (p. 147)s created under its umbrella, and (3) payment for the AWS activity related to those resources. The AWS account has permission to do anything and everything with all the AWS account resources. This is in contrast to a user (p. 156), which is an entity contained within the account.
account activity	A web page showing your month-to-date AWS usage and costs. The account activity page is located at https://aws.amazon.com/account-activity/.
ACL	See access control list (ACL).
ACM	See AWS Certificate Manager (ACM).
action	An API function. Also called *operation* or *call*. The activity the principal (p. 144) has permission to perform. The action is B in the statement "A has permission to do B to C where D applies." For example, Jane sends a request to Amazon SQS (p. 113) with Action=ReceiveMessage. Amazon CloudWatch (p. 110): The response initiated by the change in an alarm's state: for example, from OK to ALARM. The state change may be triggered by a metric reaching the alarm threshold, or by a SetAlarmState request. Each alarm can have one or more actions assigned to each state. Actions are performed once each time the alarm changes to a state that has an action assigned, such as an Amazon Simple Notification Service (p. 113) notification, an Auto Scaling (p. 115) policy (p. 143) execution or an Amazon EC2 (p. 111) instance (p. 135) stop/terminate action.
active trusted signers	A list showing each of the trusted signers you've specified and the IDs of the corresponding active key pairs that Amazon CloudFront (p. 110) is aware of. To be able to create working signed URLs, a trusted signer must appear in this list with at least one key pair ID.
additional authenticated data	Information that is checked for integrity but not encrypted, such as headers or other contextual metadata.
administrative suspension	Auto Scaling (p. 115) might suspend processes for Auto Scaling group (p. 115) that repeatedly fail to launch instances. Auto Scaling groups that most commonly experience administrative suspension have zero running instances, have been trying to launch instances for more than 24 hours, and have not succeeded in that time.
alarm	An item that watches a single metric over a specified time period, and triggers an Amazon SNS (p. 113) topic (p. 156) or an Auto Scaling (p. 115) policy (p. 143) if the value of the metric crosses a threshold value over a predetermined number of time periods.
allow	One of two possible outcomes (the other is deny (p. 127)) when an IAM (p. 118) access policy (p. 143) is evaluated. When a user makes a request to AWS, AWS evaluates the request based on all permissions that apply to the user and then returns either allow or deny.
Amazon API Gateway	A fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. See Also https://aws.amazon.com/api-gateway.

Amazon AppStream	A web service for streaming existing Windows applications from the cloud to any device. See Also https://aws.amazon.com/appstream/.
Amazon Aurora	A fully managed MySQL-compatible relational database engine that combines the speed and availability of commercial databases with the simplicity and cost-effectiveness of open source databases. See Also https://aws.amazon.com/rds/aurora/.
Amazon CloudFront	An AWS content delivery service that helps you improve the performance, reliability, and availability of your websites and applications. See Also https://aws.amazon.com/cloudfront.
Amazon CloudSearch	A fully managed service in the AWS cloud that makes it easy to set up, manage, and scale a search solution for your website or application.
Amazon CloudWatch	A web service that enables you to monitor and manage various metrics, and configure alarm actions based on data from those metrics. See Also https://aws.amazon.com/cloudwatch.
Amazon CloudWatch Events	A web service that enables you to deliver a timely stream of system events that describe changes in AWS resource (p. 147)s to AWS Lambda (p. 118) functions, streams in Amazon Kinesis Data Streams (p. 112), Amazon Simple Notification Service (p. 113) topics, or built-in targets. See Also https://aws.amazon.com/cloudwatch.
Amazon CloudWatch Logs	A web service for monitoring and troubleshooting your systems and applications from your existing system, application, and custom log files. You can send your existing log files to CloudWatch Logs and monitor these logs in near real-time. See Also https://aws.amazon.com/cloudwatch.
Amazon Cognito	A web service that makes it easy to save mobile user data, such as app preferences or game state, in the AWS cloud without writing any back-end code or managing any infrastructure. Amazon Cognito offers mobile identity management and data synchronization across devices. See Also https://aws.amazon.com/cognito/.
Amazon DynamoDB	A fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. See Also https://aws.amazon.com/dynamodb/.
Amazon DynamoDB Storage Backend for Titan	A storage backend for the Titan graph database implemented on top of Amazon DynamoDB. Titan is a scalable graph database optimized for storing and querying graphs. See Also https://aws.amazon.com/dynamodb/.
Amazon DynamoDB Streams	An AWS service that captures a time-ordered sequence of item-level modifications in any Amazon DynamoDB table, and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near real time. See Also https://aws.amazon.com/dynamodb/.
Amazon Elastic Block Store (Amazon EBS)	A service that provides block level storage volume (p. 157)s for use with EC2 instance (p. 129)s. See Also https://aws.amazon.com/ebs.
Amazon EBS-backed AMI	A type of Amazon Machine Image (AMI) (p. 112) whose instance (p. 135)s use an Amazon EBS (p. 110) volume (p. 157) as their root device. Compare this with instances launched from instance store-backed AMI (p. 135)s, which use the instance store (p. 135) as the root device.

Amazon Elastic Container Registry (Amazon ECR)	A fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon Elastic Container Service (Amazon ECS) (p. 111) and AWS Identity and Access Management (IAM) (p. 118). See Also https://aws.amazon.com/ecr.
Amazon Elastic Container Service (Amazon ECS)	A highly scalable, fast, container (p. 124) management service that makes it easy to run, stop, and manage Docker containers on a cluster (p. 123) of EC2 instance (p. 129)s. See Also https://aws.amazon.com/ecs.
Amazon ECS service	A service for running and maintaining a specified number of task (p. 155)s (instantiations of a task definition (p. 155)) simultaneously.
Amazon EC2 VM Import Connector	See https://aws.amazon.com/ec2/vm-import.
Amazon Elastic Compute Cloud (Amazon EC2)	A web service that enables you to launch and manage Linux/UNIX and Windows server instance (p. 135)s in Amazon's data centers. See Also https://aws.amazon.com/ec2.
Amazon Elastic File System (Amazon EFS)	A file storage service for EC2 (p. 111) instance (p. 135)s. Amazon EFS is easy to use and provides a simple interface with which you can create and configure file systems. Amazon EFS storage capacity grows and shrinks automatically as you add and remove files. See Also https://aws.amazon.com/efs/.
Amazon EMR (Amazon EMR)	A web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop (p. 133) processing combined with several AWS products to do such tasks as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing. See Also https://aws.amazon.com/elasticmapreduce.
Amazon Elastic Transcoder	A cloud-based media transcoding service. Elastic Transcoder is a highly scalable tool for converting (or *transcoding*) media files from their source format into versions that will play on devices like smartphones, tablets, and PCs. See Also https://aws.amazon.com/elastictranscoder/.
Amazon ElastiCache	A web service that simplifies deploying, operating, and scaling an in-memory cache in the cloud. The service improves the performance of web applications by providing information retrieval from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases. See Also https://aws.amazon.com/elasticache/.
Amazon Elasticsearch Service (Amazon ES)	An AWS managed service for deploying, operating, and scaling Elasticsearch, an open-source search and analytics engine, in the AWS Cloud. Amazon Elasticsearch Service (Amazon ES) also offers security options, high availability, data durability, and direct access to the Elasticsearch APIs. See Also https://aws.amazon.com/elasticsearch-service.
Amazon GameLift	A managed service for deploying, operating, and scaling session-based multiplayer games. See Also https://aws.amazon.com/gamelift/.
Amazon Glacier	A secure, durable, and low-cost storage service for data archiving and long-term backup. You can reliably store large or small amounts of data for significantly less than on-premises solutions. Amazon Glacier is optimized for infrequently accessed data, where a retrieval time of several hours is suitable. See Also https://aws.amazon.com/glacier/.

Amazon GuardDuty	A continuous security monitoring service. Amazon GuardDuty can help to identify unexpected and potentially unauthorized or malicious activity in your AWS environment. See Also https://aws.amazon.com/guardduty/.
Amazon Inspector	An automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed report with prioritized steps for remediation. See Also https://aws.amazon.com/inspector.
Amazon Kinesis	A platform for streaming data on AWS. Kinesis offers services that simplify the loading and analysis of streaming data. See Also https://aws.amazon.com/kinesis/.
Amazon Kinesis Data Firehose	A fully managed service for loading streaming data into AWS. Kinesis Data Firehose can capture and automatically load streaming data into Amazon S3 (p. 113) and Amazon Redshift (p. 113), enabling near real-time analytics with existing business intelligence tools and dashboards. Kinesis Data Firehose automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it. See Also https://aws.amazon.com/kinesis/firehose/.
Amazon Kinesis Data Streams	A web service for building custom applications that process or analyze streaming data for specialized needs. Amazon Kinesis Data Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources. See Also https://aws.amazon.com/kinesis/streams/.
Amazon Lightsail	Lightsail is designed to be the easiest way to launch and manage a virtual private server with AWS. Lightsail offers bundled plans that include everything you need to deploy a virtual private server, for a low monthly rate. See Also https://aws.amazon.com/lightsail/.
Amazon Lumberyard	A cross-platform, 3D game engine for creating high-quality games. You can connect games to the compute and storage of the AWS cloud and engage fans on Twitch. See Also https://aws.amazon.com/lumberyard/.
Amazon Machine Image (AMI)	An encrypted machine image stored in Amazon Elastic Block Store (Amazon EBS) (p. 110) or Amazon Simple Storage Service (p. 113). AMIs are like a template of a computer's root drive. They contain the operating system and can also include software and layers of your application, such as database servers, middleware, web servers, and so on.
Amazon Machine Learning	A cloud-based service that creates machine learning (ML) models by finding patterns in your data, and uses these models to process new data and generate predictions. See Also http://aws.amazon.com/machine-learning/.
Amazon Macie	A security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. See Also http://aws.amazon.com/macie/.
Amazon ML	See Amazon Machine Learning.
Amazon Mobile Analytics	A service for collecting, visualizing, understanding, and extracting mobile app usage data at scale.

	See Also https://aws.amazon.com/mobileanalytics.
Amazon MQ	A managed message broker service for Apache ActiveMQ that makes it easy to set up and operate message brokers in the cloud. See Also https://aws.amazon.com/amazon-mq/.
Amazon Redshift	A fully managed, petabyte-scale data warehouse service in the cloud. With Amazon Redshift you can analyze your data using your existing business intelligence tools. See Also https://aws.amazon.com/redshift/.
Amazon Relational Database Service (Amazon RDS)	A web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. See Also https://aws.amazon.com/rds.
Amazon Resource Name (ARN)	A standardized way to refer to an AWS resource (p. 147). For example: arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Bob.
Amazon Route 53	A web service you can use to create a new DNS service or to migrate your existing DNS service to the cloud. See Also https://aws.amazon.com/route53.
Amazon S3	See Amazon Simple Storage Service (Amazon S3).
Amazon S3-Backed AMI	See instance store-backed AMI.
Amazon Silk	A next-generation web browser available only on Fire OS tablets and phones. Built on a split architecture that divides processing between the client and the AWS cloud, Amazon Silk is designed to create a faster, more responsive mobile browsing experience.
Amazon Simple Email Service (Amazon SES)	An easy-to-use, cost-effective email solution for applications. See Also https://aws.amazon.com/ses.
Amazon Simple Notification Service (Amazon SNS)	A web service that enables applications, end-users, and devices to instantly send and receive notifications from the cloud. See Also https://aws.amazon.com/sns.
Amazon Simple Queue Service (Amazon SQS)	Reliable and scalable hosted queues for storing messages as they travel between computers. See Also https://aws.amazon.com/sqs.
Amazon Simple Storage Service (Amazon S3)	Storage for the internet. You can use it to store and retrieve any amount of data at any time, from anywhere on the web. See Also https://aws.amazon.com/s3.
Amazon Simple Workflow Service (Amazon SWF)	A fully managed service that helps developers build, run, and scale background jobs that have parallel or sequential steps. Amazon SWF is like a state tracker and task coordinator in the cloud. See Also https://aws.amazon.com/swf/.
Amazon Sumerian	A set of tools for creating and running high-quality 3D, augmented reality (AR), and virtual reality (VR) applications on the web. See Also https://aws.amazon.com/sumerian/.
Amazon Virtual Private Cloud (Amazon VPC)	A web service for provisioning a logically isolated section of the AWS cloud where you can launch AWS resource (p. 147)s in a virtual network that you define. You control your virtual networking environment, including selection of your

	own IP address range, creation of subnet (p. 154)s, and configuration of route table (p. 148)s and network gateways. See Also https://aws.amazon.com/vpc.
Amazon VPC	See Amazon Virtual Private Cloud (Amazon VPC).
Amazon Web Services (AWS)	An infrastructure web services platform in the cloud for companies of all sizes. See Also https://aws.amazon.com/what-is-cloud-computing/.
Amazon WorkDocs	A managed, secure enterprise document storage and sharing service with administrative controls and feedback capabilities. See Also https://aws.amazon.com/workdocs/.
Amazon WorkMail	A managed, secure business email and calendar service with support for existing desktop and mobile email clients. See Also https://aws.amazon.com/workmail/.
Amazon WorkSpaces	A managed, secure desktop computing service for provisioning cloud-based desktops and providing users access to documents, applications, and resource (p. 147)s from supported devices. See Also https://aws.amazon.com/workspaces/.
Amazon WorkSpaces Application Manager (Amazon WAM)	A web service for deploying and managing applications for Amazon WorkSpaces. Amazon WAM accelerates software deployment, upgrades, patching, and retirement by packaging Windows desktop applications into virtualized application containers. See Also https://aws.amazon.com/workspaces/applicationmanager.
AMI	See Amazon Machine Image (AMI).
analysis scheme	Amazon CloudSearch (p. 110): Language-specific text analysis options that are applied to a text field to control stemming and configure stopwords and synonyms.
application	AWS Elastic Beanstalk (p. 117): A logical collection of components, including environments, versions, and environment configurations. An application is conceptually similar to a folder. AWS CodeDeploy (p. 116): A name that uniquely identifies the application to be deployed. AWS CodeDeploy uses this name to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment.
Application Billing	The location where your customers manage the Amazon DevPay products they've purchased. The web address is http://www.amazon.com/dp-applications.
application revision	AWS CodeDeploy (p. 116): An archive file containing source content—such as source code, web pages, executable files, and deployment scripts—along with an application specification file (p. 114). Revisions are stored in Amazon S3 (p. 113) bucket (p. 121)s or GitHub (p. 133) repositories. For Amazon S3, a revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. For GitHub, a revision is uniquely identified by its commit ID.
application specification file	AWS CodeDeploy (p. 116): A YAML-formatted file used to map the source files in an application revision to destinations on the instance; specify custom permissions for deployed files; and specify scripts to be run on each instance at various stages of the deployment process.
application version	AWS Elastic Beanstalk (p. 117): A specific, labeled iteration of an application that represents a functionally consistent set of deployable application code. A

	version points to an Amazon S3 (p. 113) object (a JAVA WAR file) that contains the application code.
AppSpec file	See application specification file.
AUC	Area Under a Curve. An industry-standard metric to evaluate the quality of a binary classification machine learning model. AUC measures the ability of the model to predict a higher score for positive examples, those that are "correct," than for negative examples, those that are "incorrect." The AUC metric returns a decimal value from 0 to 1. AUC values near 1 indicate an ML model that is highly accurate.
ARN	See Amazon Resource Name (ARN).
artifact	AWS CodePipeline (p. 116): A copy of the files or changes that will be worked upon by the pipeline.
asymmetric encryption	Encryption (p. 130) that uses both a public key and a private key.
asynchronous bounce	A type of bounce (p. 121) that occurs when a receiver (p. 146) initially accepts an email message for delivery and then subsequently fails to deliver it.
atomic counter	DynamoDB: A method of incrementing or decrementing the value of an existing attribute without interfering with other write requests.
attribute	A fundamental data element, something that does not need to be broken down any further. In DynamoDB, attributes are similar in many ways to fields or columns in other database systems. Amazon Machine Learning: A unique, named property within an observation in a data set. In tabular data, such as spreadsheets or comma-separated values (.csv) files, the column headings represent the attributes, and the rows contain values for each attribute.
Aurora	See Amazon Aurora.
authenticated encryption	Encryption (p. 130) that provides confidentiality, data integrity, and authenticity assurances of the encrypted data.
authentication	The process of proving your identity to a system.
Auto Scaling	A web service designed to launch or terminate instance (p. 135)s automatically based on user-defined policies (p. 143), schedules, and health check (p. 133)s. See Also https://aws.amazon.com//autoscaling.
Auto Scaling group	A representation of multiple EC2 instance (p. 129)s that share similar characteristics, and that are treated as a logical grouping for the purposes of instance scaling and management.
Availability Zone	A distinct location within a Region (p. 146) that is insulated from failures in other Availability Zones, and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.
AWS	See Amazon Web Services (AWS).
AWS Application Discovery Service	A web service that helps you plan to migrate to AWS by identifying IT assets in a data center—including servers, virtual machines, applications, application dependencies, and network infrastructure. See Also https://aws.amazon.com/about-aws/whats-new/2016/04/aws-application-discovery-service/.

AWS AppSync	An enterprise level, fully managed GraphQL service with real-time data synchronization and offline programming features. See Also https://aws.amazon.com/appsync/.
AWS Billing and Cost Management	The AWS cloud computing model in which you pay for services on demand and use as much or as little at any given time as you need. While resource (p. 147)s are active under your account, you pay for the cost of allocating those resources and for any incidental usage associated with those resources, such as data transfer or allocated storage. See Also https://aws.amazon.com/billing/new-user-faqs/.
AWS Certificate Manager (ACM)	A web service for provisioning, managing, and deploying Secure Sockets Layer/Transport Layer Security (p. 156) (SSL/TLS) certificates for use with AWS services. See Also https://aws.amazon.com/certificate-manager/.
AWS Cloud9	A cloud-based integrated development environment (IDE) that you use to write, run, and debug code. See Also https://aws.amazon.com/cloud9/.
AWS CloudFormation	A service for writing or changing templates that create and delete related AWS resource (p. 147)s together as a unit. See Also https://aws.amazon.com/cloudformation.
AWS CloudHSM	A web service that helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated hardware security module (HSM) appliances within the AWS cloud. See Also https://aws.amazon.com/cloudhsm/.
AWS CloudTrail	A web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. See Also https://aws.amazon.com/cloudtrail/.
AWS CodeCommit	A fully managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories. See Also https://aws.amazon.com/codecommit.
AWS CodeDeploy	A service that automates code deployments to any instance, including EC2 instance (p. 129)s and instance (p. 135)s running on-premises. See Also https://aws.amazon.com/codedeploy.
AWS CodeDeploy agent	A software package that, when installed and configured on an instance, enables that instance to be used in AWS CodeDeploy deployments.
AWS CodePipeline	A continuous delivery service for fast and reliable application updates. See Also https://aws.amazon.com/codepipeline.
AWS Command Line Interface (AWS CLI)	A unified downloadable and configurable tool for managing AWS services. Control multiple AWS services from the command line and automate them through scripts. See Also https://aws.amazon.com/cli/.
AWS Config	A fully managed service that provides an AWS resource (p. 147) inventory, configuration history, and configuration change notifications for better security and governance. You can create rules that automatically check the configuration of AWS resources that AWS Config records. See Also https://aws.amazon.com/config/.

AWS Elemental MediaConvert	A file-based video conversion service that transforms media into formats required for traditional broadcast and for internet streaming to multi-screen devices. See Also https://aws.amazon.com/mediaconvert.
AWS Elemental MediaLive	A video service that lets you easily create live outputs for broadcast and streaming delivery. See Also https://aws.amazon.com/medialive.
AWS Elemental MediaPackage	A just-in-time packaging and origination service that lets you format highly secure and reliable live outputs for a variety of devices. See Also https://aws.amazon.com/mediapackage.
AWS Elemental MediaStore	A storage service optimized for media that provides the performance, consistency, and low latency required to deliver live and on-demand video content at scale. See Also https://aws.amazon.com/mediastore.
AWS Elemental MediaTailor	A video service that lets you serve targeted ads to viewers while maintaining broadcast quality in over-the-top (OTT) video applications. See Also https://aws.amazon.com/mediatailor.
AWS Database Migration Service	A web service that can help you migrate data to and from many widely used commercial and open-source databases. See Also https://aws.amazon.com/dms.
AWS Data Pipeline	A web service for processing and moving data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals. See Also https://aws.amazon.com/datapipeline.
AWS Device Farm	An app testing service that allows developers to test Android, iOS, and Fire OS devices on real, physical phones and tablets that are hosted by AWS. See Also https://aws.amazon.com/device-farm.
AWS Direct Connect	A web service that simplifies establishing a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or colocation environment. See Also https://aws.amazon.com/directconnect.
AWS Directory Service	A managed service for connecting your AWS resource (p. 147)s to an existing on-premises Microsoft Active Directory or to set up and operate a new, standalone directory in the AWS cloud. See Also https://aws.amazon.com/directoryservice.
AWS Elastic Beanstalk	A web service for deploying and managing applications in the AWS Cloud without worrying about the infrastructure that runs those applications. See Also https://aws.amazon.com/elasticbeanstalk.
AWS Glue	A fully managed extract, transform, and load (ETL) (p. 131) service that you can use to catalog data and load it for analytics. With AWS Glue, you can discover your data, develop scripts to transform sources into targets, and schedule and run ETL jobs in a serverless environment. See Also https://aws.amazon.com/glue.
AWS GovCloud (US)	An isolated AWS Region designed to host sensitive workloads in the cloud, ensuring that this work meets the US government's regulatory and compliance requirements. The AWS GovCloud (US) Region adheres to United States International Traffic in Arms Regulations (ITAR), Federal Risk and Authorization Management Program (FedRAMP) requirements, Department of Defense (DOD) Cloud Security Requirements Guide (SRG) Levels 2 and 4, and Criminal Justice Information Services (CJIS) Security Policy requirements.

	See Also https://aws.amazon.com/govcloud-us/.
AWS Identity and Access Management (IAM)	A web service that enables Amazon Web Services (AWS) (p. 114) customers to manage users and user permissions within AWS. See Also https://aws.amazon.com/iam.
AWS Import/Export	A service for transferring large amounts of data between AWS and portable storage devices. See Also https://aws.amazon.com/importexport.
AWS IoT	A managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices. See Also https://aws.amazon.com/iot.
AWS Key Management Service (AWS KMS)	A managed service that simplifies the creation and control of encryption (p. 130) keys that are used to encrypt data. See Also https://aws.amazon.com/kms.
AWS Lambda	A web service that lets you run code without provisioning or managing servers. You can run code for virtually any type of application or back-end service with zero administration. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app. See Also https://aws.amazon.com/lambda/.
AWS managed key	One of two types of customer master key (CMK) (p. 126)s in AWS Key Management Service (AWS KMS) (p. 118).
AWS managed policy	An IAM (p. 118) managed policy (p. 138) that is created and managed by AWS.
AWS Management Console	A graphical interface to manage compute, storage, and other cloud resource (p. 147)s. See Also https://aws.amazon.com/console.
AWS Management Portal for vCenter	A web service for managing your AWS resource (p. 147)s using VMware vCenter. You install the portal as a vCenter plug-in within your existing vCenter environment. Once installed, you can migrate VMware VMs to Amazon EC2 (p. 111) and manage AWS resources from within vCenter. See Also https://aws.amazon.com/ec2/vcenter-portal/.
AWS Marketplace	A web portal where qualified partners to market and sell their software to AWS customers. AWS Marketplace is an online software store that helps customers find, buy, and immediately start using the software and services that run on AWS. See Also https://aws.amazon.com/partners/aws-marketplace/.
AWS Mobile Hub	An integrated console that for building, testing, and monitoring mobile apps. See Also https://aws.amazon.com/mobile.
AWS Mobile SDK	A software development kit whose libraries, code samples, and documentation help you build high quality mobile apps for the iOS, Android, Fire OS, Unity, and Xamarin platforms. See Also https://aws.amazon.com/mobile/sdk.
AWS OpsWorks	A configuration management service that helps you use Chef to configure and operate groups of instances and applications. You can define the application's architecture and the specification of each component including package installation, software configuration, and resource (p. 147)s such as storage. You can automate tasks based on time, load, lifecycle events, and more. See Also https://aws.amazon.com/opsworks/.
AWS Organizations	An account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage.

	See Also https://aws.amazon.com/organizations/.
AWS SDK for C++	A software development kit for that provides C++ APIs for many AWS services including Amazon S3 (p. 113), Amazon EC2 (p. 111), Amazon DynamoDB (p. 110), and more. The single, downloadable package includes the AWS C++ library, code samples, and documentation. See Also https://aws.amazon.com/sdk-for-cpp/.
AWS SDK for Go	A software development kit for integrating your Go application with the full suite of AWS services. See Also https://aws.amazon.com/sdk-for-go/.
AWS SDK for Java	A software development kit that provides Java APIs for many AWS services including Amazon S3 (p. 113), Amazon EC2 (p. 111), Amazon DynamoDB (p. 110), and more. The single, downloadable package includes the AWS Java library, code samples, and documentation. See Also https://aws.amazon.com/sdk-for-java/.
AWS SDK for JavaScript in the Browser	A software development kit for accessing AWS services from JavaScript code running in the browser. Authenticate users through Facebook, Google, or Login with Amazon using web identity federation. Store application data in Amazon DynamoDB (p. 110), and save user files to Amazon S3 (p. 113). See Also http://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/.
AWS SDK for JavaScript in Node.js	A software development kit for accessing AWS services from JavaScript in Node.js. The SDK provides JavaScript objects for AWS services, including Amazon S3 (p. 113), Amazon EC2 (p. 111), Amazon DynamoDB (p. 110), and Amazon Simple Workflow Service (Amazon SWF) (p. 113) . The single, downloadable package includes the AWS JavaScript library and documentation. See Also http://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/.
AWS SDK for .NET	A software development kit that provides .NET API actions for AWS services including Amazon S3 (p. 113), Amazon EC2 (p. 111), IAM (p. 118), and more. You can download the SDK as multiple service-specific packages on NuGet. See Also https://aws.amazon.com/sdk-for-net/.
AWS SDK for PHP	A software development kit and open-source PHP library for integrating your PHP application with AWS services like Amazon S3 (p. 113), Amazon Glacier (p. 111), and Amazon DynamoDB (p. 110). See Also https://aws.amazon.com/sdk-for-php/.
AWS SDK for Python (Boto)	A software development kit for using Python to access AWS services like Amazon EC2 (p. 111), Amazon EMR (p. 111), Auto Scaling (p. 115), Amazon Kinesis (p. 112), AWS Lambda (p. 118), and more. See Also http://boto.readthedocs.org/en/latest/.
AWS SDK for Ruby	A software development kit for accessing AWS services from Ruby. The SDK provides Ruby classes for many AWS services including Amazon S3 (p. 113), Amazon EC2 (p. 111), Amazon DynamoDB (p. 110). and more. The single, downloadable package includes the AWS Ruby Library and documentation. See Also https://aws.amazon.com/sdk-for-ruby/.
AWS Security Token Service (AWS STS)	A web service for requesting temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) (p. 118) users or for users that you authenticate (federated users (p. 132)). See Also https://aws.amazon.com/iam/.
AWS Service Catalog	A web service that helps organizations create and manage catalogs of IT services that are approved for use on AWS. These IT services can include everything from

	virtual machine images, servers, software, and databases to complete multitier application architectures. See Also https://aws.amazon.com/servicecatalog/.
AWS Step Functions	A web service that coordinates the components of distributed applications as a series of steps in a visual workflow. See Also https://aws.amazon.com/step-functions/.
AWS Storage Gateway	A web service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS's storage infrastructure. See Also https://aws.amazon.com/storagegateway/.
AWS Toolkit for Eclipse	An open-source plug-in for the Eclipse Java IDE that makes it easier for developers to develop, debug, and deploy Java applications using Amazon Web Services. See Also https://aws.amazon.com/eclipse/.
AWS Toolkit for Visual Studio	An extension for Microsoft Visual Studio that helps developers develop, debug, and deploy .NET applications using Amazon Web Services. See Also https://aws.amazon.com/visualstudio/.
AWS Tools for Windows PowerShell	A set of PowerShell cmdlets to help developers and administrators manage their AWS services from the Windows PowerShell scripting environment. See Also https://aws.amazon.com/powershell/.
AWS Tools for Microsoft Visual Studio Team Services	Provides tasks you can use in build and release definitions in VSTS to interact with AWS services. See Also https://aws.amazon.com/vsts/.
AWS Trusted Advisor	A web service that inspects your AWS environment and makes recommendations for saving money, improving system availability and performance, and helping to close security gaps. See Also https://aws.amazon.com/premiumsupport/trustedadvisor/.
AWS VPN CloudHub	Enables secure communication between branch offices using a simple hub-and-spoke model, with or without a VPC (p. 158).
AWS WAF	A web application firewall service that controls access to content by allowing or blocking web requests based on criteria that you specify, such as header values or the IP addresses that the requests originate from. AWS WAF helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. See Also https://aws.amazon.com/waf/.
AWS X-Ray	A web service that collects data about requests that your application serves, and provides tools you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimization. See Also https://aws.amazon.com/xray/.

B

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

basic monitoring	Monitoring of AWS provided metrics derived at a 5-minute frequency.

batch	See document batch.
BGP ASN	Border Gateway Protocol Autonomous System Number. A unique identifier for a network, for use in BGP routing. Amazon EC2 (p. 111) supports all 2-byte ASN numbers in the range of 1 – 65335, with the exception of 7224, which is reserved.
batch prediction	Amazon Machine Learning: An operation that processes multiple input data observations at one time (asynchronously). Unlike real-time predictions, batch predictions are not available until all predictions have been processed. See Also real-time predictions.
billing	See AWS Billing and Cost Management.
binary attribute	Amazon Machine Learning: An attribute for which one of two possible values is possible. Valid positive values are 1, y, yes, t, and true answers. Valid negative values are 0, n, no, f, and false. Amazon Machine Learning outputs 1 for positive values and 0 for negative values. See Also attribute.
binary classification model	Amazon Machine Learning: A machine learning model that predicts the answer to questions where the answer can be expressed as a binary variable. For example, questions with answers of “1” or “0”, “yes” or “no”, “will click” or “will not click” are questions that have binary answers. The result for a binary classification model is always either a “1” (for a “true” or affirmative answers) or a “0” (for a “false” or negative answers).
blacklist	A list of IP addresses, email addresses, or domains that an internet service provider (p. 135) suspects to be the source of spam (p. 152). The ISP blocks incoming email from these addresses or domains.
block	A data set. Amazon EMR (p. 111) breaks large amounts of data into subsets. Each subset is called a data block. Amazon EMR assigns an ID to each block and uses a hash table to keep track of block processing.
block device	A storage device that supports reading and (optionally) writing data in fixed-size blocks, sectors, or clusters.
block device mapping	A mapping structure for every AMI (p. 112) and instance (p. 135) that specifies the block devices attached to the instance.
blue/green deployment	AWS CodeDeploy: A deployment method in which the instances in a deployment group (the original environment) are replaced by a different set of instances (the replacement environment).
bootstrap action	A user-specified default or custom action that runs a script or an application on all nodes of a job flow before Hadoop (p. 133) starts.
Border Gateway Protocol Autonomous System Number	See BGP ASN.
bounce	A failed email delivery attempt.
breach	Auto Scaling (p. 115): The condition in which a user-set threshold (upper or lower boundary) is passed. If the duration of the breach is significant, as set by a breach duration parameter, it can possibly start a scaling activity (p. 149).
bucket	Amazon Simple Storage Service (Amazon S3) (p. 113): A container for stored objects. Every object is contained in a bucket. For example, if the object named `photos/puppy.jpg` is stored in the `johnsmith` bucket, then authorized users can access the object with the URL `http://johnsmith.s3.amazonaws.com/photos/puppy.jpg`.

bucket owner	The person or organization that owns a bucket (p. 121) in Amazon S3 (p. 113). Just as Amazon is the only owner of the domain name Amazon.com, only one person or organization can own a bucket.
bundling	A commonly used term for creating an Amazon Machine Image (AMI) (p. 112). It specifically refers to creating instance store-backed AMI (p. 135)s.

C

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

cache cluster	A logical cache distributed over multiple cache node (p. 122)s. A cache cluster can be set up with a specific number of cache nodes.
cache cluster identifier	Customer-supplied identifier for the cache cluster that must be unique for that customer in an AWS Region (p. 146).
cache engine version	The version of the Memcached service that is running on the cache node.
cache node	A fixed-size chunk of secure, network-attached RAM. Each cache node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of cache nodes are supported, each with varying amounts of associated memory.
cache node type	An EC2 instance (p. 129) type used to run the cache node.
cache parameter group	A container for cache engine parameter values that can be applied to one or more cache clusters.
cache security group	A group maintained by ElastiCache that combines ingress authorizations to cache nodes for hosts belonging to Amazon EC2 (p. 111) security group (p. 150)s specified through the console or the API or command line tools.
canned access policy	A standard access control policy that you can apply to a bucket (p. 121) or object. Options include: private, public-read, public-read-write, and authenticated-read.
canonicalization	The process of converting data into a standard format that a service such as Amazon S3 (p. 113) can recognize.
capacity	The amount of available compute size at a given time. Each Auto Scaling group (p. 115) is defined with a minimum and maximum compute size. A scaling activity (p. 149) increases or decreases the capacity within the defined minimum and maximum values.
cartesian product processor	A processor that calculates a cartesian product. Also known as a *cartesian data processor*.
cartesian product	A mathematical operation that returns a product from multiple sets.
CDN	See content delivery network (CDN).
certificate	A credential that some AWS products use to authenticate AWS account (p. 109)s and users. Also known as an X.509 certificate (p. 158) . The certificate is paired with a private key.

chargeable resources	Features or services whose use incurs fees. Although some AWS products are free, others include charges. For example, in an AWS CloudFormation (p. 116) stack (p. 152), AWS resource (p. 147)s that have been created incur charges. The amount charged depends on the usage load. Use the Amazon Web Services Simple Monthly Calculator at http://calculator.s3.amazonaws.com/calc5.html to estimate your cost prior to creating instances, stacks, or other resources.
CIDR block	Classless Inter-Domain Routing. An internet protocol address allocation and route aggregation methodology. See Also Classless Inter-Domain Routing in Wikipedia.
ciphertext	Information that has been encrypted (p. 130), as opposed to plaintext (p. 143), which is information that has not.
ClassicLink	A feature for linking an EC2-Classic instance (p. 135) to a VPC (p. 158), allowing your EC2-Classic instance to communicate with VPC instances using private IP addresses. See Also link to VPC, unlink from VPC.
classification	In machine learning, a type of problem that seeks to place (classify) a data sample into a single category or “class.” Often, classification problems are modeled to choose one category (class) out of two. These are binary classification problems. Problems where more than two categories (classes) are available are called "multiclass classification" problems. See Also binary classification model, multiclass classification model.
cloud service provider	A company that provides subscribers with access to internet-hosted computing, storage, and software services.
CloudHub	See AWS VPN CloudHub.
CLI	See AWS Command Line Interface (AWS CLI).
cluster	A logical grouping of container instance (p. 124)s that you can place task (p. 155)s on. Amazon Elasticsearch Service (Amazon ES) (p. 111): A logical grouping of one or more data nodes, optional dedicated master nodes, and storage required to run Amazon Elasticsearch Service (Amazon ES) and operate your Amazon ES domain. See Also data node, dedicated master node, node.
cluster compute instance	A type of instance (p. 135) that provides a great amount of CPU power coupled with increased networking performance, making it well suited for High Performance Compute (HPC) applications and other demanding network-bound applications.
cluster placement group	A logical cluster compute instance (p. 123) grouping to provide lower latency and high-bandwidth connectivity between the instance (p. 135)s.
cluster status	Amazon Elasticsearch Service (Amazon ES) (p. 111): An indicator of the health of a cluster. A status can be green, yellow, or red. At the shard level, green means that all shards are allocated to nodes in a cluster, yellow means that the primary shard is allocated but the replica shards are not, and red means that the primary and replica shards of at least one index are not allocated. The shard status determines the index status, and the index status determines the cluster status.
CMK	See customer master key (CMK).
CNAME	Canonical Name Record. A type of resource record (p. 147) in the Domain Name System (DNS) that specifies that the domain name is an alias of another,

	canonical domain name. More simply, it is an entry in a DNS table that lets you alias one fully qualified domain name to another.
complaint	The event in which a recipient (p. 146) who does not want to receive an email message clicks "Mark as Spam" within the email client, and the internet service provider (p. 135) sends a notification to Amazon SES (p. 113).
compound query	Amazon CloudSearch (p. 110): A search request that specifies multiple search criteria using the Amazon CloudSearch structured search syntax.
condition	IAM (p. 118): Any restriction or detail about a permission. The condition is *D* in the statement "A has permission to do B to C where D applies." AWS WAF (p. 120): A set of attributes that AWS WAF searches for in web requests to AWS resource (p. 147)s such as Amazon CloudFront (p. 110) distributions. Conditions can include values such as the IP addresses that web requests originate from or values in request headers. Based on the specified conditions, you can configure AWS WAF to allow or block web requests to AWS resources.
conditional parameter	See mapping.
configuration API	Amazon CloudSearch (p. 110): The API call that you use to create, configure, and manage search domains.
configuration template	A series of key–value pairs that define parameters for various AWS products so that AWS Elastic Beanstalk (p. 117) can provision them for an environment.
consistency model	The method a service uses to achieve high availability. For example, it could involve replicating data across multiple servers in a data center. See Also eventual consistency.
console	See AWS Management Console.
consolidated billing	A feature of the AWS Organizations service for consolidating payment for multiple AWS accounts. You create an organization that contains your AWS accounts, and you use the master account of your organization to pay for all member accounts. You can see a combined view of AWS costs that are incurred by all accounts in your organization, and you can get detailed cost reports for individual accounts.
container	A Linux container that was created from a Docker image as part of a task (p. 155).
container definition	Specifies which Docker image (p. 128) to use for a container (p. 124), how much CPU and memory the container is allocated, and more options. The container definition is included as part of a task definition (p. 155).
container instance	An EC2 instance (p. 129) that is running the Amazon Elastic Container Service (Amazon ECS) (p. 111) agent and has been registered into a cluster (p. 123). Amazon ECS task (p. 155)s are placed on active container instances.
container registry	Stores, manages, and deploys Docker image (p. 128)s.
content delivery network (CDN)	A web service that speeds up distribution of your static and dynamic web content —such as .html, .css, .js, media files, and image files—to your users by using a worldwide network of data centers. When a user requests your content, the request is routed to the data center that provides the lowest latency (time delay). If the content is already in the location with the lowest latency, the CDN delivers it immediately. If not, the CDN retrieves it from an origin that you specify (for

	example, a web server or an Amazon S3 bucket). With some CDNs, you can help secure your content by configuring an HTTPS connection between users and data centers, and between data centers and your origin. Amazon CloudFront is an example of a CDN.
continuous delivery	A software development practice in which code changes are automatically built, tested, and prepared for a release to production. See Also https://aws.amazon.com/devops/continuous-delivery/.
continuous integration	A software development practice in which developers regularly merge code changes into a central repository, after which automated builds and tests are run. See Also https://aws.amazon.com/devops/continuous-integration/.
cooldown period	Amount of time during which Auto Scaling (p. 115) does not allow the desired size of the Auto Scaling group (p. 115) to be changed by any other notification from an Amazon CloudWatch (p. 110) alarm (p. 109).
core node	An EC2 instance (p. 129) that runs Hadoop (p. 133) map and reduce tasks and stores data using the Hadoop Distributed File System (HDFS). Core nodes are managed by the master node (p. 139), which assigns Hadoop tasks to nodes and monitors their status. The EC2 instances you assign as core nodes are capacity that must be allotted for the entire job flow run. Because core nodes store data, you can't remove them from a job flow. However, you can add more core nodes to a running job flow. Core nodes run both the DataNodes and TaskTracker Hadoop daemons.
corpus	Amazon CloudSearch (p. 110): A collection of data that you want to search.
credential helper	AWS CodeCommit (p. 116): A program that stores credentials for repositories and supplies them to Git when making connections to those repositories. The AWS CLI (p. 116) includes a credential helper that you can use with Git when connecting to AWS CodeCommit repositories.
credentials	Also called *access credentials* or *security credentials*. In authentication and authorization, a system uses credentials to identify who is making a call and whether to allow the requested access. In AWS, these credentials are typically the access key ID (p. 108) and the secret access key (p. 150).
cross-account access	The process of permitting limited, controlled use of resource (p. 147)s in one AWS account (p. 109) by a user in another AWS account. For example, in AWS CodeCommit (p. 116) and AWS CodeDeploy (p. 116) you can configure cross-account access so that a user in AWS account A can access an AWS CodeCommit repository created by account B. Or a pipeline in AWS CodePipeline (p. 116) created by account A can use AWS CodeDeploy resources created by account B. In IAM (p. 118) you use a role (p. 148) to delegate (p. 127) temporary access to a user (p. 156) in one account to resources in another.
cross-Region replication	A client-side solution for maintaining identical copies of Amazon DynamoDB (p. 110) tables across different AWS Region (p. 146)s, in near real time.
customer gateway	A router or software application on your side of a VPN tunnel that is managed by Amazon VPC (p. 113). The internal interfaces of the customer gateway are attached to one or more devices in your home network. The external interface is attached to the virtual private gateway (p. 157) across the VPN tunnel.
customer managed policy	An IAM (p. 118) managed policy (p. 138) that you create and manage in your AWS account (p. 109).

customer master key (CMK) — The fundamental resource (p. 147) that AWS Key Management Service (AWS KMS) (p. 118) manages. CMKs can be either customer managed keys or AWS managed keys. Use CMKs inside AWS KMS to encrypt (p. 130) or decrypt up to 4 kilobytes of data directly or to encrypt generated data keys, which are then used to encrypt or decrypt larger amounts of data outside of the service.

D

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

dashboard — See service health dashboard.

data consistency — A concept that describes when data is written or updated successfully and all copies of the data are updated in all AWS Region (p. 146)s. However, it takes time for the data to propagate to all storage locations. To support varied application requirements, Amazon DynamoDB (p. 110) supports both eventually consistent and strongly consistent reads.
See Also eventual consistency, eventually consistent read, strongly consistent read.

data node — Amazon Elasticsearch Service (Amazon ES) (p. 111): An Elasticsearch instance that holds data and responds to data upload requests.
See Also dedicated master node, node.

data schema — See schema.

data source — The database, file, or repository that provides information required by an application or database. For example, in AWS OpsWorks (p. 118), valid data sources include an instance (p. 135) for a stack's MySQL layer or a stack's Amazon RDS (p. 113) service layer. In Amazon Redshift (p. 113), valid data sources include text files in an Amazon S3 (p. 113) bucket (p. 121), in an Amazon EMR (p. 111) cluster, or on a remote host that a cluster can access through an SSH connection.
See Also datasource.

database engine — The database software and version running on the DB instance (p. 126).

database name — The name of a database hosted in a DB instance (p. 126). A DB instance can host multiple databases, but databases hosted by the same DB instance must each have a unique name within that instance.

datasource — Amazon Machine Learning (p. 112): An object that contains metadata about the input data. Amazon ML reads the input data, computes descriptive statistics on its attributes, and stores the statistics—along with a schema and other information—as part of the datasource object. Amazon ML uses datasources to train and evaluate a machine learning model and generate batch predictions.
See Also data source.

DB compute class — Size of the database compute platform used to run the instance.

DB instance — An isolated database environment running in the cloud. A DB instance can contain multiple user-created databases.

DB instance identifier — User-supplied identifier for the DB instance. The identifier must be unique for that user in an AWS Region (p. 146).

DB parameter group	A container for database engine parameter values that apply to one or more DB instance (p. 126)s.
DB security group	A method that controls access to the DB instance (p. 126). By default, network access is turned off to DB instances. After ingress is configured for a security group (p. 150), the same rules apply to all DB instances associated with that group.
DB snapshot	A user-initiated point backup of a DB instance (p. 126).
Dedicated Host	A physical server with EC2 instance (p. 129) capacity fully dedicated to a user.
Dedicated Instance	An instance (p. 135) that is physically isolated at the host hardware level and launched within a VPC (p. 158).
dedicated master node	Amazon Elasticsearch Service (Amazon ES) (p. 111): An Elasticsearch instance that performs cluster management tasks, but does not hold data or respond to data upload requests. Amazon Elasticsearch Service (Amazon ES) uses dedicated master nodes to increase cluster stability. See Also data node, node.
Dedicated Reserved Instance	An option that you purchase to guarantee that sufficient capacity will be available to launch Dedicated Instance (p. 127)s into a VPC (p. 158).
delegation	Within a single AWS account (p. 109): Giving AWS user (p. 156)s access to resource (p. 147)s in your AWS account. Between two AWS accounts: Setting up a trust between the account that owns the resource (the trusting account), and the account that contains the users that need to access the resource (the trusted account). See Also trust policy.
delete marker	An object with a key and version ID, but without content. Amazon S3 (p. 113) inserts delete markers automatically into versioned bucket (p. 121)s when an object is deleted.
deliverability	The likelihood that an email message will arrive at its intended destination.
deliveries	The number of email messages, sent through Amazon SES (p. 113), that were accepted by an internet service provider (p. 135) for delivery to recipient (p. 146)s over a period of time.
deny	The result of a policy (p. 143) statement that includes deny as the effect, so that a specific action or actions are expressly forbidden for a user, group, or role. Explicit deny take precedence over explicit allow (p. 109).
deployment configuration	AWS CodeDeploy (p. 116): A set of deployment rules and success and failure conditions used by the service during a deployment.
deployment group	AWS CodeDeploy (p. 116): A set of individually tagged instance (p. 135)s, EC2 instance (p. 129)s in Auto Scaling group (p. 115)s, or both.
detailed monitoring	Monitoring of AWS provided metrics derived at a 1-minute frequency.
Description property	A property added to parameters, resource (p. 147)s, resource properties, mappings, and outputs to help you to document AWS CloudFormation (p. 116) template elements.
dimension	A name–value pair (for example, InstanceType=m1.small, or EngineName=mysql), that contains additional information to identify a metric.

discussion forums	A place where AWS users can post technical questions and feedback to help accelerate their development efforts and to engage with the AWS community. The discussion forums are located at https://aws.amazon.com/forums/.
distribution	A link between an origin server (such as an Amazon S3 (p. 113) bucket (p. 121)) and a domain name, which CloudFront (p. 110) automatically assigns. Through this link, CloudFront identifies the object you have stored in your origin server (p. 142).
DKIM	DomainKeys Identified Mail. A standard that email senders use to sign their messages. ISPs use those signatures to verify that messages are legitimate. For more information, see http://www.dkim.org.
DNS	See Domain Name System.
Docker image	A layered file system template that is the basis of a Docker container (p. 124). Docker images can comprise specific operating systems or applications.
document	Amazon CloudSearch (p. 110): An item that can be returned as a search result. Each document has a collection of fields that contain the data that can be searched or returned. The value of a field can be either a string or a number. Each document must have a unique ID and at least one field.
document batch	Amazon CloudSearch (p. 110): A collection of add and delete document operations. You use the document service API to submit batches to update the data in your search domain.
document service API	Amazon CloudSearch (p. 110): The API call that you use to submit document batches to update the data in a search domain.
document service endpoint	Amazon CloudSearch (p. 110): The URL that you connect to when sending document updates to an Amazon CloudSearch domain. Each search domain has a unique document service endpoint that remains the same for the life of the domain.
domain	Amazon Elasticsearch Service (Amazon ES) (p. 111): The hardware, software, and data exposed by Amazon Elasticsearch Service (Amazon ES) endpoints. An Amazon ES domain is a service wrapper around an Elasticsearch cluster. An Amazon ES domain encapsulates the engine instances that process Amazon ES requests, the indexed data that you want to search, snapshots of the domain, access policies, and metadata. See Also cluster, Elasticsearch.
Domain Name System	A service that routes internet traffic to websites by translating friendly domain names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other.
Donation button	An HTML-coded button to provide an easy and secure way for US-based, IRS-certified 501(c)3 nonprofit organizations to solicit donations.
DynamoDB stream	An ordered flow of information about changes to items in anAmazon DynamoDB (p. 110) table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table. See Also Amazon DynamoDB Streams.

E

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141)

| P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

EBS	See Amazon Elastic Block Store (Amazon EBS).
EC2	See Amazon Elastic Compute Cloud (Amazon EC2).
EC2 compute unit	An AWS standard for compute CPU and memory. You can use this measure to evaluate the CPU capacity of different EC2 instance (p. 129) types.
EC2 instance	A compute instance (p. 135) in the Amazon EC2 (p. 111) service. Other AWS services use the term *EC2 instance* to distinguish these instances from other types of instances they support.
ECR	See Amazon Elastic Container Registry (Amazon ECR).
ECS	See Amazon Elastic Container Service (Amazon ECS).
edge location	A site that CloudFront (p. 110) uses to cache copies of your content for faster delivery to users at any location.
EFS	See Amazon Elastic File System (Amazon EFS).
Elastic	A company that provides open-source solutions—including Elasticsearch, Logstash, Kibana, and Beats—that are designed to take data from any source and search, analyze, and visualize it in real time. Amazon Elasticsearch Service (Amazon ES) is an AWS managed service for deploying, operating, and scaling Elasticsearch in the AWS Cloud. See Also Amazon Elasticsearch Service (Amazon ES), Elasticsearch.
Elastic Block Store	See Amazon Elastic Block Store (Amazon EBS).
Elastic IP address	A fixed (static) IP address that you have allocated in Amazon EC2 (p. 111) or Amazon VPC (p. 113) and then attached to an instance (p. 135). Elastic IP addresses are associated with your account, not a specific instance. They are *elastic* because you can easily allocate, attach, detach, and free them as your needs change. Unlike traditional static IP addresses, Elastic IP addresses allow you to mask instance or Availability Zone (p. 115) failures by rapidly remapping your public IP addresses to another instance.
Elastic Load Balancing	A web service that improves an application's availability by distributing incoming traffic between two or more EC2 instance (p. 129)s. See Also https://aws.amazon.com/elasticloadbalancing.
elastic network interface	An additional network interface that can be attached to an instance (p. 135). Elastic network interfaces include a primary private IP address, one or more secondary private IP addresses, an elastic IP address (optional), a MAC address, membership in specified security group (p. 150)s, a description, and a source/destination check flag. You can create an elastic network interface, attach it to an instance, detach it from an instance, and attach it to another instance.
Elasticsearch	An open source, real-time distributed search and analytics engine used for full-text search, structured search, and analytics. Elasticsearch was developed by the Elastic company. Amazon Elasticsearch Service (Amazon ES) is an AWS managed service for deploying, operating, and scaling Elasticsearch in the AWS Cloud. See Also Amazon Elasticsearch Service (Amazon ES), Elastic.
EMR	See Amazon EMR (Amazon EMR).

encrypt	To use a mathematical algorithm to make data unintelligible to unauthorized user (p. 156)s while allowing authorized users a method (such as a key or password) to convert the altered data back to its original state.
encryption context	A set of key–value pairs that contains additional information associated with AWS Key Management Service (AWS KMS) (p. 118)–encrypted information.
endpoint	A URL that identifies a host and port as the entry point for a web service. Every web service request contains an endpoint. Most AWS products provide endpoints for a Region to enable faster connectivity. Amazon ElastiCache (p. 111): The DNS name of a cache node (p. 122). Amazon RDS (p. 113): The DNS name of a DB instance (p. 126). AWS CloudFormation (p. 116): The DNS name or IP address of the server that receives an HTTP request.
endpoint port	Amazon ElastiCache (p. 111): The port number used by a cache node (p. 122). Amazon RDS (p. 113): The port number used by a DB instance (p. 126).
envelope encryption	The use of a master key and a data key to algorithmically protect data. The master key is used to encrypt and decrypt the data key and the data key is used to encrypt and decrypt the data itself.
environment	AWS Elastic Beanstalk (p. 117): A specific running instance of an application (p. 114). The application has a CNAME and includes an application version and a customizable configuration (which is inherited from the default container type). AWS CodeDeploy (p. 116): Instances in a deployment group in a blue/green deployment. At the start of a blue/green deployment, the deployment group is made up of instances in the original environment. At the end of the deployment, the deployment group is made up of instances in the replacement environment.
environment configuration	A collection of parameters and settings that define how an environment and its associated resources behave.
ephemeral store	See instance store.
epoch	The date from which time is measured. For most Unix environments, the epoch is January 1, 1970.
ETL	See extract, transform, and load (ETL).
evaluation	Amazon Machine Learning: The process of measuring the predictive performance of a machine learning (ML) model. Also a machine learning object that stores the details and result of an ML model evaluation.
evaluation datasource	The data that Amazon Machine Learning uses to evaluate the predictive accuracy of a machine learning model.
eventual consistency	The method through which AWS products achieve high availability, which involves replicating data across multiple servers in Amazon's data centers. When data is written or updated and `Success` is returned, all copies of the data are updated. However, it takes time for the data to propagate to all storage locations. The data will eventually be consistent, but an immediate read might not show the change. Consistency is usually reached within seconds.

	See Also data consistency, eventually consistent read, strongly consistent read.
eventually consistent read	A read process that returns data from only one region and might not show the most recent write information. However, if you repeat your read request after a short time, the response should eventually return the latest data. See Also data consistency, eventual consistency, strongly consistent read.
eviction	The deletion by CloudFront (p. 110) of an object from an edge location (p. 129) before its expiration time. If an object in an edge location isn't frequently requested, CloudFront might evict the object (remove the object before its expiration date) to make room for objects that are more popular.
exbibyte	A contraction of exa binary byte, an exbibyte is 2^60 or 1,152,921,504,606,846,976 bytes. An exabyte (EB) is 10^18 or 1,000,000,000,000,000,000 bytes. 1,024 EiB is a zebibyte (p. 158).
expiration	For CloudFront (p. 110) caching, the time when CloudFront stops responding to user requests with an object. If you don't use headers or CloudFront distribution (p. 128) settings to specify how long you want objects to stay in an edge location (p. 129), the objects expire after 24 hours. The next time a user requests an object that has expired, CloudFront forwards the request to the origin (p. 142).
explicit launch permission	An Amazon Machine Image (AMI) (p. 112) launch permission granted to a specific AWS account (p. 109).
exponential backoff	A strategy that incrementally increases the wait between retry attempts in order to reduce the load on the system and increase the likelihood that repeated requests will succeed. For example, client applications might wait up to 400 milliseconds before attempting the first retry, up to 1600 milliseconds before the second, up to 6400 milliseconds (6.4 seconds) before the third, and so on.
expression	Amazon CloudSearch (p. 110): A numeric expression that you can use to control how search hits are sorted. You can construct Amazon CloudSearch expressions using numeric fields, other rank expressions, a document's default relevance score, and standard numeric operators and functions. When you use the `sort` option to specify an expression in a search request, the expression is evaluated for each search hit and the hits are listed according to their expression values.
extract, transform, and load (ETL)	A process that is used to integrate data from multiple sources. Data is collected from sources (extract), converted to an appropriate format (transform), and written to a target data store (load) for purposes of analysis and querying. ETL tools combine these three functions to consolidate and move data from one environment to another. AWS Glue (p. 117) is a fully managed ETL service for discovering and organizing data, transforming it, and making it available for search and analytics.

F

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

facet	Amazon CloudSearch (p. 110): An index field that represents a category that you want to use to refine and filter search results.

facet enabled	Amazon CloudSearch (p. 110): An index field option that enables facet information to be calculated for the field.
FBL	See feedback loop.
feature transformation	Amazon Machine Learning: The machine learning process of constructing more predictive input representations or “features” from the raw input variables to optimize a machine learning model’s ability to learn and generalize. Also known as *data transformation* or *feature engineering*.
federated identity management	Allows individuals to sign in to different networks or services, using the same group or personal credentials to access data across all networks. With identity federation in AWS, external identities (federated users) are granted secure access to resource (p. 147)s in an AWS account (p. 109) without having to create IAM user (p. 156)s. These external identities can come from a corporate identity store (such as LDAP or Windows Active Directory) or from a third party (such as Login with Amazon, Facebook, or Google). AWS federation also supports SAML 2.0.
federated user	See federated identity management.
federation	See federated identity management.
feedback loop	The mechanism by which a mailbox provider (for example, an internet service provider (p. 135)) forwards a recipient (p. 146)'s complaint (p. 124) back to the sender (p. 150).
field weight	The relative importance of a text field in a search index. Field weights control how much matches in particular text fields affect a document's relevance score.
filter	A criterion that you specify to limit the results when you list or describe your Amazon EC2 (p. 111) resource (p. 147)s.
filter query	A way to filter search results without affecting how the results are scored and sorted. Specified with the Amazon CloudSearch (p. 110) `fq` parameter.
FIM	See federated identity management.
Firehose	See Amazon Kinesis Data Firehose.
format version	See template format version.
forums	See discussion forums.
function	See intrinsic function.
fuzzy search	A simple search query that uses approximate string matching (fuzzy matching) to correct for typographical errors and misspellings.

G

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

geospatial search	A search query that uses locations specified as a latitude and longitude to determine matches and sort the results.

gibibyte	A contraction of giga binary byte, a gibibyte is 2^30 or 1,073,741,824 bytes. A gigabyte (GB) is 10^9 or 1,000,000,000 bytes. 1,024 GiB is a tebibyte (p. 155).
GitHub	A web-based repository that uses Git for version control.
global secondary index	An index with a partition key and a sort key that can be different from those on the table. A global secondary index is considered global because queries on the index can span all of the data in a table, across all partitions. See Also local secondary index.
grant	AWS Key Management Service (AWS KMS) (p. 118): A mechanism for giving AWS principal (p. 144)s long-term permissions to use customer master key (CMK) (p. 126)s.
grant token	A type of identifier that allows the permissions in a grant (p. 133) to take effect immediately.
ground truth	The observations used in the machine learning (ML) model training process that include the correct value for the target attribute. To train an ML model to predict house sales prices, the input observations would typically include prices of previous house sales in the area. The sale prices of these houses constitute the ground truth.
group	A collection of IAM (p. 118) user (p. 156)s. You can use IAM groups to simplify specifying and managing permissions for multiple users.

H

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

Hadoop	Software that enables distributed processing for big data by using clusters and simple programming models. For more information, see http://hadoop.apache.org.
hard bounce	A persistent email delivery failure such as "mailbox does not exist."
hardware VPN	A hardware-based IPsec VPN connection over the internet.
health check	A system call to check on the health status of each instance in an Auto Scaling (p. 115) group.
high-quality email	Email that recipients find valuable and want to receive. Value means different things to different recipients and can come in the form of offers, order confirmations, receipts, newsletters, etc.
highlights	Amazon CloudSearch (p. 110): Excerpts returned with search results that show where the search terms appear within the text of the matching documents.
highlight enabled	Amazon CloudSearch (p. 110): An index field option that enables matches within the field to be highlighted.
hit	A document that matches the criteria specified in a search request. Also referred to as a *search result*.
HMAC	Hash-based Message Authentication Code. A specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key. You can use it to verify both the data integrity and

	the authenticity of a message at the same time. AWS calculates the HMAC using a standard, cryptographic hash algorithm, such as SHA-256.
hosted zone	A collection of resource record (p. 147) sets that Amazon Route 53 (p. 113) hosts. Like a traditional DNS zone file, a hosted zone represents a collection of records that are managed together under a single domain name.
HVM virtualization	Hardware Virtual Machine virtualization. Allows the guest VM to run as though it is on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance. See Also PV virtualization.

I

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

IAM	See AWS Identity and Access Management (IAM).
IAM group	See group.
IAM policy simulator	See policy simulator.
IAM role	See role.
IAM user	See user.
Identity and Access Management	See AWS Identity and Access Management (IAM).
identity provider (IdP)	An IAM (p. 118) entity that holds metadata about external identity providers.
IdP	See identity provider (IdP) .
image	See Amazon Machine Image (AMI).
import/export station	A machine that uploads or downloads your data to or from Amazon S3 (p. 113).
import log	A report that contains details about how AWS Import/Export (p. 118) processed your data.
in-place deployment	AWS CodeDeploy: A deployment method in which the application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated. You can choose to use a load balancer so each instance is deregistered during its deployment and then restored to service after the deployment is complete.
index	See search index.
index field	A name–value pair that is included in an Amazon CloudSearch (p. 110) domain's index. An index field can contain text or numeric data, dates, or a location.
indexing options	Configuration settings that define an Amazon CloudSearch (p. 110) domain's index fields, how document data is mapped to those index fields, and how the index fields can be used.
inline policy	An IAM (p. 118) policy (p. 143) that is embedded in a single IAM user (p. 156), group (p. 133), or role (p. 148).

input data	Amazon Machine Learning: The observations that you provide to Amazon Machine Learning to train and evaluate a machine learning model and generate predictions.
instance	A copy of an Amazon Machine Image (AMI) (p. 112) running as a virtual server in the AWS cloud.
instance family	A general instance type (p. 135) grouping using either storage or CPU capacity.
instance group	A Hadoop (p. 133) cluster contains one master instance group that contains one master node (p. 139), a core instance group containing one or more core node (p. 125) and an optional task node (p. 155) instance group, which can contain any number of task nodes.
instance profile	A container that passes IAM (p. 118) role (p. 148) information to an EC2 instance (p. 129) at launch.
instance store	Disk storage that is physically attached to the host computer for an EC2 instance (p. 129), and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.
instance store-backed AMI	A type of Amazon Machine Image (AMI) (p. 112) whose instance (p. 135)s use an instance store (p. 135) volume (p. 157) as the root device. Compare this with instances launched from Amazon EBS (p. 110)-backed AMIs, which use an Amazon EBS volume as the root device.
instance type	A specification that defines the memory, CPU, storage capacity, and usage cost for an instance (p. 135). Some instance types are designed for standard applications, whereas others are designed for CPU-intensive, memory-intensive applications, and so on.
internet gateway	Connects a network to the internet. You can route traffic for IP addresses outside your VPC (p. 158) to the internet gateway.
internet service provider	A company that provides subscribers with access to the internet. Many ISPs are also mailbox provider (p. 138)s. Mailbox providers are sometimes referred to as ISPs, even if they only provide mailbox services.
intrinsic function	A special action in a AWS CloudFormation (p. 116) template that assigns values to properties not available until runtime. These functions follow the format *Fn::Attribute*, such as `Fn::GetAtt`. Arguments for intrinsic functions can be parameters, pseudo parameters, or the output of other intrinsic functions.
IP address	A numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). All EC2 instance (p. 129)s are assigned two IP addresses at launch, which are directly mapped to each other through network address translation (NAT (p. 140)): a private IP address (following RFC 1918) and a public IP address. Instances launched in a VPC (p. 113) are assigned only a private IP address. Instances launched in your default VPC are assigned both a private IP address and a public IP address.
IP match condition	AWS WAF (p. 120): An attribute that specifies the IP addresses or IP address ranges that web requests originate from. Based on the specified IP addresses, you can configure AWS WAF to allow or block web requests to AWS resource (p. 147)s such as Amazon CloudFront (p. 110) distributions.
ISP	See internet service provider.
issuer	The person who writes a policy (p. 143) to grant permissions to a resource (p. 147). The issuer (by definition) is always the resource owner. AWS

does not permit Amazon SQS (p. 113) users to create policies for resources they don't own. If John is the resource owner, AWS authenticates John's identity when he submits the policy he's written to grant permissions for that resource.

item — A group of attributes that is uniquely identifiable among all of the other items. Items in Amazon DynamoDB (p. 110) are similar in many ways to rows, records, or tuples in other database systems.

J

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

job flow — Amazon EMR (p. 111): One or more step (p. 153)s that specify all of the functions to be performed on the data.

job ID — A five-character, alphanumeric string that uniquely identifies an AWS Import/Export (p. 118) storage device in your shipment. AWS issues the job ID in response to a `CREATE JOB` email command.

job prefix — An optional string that you can add to the beginning of an AWS Import/Export (p. 118) log file name to prevent collisions with objects of the same name.
See Also key prefix.

JSON — JavaScript Object Notation. A lightweight data interchange format. For information about JSON, see http://www.json.org/.

junk folder — The location where email messages that various filters determine to be of lesser value are collected so that they do not arrive in the recipient (p. 146)'s inbox but are still accessible to the recipient. This is also referred to as a spam (p. 152) or bulk folder.

K

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

key — A credential that identifies an AWS account (p. 109) or user (p. 156) to AWS (such as the AWS secret access key (p. 150)).

Amazon Simple Storage Service (Amazon S3) (p. 113), Amazon EMR (Amazon EMR) (p. 111): The unique identifier for an object in a bucket (p. 121). Every object in a bucket has exactly one key. Because a bucket and key together uniquely identify each object, you can think of Amazon S3 as a basic data map between the *bucket + key*, and the object itself. You can uniquely address every object in Amazon S3 through the combination of the web service endpoint, bucket name, and key, as in this example: `http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl`, where `doc` is the name of the bucket, and `2006-03-01/AmazonS3.wsdl` is the key.

AWS Import/Export (p. 118): The name of an object in Amazon S3. It is a sequence of Unicode characters whose UTF-8 encoding cannot exceed 1024

bytes. If a key, for example, logPrefix + import-log-JOBID, is longer than 1024 bytes, AWS Elastic Beanstalk (p. 117) returns an `InvalidManifestField` error.

IAM (p. 118): In a policy (p. 143), a specific characteristic that is the basis for restricting access (such as the current time, or the IP address of the requester).

Tagging resources: A general tag (p. 154) label that acts like a category for more specific tag values. For example, you might have EC2 instance (p. 129) with the tag key of *Owner* and the tag value of *Jan*. You can tag an AWS resource (p. 147) with up to 10 key–value pairs. Not all AWS resources can be tagged.

key pair — A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key.

key prefix — A logical grouping of the objects in a bucket (p. 121). The prefix value is similar to a directory name that enables you to store similar data under the same directory in a bucket.

kibibyte — A contraction of kilo binary byte, a kibibyte is 2^10 or 1,024 bytes. A kilobyte (KB) is 10^3 or 1,000 bytes. 1,024 KiB is a mebibyte (p. 139).

KMS — See AWS Key Management Service (AWS KMS).

L

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

labeled data — In machine learning, data for which you already know the target or "correct" answer.

launch configuration — A set of descriptive parameters used to create new EC2 instance (p. 129)s in an Auto Scaling (p. 115) activity.

A template that an Auto Scaling group (p. 115) uses to launch new EC2 instances. The launch configuration contains information such as the Amazon Machine Image (AMI) (p. 112) ID, the instance type, key pairs, security group (p. 150)s, and block device mappings, among other configuration settings.

launch permission — An Amazon Machine Image (AMI) (p. 112) attribute that allows users to launch an AMI.

lifecycle — The lifecycle state of the EC2 instance (p. 129) contained in an Auto Scaling group (p. 115). EC2 instances progress through several states over their lifespan; these include *Pending*, *InService*, *Terminating* and *Terminated*.

lifecycle action — An action that can be paused by Auto Scaling, such as launching or terminating an EC2 instance.

lifecycle hook — Enables you to pause Auto Scaling after it launches or terminates an EC2 instance so that you can perform a custom action while the instance is not in service.

link to VPC — The process of linking (or attaching) an EC2-Classic instance (p. 135) to a ClassicLink-enabled VPC (p. 158).
See Also ClassicLink, unlink from VPC.

load balancer — A DNS name combined with a set of ports, which together provide a destination for all requests intended for your application. A load balancer can distribute traffic to multiple application instances across every Availability Zone (p. 115) within a Region (p. 146). Load balancers can span multiple Availability Zones within an AWS Region into which an Amazon EC2 (p. 111) instance was launched. But load balancers cannot span multiple Regions.

local secondary index — An index that has the same partition key as the table, but a different sort key. A local secondary index is local in the sense that every partition of a local secondary index is scoped to a table partition that has the same partition key value.
See Also local secondary index.

logical name — A case-sensitive unique string within an AWS CloudFormation (p. 116) template that identifies a resource (p. 147), mapping (p. 138), parameter, or output. In an AWS CloudFormation template, each parameter, resource (p. 147), property, mapping, and output must be declared with a unique logical name. You use the logical name when dereferencing these items using the `Ref` function.

M

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

Mail Transfer Agent (MTA) — Software that transports email messages from one computer to another by using a client-server architecture.

mailbox provider — An organization that provides email mailbox hosting services. Mailbox providers are sometimes referred to as internet service provider (p. 135)s, even if they only provide mailbox services.

mailbox simulator — A set of email addresses that you can use to test an Amazon SES (p. 113)-based email sending application without sending messages to actual recipients. Each email address represents a specific scenario (such as a bounce or complaint) and generates a typical response that is specific to the scenario.

main route table — The default route table (p. 148) that any new VPC (p. 158) subnet (p. 154) uses for routing. You can associate a subnet with a different route table of your choice. You can also change which route table is the main route table.

managed policy — A standalone IAM (p. 118) policy (p. 143) that you can attach to multiple user (p. 156)s, group (p. 133)s, and role (p. 148)s in your IAM account (p. 109). Managed policies can either be AWS managed policies (which are created and managed by AWS) or customer managed policies (which you create and manage in your AWS account).

manifest — When sending a *create job* request for an import or export operation, you describe your job in a text file called a manifest. The manifest file is a YAML-formatted file that specifies how to transfer data between your storage device and the AWS cloud.

manifest file — Amazon Machine Learning: The file used for describing batch predictions. The manifest file relates each input data file with its associated batch prediction results. It is stored in the Amazon S3 output location.

mapping — A way to add conditional parameter values to an AWS CloudFormation (p. 116) template. You specify mappings in the template's optional Mappings section and retrieve the desired value using the `FN::FindInMap` function.

marker	See pagination token.
master node	A process running on an Amazon Machine Image (AMI) (p. 112) that keeps track of the work its core and task nodes complete.
maximum price	The maximum price you will pay to launch one or more Spot Instance (p. 152)s. If your maximum price exceeds the current Spot price (p. 152) and your restrictions are met, Amazon EC2 (p. 111) launches instances on your behalf.
maximum send rate	The maximum number of email messages that you can send per second using Amazon SES (p. 113).
mebibyte	A contraction of mega binary byte, a mebibyte is 2^20 or 1,048,576 bytes. A megabyte (MB) is 10^6 or 1,000,000 bytes. 1,024 MiB is a gibibyte (p. 133).
member resources	See resource.
message ID	Amazon Simple Email Service (Amazon SES) (p. 113): A unique identifier that is assigned to every email message that is sent. Amazon Simple Queue Service (Amazon SQS) (p. 113): The identifier returned when you send a message to a queue.
metadata	Information about other data or objects. In Amazon Simple Storage Service (Amazon S3) (p. 113) and Amazon EMR (Amazon EMR) (p. 111) metadata takes the form of name–value pairs that describe the object. These include default metadata such as the date last modified and standard HTTP metadata such as Content-Type. Users can also specify custom metadata at the time they store an object. In Amazon Elastic Compute Cloud (Amazon EC2) (p. 111) metadata includes data about an EC2 instance (p. 129) that the instance can retrieve to determine things about itself, such as the instance type, the IP address, and so on.
metric	An element of time-series data defined by a unique combination of exactly one namespace (p. 140), exactly one metric name, and between zero and ten dimensions. Metrics and the statistics derived from them are the basis of Amazon CloudWatch (p. 110).
metric name	The primary identifier of a metric, used in combination with a namespace (p. 140) and optional dimensions.
MFA	See multi-factor authentication (MFA).
micro instance	A type of EC2 instance (p. 129) that is more economical to use if you have occasional bursts of high CPU activity.
MIME	See Multipurpose Internet Mail Extensions (MIME).
ML model	In machine learning (ML), a mathematical model that generates predictions by finding patterns in data. Amazon Machine Learning supports three types of ML models: binary classification, multiclass classification, and regression. Also known as a *predictive model*. See Also binary classification model, multiclass classification model, regression model.
MTA	See Mail Transfer Agent (MTA).
Multi-AZ deployment	A primary DB instance (p. 126) that has a synchronous standby replica in a different Availability Zone (p. 115). The primary DB instance is synchronously replicated across Availability Zones to the standby replica.

multiclass classification model	A machine learning model that predicts values that belong to a limited, pre-defined set of permissible values. For example, "Is this product a book, movie, or clothing?"
multi-factor authentication (MFA)	An optional AWS account (p. 109) security feature. Once you enable AWS MFA, you must provide a six-digit, single-use code in addition to your sign-in credentials whenever you access secure AWS webpages or the AWS Management Console (p. 118). You get this single-use code from an authentication device that you keep in your physical possession. See Also https://aws.amazon.com/mfa/.
multi-valued attribute	An attribute with more than one value.
multipart upload	A feature that allows you to upload a single object as a set of parts.
Multipurpose Internet Mail Extensions (MIME)	An internet standard that extends the email protocol to include non-ASCII text and nontext elements like attachments.
Multitool	A cascading application that provides a simple command-line interface for managing large datasets.

N

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

namespace	An abstract container that provides context for the items (names, or technical terms, or words) it holds, and allows disambiguation of homonym items residing in different namespaces.
NAT	Network address translation. A strategy of mapping one or more IP addresses to another while data packets are in transit across a traffic routing device. This is commonly used to restrict internet communication to private instances while allowing outgoing traffic. See Also Network Address Translation and Protocol Translation, NAT gateway, NAT instance.
NAT gateway	A NAT (p. 140) device, managed by AWS, that performs network address translation in a private subnet (p. 154), to secure inbound internet traffic. A NAT gateway uses both NAT and port address translation. See Also NAT instance.
NAT instance	A NAT (p. 140) device, configured by a user, that performs network address translation in a VPC (p. 158) public subnet (p. 154) to secure inbound internet traffic. See Also NAT gateway.
network ACL	An optional layer of security that acts as a firewall for controlling traffic in and out of a subnet (p. 154). You can associate multiple subnets with a single network ACL (p. 108), but a subnet can be associated with only one network ACL at a time.
Network Address Translation and Protocol Translation	(NAT (p. 140)-PT) An internet protocol standard defined in RFC 2766. See Also NAT instance, NAT gateway.
n-gram processor	A processor that performs n-gram transformations. See Also n-gram transformation.

n-gram transformation — Amazon Machine Learning: A transformation that aids in text string analysis. An n-gram transformation takes a text variable as input and outputs strings by sliding a window of size *n* words, where *n* is specified by the user, over the text, and outputting every string of words of size *n* and all smaller sizes. For example, specifying the n-gram transformation with window size =2 returns all the two-word combinations and all of the single words.

node — Amazon Elasticsearch Service (Amazon ES) (p. 111): An Elasticsearch instance. A node can be either a data instance or a dedicated master instance.
See Also dedicated master node.

NoEcho — A property of AWS CloudFormation (p. 116) parameters that prevent the otherwise default reporting of names and values of a template parameter. Declaring the `NoEcho` property causes the parameter value to be masked with asterisks in the report by the `cfn-describe-stacks` command.

NoSQL — Nonrelational database systems that are highly available, scalable, and optimized for high performance. Instead of the relational model, NoSQL databases (like Amazon DynamoDB (p. 110)) use alternate models for data management, such as key–value pairs or document storage.

null object — A null object is one whose version ID is null. Amazon S3 (p. 113) adds a null object to a bucket (p. 121) when versioning (p. 157) for that bucket is suspended. It is possible to have only one null object for each key in a bucket.

number of passes — The number of times that you allow Amazon Machine Learning to use the same data records to train a machine learning model.

O

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

object — Amazon Simple Storage Service (Amazon S3) (p. 113): The fundamental entity type stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3.

Amazon CloudFront (p. 110): Any entity that can be served either over HTTP or a version of RTMP.

observation — Amazon Machine Learning: A single instance of data that Amazon Machine Learning (Amazon ML) uses to either train a machine learning model how to predict or to generate a prediction. Each row in an Amazon ML input data file is an observation.

On-Demand Instance — An Amazon EC2 (p. 111) pricing option that charges you for compute capacity by the hour with no long-term commitment.

operation — An API function. Also called an *action*.

optimistic locking — A strategy to ensure that an item that you want to update has not been modified by others before you perform the update. For Amazon DynamoDB (p. 110), optimistic locking support is provided by the AWS SDKs.

organization — AWS Organizations (p. 118): An entity that you create to consolidate and manage your AWS accounts. An organization has one master account along with zero or more member accounts.

organizational unit — AWS Organizations (p. 118): A container for accounts within a root (p. 148) of an organization. An organizational unit (OU) can contain other OUs.

origin access identity — Also called OAI. When using Amazon CloudFront (p. 110) to serve content with an Amazon S3 (p. 113) bucket (p. 121) as the origin, a virtual identity that you use to require users to access your content through CloudFront URLs instead of Amazon S3 URLs. Usually used with CloudFront private content (p. 144).

origin server — The Amazon S3 (p. 113) bucket (p. 121) or custom origin containing the definitive original version of the content you deliver through CloudFront (p. 110).

original environment — The instances in a deployment group at the start of an AWS CodeDeploy blue/green deployment.

OSB transformation — Orthogonal sparse bigram transformation. In machine learning, a transformation that aids in text string analysis and that is an alternative to the n-gram transformation. OSB transformations are generated by sliding the window of size *n* words over the text, and outputting every pair of words that includes the first word in the window.
See Also n-gram transformation.

OU — See organizational unit.

output location — Amazon Machine Learning: An Amazon S3 location where the results of a batch prediction are stored.

P

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

pagination — The process of responding to an API request by returning a large list of records in small separate parts. Pagination can occur in the following situations:

- The client sets the maximum number of returned records to a value below the total number of records.
- The service has a default maximum number of returned records that is lower than the total number of records.

When an API response is paginated, the service sends a subset of the large list of records and a pagination token that indicates that more records are available. The client includes this pagination token in a subsequent API request, and the service responds with the next subset of records. This continues until the service responds with a subset of records and no pagination token, indicating that all records have been sent.

pagination token — A marker that indicates that an API response contains a subset of a larger list of records. The client can return this marker in a subsequent API request to retrieve the next subset of records until the service responds with a subset of records and no pagination token, indicating that all records have been sent.
See Also pagination.

paid AMI — An Amazon Machine Image (AMI) (p. 112) that you sell to other Amazon EC2 (p. 111) users on AWS Marketplace (p. 118).

paravirtual virtualization	See PV virtualization.
part	A contiguous portion of the object's data in a multipart upload request.
partition key	A simple primary key, composed of one attribute (also known as a *hash attribute*). See Also partition key, sort key.
PAT	Port address translation.
pebibyte	A contraction of peta binary byte, a pebibyte is 2^50 or 1,125,899,906,842,624 bytes. A petabyte (PB) is 10^15 or 1,000,000,000,000,000 bytes. 1,024 PiB is an exbibyte (p. 131).
period	See sampling period.
permission	A statement within a policy (p. 143) that allows or denies access to a particular resource (p. 147). You can state any permission like this: "A has permission to do B to C." For example, Jane (A) has permission to read messages (B) from John's Amazon SQS (p. 113) queue (C). Whenever Jane sends a request to Amazon SQS to use John's queue, the service checks to see if she has permission and if the request satisfies the conditions John set forth in the permission.
persistent storage	A data storage solution where the data remains intact until it is deleted. Options within AWS (p. 114) include: Amazon S3 (p. 113), Amazon RDS (p. 113), Amazon DynamoDB (p. 110), and other services.
physical name	A unique label that AWS CloudFormation (p. 116) assigns to each resource (p. 147) when creating a stack (p. 152). Some AWS CloudFormation commands accept the physical name as a value with the `--physical-name` parameter.
pipeline	AWS CodePipeline (p. 116): A workflow construct that defines the way software changes go through a release process.
plaintext	Information that has not been encrypted (p. 130), as opposed to ciphertext (p. 123).
policy	IAM (p. 118): A document defining permissions that apply to a user, group, or role; the permissions in turn determine what users can do in AWS. A policy typically allow (p. 109)s access to specific actions, and can optionally grant that the actions are allowed for specific resource (p. 147)s, like EC2 instance (p. 129)s, Amazon S3 (p. 113) bucket (p. 121)s, and so on. Policies can also explicitly deny (p. 127) access. Auto Scaling (p. 115): An object that stores the information needed to launch or terminate instances for an Auto Scaling group. Executing the policy causes instances to be launched or terminated. You can configure an alarm (p. 109) to invoke an Auto Scaling policy.
policy generator	A tool in the IAM (p. 118) AWS Management Console (p. 118) that helps you build a policy (p. 143) by selecting elements from lists of available options.
policy simulator	A tool in the IAM (p. 118) AWS Management Console (p. 118) that helps you test and troubleshoot policies (p. 143) so you can see their effects in real-world scenarios.
policy validator	A tool in the IAM (p. 118) AWS Management Console (p. 118) that examines your existing IAM access control policies (p. 143) to ensure that they comply with the IAM policy grammar.

presigned URL	A web address that uses query string authentication (p. 145).
prefix	See job prefix.
Premium Support	A one-on-one, fast-response support channel that AWS customers can subscribe to for support for AWS infrastructure services. See Also https://aws.amazon.com/premiumsupport/.
primary key	One or two attributes that uniquely identify each item in a Amazon DynamoDB (p. 110) table, so that no two items can have the same key. See Also partition key, sort key.
primary shard	See shard.
principal	The user (p. 156), service, or account (p. 109) that receives permissions that are defined in a policy (p. 143). The principal is A in the statement "A has permission to do B to C."
private content	When using Amazon CloudFront (p. 110) to serve content with an Amazon S3 (p. 113) bucket (p. 121) as the origin, a method of controlling access to your content by requiring users to use signed URLs. Signed URLs can restrict user access based on the current date and time and/or the IP addresses that the requests originate from.
private IP address	A private numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). All EC2 instance (p. 129)ss are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT (p. 140)): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC (p. 113) are assigned only a private IP address.
private subnet	A VPC (p. 158) subnet (p. 154) whose instances cannot be reached from the internet.
product code	An identifier provided by AWS when you submit a product to AWS Marketplace (p. 118).
properties	See resource property.
property rule	A JSON (p. 136)-compliant markup standard for declaring properties, mappings, and output values in an AWS CloudFormation (p. 116) template.
Provisioned IOPS	A storage option designed to deliver fast, predictable, and consistent I/O performance. When you specify an IOPS rate while creating a DB instance, Amazon RDS (p. 113) provisions that IOPS rate for the lifetime of the DB instance.
pseudo parameter	A predefined setting, such as `AWS:StackName` that can be used in AWS CloudFormation (p. 116) templates without having to declare them. You can use pseudo parameters anywhere you can use a regular parameter.
public AMI	An Amazon Machine Image (AMI) (p. 112) that all AWS account (p. 109)s have permission to launch.
public data set	A large collection of public information that can be seamlessly integrated into AWS cloud-based applications. Amazon stores public data sets at no charge to the community and, like all AWS services, users pay only for the compute and storage they use for their own applications. These data sets currently include data from the Human Genome Project, the U.S. Census, Wikipedia, and other sources. See Also https://aws.amazon.com/publicdatasets.

public IP address	A pubic numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). EC2 instance (p. 129)s are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT (p. 140)): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC (p. 113) are assigned only a private IP address.
public subnet	A subnet (p. 154) whose instances can be reached from the internet.
PV virtualization	Paravirtual virtualization. Allows guest VMs to run on host systems that do not have special support extensions for full hardware and CPU virtualization. Because PV guests run a modified operating system that does not use hardware emulation, they cannot provide hardware-related features such as enhanced networking or GPU support. See Also HVM virtualization.

Q

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

quartile binning transformation	Amazon Machine Learning: A process that takes two inputs, a numerical variable and a parameter called a bin number, and outputs a categorical variable. Quartile binning transformations discover non-linearity in a variable's distribution by enabling the machine learning model to learn separate importance values for parts of the numeric variable's distribution.
Query	A type of web service that generally uses only the GET or POST HTTP method and a query string with parameters in the URL. See Also REST.
query string authentication	An AWS feature that lets you place the authentication information in the HTTP request query string instead of in the `Authorization` header, which enables URL-based access to objects in a bucket (p. 121).
queue	A sequence of messages or jobs that are held in temporary storage awaiting transmission or processing.
queue URL	A web address that uniquely identifies a queue.
quota	Amazon RDS (p. 113): The maximum number of DB instance (p. 126)s and available storage you can use. Amazon ElastiCache (p. 111): The maximum number of the following items: • The number of cache clusters for each AWS account (p. 109) • The number of cache nodes per cache cluster • The total number of cache nodes per AWS account across all cache clusters created by that AWS account

R

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141)

| P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

range GET	A request that specifies a byte range of data to get for a download. If an object is large, you can break up a download into smaller units by sending multiple range GET requests that each specify a different byte range to GET.
raw email	A type of *sendmail* request with which you can specify the email headers and MIME types.
RDS	See Amazon Relational Database Service (Amazon RDS).
read replica	Amazon RDS (p. 113): An active copy of another DB instance. Any updates to the data on the source DB instance are replicated to the read replica DB instance using the built-in replication feature of MySQL 5.1.
real-time predictions	Amazon Machine Learning: Synchronously generated predictions for individual data observations. See Also batch prediction.
receipt handle	Amazon SQS (p. 113): An identifier that you get when you receive a message from the queue. This identifier is required to delete a message from the queue or when changing a message's visibility timeout.
receiver	The entity that consists of the network systems, software, and policies that manage email delivery for a recipient (p. 146).
recipient	Amazon Simple Email Service (Amazon SES) (p. 113): The person or entity receiving an email message. For example, a person named in the "To" field of a message.
Redis	A fast, open source, in-memory key-value data structure store. Redis comes with a set of versatile in-memory data structures with which you can easily create a variety of custom applications.
reference	A means of inserting a property from one AWS resource (p. 147) into another. For example, you could insert an Amazon EC2 (p. 111) security group (p. 150) property into an Amazon RDS (p. 113) resource.
Region	A named set of AWS resource (p. 147)s in the same geographical area. A Region comprises at least two Availability Zone (p. 115)s.
regression model	Amazon Machine Learning: Preformatted instructions for common data transformations that fine-tune machine learning model performance.
regression model	A type of machine learning model that predicts a numeric value, such as the exact purchase price of a house.
regularization	A machine learning (ML) parameter that you can tune to obtain higher-quality ML models. Regularization helps prevent ML models from memorizing training data examples instead of learning how to generalize the patterns it sees (called overfitting). When training data is overfitted, the ML model performs well on the training data but does not perform well on the evaluation data or on new data.
replacement environment	The instances in a deployment group after the AWS CodeDeploy blue/green deployment.
replica shard	See shard.
reply path	The email address to which an email reply is sent. This is different from the return path (p. 148).

representational state transfer	See REST.
reputation	1. An Amazon SES (p. 113) metric, based on factors that might include bounce (p. 121)s, complaint (p. 124)s, and other metrics, regarding whether or not a customer is sending high-quality email. 2. A measure of confidence, as judged by an internet service provider (p. 135) or other entity that an IP address that they are receiving email from is not the source of spam (p. 152).
requester	The person (or application) that sends a request to AWS to perform a specific action. When AWS receives a request, it first evaluates the requester's permissions to determine whether the requester is allowed to perform the request action (if applicable, for the requested resource (p. 147)).
Requester Pays	An Amazon S3 (p. 113) feature that allows a bucket owner (p. 122) to specify that anyone who requests access to objects in a particular bucket (p. 121) must pay the data transfer and request costs.
reservation	A collection of EC2 instance (p. 129)s started as part of the same launch request. Not to be confused with a Reserved Instance (p. 147).
Reserved Instance	A pricing option for EC2 instance (p. 129)s that discounts the on-demand (p. 141) usage charge for instances that meet the specified parameters. Customers pay for the entire term of the instance, regardless of how they use it.
Reserved Instance Marketplace	An online exchange that matches sellers who have reserved capacity that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instance (p. 147)s that you purchase from third-party sellers have less than a full standard term remaining and can be sold at different upfront prices. The usage or reoccurring fees remain the same as the fees set when the Reserved Instances were originally purchased. Full standard terms for Reserved Instances available from AWS run for one year or three years.
resource	An entity that users can work with in AWS, such as an EC2 instance (p. 129), an Amazon DynamoDB (p. 110) table, an Amazon S3 (p. 113) bucket (p. 121), an IAM (p. 118) user, an AWS OpsWorks (p. 118) stack (p. 152), and so on.
resource property	A value required when including an AWS resource (p. 147) in an AWS CloudFormation (p. 116) stack (p. 152). Each resource may have one or more properties associated with it. For example, an `AWS::EC2::Instance` resource may have a `UserData` property. In an AWS CloudFormation template, resources must declare a properties section, even if the resource has no properties.
resource record	Also called *resource record set*. The fundamental information elements in the Domain Name System (DNS). See Also Domain Name System in Wikipedia.
REST	Representational state transfer. A simple stateless architecture that generally runs over HTTPS/TLS. REST emphasizes that resources have unique and hierarchical identifiers (URIs), are represented by common media types (HTML, XML, JSON (p. 136), and so on), and that operations on the resources are either predefined or discoverable within the media type. In practice, this generally results in a limited number of operations. See Also Query, WSDL, SOAP.
RESTful web service	Also known as RESTful API. A web service that follows REST (p. 147) architectural constraints. The API operations must use HTTP methods explicitly; expose hierarchical URIs; and transfer either XML, JSON (p. 136), or both.

HTTP-Query — See Query.

return enabled — Amazon CloudSearch (p. 110): An index field option that enables the field's values to be returned in the search results.

return path — The email address to which bounced email is returned. The return path is specified in the header of the original email. This is different from the reply path (p. 146).

revision — AWS CodePipeline (p. 116): A change made to a source that is configured in a source action, such as a pushed commit to a GitHub (p. 133) repository or an update to a file in a versioned Amazon S3 (p. 113) bucket (p. 121).

role — A tool for giving temporary access to AWS resource (p. 147)s in your AWS account (p. 109).

rollback — A return to a previous state that follows the failure to create an object, such as AWS CloudFormation (p. 116) stack (p. 152). All resource (p. 147)s associated with the failure are deleted during the rollback. For AWS CloudFormation, you can override this behavior using the `--disable-rollback` option on the command line.

root — AWS Organizations (p. 118): A parent container for the accounts in your organization. If you apply a service control policy (p. 150) to the root, it applies to every organizational unit (p. 142) and account in the organization.

root credentials — Authentication information associated with the AWS account (p. 109) owner.

root device volume — A volume (p. 157) that contains the image used to boot the instance (p. 135) (also known as a *root device*). If you launched the instance from an AMI (p. 112) backed by instance store (p. 135), this is an instance store volume (p. 157) created from a template stored in Amazon S3 (p. 113). If you launched the instance from an AMI backed by Amazon EBS (p. 110), this is an Amazon EBS volume created from an Amazon EBS snapshot.

route table — A set of routing rules that controls the traffic leaving any subnet (p. 154) that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.

row identifier — row ID.Amazon Machine Learning: An attribute in the input data that you can include in the evaluation or prediction output to make it easier to associate a prediction with an observation.

rule — AWS WAF (p. 120): A set of conditions that AWS WAF searches for in web requests to AWS resource (p. 147)s such as Amazon CloudFront (p. 110) distributions. You add rules to a web ACL (p. 158), and then specify whether you want to allow or block web requests based on each rule.

S

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

S3 — See Amazon Simple Storage Service (Amazon S3).

sampling period — A defined duration of time, such as one minute, over which Amazon CloudWatch (p. 110) computes a statistic (p. 153).

sandbox	A testing location where you can test the functionality of your application without affecting production, incurring charges, or purchasing products. Amazon SES (p. 113): An environment that is designed for developers to test and evaluate the service. In the sandbox, you have full access to the Amazon SES API, but you can only send messages to verified email addresses and the mailbox simulator. To get out of the sandbox, you need to apply for production access. Accounts in the sandbox also have lower sending limits (p. 150) than production accounts.
scale in	To remove EC2 instances from an Auto Scaling group (p. 115).
scale out	To add EC2 instances to an Auto Scaling group (p. 115).
scaling policy	A description of how Auto Scaling should automatically scale an Auto Scaling group (p. 115) in response to changing demand. See Also scale in, scale out.
scaling activity	A process that changes the size, configuration, or makeup of an Auto Scaling group (p. 115) by launching or terminating instances.
scheduler	The method used for placing task (p. 155)s on container instance (p. 124)s.
schema	Amazon Machine Learning: The information needed to interpret the input data for a machine learning model, including attribute names and their assigned data types, and the names of special attributes.
score cut-off value	Amazon Machine Learning: A binary classification models output a score that ranges from 0 to 1. To decide whether an observation should be classified as 1 or 0, you pick a classification threshold, or cut-off, and Amazon ML compares the score against it. Observations with scores higher than the cut-off are predicted as target equals 1, and scores lower than the cut-off are predicted as target equals 0.
SCP	See service control policy.
search API	Amazon CloudSearch (p. 110): The API that you use to submit search requests to a search domain (p. 149).
search domain	Amazon CloudSearch (p. 110): Encapsulates your searchable data and the search instances that handle your search requests. You typically set up a separate Amazon CloudSearch domain for each different collection of data that you want to search.
search domain configuration	Amazon CloudSearch (p. 110): An domain's indexing options, analysis scheme (p. 114)s, expression (p. 131)s, suggester (p. 154)s, access policies, and scaling and availability options.
search enabled	Amazon CloudSearch (p. 110): An index field option that enables the field data to be searched.
search endpoint	Amazon CloudSearch (p. 110): The URL that you connect to when sending search requests to a search domain. Each Amazon CloudSearch domain has a unique search endpoint that remains the same for the life of the domain.
search index	Amazon CloudSearch (p. 110): A representation of your searchable data that facilitates fast and accurate data retrieval.
search instance	Amazon CloudSearch (p. 110): A compute resource (p. 147) that indexes your data and processes search requests. An Amazon CloudSearch domain has one or more search instances, each with a finite amount of RAM and CPU

	resources. As your data volume grows, more search instances or larger search instances are deployed to contain your indexed data. When necessary, your index is automatically partitioned across multiple search instances. As your request volume or complexity increases, each search partition is automatically replicated to provide additional processing capacity.
search request	Amazon CloudSearch (p. 110): A request that is sent to an Amazon CloudSearch domain's search endpoint to retrieve documents from the index that match particular search criteria.
search result	Amazon CloudSearch (p. 110): A document that matches a search request. Also referred to as a *search hit*.
secret access key	A key that is used in conjunction with the access key ID (p. 108) to cryptographically sign programmatic AWS requests. Signing a request identifies the sender and prevents the request from being altered. You can generate secret access keys for your AWS account (p. 109), individual IAM user (p. 156)s, and temporary sessions.
security group	A named set of allowed inbound network connections for an instance. (Security groups in Amazon VPC (p. 113) also include support for outbound connections.) Each security group consists of a list of protocols, ports, and IP address ranges. A security group can apply to multiple instances, and multiple groups can regulate a single instance.
sender	The person or entity sending an email message.
Sender ID	A Microsoft-controlled version of SPF (p. 152). An email authentication and anti-spoofing system. For more information about Sender ID, see Sender ID in Wikipedia.
sending limits	The sending quota (p. 150) and maximum send rate (p. 139) that are associated with every Amazon SES (p. 113) account.
sending quota	The maximum number of email messages that you can send using Amazon SES (p. 113) in a 24-hour period.
server-side encryption (SSE)	The encrypting (p. 130) of data at the server level. Amazon S3 (p. 113) supports three modes of server-side encryption: SSE-S3, in which Amazon S3 manages the keys; SSE-C, in which the customer manages the keys; and SSE-KMS, in which AWS Key Management Service (AWS KMS) (p. 118) manages keys.
service	See Amazon ECS service.
service control policy	AWS Organizations (p. 118): A policy-based control that specifies the services and actions that users and roles can use in the accounts that the service control policy (SCP) affects.
service endpoint	See endpoint.
service health dashboard	A web page showing up-to-the-minute information about AWS service availability. The dashboard is located at http://status.aws.amazon.com/.
service role	An IAM (p. 118) role (p. 148) that grants permissions to an AWS service so it can access AWS resource (p. 147)s. The policies that you attach to the service role determine which AWS resources the service can access and what it can do with those resources.
SES	See Amazon Simple Email Service (Amazon SES).

session	The period during which the temporary security credentials provided by AWS Security Token Service (AWS STS) (p. 119) allow access to your AWS account.
SHA	Secure Hash Algorithm. SHA1 is an earlier version of the algorithm, which AWS has deprecated in favor of SHA256.
shard	Amazon Elasticsearch Service (Amazon ES) (p. 111): A partition of data in an index. You can split an index into multiple shards, which can include primary shards (original shards) and replica shards (copies of the primary shards). Replica shards provide failover, which means that a replica shard is promoted to a primary shard if a cluster node that contains a primary shard fails. Replica shards also can handle requests.
shared AMI	An Amazon Machine Image (AMI) (p. 112) that a developer builds and makes available for others to use.
shutdown action	Amazon EMR (p. 111): A predefined bootstrap action that launches a script that executes a series of commands in parallel before terminating the job flow.
signature	Refers to a *digital signature*, which is a mathematical way to confirm the authenticity of a digital message. AWS uses signatures to authenticate the requests you send to our web services. For more information, to https://aws.amazon.com/security.
SIGNATURE file	AWS Import/Export (p. 118): A file you copy to the root directory of your storage device. The file contains a job ID, manifest file, and a signature.
Signature Version 4	Protocol for authenticating inbound API requests to AWS services in all AWS Regions.
Simple Mail Transfer Protocol	See SMTP.
Simple Object Access Protocol	See SOAP.
Simple Storage Service	See Amazon Simple Storage Service (Amazon S3).
Single-AZ DB instance	A standard (non-Multi-AZ) DB instance (p. 126) that is deployed in one Availability Zone (p. 115), without a standby replica in another Availability Zone. See Also Multi-AZ deployment.
sloppy phrase search	A search for a phrase that specifies how close the terms must be to one another to be considered a match.
SMTP	Simple Mail Transfer Protocol. The standard that is used to exchange email messages between internet hosts for the purpose of routing and delivery.
snapshot	Amazon Elastic Block Store (Amazon EBS) (p. 110): A backup of your volume (p. 157)s that is stored in Amazon S3 (p. 113). You can use these snapshots as the starting point for new Amazon EBS volumes or to protect your data for long-term durability. See Also DB snapshot.
SNS	See Amazon Simple Notification Service (Amazon SNS).
Snowball	An AWS Import/Export (p. 118) feature that uses Amazon-owned Snowball appliances for transferring your data. See Also https://aws.amazon.com/importexport.
SOAP	Simple Object Access Protocol. An XML-based protocol that lets you exchange information over a particular protocol (HTTP or SMTP, for example) between applications.

	See Also REST, WSDL.
soft bounce	A temporary email delivery failure such as one resulting from a full mailbox.
software VPN	A software appliance-based VPN connection over the internet.
sort enabled	Amazon CloudSearch (p. 110): An index field option that enables a field to be used to sort the search results.
sort key	An attribute used to sort the order of partition keys in a composite primary key (also known as a *range attribute*). See Also partition key, primary key.
source/destination checking	A security measure to verify that an EC2 instance (p. 129) is the origin of all traffic that it sends and the ultimate destination of all traffic that it receives; that is, that the instance is not relaying traffic. Source/destination checking is enabled by default. For instances that function as gateways, such as VPC (p. 158) NAT (p. 140) instances, source/destination checking must be disabled.
spam	Unsolicited bulk email.
spamtrap	An email address that is set up by an anti-spam (p. 152) entity, not for correspondence, but to monitor unsolicited email. This is also called a *honeypot*.
SPF	Sender Policy Framework. A standard for authenticating email. See Also http://www.openspf.org.
Spot Instance	A type of EC2 instance (p. 129) that you can bid on to take advantage of unused Amazon EC2 (p. 111) capacity.
Spot price	The price for a Spot Instance (p. 152) at any given time. If your maximum price exceeds the current price and your restrictions are met, Amazon EC2 (p. 111) launches instances on your behalf.
SQL injection match condition	AWS WAF (p. 120): An attribute that specifies the part of web requests, such as a header or a query string, that AWS WAF inspects for malicious SQL code. Based on the specified conditions, you can configure AWS WAF to allow or block web requests to AWS resource (p. 147)s such as Amazon CloudFront (p. 110) distributions.
SQS	See Amazon Simple Queue Service (Amazon SQS).
SSE	See server-side encryption (SSE).
SSL	Secure Sockets Layer See Also Transport Layer Security.
stack	AWS CloudFormation (p. 116): A collection of AWS resource (p. 147)s that you create and delete as a single unit. AWS OpsWorks (p. 118): A set of instances that you manage collectively, typically because they have a common purpose such as serving PHP applications. A stack serves as a container and handles tasks that apply to the group of instances as a whole, such as managing applications and cookbooks.
station	AWS CodePipeline (p. 116): A portion of a pipeline workflow where one or more actions are performed.
station	A place at an AWS facility where your AWS Import/Export data is transferred on to, or off of, your storage device.

statistic	One of five functions of the values submitted for a given sampling period (p. 148). These functions are `Maximum`, `Minimum`, `Sum`, `Average`, and `SampleCount`.
stem	The common root or substring shared by a set of related words.
stemming	The process of mapping related words to a common stem. This enables matching on variants of a word. For example, a search for "horse" could return matches for horses, horseback, and horsing, as well as horse. Amazon CloudSearch (p. 110) supports both dictionary based and algorithmic stemming.
step	Amazon EMR (p. 111): A single function applied to the data in a job flow (p. 136). The sum of all steps comprises a job flow.
step type	Amazon EMR (p. 111): The type of work done in a step. There are a limited number of step types, such as moving data from Amazon S3 (p. 113) to Amazon EC2 (p. 111) or from Amazon EC2 to Amazon S3.
sticky session	A feature of the Elastic Load Balancing (p. 129) load balancer that binds a user's session to a specific application instance so that all requests coming from the user during the session are sent to the same application instance. By contrast, a load balancer defaults to route each request independently to the application instance with the smallest load.
stopping	The process of filtering stop words from an index or search request.
stopword	A word that is not indexed and is automatically filtered out of search requests because it is either insignificant or so common that including it would result in too many matches to be useful. Stop words are language-specific.
streaming	Amazon EMR (Amazon EMR) (p. 111): A utility that comes with Hadoop (p. 133) that enables you to develop MapReduce executables in languages other than Java. Amazon CloudFront (p. 110): The ability to use a media file in real time—as it is transmitted in a steady stream from a server.
streaming distribution	A special kind of distribution (p. 128) that serves streamed media files using a Real Time Messaging Protocol (RTMP) connection.
Streams	See Amazon Kinesis Data Streams.
string-to-sign	Before you calculate an HMAC (p. 133) signature, you first assemble the required components in a canonical order. The preencrypted string is the string-to-sign.
string match condition	AWS WAF (p. 120): An attribute that specifies the strings that AWS WAF searches for in a web request, such as a value in a header or a query string. Based on the specified strings, you can configure AWS WAF to allow or block web requests to AWS resource (p. 147)s such as CloudFront (p. 110) distributions.
strongly consistent read	A read process that returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful—regardless of the region. See Also data consistency, eventual consistency, eventually consistent read.
structured query	Search criteria specified using the Amazon CloudSearch (p. 110) structured query language. You use the structured query language to construct compound queries that use advanced search options and combine multiple search criteria using Boolean operators.
STS	See AWS Security Token Service (AWS STS).

subnet — A segment of the IP address range of a VPC (p. 158) that EC2 instance (p. 129)s can be attached to. You can create subnets to group instances according to security and operational needs.

Subscription button — An HTML-coded button that enables an easy way to charge customers a recurring fee.

suggester — Amazon CloudSearch (p. 110): Specifies an index field you want to use to get autocomplete suggestions and options that can enable fuzzy matches and control how suggestions are sorted.

suggestions — Documents that contain a match for the partial search string in the field designated by the suggester (p. 154). Amazon CloudSearch (p. 110) suggestions include the document IDs and field values for each matching document. To be a match, the string must match the contents of the field starting from the beginning of the field.

supported AMI — An Amazon Machine Image (AMI) (p. 112) similar to a paid AMI (p. 142), except that the owner charges for additional software or a service that customers use with their own AMIs.

SWF — See Amazon Simple Workflow Service (Amazon SWF).

symmetric encryption — Encryption (p. 130) that uses a private key only.
See Also asymmetric encryption.

synchronous bounce — A type of bounce (p. 121) that occurs while the email servers of the sender (p. 150) and receiver (p. 146) are actively communicating.

synonym — A word that is the same or nearly the same as an indexed word and that should produce the same results when specified in a search request. For example, a search for "Rocky Four" or "Rocky 4" should return the fourth *Rocky* movie. This can be done by designating that `four` and `4` are synonyms for `IV`. Synonyms are language-specific.

T

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

table — A collection of data. Similar to other database systems, DynamoDB stores data in tables.

tag — Metadata that you can define and assign to AWS resource (p. 147)s, such as an EC2 instance (p. 129). Not all AWS resources can be tagged.

tagging — Tagging resources: Applying a tag (p. 154) to an AWS resource (p. 147).

Amazon SES (p. 113): Also called *labeling*. A way to format return path (p. 148) email addresses so that you can specify a different return path for each recipient of a message. Tagging enables you to support VERP (p. 157). For example, if Andrew manages a mailing list, he can use the return paths andrew+recipient1@example.net and andrew+recipient2@example.net so that he can determine which email bounced.

target attribute — Amazon Machine Learning (Amazon ML): The attribute in the input data that contains the “correct” answers. Amazon ML uses the target attribute to learn how

	to make predictions on new data. For example, if you were building a model for predicting the sale price of a house, the target attribute would be “target sale price in USD.”
target revision	AWS CodeDeploy (p. 116): The most recent version of the application revision that has been uploaded to the repository and will be deployed to the instances in a deployment group. In other words, the application revision currently targeted for deployment. This is also the revision that will be pulled for automatic deployments.
task	An instantiation of a task definition (p. 155) that is running on a container instance (p. 124).
task definition	The blueprint for your task. Specifies the name of the task (p. 155), revisions, container definition (p. 124)s, and volume (p. 157) information.
task node	An EC2 instance (p. 129) that runs Hadoop (p. 133) map and reduce tasks, but does not store data. Task nodes are managed by the master node (p. 139), which assigns Hadoop tasks to nodes and monitors their status. While a job flow is running you can increase and decrease the number of task nodes. Because they don't store data and can be added and removed from a job flow, you can use task nodes to manage the EC2 instance capacity your job flow uses, increasing capacity to handle peak loads and decreasing it later. Task nodes only run a TaskTracker Hadoop daemon.
tebibyte	A contraction of tera binary byte, a tebibyte is 2^40 or 1,099,511,627,776 bytes. A terabyte (TB) is 10^12 or 1,000,000,000,000 bytes. 1,024 TiB is a pebibyte (p. 143).
template format version	The version of an AWS CloudFormation (p. 116) template design that determines the available features. If you omit the `AWSTemplateFormatVersion` section from your template, AWS CloudFormation assumes the most recent format version.
template validation	The process of confirming the use of JSON (p. 136) code in an AWS CloudFormation (p. 116) template. You can validate any AWS CloudFormation template using the `cfn-validate-template` command.
temporary security credentials	Authentication information that is provided by AWS STS (p. 119) when you call an STS API action. Includes an access key ID (p. 108), a secret access key (p. 150), a session (p. 151) token, and an expiration time.
throttling	The automatic restricting or slowing down of a process based on one or more limits. Examples: Amazon Kinesis Data Streams (p. 112) throttles operations if an application (or group of applications operating on the same stream) attempts to get data from a shard at a rate faster than the shard limit. Amazon API Gateway (p. 109) uses throttling to limit the steady-state request rates for a single account. Amazon SES (p. 113) uses throttling to reject attempts to send email that exceeds the sending limits (p. 150).
time series data	Data provided as part of a metric. The time value is assumed to be when the value occurred. A metric is the fundamental concept for Amazon CloudWatch (p. 110) and represents a time-ordered set of data points. You publish metric data points into CloudWatch and later retrieve statistics about those data points as a time-series ordered data set.
time stamp	A date/time string in ISO 8601 format.
TLS	See Transport Layer Security.

tokenization	The process of splitting a stream of text into separate tokens on detectable boundaries such as whitespace and hyphens.
topic	A communication channel to send messages and subscribe to notifications. It provides an access point for publishers and subscribers to communicate with each other.
training datasource	A datasource that contains the data that Amazon Machine Learning uses to train the machine learning model to make predictions.
transition	AWS CodePipeline (p. 116): The act of a revision in a pipeline continuing from one stage to the next in a workflow.
Transport Layer Security	A cryptographic protocol that provides security for communication over the internet. Its predecessor is Secure Sockets Layer (SSL).
trust policy	An IAM (p. 118) policy (p. 143) that is an inherent part of an IAM role (p. 148). The trust policy specifies which principal (p. 144)s are allowed to use the role.
trusted signers	AWS account (p. 109)s that the CloudFront (p. 110) distribution owner has given permission to create signed URLs for a distribution's content.
tuning	Selecting the number and type of AMIs (p. 112) to run a Hadoop (p. 133) job flow most efficiently.
tunnel	A route for transmission of private network traffic that uses the internet to connect nodes in the private network. The tunnel uses encryption and secure protocols such as PPTP to prevent the traffic from being intercepted as it passes through public routing nodes.

U

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

unbounded	The number of potential occurrences is not limited by a set number. This value is often used when defining a data type that is a list (for example, `maxOccurs="unbounded"`), in WSDL (p. 158).
unit	Standard measurement for the values submitted to Amazon CloudWatch (p. 110) as metric data. Units include seconds, percent, bytes, bits, count, bytes/second, bits/second, count/second, and none.
unlink from VPC	The process of unlinking (or detaching) an EC2-Classic instance (p. 135) from a ClassicLink-enabled VPC (p. 158). See Also ClassicLink, link to VPC.
usage report	An AWS record that details your usage of a particular AWS service. You can generate and download usage reports from https://aws.amazon.com/usage-reports/.
user	A person or application under an account (p. 109) that needs to make API calls to AWS products. Each user has a unique name within the AWS account, and a set of security credentials not shared with other users. These credentials are separate from the AWS account's security credentials. Each user is associated with one and only one AWS account.

V

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

validation
: See template validation.

value
: Instances of attributes (p. 115) for an item, such as cells in a spreadsheet. An attribute might have multiple values.

  Tagging resources: A specific tag (p. 154) label that acts as a descriptor within a tag category (key). For example, you might have EC2 instance (p. 129) with the tag key of *Owner* and the tag value of *Jan*. You can tag an AWS resource (p. 147) with up to 10 key–value pairs. Not all AWS resources can be tagged.

Variable Envelope Return Path
: See VERP.

verification
: The process of confirming that you own an email address or a domain so that you can send email from or to it.

VERP
: Variable Envelope Return Path. A way in which email sending applications can match bounce (p. 121)d email with the undeliverable address that caused the bounce by using a different return path (p. 148) for each recipient. VERP is typically used for mailing lists. With VERP, the recipient's email address is embedded in the address of the return path, which is where bounced email is returned. This makes it possible to automate the processing of bounced email without having to open the bounce messages, which may vary in content.

versioning
: Every object in Amazon S3 (p. 113) has a key and a version ID. Objects with the same key, but different version IDs can be stored in the same bucket (p. 121). Versioning is enabled at the bucket layer using PUT Bucket versioning.

VGW
: See virtual private gateway.

virtualization
: Allows multiple guest virtual machines (VM) to run on a host operating system. Guest VMs can run on one or more levels above the host hardware, depending on the type of virtualization.
  See Also PV virtualization, HVM virtualization.

virtual private cloud
: See VPC.

virtual private gateway
: (VGW) The Amazon side of a VPN connection (p. 158) that maintains connectivity. The internal interfaces of the virtual private gateway connect to your VPC (p. 158) via the VPN attachment and the external interfaces connect to the VPN connection, which leads to the customer gateway (p. 125).

visibility timeout
: The period of time that a message is invisible to the rest of your application after an application component gets it from the queue. During the visibility timeout, the component that received the message usually processes it, and then deletes it from the queue. This prevents multiple components from processing the same message.

volume
: A fixed amount of storage on an instance (p. 135). You can share volume data between container (p. 124)s and persist the data on the container instance (p. 124) when the containers are no longer running.

VPC	Virtual private cloud. An elastic network populated by infrastructure, platform, and application services that share common security and interconnection.
VPC endpoint	A feature that enables you to create a private connection between your VPC (p. 158) and an another AWS service without requiring access over the internet, through a NAT (p. 140) instance, a VPN connection (p. 158), or AWS Direct Connect (p. 117).
VPG	See virtual private gateway.
VPN CloudHub	See AWS VPN CloudHub.
VPN connection	Amazon Web Services (AWS) (p. 114): The IPsec connection between a VPC (p. 158) and some other network, such as a corporate data center, home network, or co-location facility.

W

Numbers and Symbols (p. 108) | A (p. 108) | B (p. 120) | C (p. 122) | D (p. 126) | E (p. 128) | F (p. 131) | G (p. 132) | H (p. 133) | I (p. 134) | J (p. 136) | K (p. 136) | L (p. 137) | M (p. 138) | N (p. 140) | O (p. 141) | P (p. 142) | Q (p. 145) | R (p. 145) | S (p. 148) | T (p. 154) | U (p. 156) | V (p. 157) | W (p. 158) | X, Y, Z (p. 158)

WAM	See Amazon WorkSpaces Application Manager (Amazon WAM).
web access control list	AWS WAF (p. 120): A set of rules that defines the conditions that AWS WAF searches for in web requests to AWS resource (p. 147)s such as Amazon CloudFront (p. 110) distributions. A web access control list (web ACL) specifies whether to allow, block, or count the requests.
Web Services Description Language	See WSDL.
WSDL	Web Services Description Language. A language used to describe the actions that a web service can perform, along with the syntax of action requests and responses. See Also REST, SOAP.

X, Y, Z

X.509 certificate	An digital document that uses the X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the entity described in the certificate (p. 122).
yobibyte	A contraction of yotta binary byte, a yobibyte is 2^80 or 1,208,925,819,614,629,174,706,176 bytes. A yottabyte (YB) is 10^24 or 1,000,000,000,000,000,000,000,000 bytes.
zebibyte	A contraction of zetta binary byte, a zebibyte is 2^70 or 1,180,591,620,717,411,303,424 bytes. A zettabyte (ZB) is 10^21 or 1,000,000,000,000,000,000,000 bytes. 1,024 ZiB is a yobibyte (p. 158).
zone awareness	Amazon Elasticsearch Service (Amazon ES) (p. 111): A configuration that distributes nodes in a cluster across two Availability Zone (p. 115)s in the same Region. Zone awareness helps to prevent data loss and minimizes downtime in the event of node and data center failure. If you enable zone awareness, you must have an even number of data instances in the instance count, and you also must

use the Amazon Elasticsearch Service Configuration API to replicate your data for your Elasticsearch cluster.